5. Dictionary Writing Systems and Beyond

Andrea Abel

Abstract

Compiling a dictionary is a challenging, laborious and time-consuming task, sometimes a real “drudgery”. For many years collecting data and compiling dictionary entries was done with pen and paper, with lots of time spent in routine tasks. Nowadays dictionaries are written on computers. As a consequence, more and more targeted software tools are being developed. These allow lexicographers to concentrate more on their core business, to guarantee the long term storage and reusability of data and, finally, to better support a consistent and structured lexicographic process from the conception phase to the final product. Any software that serves these purposes is generally referred to as a dictionary writing system (DWS).

This contribution aims at giving a general overview over the topic of DWSs as well as illustrating the state of the art. Special attention will be given to all additional tools and features that go beyond a simple DWS.
1 Introduction

For many years, dictionary writing was a task accomplished with pen and paper. Nowadays, dictionaries are written on computers with the help of the increasingly targeted software tools which are being developed for this purpose. This allows lexicographers to concentrate better on their core business, that is, collecting data and compiling dictionary entries. Furthermore, the long term storage and reusability of data can be guaranteed and, finally, a consistent and structured lexicographic process from the conception phase up to the final product, can be supported more efficiently. Depending on the context in which a specific project is being developed, a range of requirements and needs have to be considered (e.g. commercial vs. academic projects, financial issues and time constraints, print vs. electronic medium). Software that meets these needs and performs these tasks is generally referred to as a dictionary writing system.

This chapter aims to give a general overview of dictionary writing systems, describing general trends rather than focusing more narrowly on single products. At the same time, the goal is to provide a holistic view of the additional tools and features that can be integrated into what we might call a ‘pure’ dictionary writing system. Section 2 provides a general overview of dictionary writing systems in an attempt to isolate the concept; it also includes some historical notes. The central functions and components of dictionary writing systems are presented in Section 3. Several recent issues are dealt with individually. Section 4 focuses on the advantages and disadvantages of off-the-shelf vis-à-vis in-house software, and combinations of the two. Possible further selection criteria for lexicographic tools are discussed, focusing on some emerging trends. In fact, the view beyond the pure dictionary
writing system, and the additional features or tools now emerging, may be a decisive factor. There is significant convergence between the various systems, particularly when commercial software systems are compared. In Section 5, the integration of a range of tools is discussed, the most important being dictionary writing systems and corpus query software. Section 6 is devoted to the creation of multifunctional databases using a dictionary writing system. The chapter ends with a few concluding remarks.

2 Dictionary writing systems: a general overview

Both dictionaries as a whole and single dictionary entries are highly structured objects with many recurring elements. Several aspects of any specific dictionary must be defined: the principles of data collection and selection, the treatment of different types of information and parts of speech, the use of a specific metalanguage, the treatment of macro-, medio- and microstructural elements, etc. Each entry in any dictionary contains a range of predetermined possible categories (e.g. headword, pronunciation and collocations) that may differ depending on the entry type (e.g. different parts of speech or reference entries) and have one or more valid locations within any given entry. A noun entry in a monolingual dictionary may, for example, contain categories such as the headword, pronunciation, variant forms, inflected forms, register labels, definitions, lexicographic examples, collocations, idioms and usage notes, while a conjunction entry does not require a category for inflected forms. Some categories have only one valid location within an entry (e.g. pronunciation usually follows the headword), while others may have several valid locations (e.g. lexicographic
Dictionary entries are also characterised by a series of navigational components and markers (e.g. font, font size and symbols). A fixed and predictable structure with regard to both content and layout is essential to allow dictionary users to quickly and easily find the information they are looking for. In order to guarantee the necessary coherence and consistency for efficient dictionary use, it is important that a set of guidelines are clearly defined during the dictionary design phase. These guidelines, which lexicographers must follow carefully, are usually described in detail in a specific style guide (e.g. Atkins and Rundell 2008: 117 ff., Svensèn 2009: 407 ff.). In the past, style guides included detailed instructions on the correct order of various parts of an entry, the font (e.g. size, style, weight etc.) to be used for each element, the precise wording of specific labels (e.g. geographical labels such as “AmE” or “American”), etc. Today, computerisation has largely relieved lexicographers of the need to pay attention to such tasks (Atkins and Rundell 2008: 212). This is one of the respects in which targeted software comes into play, in particular the dictionary writing systems which are the subject of this chapter.

The first generation of dedicated dictionary writing systems was developed at the beginning of the 1990s, with the aim of making life easier on the entry-writing front (Rundell and Kilgarriff 2011). This still seems to be the main task of dictionary writing systems, whether the software is produced commercially or in-house. In addition to the requirement of supporting lexicographers in their work, sustainable data storage and – especially in larger projects – an efficient project management system (which may range from the conception phase to the final product) of the whole workflow are challenging demands that computerisation should be able to meet (see...

Atkins and Rundell 2008). These three aspects should be taken into consideration when trying to outline the main features of a dictionary writing system.

If we compare some definitions of dictionary writing systems, we notice that, for the most part, they agree. According to Kilgarriff (2006: 7), a dictionary writing system is “a piece of software for writing and producing a dictionary. It might include an editor, a database, a Web interface and various management tools […]. It operates with a dictionary grammar, which specifies the structure of the dictionary”. These key components are found in various definitions. For Atkins and Rundell (2008: 103), a dictionary writing system is “a program that enables lexicographers to compile and edit dictionary text, as well as facilitating project management and (later in the process) typesetting and output to printed or electronic media.”

