

Jacek Jabłoński
University of Wrocław

Componential Analysis in Translation of Computer Software

1. Componential analysis in general

In linguistics, componential analysis [CA] is a way of “splitting up various senses of a word into sense-components, which may or may not be universals” (Newmark 1995: 114). The semantic components are represented in a binary manner with a word either having a particular distinctive feature or not, which is marked as (+) or (–). However, “as a theory which sought to isolate universal semantic features ... CA has been a disappointment” (Bell 1991: 87). Bell continues to state that “the features proposed for the analysis of any problem are arbitrary and the binary nature of the features ... limits the application of the analysis to items which are clearly distinguishable in such terms” (1991: 90). Therefore, other criteria have been developed. Apart from the original binary taxonomy (+/–), there were attempts to introduce multiple taxonomy, hierarchy, polarity, relation and other criteria. However, CA was still not a viable manner of investigating universal features.

On the other hand, as Newmark puts it, CA may also be used as a translation tool. In that case it is a method of comparing the SL word or term with a proposed TL one “which has a similar meaning, but is not an obvious one-to-one equivalent, by demonstrating first their common and then their differing sense components” (1995: 114). It is crucial to mention that the distinguishable features of an SL term may include, as Newmark describes it, “composition, shape, size and function of the referent; ... its cultural context and connotations, as well as its currency, period, class usage and its degree of formality, emotional tone, generality or technicality, and, finally, the pragmatic effect of its sound composition” (1995: 114).

Bell draws a slightly different image of a full CA entry for a given word by enriching the syntactic information with grammatical and phonological aspects and states that a fuller record for an item should include pronunciation, orthography,

syntactic information, any significant morphological information and semantic sense (1991: 88).

While in semantics comparing words using CA involves using a fixed number of distinctive features, in translation the comparison revolves around features which are needed at a time. The idea is not to describe a word as accurately as possible but to find the difference between (or the common properties of) the SL and the TL term. For instance, comparing the sense components of the English term *multiplayer* with Polish *gra wieloosobowa*, will show that both terms share the [+ more than one player] feature and both have [+ game] as well. As these two components are crucial for the meaning of the message, the terms are equivalent enough to make a fulfilling translation. In this case CA may serve either as a technique of producing a proper TL term or as a tool of evaluating an existing translation of *multiplayer* (which, until some time ago, did not have a standard translation).

2. Componential analysis in translation of computer software

Computer applications have a unique feature distinguishing them from other text types – interface. It is a medium with which the user communicates with the machine/program/operating system and with which the M/P/OS communicates back. Popular modern interfaces usually make use of language (to describe them) and of audio-visual feedback (to strengthen them).

The language of the interface is very much standardized. Through the years the commands available to the user have been shortened and simplified to make them as comprehensible as possible. And so the typical phrases present in the interfaces are for instance: *minimalize*, *load*, *quit*, etc. Of course, many applications introduce some innovations to the known forms of interface but its language usually stays the same or very similar.

When dealing with the interface, the intention of the translator is to maintain the meanings of the original commands and, at the same time, to avoid breaking the patterns created by other, similar programs. The user should, by all means, be capable of intuitively moving around the interface due to the presence of well known commands. And so, in terms of CA, the TL command of a given interface should optimally have the [+standardized language] feature. For example, in various operating systems there is a command in SL – *create folder*. In a Polish translation two versions are standard, namely: *utwórz katalog (folder)*, *nowy katalog (folder)*. All the above-mentioned TL terms are [+ standardized language]. If effort is made to introduce a new translation, such as *nowa teczka*, it may not be communicative or entirely comprehensible to the addressee. Although the graphical representation clearly shows a *teczka* and though known dictionaries do suggest that a *folder* (Eng.) and *teczka* are the same, the translation is somehow missed. Using CA clearly shows why.

