Chapter 2 A general overview of state-of-the-art MAHT workbenches.

2.1 Tools selected.

The tools discussed in this chapter are as follows:

- **Trados 5.5 by TRADOS**
- **SDLX Translation Suite 4.2.1 by SDL International**
- **Déjà Vu Interactive 3.0.25 by Atril**
- **Wordfast 4.20 with +Tools 2.9e by Yves Champollion**

The tools listed above were selected as they represent the latest developments in CAT software (as of late 2002) and are definitely the most commonly used software packages on the Polish translation market. (cf. Feder 2002a, opinion poll on http://www.proz.com[5]). Furthermore, these products are updated and further developed by their producers, and are easily available to prospective users.

As CAT software comes in a variety of versions tailored to the needs of a particular customer, such as big translation companies, small groups of translators, in-house translators or freelance translators, functionalities offered by particular versions also differ. Therefore, the author had to further limit his overview to specific versions of CAT programs and decided to focus on freelance versions (for individual translators performing the translation work independently) of the aforementioned packages, as they all contain features essential to the translation process and do not possess only a handful of advanced project management options which do not have a considerable impact on the linguistic performance of the tools discussed.

[5] An Internet portal for the translators' community.

2.2 Structure and functionalities.

Despite the fact that each MAHT workbench has a number of characteristic features (see chapter 3), all of them have a similar structure and offer a comparable set of functions that assist the work of a translator.

2.2.1 Common considerations

2.2.1.1 Accepted file types.

As the translators need to be able to work with a huge variety of document formats used by their clients, it is obvious that MAHT tools should accept (*import*) as many formats as possible. Furthermore, an MAHT tool has to ensure that all non-translation information contained in the SL files is kept (e.g. formatting information in the translated files) and reproduced in the TL files (*exported*); such a functionality is labeled *tag protection*. The workbenches under scrutiny accept all major file types (such as *.doc, *.rtf, *.html etc.) either directly, or by means of filtering/conversion applications.

2.2.1.2 Supported languages.

All the latest MAHT tools support most of the existing languages in their TM and TMS (Terminology Management System) modules; their number is limited only by the availability of spell checking modules, thesauri etc. and fonts supported by the system. Unfortunately, such support might sometimes be a problem for minority languages such as Polish, which has a number of nonstandard characters. A good case in point is that of Wordfast – an MAHT tool that only recently provided a properly functioning recognition of Polish characters.

2.2.2 System Components

MAHT packages usually offer a number of modules which can all be run as separate programs in the Windows environment, but are at the same time cross-linked, letting the user easily access functions of one module from another. Although, obviously, the level of integration of program components varies, from full integration in e.g. SLDX to minimal

integration, as in Trados; from a functional point of view, one can distinguish four major components of MAHT packages, which are discussed in detail below.

