MULTIMEDIA REVIEW

Terminology Management Systems for the development of (specialised) dictionaries: a focus on WordSmith Tools and Termstar XV

Mike Scott, 2011
<http://www.lexically.net/downloads/version6/HTML/index.html>
Star Servicios Lingüísticos, 2011
<http://www.star-spain.com/es/tecnologia/term.php>

Reviewed by Nuria Edo Marzá
nedo@ang.uji.es
Universitat Jaume I, Spain

This review aims to focus on the analysis of the technical possibilities offered by two of the main Terminology Management Systems (TMSs) – the corpus-query program *WordSmith Tools* (currently in its 6.0 version) and the multilingual terminological database *TermStar XV*. Subsequently, they will be compared with other similar systems that are currently available, as well as in terms of their potential for the development of (specialised) dictionaries.

Terminology management includes a series of activities ranging from terminology extraction to the creation and validation of terminology, including the classification, retrieval and exchange of such terminology (Mesa-Lao 2008). Therefore, being aware of the most appropriate TMS according to one's particular needs is paramount for three main types of users: terminologists, translators and authors. In this review, our attention will be focused on terminologists' needs. Consequently, the software tools or TMSs analysed here were chosen because of their potential in the two main stages generally involved in the dictionary-making process: 1) term extraction and term in-corpus analysis, and 2) data processing, management and storage.

For the first main stage, a closer look will be taken at *WordSmith Tools* (WST), *MonoConc Pro* and *AntConc*, some of the more readily available and reasonably priced packages for working with corpora, with the aim of contrasting the different options they provide. Then, for the second big stage mentioned, *TermStar XV* will be analysed

and compared with other similar software systems such as *AnyLexic*, *SDL MultiTerm*, *Multitrans TermBase*, *Déjà Vu X Termbase* and *Gesterm*.

The main aspects of these software tools that will be reviewed will be mostly those related with the possibilities offered as regards their functionality and management, their potential for the creation of terminological cards and for the retrieval of specific information (specific searches or data filters), the management of export and import tasks, and the user-friendliness of the environment, among others.

The first main stage in the development of any specialised dictionary, i.e. that of term extraction and term in-corpus analysis, is normally carried out by means of corpus-query programs or software concordance programs like *WordSmith Tools*. WST is an integrated suite of programs for looking at how words behave in texts (Scott 2011), apart from providing varied corpus counts which may be useful for different purposes. Hence, WST is a corpus-query program capable of processing large numbers of texts with the aim of identifying characters or chains of characters that could be potential terms. Term extraction is thus "an operation which takes a document as input and produces a list of term candidates as output" (Streiter et al. 2003: 2). Those terms are then analysed in context in order to verify or revoke their "term status" in real use.

The software concordance program *WordSmith Tools* is a collection of three programs or applications: Wordlist, Concord and KeyWords. With Wordlist the user can create frequency and alphabetical lists and even a combination of the two; it also reveals relevant statistical and numerical data, and different wordlists can be compared. Furthermore, Wordlist offers the possibility of easily showing how many of our texts each word occurred in. This is important because frequency does not always imply importance or relevance in discourse – it may simply be due to some author's idiosyncrasies – and this is easily noticeable if we check that a top frequency word is top-frequent only in a given text from the corpus. Wordlist also allows the user to lemmatise and to make a word list with pairs or triplets of words (n-grams), for which he/she will first need to compute an index file.

Concord is the pure concordance application of WST and thus the one in charge of generating lists of concordance lines (also known as *Key Word in Context* – KWIC), apart from automatically identifying words that appear jointly a given number of times:

collocations, clusters (groups) and patterns (structures). For instance, Concord enables researchers to find recurring clusters, i.e. multi-word units, from within the entire corpus. It also allows users to perform multi-word queries and provides the plots (or distributions across the corpus) of the lexical units analysed. The Concord application Concordance also generates polylexical lists in which the degree of interdependence or the degree of the link or relation between words is established through the measure "Mutual Information". Concord also has *sort* functions that allow users to sort concordance lines in several ways with respect to the search word, which can provide insights on word uses and senses.