Table 1.

	creation of a new element	accordance with the graphical representation	belongs to standard IT vocabulary
create folder	+	+	+
utwórz katalog	+	+	+
nowy folder	+	+	+
nowa teczka	+	+	–

Another table illustrates how one could analyze possible translations of SL *properties*, a command often encountered in popular operating systems, leading to the display of a given file's/folder's characteristics. Again, the TL is Polish. Other possible TL terms come from popular online bilingual dictionaries under *properties*.

Table 2.

	'characteristics'	can be used with 'files' and 'folders'	belongs to standard IT vocabulary
properties	+	+	+
właściwości	+	+	+
charakterystyka	+	+	–
cechy	+	+	–
własności	+	+	–

Obviously, one does not have to draw tables to learn that *charakterystyka* is not a well-functioning translation in this case, but clearly CA is still a useful tool for showing and comparing possible TL terms.

What is more, texts of computer software are rich in short, concise commands or descriptions. Whenever a translator deals with phrases such as: *save as*, *new game*, *shut down system*, etc., he or she deals with a whole translation unit. Each of the commands is "the smallest segment of a SL text which can be translated as a whole, in isolation from other segments" (Newmark 1988: 285). What follows is that in the process of translating computer software, CA is used not only to compare or analyze particular words, but also to work on whole translation units and to seek their proper TL equivalents.

CA is also very useful when it comes to other software-related texts, computer games for example. Video games have plenty of stand-alone terms which may, just as interface commands, function as units of translation, e.g.: names of creatures, weapons, equipment, names of non-existing worlds, etc.

Diablo, a popular role-playing game may be a very good example to illustrate the convenience of using CA when dealing with computer games. *Diablo* has not been officially translated into Polish, so the example shows a 'fresh' attempt at the source text.

As a game where players explore dangerous dungeons and fight fantasy creatures, *Diablo* is very rich in names of medieval weapons. Bastard swords, shields, bucklers, axes or falchions are amongst the items the players gather during gameplay. One weapon, however, is *blade*, which belongs to the category of swords with *two-handed swords*, *sabres*, *long swords*, *broad swords* and others. While finding an equivalent to SL *sabre* is relatively easy, translating *blade* into TL may benefit from CA. It needs mentioning that every weapon in *Diablo* is depicted and that the image of *blade* shows a medium-sized two-edged sword. The table below depicts contrasts between various possible translations.

Table 3.

	corresponds with the picture	sword	medium size	double- edged	European **	specialist name
blade	–	–	+	+	+	–
ostrze	–	–	+	+	+	–
miecz szeroki*	+	+	+ / –	+	+	+ / –
miecz długi*	+	+	+	+	+	+ / –
katzbalger	+	+	+	+	+	+
miecz prosty	+	+	+	+	+	+ / –
jian	+	+	+	+	–	+

* Clear equivalents for other weapons in *Diablo*, hence inadequate as translations of “blade.”

** European = more recognizable to the Western Culture addressees.

The table above clearly shows that *miecz prosty* is the most suitable TL term. By analyzing the detailed components of given terms, the translator is able not only to choose the closest TL word but also to find a reasonable way of improving certain ambiguities of the original text (blade is not a type of sword).

3. Componential analysis in translation criticism

CA is also very useful in translation criticism. However, while in the process of translation itself CA helps search for optimal TL terms/words, in translation criticism it helps visualize and document the faults and strengths of the TL text.

The vocabulary of operating systems or popular programs is to a great extent standardized and there is not much to criticize but in the field of computer games there is a whole spectrum of texts needing improvement or, by contrast, showing good translation work.

Heroes of Might and Magic III is a game whose Polish edition may give numerous examples of how CA helps evaluate an existing translation. One of the creatures, for instance, a *war unicorn*, is called *jednorożec bitewny* in the TT.

At first, the translation may appear successful but only as long as connotations, as one of the sense components, are not analyzed. For the sake of this example the highlighted features only include those of relevance to the decision between *bitewny* / *wojenny* / *bojowy*.