Several terms are used to refer to this type of application: here the focus will be on English ones. They usually designate somewhat complex software that includes several components in addition to the simple entry-writing editor, particularly in the case of commercial products, or a software program that is closely connected to a series of applications that together serve to assist dictionary production. Thus, besides ‘dictionary writing system’ (used, for example, by Atkins and Rundell 2008: 103), other terms include: ‘dictionary editing system’ (Svensén 2009: 422), ‘dictionary compilation software’, ‘lexicography software’ or ‘dictionary production software’ (de Schryver and Joffe 2006: 41, Joffe and de Schryver 2004: 17), ‘lexicographic workbench’ (Ridings 2003: 204), ‘dictionary management system’ or ‘lexicographer’s workbench’ (Langemets, Loopmann and Viiks 2010: 425), ‘dictionary editing tool’ (Krek 2010 928), ‘dictionary building software’ (Mangeot 2006: 185) or simply ‘editorial system’ (Tittel 2010: 298).
In the past, a number of software packages have been developed, the best known being commercial products such as IDM DPS, TLex, ABBY Lingvo Content and iLEX offering off-the-shelf solutions suitable for the production of a great variety of dictionary products. In addition, a large number of in-house software programs exist that have been partly created for one dictionary publisher or one specific project (for example the DicSy for Norstedts, Sweden’s leading dictionary publisher (Svensèn 2009: 423, based on the Compulexis system), the ‘Wissensnetz deutsche Sprache’ (German Language Knowledge Network) for the German publisher Duden (Alexa et al. 2002, the ‘Duden Ontology’ that is being developed in cooperation with the software company Intelligent Views using a special software for knowledge management, to mention only a few). Table 1 lists a number of specific dictionary writing systems being created and used in academic and non-commercial contexts. In such environments, the in-house solution is the most common approach.

<table>
<thead>
<tr>
<th>Dictionary writing system</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>EELex</td>
<td>Dictionary writing system developed at the Institute for Estonian Language (Langemets et al., 2010)</td>
</tr>
<tr>
<td>“Algemeen Nederlands Woordenboek” (ANW) Article Editor</td>
<td>Dictionary writing system developed at the Instituut voor Nederlandse Lexicologie, the Netherlands (Niestadt 2009)</td>
</tr>
<tr>
<td>Onoma</td>
<td>Dictionary writing system used in the compilation of the Woordeboek van die Afrikaanse Taal (Ridings 2003, Mongwe 2006: 20)</td>
</tr>
<tr>
<td>Jibiki platform</td>
<td>A generic online, open source environment suitable for</td>
</tr>
</tbody>
</table>
Table 1: In-house dictionary writing systems

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Dictionary Editor and Browser” (DEB)</td>
<td>An open source platform developed at Masaryk University (Horák and Rambousek 2007)</td>
</tr>
<tr>
<td>Termania</td>
<td>A free online dictionary portal with integrated dictionary browsing and editing tools (Krek 2010)</td>
</tr>
<tr>
<td>Different projects developed at the University of Savoy (Mangeot 2006),</td>
<td></td>
</tr>
</tbody>
</table>

In addition to these projects, there are countless tailor-made solutions that do not have proper names. Examples include the lexicographical-lexicological project elexiko at the Institut für Deutsche Sprache (Hahn et al. 2008, Müller-Spitzer 2011), the digital lexical system for the German language “Digitales Wörterbuch der deutschen Sprache” (DWDS) at the Berlin-Brandenburgische Akademie der Wissenschaften (Klein and Geyken 2010), the etymological dictionary of Old French “Dictionnaire étymologique de l’ancien français” (DEAF) at the University of Heidelberg (Tittel 2010). There are also smaller projects such as the electronic learner’s dictionary for German and Italian “Elektronisches Lernerwörterbuch Deutsch-Italienisch” (ELDIT) at EURAC (Abel and Weber 2000) or dictionaries for minority languages like the German-Lower Sorbian Dictionary at the Sorbian Institute (Bartels 2010). Finally, special solutions exist where commercial dictionary writing systems are optimised for specific dictionary needs, as is the case with Pasadena, which has been specifically developed by IDM for the Oxford English Dictionary as a variant of the commercial software package DPS (Thompson 2005).
A dictionary writing system should never be viewed in isolation, but rather within the framework of a specific dictionary project and its whole environment, including related projects, long term perspectives, etc. A dictionary project can be divided into three work phases: planning, implementation and follow-up (see Svensén 2009: 398 ff., Atkins and Rundell 2008: 18 ff., 97 ff.). A series of aspects have to be considered for each phase, with a clear distinction between commercial projects that concentrate primarily on the market and academic projects that focus on recent research trends. During the planning phase, several aspects must be taken into account: the market-driven or research-oriented ‘demand’, users, contents, entry layout and design, distribution medium, dictionary layout, budget, time schedule, team, workflows, resources, software tools, etc. The implementation phase includes activities such as data selection, input, revisions, proofreading, (automated) typesetting, etc. Finally, issues such as maintenance and re-usability of data are crucial. Thus, the dictionary writing system can be seen as an aid or tool embedded in the entire dictionary writing process, while never losing its function for data synthesis, that is data input and editing.

While a dictionary writing system, being a dedicated system and more than a generic editor, basically supports dictionary compilation and dictionary entry editing, there is another type of software that supports lexicographers in dictionary making, namely corpus-query systems. These are frequently used for data analysis and selection (see Kilgarriff and Kosem, this volume). In recent years, corpus-query systems have become a standard tool in lexicographic work (Atkins and Rundell 2008). A corpus-query system can be used in addition to a dictionary writing system or be an integral part of it (see below).
3 Main characteristics of a dictionary writing system

The core function of a dictionary writing system is entry editing: nowadays dictionaries are written on computers and a dictionary writing system should first and foremost streamline entry writing. It should also be able to cope with the particular demands of complex dictionary writing projects (Atkins and Rundell 2008: 114).

As for dictionary writing, it is important to distinguish between three aspects, which we will briefly summarise here. The most crucial aspect is the content of the dictionary. Second, each dictionary has a specific configuration of its different components or a structured data model. We labelled this a specific “syntax” earlier, but it might also be termed the “dictionary grammar” or, using a more technical term, the document type definition (DTD). Third, there is the data presentation aspect, the formatting and style (see e.g. de Schryver and Joffe 2006: 41). These three aspects should be considered individually, as specific programs are best suited to work on each of them.

Data inputting and editing in dictionaries can happen in many different ways, each of which has advantages and drawbacks. It is, in principle, possible to enter the entire dictionary text in one sequence into any word-processing system. In this way, text is processed and stored linearly, exactly in the way it should appear in the final product. Until recently, both institutions with smaller projects, and large publishing houses have used this method, and to some extent they still do so today in specific circumstances (e.g. the Sorbian Institute works with informants who may be elderly people lacking IT skills and lexicographic expertise).