Finally, the Keywords application retrieves a series of key words from the corpus and this keyness is established by determining those words from the corpus which occur unusually frequently in comparison with some kind of reference corpus. Collocates, plots, patterns and clusters can also be analysed with Keywords.

Nonetheless, apart from WST, nowadays there are many other alternative corpus query programs with similar applications and possibilities. *AntConc* and *MonoConc Pro* are just a couple of examples from the many software packages currently available to carry out corpus-based research. All of them offer the basic functions expected of any concordance software program: frequency and KWIC lists generation, clusters and collocates retrieval, concordance plots generation, different sorting possibilities, and so forth. The differences have mainly to do with the user-friendliness of the programs, the displays of data offered and their specific ability to carry out certain tasks.

In general, the three programs mentioned here for term extraction and term in-corpus analysis are valid and reliable, even when WST seems to show a greater potential with respect to the other two in terms of the number of functions it is able to perform. *MonoConc Pro* is a fast concordance program with a really good user-interface. Apart from the intuitive nature of its interface, *MonoConc Pro* also presents a feature not shared by the other two that makes it particularly attractive for researchers, namely: the split screen which allows users to expand the context of an entry line when highlighting it, the fuller context being displayed in the upper window. As Reppen (2001) states, in WST, the entire display must be expanded or reduced, so the context is expanded for all of the entries being viewed rather than for a single highlighted entry. *MonoConc*

Pro is thus easy to use (in fact it is the program that is generally used nowadays for language learning purposes) but it also comes with a range of powerful features such as context search, regular expression search, part-of-speech tag search, collocations and corpus comparison. Its simplified version, *MonoConc Easy*, however, has many of the features of *MonoConc Pro*, but does not include some of the advanced features such as the advanced sort and corpus comparison. *MonoConc Pro* is known for its intuitive interface but *MonoConc Easy* is even easier to use, as its name indicates, and is therefore a good choice for less experienced concordance users. It is thus very useful for general concordancing and for use in computer labs, but it is probably not the best option for terminologists and terminographers, since the program is targeted more towards student and teaching use than for in-depth, professional corpus research.

Therefore, the main advantage of *MonoConc Pro* over *WordSmith Tools* is that it is much easier to use. For example, when *MonoConc Pro* is launched, a clear easy-to-use screen appears with a bar across the top, providing the options available. The screens are clearer, and since they resemble the screens of many word processing programs, users, especially those starting out in corpus analysis, may feel more comfortable. Nevertheless, when *WordSmith* is launched there are many screens that appear, and it may be more time-consuming and a bit challenging until the user becomes familiar with the program.

However, in addition to the functions that these programs have in common, *WordSmith* is able to perform a number of useful tasks that *MonoConc Pro* and *AntConc* are not, apart from providing a greater range of features and possibilities in terms of establishing and working with personalised settings:

> For example, *WordSmith* can provide information about the distribution of a feature in a single text or across texts. Distributions are shown with a graph that plots the occurrences of the target item in the text or corpus […]. The distribution of a particular lexical or grammatical feature across a text or series of texts can provide interesting information about the text structure and also about how the feature functions across various texts (Reppen 2001: 34).

To sum up, all three programs – *WST, MonoConc Pro* and *AntConc* – include many of the same features, such as the ability to create word lists (in both alphabetical order and order of frequency), generate concordance output and give collocation information. In addition, they can all easily handle large corpora and work with either tagged or untagged texts. However, the three programs have different strengths: *Antconc* and

MonoConc Pro have the added advantage of being free software packages that are quite easy to manage and conceptually, for users who feel less comfortable with computers, *AntConc*'s and *MonoConc Pro's* interfaces are far more user-friendly than that of *WordSmith.* In fact, *AntConc* is probably the simplest to use and performs the basic functions, but has the shortcoming of not offering many ways of saving the results. However, despite the fact that *WST* may seem less user-friendly at first sight, it is also easy to use once you have spent a little time with it and its potential – in terms of the number of features offered and options available – is much bigger than that of the other two programs. Obviously, it is the terminographers themselves who have to make the final choice as to which one best suits their needs but, in general, WST would be the best choice for terminologists and for the more professional researcher and terminologist.