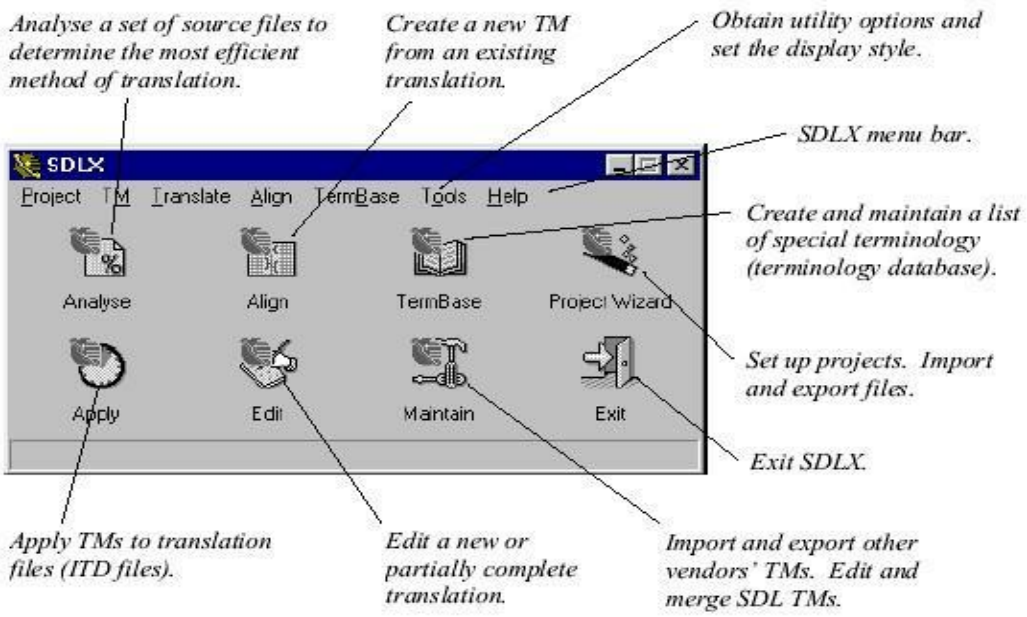

Figure 5: Components of a CAT tool in SDLX (SDL International 2002: 9)

2.2.2.1 Alignment

Alignment modules let the user create a TM from pairs of files: SL files and their translations, by matching the target segments with source segments, and in this way, take advantage of the translation work performed before the user acquired the CAT tool (see 1.4).

Unfortunately, the process is not fully automated, as the automatic alignment using the so-called segmentation rules set by the user (the user needs to specify what should count for the program as the beginning and the end of a segment, e.g. a question mark, a full stop, a colon would mark the end of a segment) does not produce correct pairs of sentences in all cases. For example, in Figure 6 below, in an alignment module of SDLX, *Dorosły liść* (*mature leaf* in Polish) in the translation segment 20 is obviously not a proper translation of the source segment 20: *6.8 surface*, but if the user does not edit the aligned pairs of segments, the system will

accept it as it is, and the resulting TM entry will be improper. To ensure correct alignment the systems enable the user to perform editing of the pairs.

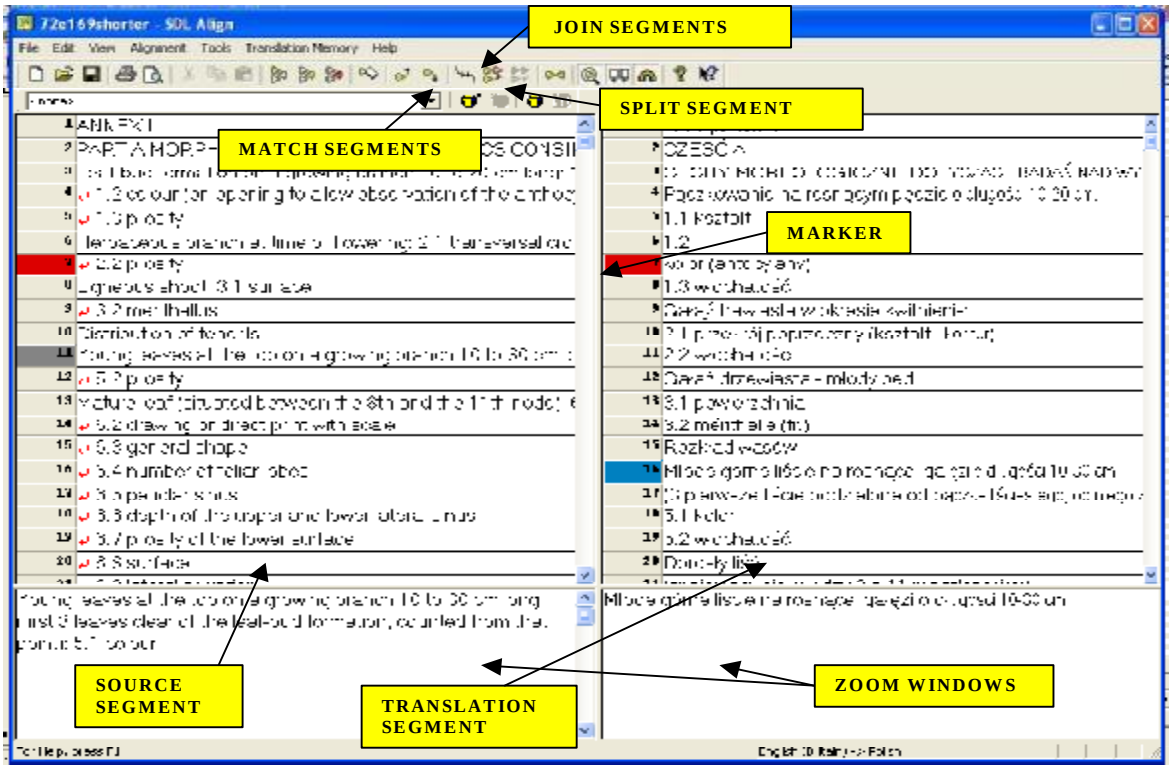

Figure 6: SDLX Align window.