When using a word processing system, details such as font type and size can and must be selected directly. So, next to dictionary content, the form becomes an

important aspect. The advantages of such a data input system lie in the high degree of flexibility, as the editors are completely free to add or change almost anything anywhere. Among possible disadvantages, different types of information are not always explicitly marked and thus the final product is not searchable; automatic checking of consistency and conformity can also only be done in a limited way. In addition, the reusability of data is not straightforward. For example, it is not possible to search for geographical labels if they are not explicitly tagged as such, or to quickly and easily check if these labels are used in a consistent way (e.g. always using non-abbreviated forms like “American” etc.). Some drawbacks relating to form and content control or data reuse can be diminished, for example by consistently using uniform labels for single data categories (Svensén 2009: 421).

To spare editors the task of checking the order in which the data are stored, a lexical database can be used where the data are organised in records, each containing a specific series of different types of information (data categories such as definitions, examples, etc.). This allows easy filtering of the entire database according to any input field. As a consequence, users may detect connections in the data that would not be evident in a data set that was stored linearly. From the point of view of the dictionary producer, such a database has the advantage of generating a great variety of products based on one and the same material (Svensén 2009: 421). For example, it would be possible to easily filter out data regarding word spelling to produce a specific spelling dictionary starting from the database of a monolingual dictionary.

Another method of dictionary writing uses a mark-up language such as XML to input, organise and edit the data. As in a database, the different types of information are kept separate. However, the data is organised in a hierarchical structure, which is not the case with the database. The hierarchy sees the entire
dictionary as the first level, followed by the single dictionary entries, the senses, etc.

XML allows data to be stored both as a file and as a database. This is, in fact, a fairly common procedure in dictionary production, since it ensures quicker and more direct access to the data than other storage systems (Svensén 2009: 421).

Mark-up languages, such as the popular XML, allow electronic documents to be structured in a machine-readable way by adding information to the text in the form of tags, that is standardised labels. These tags may refer to the text and its features, or to the linguistic phenomena it contains. It is possible to define a basically unlimited number of tags for a document, and to define the relationships between the tags within the overall structure. Tagging covers document structure and content, but not appearance: its great advantage is that it is easy to change the appearance of a document without affecting its content (Svensén 2009: 49–50).

Different types of input software can be used when working with a mark-up language such as XML. Some dictionary projects use a generic XML editing tool such as Emacs, which can be adapted and used for lexicography. Figure 1 shows an extract from a possible template for a simple entry: the text is inserted by the lexicographer between an opening and a closing tag (in the example, an “x” is used as a place holder where the text can be entered).
Although they are efficient and popular programs, these generic tools do not necessarily meet the needs of complex dictionary projects, because they were not specifically designed for lexicographic work. For example, they are quite sensitive to errors made by the editors and often do not have very user-friendly interfaces.

Another option is to develop a new system based on an XML editor and adapt it to dictionary production (this can be commercial, freeware or open software, such as Oxygen, or Xmetal). Several major publishers have followed this approach, refining their programs over the years and collecting input from lexicographers and different projects. However, as specific off-the-shelf dictionary writing system packages have been marketed in the last years, publishers now tend to switch to these (see Atkins and Rundell 2008: 113–114).
Commercial dictionary writing systems are designed to manage the entire dictionary production workflow, from the first entry to the final product ready for publication in print or electronic format. They typically consist of three main components:

- a text editing interface, used by lexicographers to create and edit entries;
- a database, where data is stored;
- a set of administrative tools for project management and publication (see Atkins and Rundell 2008: 113–114).

From a conceptual point of view it is not always possible to separate these three components, as they are closely linked to each other in various phases of a dictionary project. Thus, it could be difficult to determine whether a certain feature is connected more to the administrative tools or the editing interface. In fact, the administrative tools can be configured so as to affect editing at the ‘front-end’ (for example, when entry templates are used). This affects entry operations and administration.

Nevertheless, the distinction is a useful starting point to describe some general requirements that a dictionary writing system should be able to meet; in this section the focus will be, not on the dictionary project with its different phases, but on the dictionary writing system itself. The overview provided in Sections 3.1 to 3.3 summarises the main characteristics of a dictionary writing system as described by Atkins and Rundell (2008: 113 ff.) and Svensén (2009: 415 ff.). Relevant webpages and literature published by the best-known commercial providers of dictionary writing systems are also taken into account.

3.1 The editing tool
The editing tool allows lexicographers to enter their text into predefined slots or spaces. A dictionary writing system usually offers one editing interface but several different ways of viewing the data. A typical screen in an editing tool may show different panes, usually presenting administrative functions and editing possibilities. Visualising the data may be as important as adding content (in order to proceed with corrections immediately).

Figure 2: An extract from a typical screen of an editing interface in a dictionary writing system (left pane: WYSIWYG view, right pane: tree-diagram view)

Different views are provided for a ‘what-you-see-is-what-you-get’ (WYSIWYG) view (or ‘preview mode’), and ‘tree-diagram’ view (Figure 2). Both views allow text
to be added or edited, and any change will automatically be stored centrally.

However, the tree diagram is usually the most widely-used editing interface. The WYSIWYG view gives a good idea of the look and feel of the final dictionary entry, but the tree view shows the structural elements of the entry (headword, word class marker, definitions, derived forms, etc.), while at the same time providing the slots where data can be entered.

A good dictionary writing system relieves the lexicographer of routine tasks and automates many ‘administrative’ procedures that had to be taken care of manually before the introduction of dictionary writing systems. The dictionary writing system allows only a limited number of values and character strings for certain fields (labels and indications of part-of-speech, word-class markers, grammar codes, register labels etc.), by using drop-down lists, for example. This helps to keep the entries consistent. Furthermore, some non-typographical structure indicators, for example commas that separate alternative meanings of a lemma, brackets around certain information types, a symbol introducing syntagmatic blocks, etc., are generated automatically. This further helps lexicographers who no longer have to worry about formal aspects (font, font style etc.), which were previously defined in detail by the style guides. Now they can focus on entering text and content into the relevant slots while the final output is generated by style sheets.

The style guide can be integrated into the dictionary writing system to make context-sensitive help available and accessible with a simple click. This is useful, for example, when lexicographers are uncertain about the rules to be applied. Furthermore, style guides can easily be updated according to the changes made during the implementation of a dictionary project, and re-issued to the editorial team. Style guides used to contain many rules concerning presentation and layout; nowadays they
offer detailed information on how to use the dictionary writing system and explain, for example, which types of data should be keyed into which fields.