Austermühl (2001: 102) defined terminology management as 'the documentation, storage, manipulation and presentation' of terminology, which could at the same time be defined as the specific vocabulary of a specialised area. Accordingly, terminologists grant a great deal of importance to the necessary creation of multilingual terminological databases, also understood here as TMSs. Such databases for managing and storing terminology are mainly assessed on the basis of their compatibility with various languages and alphabets, on the possibility of carrying out global changes, and on the flexibility of management tasks. Therefore, the very definition of terminological database may help us understand its importance for terminological tasks:

> a computerised storage system of lexical elements that are structured according to a series of criteria (alphabetical order, conceptual hierarchy, etc.), according to the users and according to the purpose of the terminological compilation, which must be flexible and accurately reflect the relationships between the hierarchies of information, making the loading of all the pertinent data and their rapid retrieval with varied possibilities of presentation feasible (Gómez González-Jover and Vargas Sierra 2003).

It is a fact that the easiest way to store terminological data is to do it in software tools or databases that do not require much training or significant expense. They must also allow data storage or simple import and export tasks to be performed using applications like a word-processor such as *MS Word*, a spreadsheet application such as *MS Excel* or a database management system such as *MS Access*. However, the potential of these tools is not comparable with that offered by other TMSs, such as *TermStar*, or other similar

software products such as *AnyLexic*, *SDL MultiTerm*, *Multitrans TermBase*, *Déjà Vu X TermBase* or *Gesterm*.

In this review and for the second big stage pointed out here in the dictionary-making process (i.e. data processing, management and storage), *TermStar XV* was the point of departure for analysis and comparison. *TermStar XV* is a terminological database, a system of multilingual terminological management oriented towards the concept. This implies that *TermStar* is completely focused on meaning and not on the terms of each language. It allows the user to open a new register (terminological card) for each concept, not for each term, since a concept may contain different terms and linguistic variants for a single object, characteristic or action. An example of this could be the term "mouse", either as a computer device or an animal: the term is the same but the concepts are different. Accordingly, with *TermStar XV*, different registers may be created for different concepts denominated by the same terminological unit. *TermStar* allows for more than 50 different fields in each register, some of them assigned by default by the program and some others which can be defined according to the users' needs and the final objective(s) of the work. In this way, a personalised distribution model of the fields (layout) may be enhanced so that the terminologist can optimise his/her work and find it easier to focus on the target aimed at. Figure 1 shows an illustrative register under development from *TermStar*, designed according to the terminographer's needs for a prospective specialised bilingual dictionary of the ceramics industry.