Typical problems and solutions.

The fundamental problem concerning alignment is the result of the fact that it is anything but easy to define, or choose, segmentation rules that would cause the software to properly "cut" the text into "pieces" desired by the translator, hence the process of alignment may prove to be a source of frustration for a user who has not developed proper usage strategies.

The CAT systems under scrutiny here provide sometimes more (e.g. WinAlign of Trados 5.5), sometimes less (+Align, a part of +Plus by Yves Champollion) advanced solutions,

that can limit the tediousness of alignment, based on certain procedures which are performed by the program according to the rules (and exceptions to the rules) specified by the user.

For example, in Trados alignment module: WinAlign, which provides a relatively advanced set of options, the user can specify the segmentation rules for both the target and source documents, telling the program where the segment should begin and end, what the exceptions should be (e.g. even though a full stop usually marks the end of a sentence, but as it is also used in abbreviations, the program has to be specifically informed that it should not view a full stop after a "g" for instance, as a segment end marker. In other words, a program needs to be provided with a list of abbreviations and segment delimiters).

To address this issue some programs like TRADOS WinAlign also attempt to use some more advanced procedures using the text structure and formatting information contained in e.g. *.doc files such as styles, or font sizes.

Another program, SDLX Align, provides the translator with a "validation tool" which compares the source and translation segments and indicates those, which have a significantly different number of words and are therefore likely to be mismatched.

Other common problems with alignment in CAT tools are:

- one segment in the SL file corresponds to more than one TL segments (e.g. Source segment 11 and Target segments 16 and 17 in Figure 3)
- more than one SL segments correspond to a single TL segment
- SL and TL segments are misplaced and need to be shuffled (most of the segments in figure 3)
- when the aligned documents are long, over a few pages long, the translator may lose track of segment pairs, especially when the layouts of SL and TL documents are not directly corresponding. For example a phrase found at the beginning of the SL document, might be at the very end of the TL document.

The standard options enabling a proper pairing of segments typically found in MAHT workbenches are:

- match – a user can select SL and TL segments (one or more) and match them –useful when SL and TL segments are far from one another
- join – the user can select two or more segments in the SL or TL text windows and join them
- split – the user can select the point at which a segment should be split in the zoom window and split the segment
- edit text in the zoom window – users can enter text freely in the zoom window when they decide that the translation provided should be modified or changed.
- move segments to the top or the end of the document - the user can select to move a SL segment, a TL segment or both.

To summarize, and that is also true for other MAHT workbenches' modules, advanced as the options provided by the alignment software may be, it is only the user's expertise that makes the difference between success and failure.

Underlying technology

Alignment algorithms used by MAHT workbenches are usually a combination of length-based and text-based statistical/probabilistic approaches.

The length-based approach exploits the simple observation that a long SL text fragment usually corresponds to a long TL text fragment, and a short TL text fragments tend to be translated by short ones.

Text-based approach attempts to "exploit translation, similarity or identity correspondences between words and other textual components such as figures, proper names and dates" (Trujillo 1999: 72).

The reason why so much manual editing of aligned texts is required is the fact that a MAHT tool often uses a simplified algorithm: it performs segmentation of SL and TL texts,

numbers (indexes) the SL and TL segments and relates them to one another according to this index (i.e. the 20th segment identified in a row in SL text is paired with the 20th identified segment of the TL text. See Figure 6).

Another, more advanced technique tries to exploit a number of structural text features "using text structure markers such as document structure and formatting (e.g. styles as in Word or formatting tags as in SGML, HTML or some DTP programmes or even formatting conventions within segments such as the number of words in bold, italics or underline typeface, etc.)" (Feder 2001: 162).

2.2.2.2 Translation Memory

Translation Memory is undoubtedly the key component of every MAHT workbench and as such it has to meet the highest demands of efficiency (see 4.1).

Structure

Although the data storage format may differ (e.g. Microsoft Access in SDLX, the program's native format in Déjà Vu) Translation Memories in MAHT tools all have a structure of a relational database, which contains pairs of source and translation segments.

