



faculty of social
sciences, arts
and humanities

SAL 40/2022

Research Journal
Studies about Languages
pp. 103-114

ISSN 1648-2824 (print)

ISSN 2029-7203 (online)

DOI 10.5755/j01.sal.1.40.31081

TRANSLATION / VERTIMAS

Lithuanisation of Software Metaphors

Received 03/2022

Accepted 05/2022



<http://dx.doi.org/10.5755/j01.sal.1.40.31081>

HOW TO CITE: Maumevičienė, D. (2022). Lithuanisation of software metaphors. *Studies about Languages / Kalbų studijos*, 40, 103–114. <http://doi.org/10.5755/j01.sal.1.40.31081>

Lithuanisation of Software Metaphors

Programinės įrangos metaforų lietuvinimas

DAINORA MAUMEVIČIENĖ, Kaunas University of Technology, Lithuania

Abstract

The majority of software that is localised, i.e., adapted or Lithuanised as a product to the Lithuanian market, language, culture, social norms and values, is originally engineered and designed in the United States of America. Despite the fact that internationalisation has to be carried out as a process of neutralising and standardising the software for further localisation of the product, it is usually carried out incompletely or in some cases might be unperformed at all, if companies have not considered the product to be sold globally. Thus, the software contains attributes, for example, writing conventions, legal norms and rules, culture-specific items, metaphors and other that reflect the source culture and language of the software. Software related texts, online help documents, resource files, messages and other contain various metaphors that are related to applications, malfunctioning, software components and menu items and hardware. At the same time, conceptual metaphors represent the worldview, and its perception is pertinent only to some specific language users. The idea of localisation is both to adapt products to target users in terms of their culture and language and make them feel that the product was designed and created for them. Thus, taking in mind that conceptual metaphors represent the perception of the worldview, the crucial question arises, if it is possible to adapt, i.e., Lithuanise, conceptual metaphors to such an extent that they would reflect the worldview and perception of the target language and culture the tropes are used to represent, categorise and depict. At the same time, it becomes interesting to examine how the trope is being changed in the target language as well as analyse the capacities of the language to change, adapt to and embrace new things it has to name.

KEYWORDS: Lithuanisation, localisation, metaphor, conceptual metaphor, software, translation strategies.

Introduction

The use of different information technologies and artificial intelligence related solutions, software programmes and packages in all areas of human activity is obvious. Despite the fact that digital literacy skills of the entire population are constantly increasing, the need to localise and adapt products is one of the imperatives that companies need to carefully reconsider, addressing both their product development and marketing strategies, if they wish their companies to become players of international and global markets. In order to address the needs of the rapidly changing society and business, and achieve a wider sales and usage of a product, i.e., software or any other, the largest companies of the language industry and providers of software solutions, such as *Microsoft*, *Google*, *Oracle*, *IBM*, *Hewlett Packard*, *SunMycrosystems*, aim at localising as many products for different target markets as they can. In this article localisation is defined and perceived as a process of adapting and tailoring a product to some specific

target market and its local laws, norms, values and regulations, consumer needs and behaviour, expectations and linguistic/cultural requirements.

The majority of software that is localised, i.e., Lithuanised to the Lithuanian market, is originally designed and developed in the United States of America; thus, the product entails features (for instance, date formats, the system of measurement, the display of weekdays, metaphors and the like) that are characteristic and peculiar to the source language and culture. Though during the process of localisation, internationalisation is a must and one of the key stages in terms of product preparation for further processes of translation and localisation as indicated by Pym (2014), it is not always carried out due to the fact that the producer of the software initially aimed at producing a product for local use and started considering entering the global market later. In such cases many source language and culture specific software components are retained. Even though, internationalisation has been performed during stages of software development for localisation, it is sometimes incomplete, and software developers and engineers tend to skip some internationalisation related tasks (Dagienė et al., 2011). Thus, the elements mentioned above might be retained, eliminated or modified by means of following specific strategies so as to make sense in the target language and culture. Carrying out this research and trying to examine the processes of Lithuanisation of enterprise resource management software *Oracle*, the author of the paper was provided the right to get acquainted with internal documents, contracts of purchasing the software from a company in the United States of America by the Lithuanian party, where terms and conditions of software Lithuanisation were presented. The confidential internal documentation indicated the fact that the original software was prototyped and adapted to the Lithuanian language and culture from the American version of the software and demonstrates the direct link between English and Lithuanian as the source and target languages and cultures.