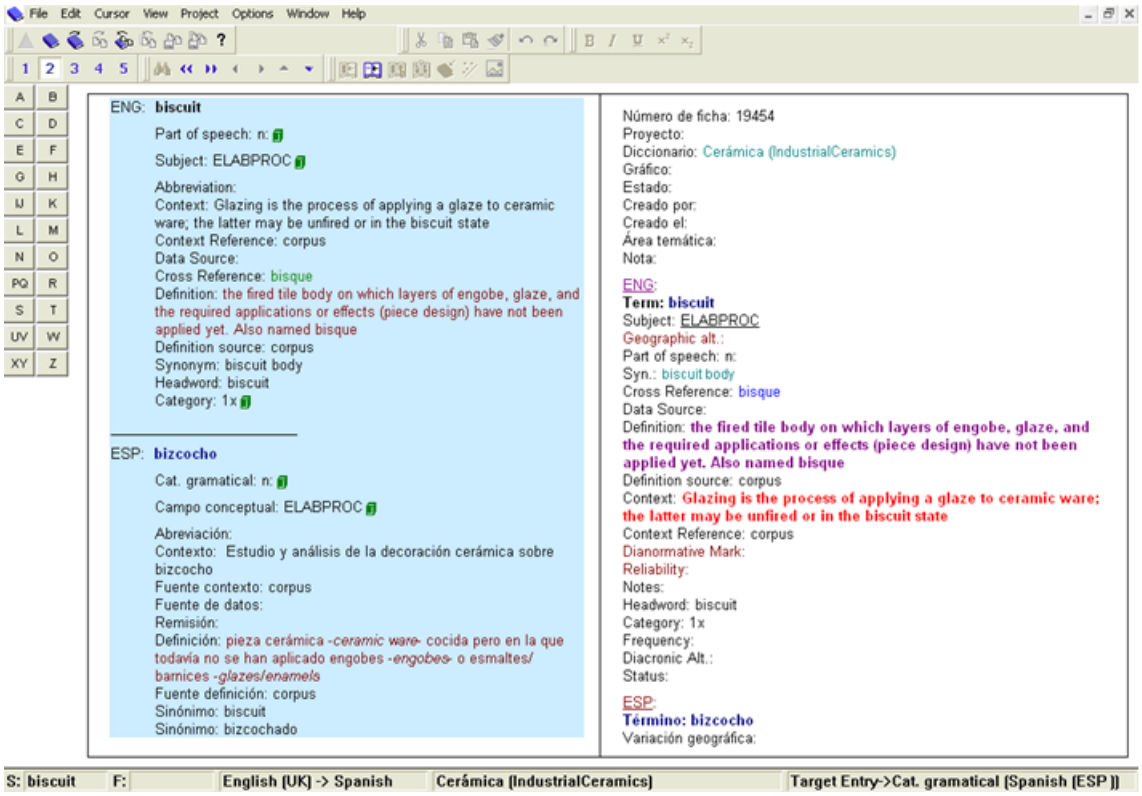

Figure 1. Register under development from *TermStar* and showing a personalised layout.

As indicated on the Star Group webpage (http://www.star-spain.com/es/inicio/), *TermStar* can be accessed as an integrated part of the translation memory and editor Transit, as a macro module of several common text-processing software products (e.g. Microsoft Word), or as a stand-alone dictionary application, which is the option presented here. *TermStar* also offers the possibility of quickly and easily creating registers and having immediate access to them. In the same way, the management carried out by the database management system allows the user to gain rapid and easy access to the data, to have these data ordered according to different criteria, to relate the different data items to each other, and so forth.

Apart from the ones already mentioned, Gómez González-Jover (2005) points out some other technical features that make *TermStar* an overall satisfactory system – despite its price – for the management of terminological data:

- The number of databases which can be created with *TermStar* is unlimited, as well as offering the possibility of opening them all at the same time if desired.

- The number of registers/terminological cards in each database is also unlimited.
- The structure of registers/terminological cards is fixed but dynamic.
- The register/terminological card contains more than 50 fields, some of them predetermined, with administrative information (for instance the number of the concept, graphics, images, entry date, etc.) and some others of a terminological nature that can be repeated in the card/register in each of the working languages.
- The number of working languages is also unlimited.
- It is possible to perform searches of truncated words with the character asterisk, as well as to specify the fields to search (term, abbreviation, synonyms, etc.).
- In addition to the search function, the program also provides, by means of filters, another way of searching for terms.
- Cross references in the form of hyperlinks can be created either manually or automatically (this option allows the terminographer to go from one card to another instantly).
- It allows the user to include non-linguistic fields (such as graphics or images) which, in spite of having no direct correspondence with the kind of information to be contained by the lexical entries of conventional dictionaries, may be useful and enlightening.
- It offers a flexible selection of sorting criteria.

Terminological databases are employed by a wide range of users with very different profiles so that their information needs are, normally, also diverse. In this sense, *TermStar* provides a high degree of flexibility that allows it to be adapted to the needs of each user, apart from offering various modes of data retrieval. However, it is quickly noticeable that the import/export processes in *TermStar* are rather complicated, since several commands from more than one menu are required. Missing a step or making a small mistake in the process implies that the whole import/export procedure fails, which is frustrating, especially for the new user or for the non-professional. Nonetheless, updating data is very user-friendly within *TermStar*, as is adding a new entry, since the whole procedure follows an intuitive logic which anyone familiar with computers can grasp.