Formats

Data contained in a TM may be stored in various formats; each program uses its native (proprietary) TM format, but all MAHT workbenches enable the user to store the TM in a number of formats, also those used by other MAHT packages, which enables some limited capability of using a TM saved in e.g. Trados Workbench format in Wordfast (which has been a source of enormous popularity of this initially free, and now relatively affordable, tool).

Exchange of TMs, needless to say, is of utmost importance to CAT tools' users, because a translator would be forced to buy all the MAHT workbenches available on the market (which are anything but cheap) in order to accept a translation assignment, which can be accompanied by a TM in any format that the customer could have chosen. Unfortunately such an exchange of TMs among a number of MAHT packages is not flawless, often some formatting data is lost, and hence there is no full compatibility.

A good attempt at finding a solution to the problem of TM exchange is the recently introduced TMX (Translation Memory eXchange) format developed by LISA (Localization Industry Standards Association[6]). TMX is (Pooley 2002) an"XML format for the interchange of translation memory data. As such, it consists of elements (with attributes) that provide information about translation 'segments'. The size of a segment is not pre-defined and it will usually be a phrase, sentence or paragraph. For most tools using TMX, the default segment size is a sentence. Within each segment of TMX, there are optionally elements that provide information about the formatting contained in the segment (change of font, hyperlink etc.). TMX also provides for the definition of text 'subflows' such as footnotes and index entries."

The introduction of TMX is an important development for CAT tools' users, as it ensures (Briggs 2002) both: "a) the reusability of data between complimentary tools, and b) the portability of data between competing tools"

Therefore the support of TMX is a must for all MAHT workbenches and all MAHT workbenches under scrutiny in this paper provide the possibility of importing/exporting a TM in TMX as well as other popular and program-specific formats (txt, Excel, Access, TWB, SDLX, Déjà Vu etc.).

[6] http://www.lisa.org

Match types

All the latest MAHT packages support both kinds of matches defined in 1.4 (exact/ fuzzy) and let the user specify at what percentage of "fuzziness" matches should be automatically entered by the program during the pre-translation stage (usually from 30 to 100%, but setting this value lower than 70-75% is not recommended by software providers as such matches would not be very relevant).

Editing/Managing TMs

TMs in all MAHT tools can be edited in a number of ways. The most common options, to name a few, are: merging a number of TMs or dividing a TM into smaller TMs, useful when for example the translator wants to use TUs created after a certain date, or wants to create a smaller TM for the needs of a certain project (e.g. a translator has a single big TM, but now wants to use a TM relevant to a certain subject area); reversing the language pair in the case of bilingual TMs (e.g. from English>Polish database to a Polish>English database); editing/deleting the text of paired segments; searching the TM according to certain criteria (e.g. date of creation, author etc.); global replacing of data etc.. TMs also contain administrative data about who, when and for whom, translated a TU. Some more advanced options allow the translator to protect TMs with a password, analyze a TM from the point of view of the leverage it may provide in a specific translation project, apply a TM to a number of SL files etc.

Underlying technology

The way a TM operates depends on the ***search*** and ***retrieval*** techniques employed by software providers. First and foremost, the program needs to be able to perform a ***similarity calculation*** (that will tell it that a SL sentence is e.g. similar in 65% to a TU in a database) that should ensure that a sentence contained and searched for in a TM will be as much semantically

and syntactically similar to the one in the query (the translator is currently working on) as possible.

One approach used utilizes "syntactic and morphological parsing and analysis to build a representation of a given SL segment (i.e. the *new* one that is being translated and thus added to the existing TM or set of TMs) and comparing it against similar representations created for the segments already stored in a TM)." (Feder 2001: 159 cf. Heyn 1995: 74)

Programs may also employ a more effective approach – the combination of "heuristics applied on syntactic features, morphological reduction, the use of classic (relational) database systems, etc." (Feder 2001: 159; cf. Heyn 1995: 74).

In detail, such a combination includes a number of statistical formulas, such as: Dice's coefficient (ratio of common words and the total words in two sentences multiplied by two), Jaccard's coefficient, Cosine coefficient, overlap coefficient, or dissimilarity coefficient (cf. Trujillo 1999: 61-67). All these formulas provide similar results. As a heuristic measure the results they provide do not guarantee perfect returns. To enhance information retrieval, search techniques employed include: ***stoplists*** that ensure "better retrieval and more intuitive results" (Trujillo 1999: 63) by removal of the most frequent words in a language (for English the list could include: a, and, an, by, from, of, or, the, that, to, with). Other means are used to calculate string/sentence and word similarity through ***stemming algorithms*** based on successor variety, table lookup, affix removal procedures, or N-gram techniques; as well as the technique of ***inverted files***[7] which enables more efficient retrieval (Trujillo 1999: 67).