A dictionary writing system also allows lexicographers to copy and paste text and to move fields or entire groups of fields, for example, to other parts of the dictionary. In some systems, lexicographers may create ‘templates’ for typical entry-types (e.g. a typical noun entry) or recurring parts of entries (e.g. collocation structures) that contain ready-made configurations of structural elements which can be used whenever needed.

While the lexicographer is entering data, the system checks that the syntax corresponds to the dictionary’s document type definition, which defines the elements that are part of the dictionary and their required sequence. A dictionary writing system therefore also has a function that validates entry structure: if text elements are inserted in the wrong order, the system alerts the user. For example, if a definition is inserted before the lemma, which always needs to appear first, the dictionary writing system warns the lexicographer of the mistake. Nevertheless, exceptions must be possible, as there may be occasions where this is desirable or necessary.

Fortunately, some complex and highly error-prone procedures can be handled automatically, thus relieving dictionary editors of these tasks. For example, reordering the senses of a polysemous word or adding a new sense in the middle of an entry might call for changes in other parts of the dictionary. Nowadays, it is the dictionary writing system that takes care of re-numbering the whole entry, as well as making the appropriate changes to the sense numbers in any cross-reference. In addition, the dictionary writing system alerts the lexicographer when a lemma is already present in the dictionary. In case of homonymy it requires a homonym number.
Real-time spellcheckers reduce the presence of typos. In addition, if the dictionary requires a restricted defining vocabulary, that is a list of words to be used in the definitions (e.g. the Longman Defining Vocabulary), a dictionary writing system will usually check the words used in definitions against the list of possibilities.

3.2 The database

Text entered and edited in the ‘front end’ of the dictionary writing system is stored in the dictionary’s database. Usually lexicographers do not work directly on the database to any great extent, but they can use it to run complex searches and filter the text with the help of the specific query language used by the dictionary writing system. It is possible, for example, to find all entries written or modified by an editor, example sentences containing specific patterns, entries including particular words, items having particular register labels, etc.

The database usually uses Unicode, an IT standard that assigns a unique and universal number to each character existing on any platform and in any program or language. In this way, all the characters of all written languages, including special characters such as those used in the International Phonetic Alphabet (IPA), can be universally recognised. This aspect is critical for the database and the editing system must support it.

Today’s dictionary writing systems are typically based on a server-client architecture: lexicographers work on computers that are connected to a server where all changes are stored centrally. It thus becomes possible to work via the Internet from any location. This feature is a cornerstone in the recent development of dictionary writing systems. In the 1990s lexicographers began working directly on computers

and it was in this period that the first generation of dictionary writing systems was created. At that time, staff usually still worked on the publisher’s premises, but in the second half of the decade a variety of technical developments (falling hardware prices, rising hardware capacity, e-mail communication, etc.) enabled collective work to be conducted remotely. From 2000 onwards, fast 24/7 Internet connections have been available. This has supported the latest developments in computational lexicography, where both corpus and dictionary text in progress are stored on a server that editors can access online (from any location) through a corpus-query system and a dictionary writing system. Thus, distributed work is easy to manage via a dictionary writing system and other equivalent environments.

Another important issue is the import and export from and into different formats: the dictionary writing system should allow the export of entire dictionaries or parts of dictionaries in formats such as XML, RTF, PDF or HTML. This is essential when selling dictionary material or producing electronic dictionaries. At the same time, it should also be possible to import other material into a dictionary in progress.5

3.3 Administrative tools

As well as providing an environment for dictionary writing, editing and storage, dictionary writing systems also offer ‘housekeeping’ tools that help to manage large projects. A ‘workflow manager’ may allocate a batch of entries to be compiled or edited by a particular lexicographer. At any time the system keeps a record of who is working on which entries. Any delay in the planned work schedule will be automatically brought to the attention of the project manager by the dictionary writing system. Complete and exported batches of work can be imported back into the
database, so that everyone working on the dictionary always has access to the latest version. The dictionary writing system allows project managers to keep track of progress against the working schedule and budget. Senior staff can also continually monitor the dictionary text as it develops and give feedback to the lexicographers.

The system should ensure that no more than one person at a time can edit an entry. Individual fields in the database should also be ‘lockable’. This is useful, for example, when fields of the same type, such as all the pronunciation or etymology fields, are to be edited only by specialists, while other team members take care of the remaining parts of the entries. In addition, the program should have different levels of authorisation so that, for example, freelance staff are allowed to edit only the dictionary project they have been hired for and not other products.

A useful feature is version control, that is a facility that makes it possible to track changes and, for example, to provide access to data that have been modified at a specific moment in time. A further aspect is scriptability, that is the possibility of automating processes through scripts. Mass data update allows, for instance, the same correction to be made to a whole set of documents with a single operation. Batch merges allow large amounts of data to be added to the database quickly and easily.

A series of further issues should be taken into consideration when choosing a dictionary writing system. These are mainly technical aspects regarding service (e.g. hosting, maintenance, backup, setup, training), the operating system (e.g. Windows, Linux, Mac), prices and licences (e.g. commercial or academic licences).

4 Off-the-shelf versus in-house software
Every lexicographic project uses some kind of dictionary writing system. Sometimes these tools are written in-house, such as an XML-editor customised for one or more dictionary projects; in other cases, off-the-shelf dictionary writing system packages are preferred. The best-known ones have already been mentioned above. Some scholars (e.g. de Schryver 2011) argue that developing an in-house tool is a version of reinventing the wheel, in view of the fact that excellent, highly sophisticated dictionary writing systems already exist. Before good off-the-shelf dictionary writing system packages became available, it was normal or rather necessary for dictionary publishers to develop their own tools in-house and/or adapt generic XML editors. The same is true for corpus-query systems. In recent years, more and more dictionary publishers have ‘walked the proprietary route’ by switching to off-the-shelf products. All the main English language publishers have now done so, including well-established ones such as Merriam Webster in the US. This works well even for very complex projects, such as the OED, which opted for a special solution on the basis of a proprietary system (Pasadena). One argument in favour of proprietary solutions is that a range of different users will contribute their own suggestions and requests for improvements, many of which will eventually be incorporated into the standard packages. So a tool or a package used by many publishers means that they will collectively contribute to improving it, thus ensuring more rapid development. In this way, both dictionary compilers and developers may benefit from the new situation (de Schryver 2011).