TermStar is thus an excellent repository for huge amounts of terminological data, since it allows numerous databases to be created, each capable of housing several bilingual and multilingual dictionaries supporting different languages. *TermStar* also allows the user to personalise the prospective microstructure of the dictionary through "entry arrangement codes", something that is especially useful for dealing with compound terms and multi-word units. The codified category "Category" (together with the category "Headword") in *TermStar* may be configured, for instance, to offer four main arrangement categories: Category 1x shows that the term in the entry has no abbreviation and has to be considered a main entry in the final dictionary layout, whereas category 1 indicates the same main entry status but referred to a terminological unit with abbreviated form(s). On the other hand, the "subentries" in the dictionary are assigned categories 2 or 2x, depending on whether they have an abbreviation or not. In the case of 2 or 2x category terms, the headword that these subentries belong to must also be specified for a correct subsequent arrangement of final dictionary entries and subentries. For instance, when creating the entry "abrasion", if the user wants "abrasion/abrasive hardness (AH)" to become a subentry of the headword (main entry) "abrasion" (category 1x), "abrasion/abrasive hardness (AH)" will be assigned to category "2" because of its abbreviated form, whereas "abrasion resistance" will be assigned to category "2X", since it does not have an abbreviation (see Figure 2). Filling in these fields correctly is the key to obtaining a successful final arrangement of dictionary entries and subentries, both with simple terms and multi-words units, and the possibilities offered by *TermStar* in this respect are very operative and practical.

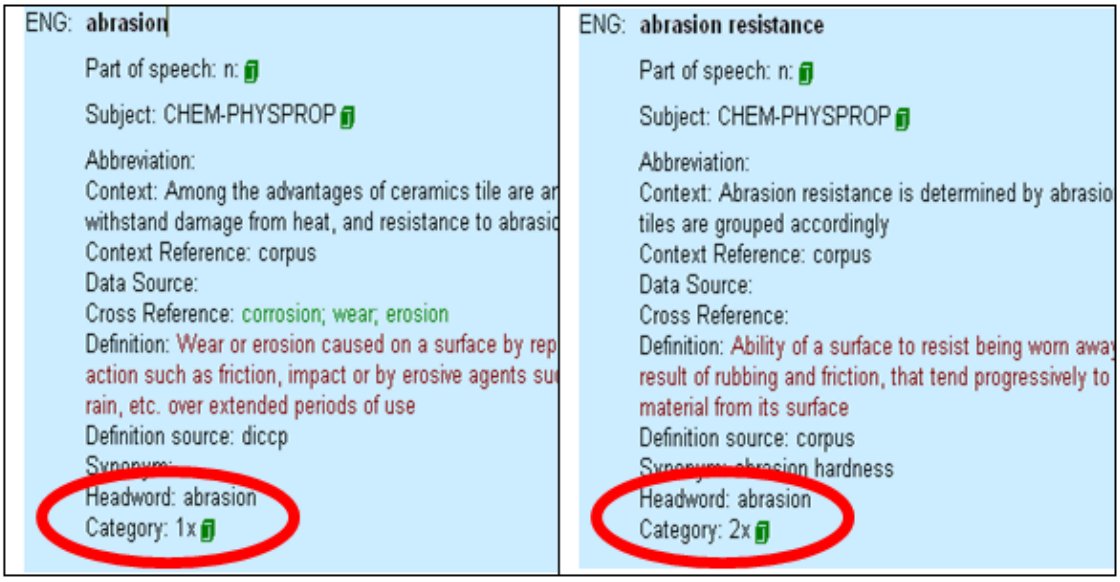

Figure 2: Example of entry and subentry arrangement through codes in *TermStar*.

The huge potential of *TermStar*, despite some of the shortcomings mentioned above, makes it a good and complete option for the second broad stage of the dictionary-making process. This may be clearly observed in Table 1, which, owing to space limitations, shows only a graphic comparison between *TermStar* and *AnyLexic*, *SDL MultiTerm*, *Multitrans TermBase*, *Déjà Vu X TermBase* and *Gesterm*. It can be seen that *TermStar* accomplishes all the functions and possesses all the features included in the table.

Table 1. Table comparing the main features of the TMSs under analysis (adapted from Mesa-Lao 2008).