Another technique used involves "learning properties from text material" (Heyn 1995: 75) "related to the idea of constructing neural networks. This approach is an artificial intelligence (AI) model and it tries to imitate, by means of software and hardware configurations, the way in which the nerve cells process information, learn and remember. The

[7] Inverted file is an index containing a list of words, with each word pointing back to all the sentences in which it is found (Trujillo 1999: 67)

learning method is based on association and recognition of certain recurrent patterns." (Feder 2001: 160).

The selection and utilization of techniques for information search and retrieval mentioned above accounts for the difference in performance of TM components of various MAHT packages (see chapter 4).

2.2.2.3 Terminology Management System

Terminology management, understood as the maintenance of "consistency and accuracy of terminology" (Somers 1997: 7), is a crucial part of each translator's work, which has, needless to say, an enormous impact on translation quality. Therefore, even in cases when the TM component is not useful to a significant extent, because e.g. there is not enough internal or external repeatability (see 1.5), the translator can still benefit from the TMS component (which is often sold/used as a stand-alone application).

In short, a TMS allows the user to create and edit multilingual terminology databases. Terminology Database (TDB) may be defined (Feder 2001: 54) as a "collection of data stored (…) in electronic form, or body of explicit vocabulary which, in its most fundamental form, is similar in structure to a dictionary". MAHT TMS components offer an array of functions helpful in managing terminology information that the user may find important during the translation process. In general, TMS modules provide the translator with advanced functionalities for: TDB structure *creation*, terminology *entry*, and terminology *retrieval/search*.

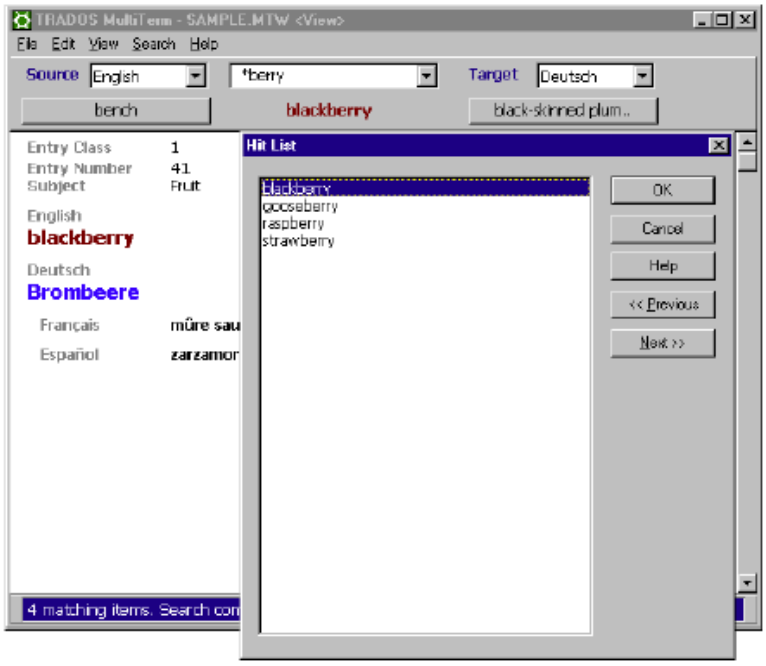

Figure 7: Global search results with a wildcard in MultiTerm.

TDB Structure and format

Structure of a TDB and a particular entry can be easily customized, e.g. in SDLX TermBase TMS (see Figure 8) or Trados MultiTerm TMS (see Figure 9) the user is free to build a multilingual TDB with as many interlinked information fields (e.g. definition, context, synonyms, gender, illustrations) as he wishes. Furthermore, the layout of a TMS, in order to facilitate clear presentation of data, may be arranged according to many user-specified criteria. Similarly to the TMX format discussed before, a TDB can be saved in an exchange format proposed by LISA, called *TBX*. However, this format, unlike TMX, is not yet widely recognized.