While many publishers have shifted to the proprietary track, several commercial and academic institutions are still using, developing and improving their own systems. In some cases such dictionary writing system tools are created within an open source and/or freely available development framework (e.g. DEB, Jibiki,

EELex), although this is not always the case (e.g. Duden “Wissensnetz deutsche Sprache”).

The decision to develop an in-house tool may be due to the fact that, especially in long-term projects, publishers keep working with their home-grown systems and adapt them progressively to their own requirements. In addition, they may integrate them with a series of further tools tailored for a specific project or a growing number of projects. One difficulty is that, if in-house systems are used, the partly free, partly commercial additional tools must not only be specifically tailored but also need to be able to work together. If, however, commercial products composed of several components are used, the companies have to ensure that these components interoperate smoothly. We will mention only two examples of such home-grown systems being developed and constantly expanded within long-term projects: the elexiko and DWDS initiatives, both under development in a German-speaking academic context (the former at the Institut für Deutsche Sprache in Mannheim, the latter at the Berlin-Brandenburgische Akademie der Wissenschaften in Berlin).

elexiko (Müller-Spitzer 2011, Müller-Spitzer and Möhrs 2008) aims to create an electronic corpus-based dictionary of contemporary German. It is included in the Online Vocabulary System of German (OWID), with the intention of developing a network of interrelated, but independent lexicographic products that share some of their data modelling features. A tailor-made, fine-grained XML-DTD containing a high number of elements was created for elexiko, and the structure was then implemented in an XML-editor that is used for entering the data (XMLMetal, a commercial product). Dictionary entries are stored in an ORACLE database and lexicographers can perform queries throughout the entire database with the help of a specific interface. In addition, a corpus-query system is used to search within the in-
house corpus COSMAS that allows lexicographers to analyse co-occurrences. Within the lexicographic working environment some workflows can be monitored, for example, those entries that are in progress. In order to support editorial work, a ‘reference manager’ was developed that allows in- and outgoing references in dictionary entries to be crosschecked. This system of coordinated products works well, even though some improvements have been suggested, such as an online preview and the use of templates for some entry parts (Müller-Spitzer and Möhrs 2008).

The DWDS is another example of a long-term lexicographic project that uses an in-house system for lexicographic work (Klein and Geyken 2010). It combines a series of tailor-made tools, such as an XML-Editor (Oxygen) for data input that has a direct interface to an administration tool for version control (currently used tool: Subversion) so that all lexicographers – including those based in remote locations – can access the central repository and check lexicographic entries in and out while the project manager keeps track of all changes. Corpus data is accessed via another in-house corpus-query system that is used together with other tools. A serious challenge posed by such large lexicographic projects with a substantial number of team members, some of whom work from different locations (as in the case of DWDS), is providing a tool to support good workflow management.

In some cases, an in-house system is developed due to the need to provide an environment for lexical database management as well as semantic networks and ontologies: that is the system should be used not only for the production of dictionaries, but also for encyclopaedias, Wordnets or similar products. Two examples deserve particular mention. First, the Dictionary Editing and Browsing (DEB) platform is an open source and freely available development framework (Horák and...

Rambousek 2007, Horák et al 2008, Vossen et al. 2007). Within this platform, which works on a client-server basis, a range of dictionary applications can be used. The tools include a dictionary browser (DEBDict), a dictionary writing system for the development of the Czech Lexical Database (PRALED) that will be linked to a corpus query system (Manatee/Bonito corpus manager and the Word Sketch Engine), a Wordnet editor and browser (DEBVisDic) including functionalities such as synset preview, remote teamwork capabilities and an administration interface. On the basis of the Wordnet editor, a specific application has been developed for a project called Cornetto, designed to build a lexical semantic database for Dutch by combining and interconnecting two electronic dictionaries (the “Referentie Bestand Nederlands” and the “Dutch Wordnet”) and thus containing, inter alia, semantic relations, combinatorial relations and an ontology.

The German publisher Duden (Alexa et al. 2002) felt the need to replace the editorial system “Reda” they had been using for more than 20 years, where each dictionary was compiled separately, by a new tool for the administration of language data that offers a new working environment for lexicographers. This innovation was based on a formal explicit representation of all Duden dictionary entries (integrating lexical and ontological information) and thus reduces the redundancy and increases the efficient maintenance of the dictionary data within a single data pool. It was also intended to support reusability for both print and electronic products, as well as the development of language technology applications. The current result is the “Wissensnetz deutsche Sprache” (German Language Knowledge Network) which contains dictionary data that are semantically interlinked through a complex system of multiple and different underlying ontologies. Dictionary entries within the “Wissensnetz” have a particular structure: an entry not only contains lemma- and
sense-related data, but also a concept-related level. The sense level represents the bridge between lemma and concepts, as each sense can be linked to a concept through an explicit semantic relation (e.g. synonymy, hyperonymy, meronymy).

There are many good reasons for choosing one solution rather than another. The basic consideration when choosing a dictionary writing system is the need or needs that it is expected to meet, and whether the whole lexicographic environment can be managed by an off-the-shelf system or not. Obviously, the kind of financial investments that are possible or foreseen within a project are also an issue. It may be argued (e.g. de Schryver 2011: 647-648) that sometimes the series of reasons given for the development of an in-house system is not in fact really decisive. Many of these aspects, such as the need for a clear overview of complex articles, user-friendliness or a system that can handle a very complex document type definition, are already standard features of any off-the-shelf tool.

There is a good deal of convergence between the various commercially available dictionary writing systems with respect to their main aspects and components (such as a text-editing interface with real-time preview, administrative tools, etc.), as it is inevitable that the positive innovations introduced by one supplier will soon be adopted by other providers as well. As a consequence, most commercial products are of good quality. Some have particular strengths. For example, ABBY (Rylova 2010, Kuzmina and Rylova 2010) has very good tools for search queries, offering a filter with tick boxes that can be easily used without knowing any special query language, while this may be necessary for other tools. TLex has an integrated corpus-query system and a ruler tool that helps ensure a balanced treatment of entries and space allocation (article length), a feature that is very useful for print dictionaries. IDM is particularly strong in its workflow and project management functions,
although the system is not entirely intuitive. In order to take advantage of the large number of functions available, a certain amount of advanced knowledge is required of the user.