Table 4.

	connotes bębny, okręt, sztuka	connotes szal, zawierucha, zapał, pole	connotes animals, rumak among others
war	+	–	+
bitewny	–	+	–
wojenny	+	–	–
bojowy	–	–	+

The table indicates in a clear way two major conclusions. Firstly, CA of chosen features suggests that connotations of the determiner *war* link with a larger amount of words than the Polish *wojenny*. Secondly, through analysis of sensual components we are able to deem the existing translation faulty and propose the more proper term *jednorożec bojowy*.

Another example comes from the same game and refers to a creature called *rogue* and, inconsequently, *rozbójnik* in the TL. The table below illustrates a brief analysis of the relevant sensual features with a two-fold aim: determining/proving the inaccuracy of the existing translation and proposing a better TL term. The suggested TL words are five closest synonyms to the one present in the Polish version of the game.

Table 5.

	behaves badly	liked by other people	connotes a folk hero	brutal
rogue	+	+ / –	–	+ / –
rozbójnik	+	–	–	+
zbójnik	+	+ / –	+	+ / –
łotr	+	–	–	–
rabuś	+	+ / –	–	–
rzezimieszek	+	–	–	+ / –
łupieżca	+	–	–	+

The criterion of connoting a folk hero is an unnecessary addition referring to the fact that *zbójnik* to some people stands for Janosik, a folk hero. That supplementing feature, however, places an even stronger emphasis on *zbójnik* being the best TL term for *rogue*.

The next example is aimed at showing that CA is also effective in evaluating successful translations and is useful even when neologisms are the object of the analysis. The SL term *dwarven thunderer* comes from a turn-based strategy game entitled *Battle for Wesnoth*. The TL term is *krasnoludzki grzmotomiot*. It needs to be mentioned that the creature is depicted as a dwarf holding a rifle, which, upon shot, produces a very loud thunderous sound.

Table 6.

	dwarf	suggests loud thunderous sound	neologism	suggests shooting / throwing
dwarven thunderer	+	+	+	–
krasnoludzki grzmotomiot	+	+	+	+

It appears that the TL variant provides the addressee with even more information and corresponds better with the picture in the game (which is an important aspect of computer games translation).

4. Summary

Componential analysis is a practical and useful technique in translation of computer software-related texts. Computer programs usually have their text layers formulated in such a precise and concise way that contrast analysis is not only aimed at single words, but at whole units of translation, each delivering a separate message to the addressee.

Another significant aspect of this use of CA is that it “allows for a particularly compact representation of meaning if the features are binary, or have a small number of values” (Krifka 2001: 6). In translation procedures utilizing CA the representation is even more compact, for the lexical items do not require full semantic/grammatical/phonological entries but usually only need specification of few key features.

As shown in the examples, CA used as a technique for finding a proper TL term enables a convenient and effective comparison of the SL term and the proposed TL one. Not only does this comparison allow to separately evaluate numerous significant features but also enables to include aspects such as correspondence with the graphical layer of the application.

When used in translation criticism, CA proves to be a very successful method of determining whether small, one-term or -phrase translation units in SL and TL are functionally and semantically equivalent.

To conclude, despite being dismissed in general linguistics, CA is a swift and efficient tool in a translator’s hand when confronted with a multi-media computer text characterised by a small translation unit.

References

- Bell, R.T. 1991. *Translation and Translating: Theory and Practice*. London: Longman Group UK Limited.
- Krifka, M. 2001. *Lexikalische Semantik*. Berlin: Humboldt Universitaet. http://amor.cms.hu-berlin.de/~h2816i3x/Lehre/2001_HS_LexikalischeSemantik/LexSemantik2.pdf.
- Newmark, P. 1988. *A Textbook of Translation*. London, Prentice Hall.
- Newmark, P. 1995. *A Textbook of Translation*. Hempsteadt, Phoenix ELT.

Sources

- Battle for Wesnoth*. 1993. Copyright *Battle for Wesnoth*.
- Diablo*. 1996. Blizzard Entertainment.
- Heroes of Might and Magic III*. 2001. 3DO. New World Computing.