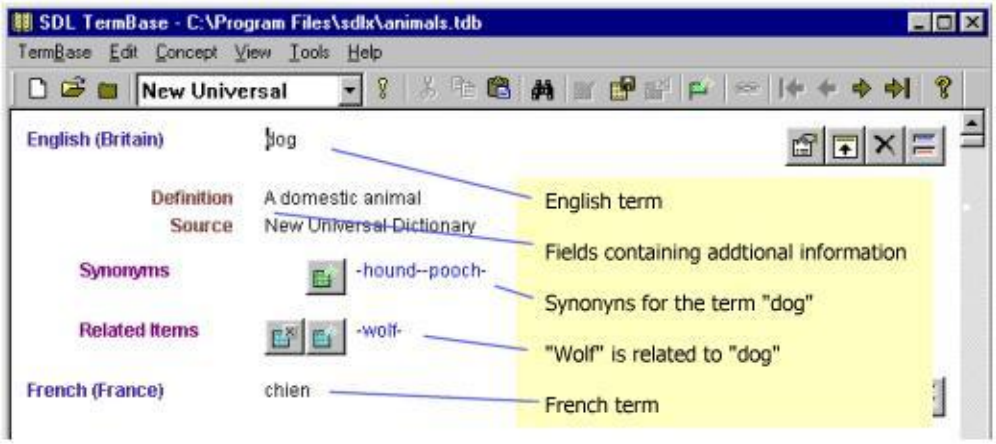

Figure 8: TermBase window[8].

Terminology search

TMS components allow for effective terminology retrieval according to the needs of the user.

Typical search options include:

- search with or without wildcards
- global search
- filter search according to a number of criteria such as: date of entry creation, author, subject field etc.

[8] SDL TermBase Help file.

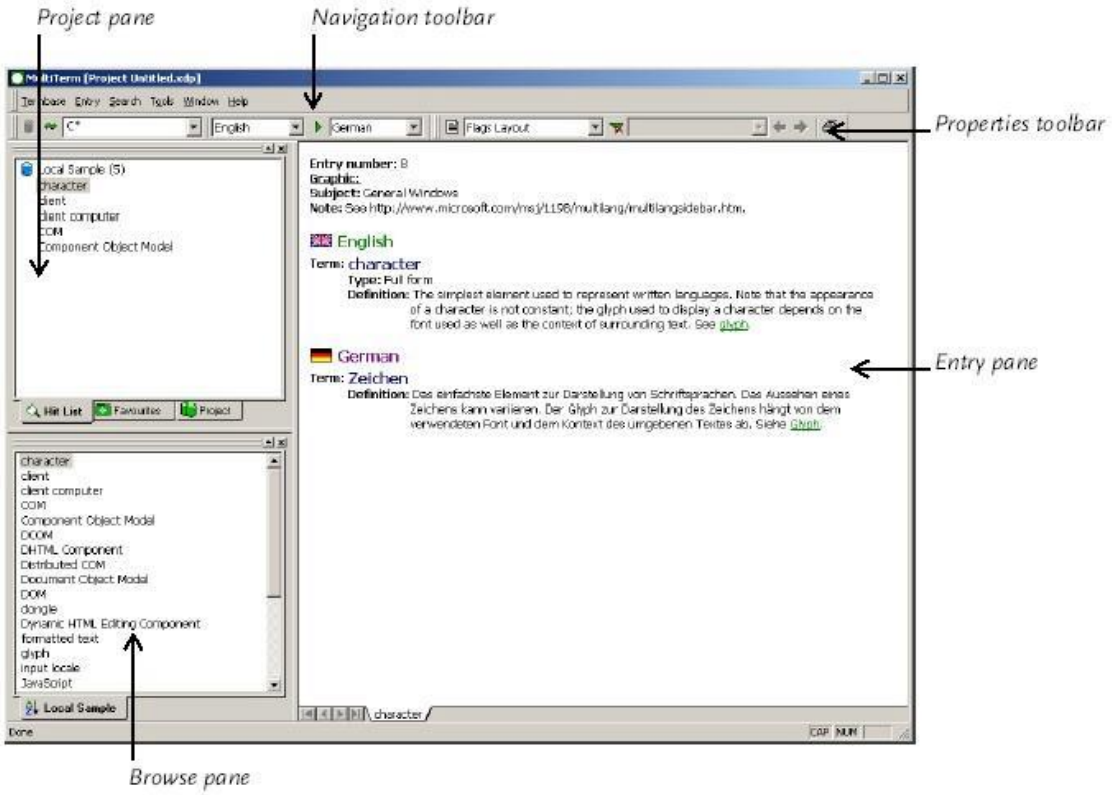

Figure 9: MultiTerm TMS (TRADOS : 54)

TDB Management

Similarly to TMs, TDBs may be imported and exported in a number of formats. TMS systems (Trados MultiTerm for instance) also allow for sharing of terminology data over a network.