Differences between the systems may also relate to quite basic aspects, such as the price and availability of academic licences (e.g. TLex has quite affordable prices for its academic licences). Such aspects may be decisive when choosing one tool over the other.

If and which new features are going to be offered or required by dictionary publishers will become clear in future. To name just one development, recent discussions focus on a supporting tool that could “activate ‘layers of restricted defining vocabulary’” (de Schryver and Prinsloo 2011: 7). The underlying idea is that “one would be able to set a certain level or ‘age range’ for a particular dictionary under compilation and that the dictionary writing system would then, during the writing of the dictionary, automatically flag or colour-code those words not belonging to the approved defining vocabulary for that level” (de Schryver and Prinsloo 2011: 7-8).

5 Combination and integration of tools

Today, lexicographers can use a wide range of tools for different phases of dictionary production. Some may be part of a dictionary writing system package, while others are stand-alone product. In any case, they should ideally interoperate smoothly with dictionary writing systems. This is especially true for corpus-query systems, which are among the most important tools for lexicographers.
Dictionary writing systems share their main features. However, some dictionary writing system suppliers offer additional tools that may be of interest for commercial publishers in a competitive market and, thus, be decisive when choosing one product or another. We will list some of these additional tools in a short excursus, before moving on to dictionary writing systems in a stricter sense.

Dictionary publishers are nowadays facing increasing competition from free online dictionaries (see Lannoy 2010a). While established dictionary publishers hire people to create content, other providers recycle and re-use this content, making it available for free. This obviously reduces the publishers’ potential market share (see Adam Kilgarriff’s report on eLEX 2009). “Free” is therefore discussed as a “leading [business] model for online dictionaries” and IDM has “entered the online dictionary market to provide publishers with the means to compete with the most successful free websites that deliver well indexed content very quickly, have web pages optimised for search engines, multilingual interfaces and put extreme care in the data” (Lannoy 2010a: 174). Free dictionary websites, which undoubtedly have a strong attraction power, must be indexed and optimised for search engines. This can enhance the visibility of a product. The quality of the content helps in “building user loyalty and depth of visit on the websites” (Lannoy 2010a: 174). This may establish the brand of free dictionary websites and enhance the upsales potential for other products.

Dictionary writing system suppliers such as IDM have recently been discussing the issue of user-generated data. Accepting user-generated content means reverting to a democratic editorial process and an open critical review of the product. It represents, in fact, an essential source of improvement and is therefore well worth considering. From the point of view of commercial publishers, an online dictionary

should be as comprehensive as possible, so that it is privileged in web searches (as it is subject to SEO [Search Engine Optimisation] principles) (Lannoy 2010b).

In this context, it is interesting to note that in recent years several authors have considered aspects of collaborative dictionary writing and its methods, in which the boundaries between the lexicographer and the user have become blurred (Abel 2006). An example of this is “fuzzy simultaneous feedback”, a concept that, in its electronic adaptation, refers to a type of intelligent and adaptive dictionary in which customisation is performed online in real time (de Schryver, in print).

A variety of tools is usually provided as part of a dictionary writing system package, ranging from entry compilation to publication. Corpus-query systems (e.g. Sketch Engine, CQP, DWDS/DDC, COSMAS etc.) are often presented as a separate tool, being considered necessary to analyse future dictionary data. In an ideal world, in order to streamline the whole process, it would also be desirable to have the corpus-query system integrated. In this context, a historical note may be interesting: the first true off-the-shelf tool was Gestorlex, developed by the software house Textware for the Danish publisher Gyldendal in the late 1980s, and later adopted by Longman. It was an integrated dictionary writing system and corpus-query system and, despite being a good system, it had the fatal flaw of running on an operating system (OS/2) that could not compete with Windows. To the best of my knowledge, the only off-the-shelf dictionary writing system package which currently has an integrated corpus-query system is TLex, which offers basic functions such as the production of concordances based on corpus queries, that is lists of keywords in context. Word sketches, however, cannot be produced by TLex: these are lexical profiles, or rather corpus-based outputs of the grammatical and collocational behaviour of a word

(Kilgarriff et al. 2004: 105 ff.), which “improve on standard collocation lists by using a grammar and a parser to find collocates in specific grammatical relations” (Atkins and Rundell 2008: 109). Rather than a single list built on grammar, they produce a list of subjects, one of objects, and so forth. The word sketches function is offered by the corpus-query system Sketch Engine that is used in many dictionary projects (e.g. in publishing houses such as Oxford University Press, Cambridge University Press, Collins, Macmillan, INL, Cornelsen) (Kilgarriff et al. 2010: 412; see also Kilgarriff and Kosem, this volume).

Some may thus opt for maximum compatibility between the dictionary writing system and corpus-query system and not for integration of the two tools. An example where this works quite smoothly is the NEID/DANTE project, where the IDM-dictionary writing system and the Sketch Engine are closely connected. This obviously requires some collaboration between the suppliers. DANTE (Database of Analysed Texts in English) is a lexicographic project that aims at building a fine-grained lexical database on the basis of the analysis of corpus data (cf. Rundell, this volume). It is targeted at lexicographers as a basis for the compilation of a concrete dictionary, but it also has potential for other uses in language technology. In this case, the database is being developed as the basis for the elaboration of a New English-Irish Dictionary8 (NEID) Atkins et al 2010, Rundell and Kilgarriff 2011). In the DANTE database, all word senses, constructions and collocations are illustrated with one or more unedited corpus examples. Lexicographers were asked to extract example sentences from the corpus system and feed them into the dictionary editing system, using a fiddly standard copy-and-paste function. Later, a button for ‘one-click copying’ was added at the end of a concordance line to allow quick and smooth
selection of the sentences, at the same time pasting them directly into the correct field in the dictionary writing system (Rundell and Kilgarriff 2011).