SEARCH OPTIONS							
TMS name	**Exact search**	**Partial search (fuzzy)**	**Truncated search (wildcards)**	**Search with Boolean operators**	**Search history**	**Showing/hiding entries through condition filters**	**Creating cross references among data registers**
Termstar	X	X	X	FILTERS	X	X	X
AnyLexic	X			X	X		
SDL MultiTerm	X	X	X	FILTERS		X	X
Multitrans TermBase	X	X			X	X	
Déjà Vu X TermBase	X		X			X	
Gesterm	X			FILTERS		X	

ENTRY MODIFICATION							
TMS name	**Terminological management or edition according to users' profiles**	**Adding entries from outside the application**	**Global modifications in the data through replacing functions**	**Copying complete entries**	**Adding illustrative graphics**	**Adding links to external hypertext. resources (Intranet/ Internet)**	**Defining predetermined values (automatic. added for new entries)**
Termstar	X	X	X	X	X	X	X
AnyLexic							
SDL MultiTerm	X	X			X		X
Multitrans TermBase	X	X	X				X
Déjà Vu X TermBase	X	X					X
Gesterm							

DISPLAY MODE			
TMS name	**Choosing the different ways of presenting the dictionaries**	**Designing personalized display modes and users' profiles**	**Configuring the visual aspects of any field within the database**
Termstar	X	X	X
AnyLexic			X
SDL MultiTerm	X		X
Multitrans TermBase			
Déjà Vu X TermBase			
Gesterm			

TERMINOLOGY MANAGEMENT (DICTIONARY ORGANISATION) AND LANGUAGE MANAGEMENT							
TMS name	**Kind of information that can be codified**	**Opening various dictionaries at a time**	**Modify/create from scratch the structure of the entries in the DB**	**UNICODE support**	**Interchange between the source-target languages of a DB**	**Using languages with alphabets different from the Latin one**	**Windows IME (Input Method Editors) support**
Termstar	Textual, graphics, hypertext	X	template	X	X	X	X
AnyLexic	Textual	X		X		X	X
SDL MultiTerm	Textual, graphics, hypertext	X	X	X	X	X	X
Multitrans TermBase	Textual	X	X	X	X	X	X
Déjà Vu X TermBase	Textual		template	X	X	X	X
Gesterm	Textual			X	X	X	X

Therefore, among the basic functions to be taken into account in order to decide on the suitability of any TMS, the terminologist should consider mainly the possibilities

offered as regards their functionality and management, their potential for the creation of terminological cards, and the data-filtering options, as well as the feasibility of export and import tasks and the user-friendliness of the environment. However, as Reppen (2001: 32) states "as with software purchase, the needs of the user should play a key role in deciding which program is most appropriate", since the value of such tools varies greatly depending on individual needs and circumstances.

REFERENCES

Austermühl, F. 2001. *Electronic Tools for Translators*. Manchester: St. Jerome.

Gómez González-Jover, A. 2005. *Terminografía, Lenguajes Profesionales y Mediación Interlingüística. Aplicación Metodológica al Léxico Especializado de la Industria del Calzado y las Industrias Afines*. Ph. D. dissertation, Alicante: Departamento de Filología Inglesa, Universidad de Alicante.

Gómez González-Jover, A. and Vargas Sierra, Ch. 2003. "Metodología para alimentar una base de datos terminológica desde las necesidades del traductor". *Proceedings of the I Congreso Internacional de la Asociación Ibérica de Estudios de Traducción e Interpretación*.

Mesa-Lao, B. 2008. "Catàleg de gestors de terminología". *Revista Tradumática. Traducció i Tecnologies de la Informació i la Comunicació*, 6.

Reppen, R. 2001. "Review of *MonoConc* pro and *WordSmith Tools"*. *Language Learning & Technology*, 5 (3), 32-36.

Scott, M. *WordSmith Tools. Version 6. Online manual*. 10 September 2011 <http://www.lexically.net/downloads/version6/HTML/index.html>

Star Servicios Lingüísticos. 9 September 2011. <http://www.star-spain.com/es/inicio/>

Streiter, O., Zielinski, D., Ties, I. and Voltmer, L. 2003. "Term extraction for Ladin: An example-based approach". In *Proceedings of TANL 2003 Workshop on Natural Language Processing of Minority Languages with few computational linguistic resources*. Batz-sur la Mer, France.

Received October 2011