2.2.2.4 Editor/ Workspace

MAHT workbenches provide the user with an integrated workspace in which the proper translation work is performed. Typically the translation process in MAHT workbenches has three basic steps (cf. 1. 4): **Import, Translation**, which can be further broken down into: ***pre-translation***, ***editing*** of ***fuzzy matches*** and ***translation*** of ***new material*** – all on a segment-by-segment basis) and **Export** (of translated files into the desired file format). All those operations are performed in a single working environment with the translator remaining in control during all stages of the process.

MAHT workbenches use either an integrated "native" editor (as in SDLX or Déjà Vu) or an external editor (MS Word), (as Trados and Wordfast) which offer the typical text editing functions (search, replace, copy, paste, undo, redo etc.)

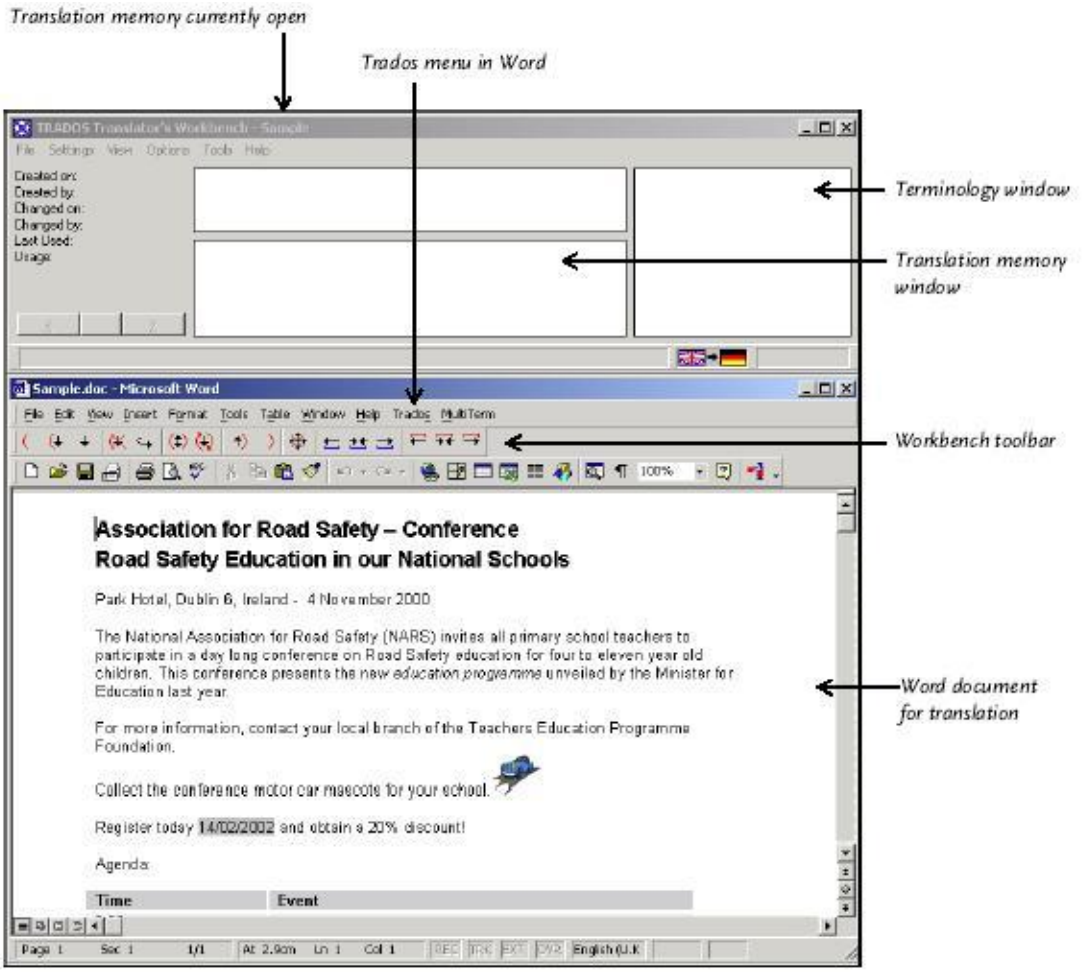

Figure 10: Workspace arrangement in Trados 5.5 with an external text editor (Trados 2002: 109)