The Tickbox Lexicography (TBL) option is a further development towards making the lexicographer’s life easier. TBL models and streamlines the process of extracting data from a corpus and inserting it into a dictionary writing system (Kilgarriff et al. 2010: 413; Kilgarriff and Kosem, this volume). Thus, it is no longer necessary to mark single concordances: by clicking on tickboxes, entire chunks of data (e.g. a collocation type plus example) can be transferred into the correct fields of the dictionary writing system in one go. Tickbox Lexicography includes another function, that is GDEX (“Good Dictionary Example eXtractor”) (Kilgarriff et al. 2008), an algorithm targeted at identifying sentences that are likely to fulfil the criteria of being a “good” example. With this function, lexicographers are not faced with long lists of concordances, as the system makes a choice of six corpus example sentences (by default, but this can be changed according to the project) (Rundell and Kilgarriff 2011; Kilgarriff et al 2010: 413). This function should spare the lexicographer from the demanding and time-consuming scanning of thousands of concordance lines, a task that has been called a “new form of drudgery for the lexicographer” (Rundell and Kilgarriff 2011: 260).

Other projects have also tried to make data transfer from the corpus to the dictionary writing system more efficient. To mention only one, elexiko has a function where example sentences from the COSMAS corpus can be inserted into the editing software with a drag-and-drop-function.

In conclusion, it is important to consider carefully which functions and features are really essential for the dictionary project that is to be started or further developed before choosing a dictionary writing system. It is in this phase that the
choice between a known working environment and a new system has to be made. It is extremely important that project leaders make this decision only after intense discussion with lexicographers and IT experts, so as to make sure that all the advantages and disadvantages are well weighed against each other.

The questions to be asked include costs, time constraints, and issues of quality, quantity and distribution; for commercial products, all these aspects are strongly market driven. In addition, the fierce competition from a growing number of resources that are freely available online must be faced. Finally, the functions and features of a dictionary writing system, as described earlier, must all be considered separately, including the modules that the dictionary writing system is expected to include or to be perfectly compatible with (e.g. workflow management, proofreading, data conversion for any electronic application, online publication, mobile devices or CD-Rom/DVD, data searching in the whole repository, hosting). Practical issues such as in-house and/or external helpdesk support should also be considered.

6 Dictionary writing systems for multifunctional databases

A new challenge for today’s dictionary writing systems is that they are increasingly used not just to write single dictionaries but to build databases that will serve as the basis for many dictionaries. Atkins and Rundell (2008: 98) describe “the ideal way to compile a corpus-based dictionary from scratch” as a two-fold process, as we can generally distinguish between analysis and synthesis. It could even be a three-fold process, if we include translation for bilingual dictionaries as well. Lexicographers compile a lexical database on the basis of corpus evidence; generally speaking database entries reflect dictionary entries, but they are much more detailed. For
example, they contain a huge range of senses per headword, a rich selection of collocations, example sentences etc. (see Atkins and Rundell 2008: 97 ff.), providing an excellent foundation for the next steps. For bilingual dictionaries, this means first and foremost translating the database; elsewhere, it means editing the entries and compiling a dictionary. This approach has been applied, for example in the NEID/DANTE project (Atkins and Rundell 2008), but also in other contexts where huge lexical databases are being developed. These include the lexical database of the Czech language in the 21st Century (“Lexikon 21”, Rangelova and Králík 2007), the database for the Dutch language (“The Referentiebestand Nederlands” RBN, Van der Vliet 2007), and the Online Vocabulary System of German (OWID) (Müller-Spitzer and Möhrs 2008), to name just a few.

In these and other contexts, huge multifunctional lexicographic databases are being built as a basis that may meet the “different types of needs of different types of users in different types of situations” (Spohr: forthcoming) – in terms of the modern theory of lexicographic functions (cf. Bergenholtz and Tarp 2002) – and thus steer the creation of specific dictionaries.

This touches on another issue, which is currently quite relevant in the context of dictionary writing systems. It is an idea introduced or further developed in the context of function theory but also beyond: Spohr (forthcoming) introduced the concept of a “pluri-monofunctional lexicographic tool”, a tool “that is capable of deriving multiple monofunctional dictionaries”. Gouws (2006: 53) describes the concept of a “Mutterwörterbuch” (“mother dictionary”) as a huge virtual dictionary that contains different dictionaries as subtexts (where the single items are marked so that it is possible to derive specific dictionaries). There are also some earlier, and to a certain extent related, ideas: the idea of multifunctional linguistic databases with
multiple uses and multifunctional dictionaries was already of topical interest in the
1980s. Multifunctionality, as well as reusability, were discussed at two levels: human
users and NLP applications (see Kruyt 2003 for an overview).10 The issue of
multifunctionality has also been raised in the context of specialised languages, in
particular in the context of the creation of terminology databases. Martin and van der
Vliet (2003: 338) describe a useful multi-purpose or multifunctional, and multi-user
approach to a database from which specific dictionaries or front-end databases can be
derived on the basis of different users and their needs (Martin 2000). Atkins (1996:
531) presented an early idea of a “virtual dictionary” to be derived from a “real”
database within the frame of a “multilingual hypertextual lexical resource”.

Lexical databases are not always compiled from scratch as in the DANTE
case; they can be built on the basis of pre-existing dictionaries. For example, the
Duden “Wissensnetz” includes almost the entire lexicographic material owned by the
publisher. In general, there is a trend towards change in the whole lexicographic
process, as lexicographers become more and more engaged in adapting and enriching
existing resources and targeting parts of resources at different user groups. In other
words, lexicographic work is now largely becoming an activity that builds on existing
material.

When creating lexicographic databases, it is preferable to work on the basis of
models for the representation of different monofunctional dictionaries in order to
assure quality (as proposed, for example, by Spohr, forthcoming). Spohr presents a
multi-layer architecture that deals dynamically with user needs, thanks to the
introduction of a layer that contains “function-specific information for each
lexicographic indication in-between the user interface and the lexicographic database.
This layer acts as a kind of filter that lets through only those indications which are
believed to be relevant to certain types of users in particular types of situations (on the basis of Tarp 2008”). All indications are presented in a form that is appropriate for particular user types, for example by using labels in different languages and varying degrees of specialised terminology. As function-related specifications are separate from dictionary content, this additional layer can dynamically generate different monofunctional views of the lexical data in the database, on the basis of different combinations of situation and user types.