The editor module of an MAHT workbench is interfaced with other modules of the program. It provides access to the TM module with standard functions such as: create a new TM, open/close an existing TM, update the opened TM with already translated segments, apply a TM (insert fuzzy and exact matches), lookup – search the TM for the translation of the current segment (marked by the translator), concordance search a TM for either the source or translation segment, propagate a single TU in the whole TL document/TM etc. (some more

advanced packages enable operation on groups of files, such as applying a TM to a batch of files). The translator can also access the TMS which provides the translator with another set of various advanced options of searching a TDB and using the terminological data stored in it. After a translator completes a project, the MAHT may provide him with statistical information on e.g. the number of pages or words translated, TUs entered or modified etc.

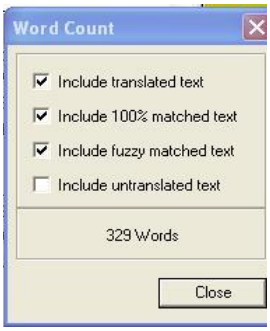

Figure 11: SDL Edit word count.

As typical windows applications MAHT workbenches use an interface of customizable windows, tool bars and menu bars, enabling the translator to arrange the workspace according to his needs and wishes (for a typical layout see Figure 10). Some of the most frequently used options may be accessible through keyboard shortcuts (the user can press Ctr+J in SDLX to join segments, for instance).

The layout can be easily customized: the main windows (such as source and translation windows) can be set either horizontally or vertically, their size can also be changed. MAHT software providers also do their best to make their programs user-friendly. Typically, help files are accessible at all times, and short popup messages (tooltips) are displayed if the mouse cursor is held over a particular screen element for some time.

Furthermore, the most common operations, for example in Trados or SDLX, are performed with the assistance of the so-called wizards ─ series of dialog boxes, leading the user step by step to his desired goal, so that even a beginner user of an MAHT workbench can manage his translation assignment.

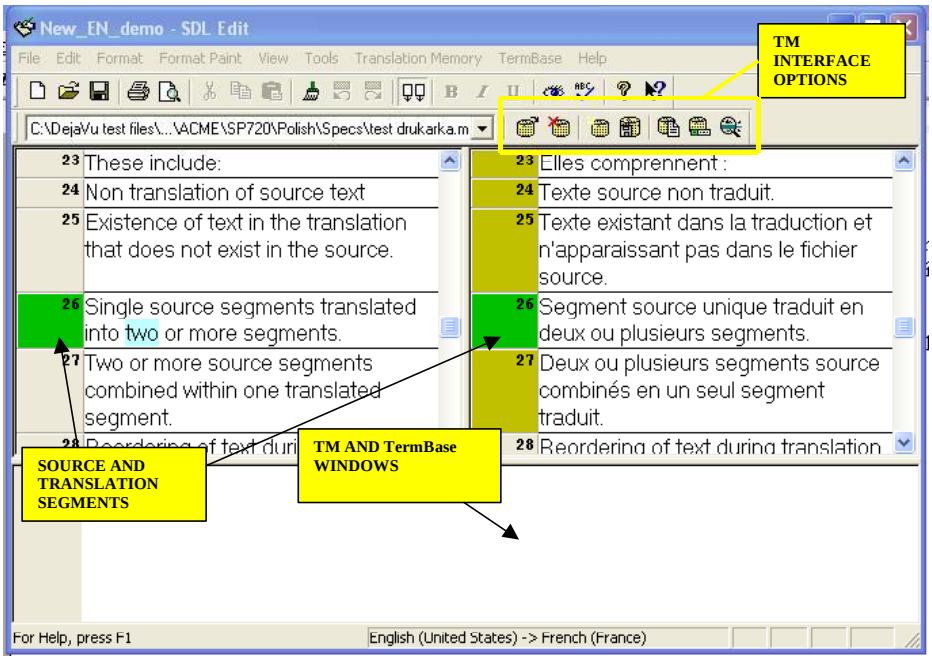

Figure 12: Workspace layout in SDLX.