In a further step, not only user groups but individual users are considered. There is also a recognition of intelligent and adaptive media (de Schryver 2010, Trap Jensen 2010, Müller-Spitzer and Möhrs 2008, Gamper and Knapp 2000, 2002), although Trap Jensen (2010: 1134) stresses that “[t]he notion of dictionary customisation is still in its infancy”. Thus, more research is needed into dictionary usability. It seems that dictionaries and dictionary entries will come to be considered more and more as dynamic entities (Spohr: forthcoming) and that dictionary writing systems will have to be able to deal with that. In many cases a “dictionary writing system” is, strictly speaking, rather a “database writing system”. In any case, it is a tool that allows lexicographers to create and manage comprehensive and increasingly dynamic resources.

7 Conclusion

Dictionary writing systems are quickly developing away from being pure editing systems and/or authoring tools towards becoming applications that include a huge range of components and modules with a great number of functions. The dictionary writing system in the sense of a mere editing aid is just a small part. This is certainly
true for the big commercial products that support the workflow and routines of dictionary projects from the planning and design phase up to the publication of the final product. They have to be as flexible as possible to the different but recurring requirements of the various target groups. Commercial off-the-shelf tools mainly serve commercial publishers and try to respond to the market. This means that they have to focus their products more and more on other media alongside printed dictionaries: electronic media, online dictionaries, CD-ROM/DVDs, and mobile phones. Remaining competitive with non-commercial entities that publish freely accessible online dictionaries is another important issue for the dictionary market.

In-house products support lexicographers in the same way as off-the-shelf software, for example, by following style guidelines in a coherent way. Essentially, they fulfil the same functions as their larger commercial siblings. Sometimes they are tailor-made applications to be used for single or smaller projects, sometimes for long-term projects and huge teams of workers. Close cooperation with a team of IT experts is always necessary, even when a commercial off-the-shelf tool has been chosen.

On the whole, when dealing with dictionary writing systems it is evident that the recent developments indicate a fundamental change in lexicographic processes and the lexicographic workplace. Processes are characterised by growing automation and the work of the lexicographers is changing: for example, more often than not, lexicographers compile lexicographic databases and update existing material. User expectations are being discussed intensively, too. End user participation in the production of dictionaries, for example, represents a significant and ongoing debate (user-generated content and/or collaborative dictionary writing, user customisation and adaptive user views etc.). Electronic media are increasingly becoming the centre of attention and will raise new questions about presentation and visualisation in all
phases of dictionary writing (for visualisation of linguistic information, see Culy and Lyding 2010, Rohrdantz et al. 2010).

The dictionary writing system plays a central role in the whole process of dictionary production: it is developing towards an increasingly versatile, multifunctional ‘all in one’ tool that works as a dashboard from where a series of processes and tasks can be controlled, managed and implemented.

References

Dictionaries and websites

ABBY Lingvo Content – http://www.abbyy.com
DWDS – www.dwds.de
ELDIT – http://www.eurac.edu/eldit
German-Lower Sorbian Dictionary – http://www.dolnoserbski.de/dnw/index.htm
iLEX – http://www.emp.dk
Intelligent Views – http://www.i-views.de


(online version of the sixth edition: http://www.archive.org/stream/dictionaryofengl02johnuoft#page/n37/mode/2up)

Leo – http://dict.leo.org/

Linguee – http://www.linguee.de/


OWID – http://www.owid.de/

Oxygen – http://www.oxygenxml.com/


Subversion – http://subversion.apache.org/

Termania Portal – http://www.termania.net/

TLex – http://tshwanedje.com

Xmetal – http://na.justsystems.com

XMLMind – http://www.xmlmind.com/

XMLSpy – http://www.altova.com/xml-editor/


(links checked on 12/06/2011)

Other references


Acknowledgements

I am grateful for many interesting and fruitful discussions and mail correspondence in the course of preparing this chapter. My particular thanks go to (in alphabetical order): Melina Alexa, Hauke Bartels, Model Benedikt, Henning Bergholtz, Philippe Climent, Chris Culy, Gilles-Maurice de Schryver, Henrik Dittmann, Birgit Eickhoff, Angelika Haller-Wolf, Alexander Geyken, Holger Hvelplund, Adam Kilgarriff, Annette Klosa, Margit Langements, Verena Lyding, Christine Möhrs, Carolin Müller-Spitzer, Michael Rundell, Dennis Spohr, Antje Tölpel, Ülle Viks. My special thanks, however, go to Elena Chiocchetti and Peter Farbridge for proofreading the text in English.

Notes

1 The huge field of well-known, sometimes even quite large and popular online dictionary products that are neither academic nor commercial are not discussed in this article, although they represent an important segment that would be worth analysing more in detail (see Nesi this volume). Such dictionaries or lexical environments may include for example Leo, Linguee, yourdictionary, etc. Within this chapter we consider only products for which scientific literature or documentation on the dictionary writing system is available.

2 A dictionary writing system should not be confused with desktop publishing (DTP) software (de Schryver and Joffe 2006: 41). These are WYSIWYG layout programs used on personal computers that produce print-ready documents (e.g. Adobe InDesign). A DTP can be added to or supported by a dictionary writing system.

3 In a DTD the tags, that is the standardised labels for text characteristics or linguistic phenomena, as well as the structure of a document, can be defined. However, there are more powerful means of
defining an XML document. Nowadays an XML schema is frequently used, an advantage being, for example, that the type of information allowed in a document can also be defined and validated. As this chapter is not targeted at computer experts, we will not go into further detail here.

4 In this example from the dictionary project “Cornelsen Schulwörterbuch ¡Apúntate!” IDM’s DPS is used as the dictionary writing system.

5 A database usually uses XML and DTD, an XML schema or its own formats based on XML, and works on the basis of ORACLE, MSSQL or Postgres databases.

6 Further research on specific user needs regarding (graded) dictionary definitions is needed. The findings of the study on dictionary definitions compiled for different age groups reported by de Schryver and Prinsloo (2011: 26) indicate that the definitions in the dictionaries examined are generally too difficult for the intended age groups.


8 The New English-Irish Dictionary is being developed for Foras na Gaeilge, Dublin, which is the official Irish government body responsible for the promotion of the Irish language (http://www.forasnagaeilge.ie/).

9 OWID could be described as a lexical database “as it contains a huge amount of lexical data and it is not designed for a specific user type and specific situations” (Müller-Spitzer and Möhrs 2008: 43).

10 The idea of multifunctional linguistic databases is discussed with reference to Zampolli 1987, and that of multifunctional dictionaries with reference to Zimmermann 1983 (Kruyt 2003: 194).