Metaphors are characteristic to both computer science, information and communication technologies as well as various computer hardware or software texts. Software systems, computer and software functions, IT and computer related malfunctioning, names of software components, menu items, tabs and fields and other frequently apply and employ buildings-related, administration, gardening, animal, illness (virus) or other conceptual metaphors (Boyd, 2003/2013). While carrying out the process of localisation, the sense that a product has been developed for the target language and culture, has to be created. This means that any parts of software should resemble the language and culture the product is localised for. All culture specific items, for example, writing conventions and standards, other requirements, including metaphors, must be characteristic to the target language and culture. Moreover, metaphors, especially multimodal metaphors, might serve as a tool “to enhance the understanding of a culture (Forceville, 2016), which indicates that software developers, translators-localisers must be aware of the fact and be well acquainted with both the source language/culture and the target language/culture. Therefore, it becomes interesting to examine and analyse how the trope is changed when the linguistic corpus of the software is localised for the Lithuanian language and culture as well as how conceptual systems that metaphors stand for are Lithuanised and adapted from one language to another, if adapted at all.

Moreover, it is relevant and interesting to analyse the capacities and possibilities of the language to both express, represent and embrace current realities that have to be expressed through new words, concepts and conceptions. In other words, it is interesting to examine the power of language evolution and adaptation. Technologies, computer science and artificial intelligence offer new perceptions, conceptions, terms, words in terms of their composition, presentation and structure, thus, it is relevant to study, how it affects the language. Secondly, metaphors reveal conceptualisation of the world that is specific for each culture and language respectively. Following the point of view on metaphors as explained by André (2011), the study of metaphors is useful since it might offer new insights, models and paradigms to study and understand ways how people understand the world around them. Therefore, the focus of the article is shifted on the Lithuanisation of software metaphors with the aim to find out the way the trope adjusted in the process of software translation and localisation when linguistic content is translated and Lithuanised so that it would be perceived and recognised by the Lithuanian users. In an attempt to conduct the analysis and examine how software metaphors are tailored to the Lithuanian language and culture, the paper employs the following research methods. The theoretical insights have been provided by means of applying descriptive, analytical and contrastive research methods. Multimodal (composed of linguistic and visual modes) and monomodal (composed of linguistics modes) metaphors have been investigated by means

of using the model of the semantic strategy of translation of *trope change* as adapted from Chesterman's (1997) model of thirty strategies of translation as well as a tool of multimodal transcriptions as provided by Thibault (2000) and further improved by Baldry and Thibault (2006). The linguistic corpus of software usually composes the biggest part of software corpus that needs to be translated after the prototyping of the target software, thus, it is purposeful to apply strategies of translation in the process of localisation. The empirical corpus of the research, composed of manually collected software language strings and screenshots of both original and Lithuanian software product, i.e., *Oracle*, has been applied to examine the way software metaphors are adapted during the process of software Lithuanianisation. The research corpus contains 84 852 translation units / language strings where the length of a language string might range from 4 characters to an extensive text (message) of 5000 characters without spaces. Metaphors composed around one third of the research sample, i.e., 25 401 (one metaphor equals 1 unit) in total, whereas 12 001 represent multimodal metaphors.

Theoretical Framework

Conceptual metaphors. People cogitate and work applying schemes – metaphors. All the system of concepts is metaphoric since by means of schemes people make up a mental picture of the reality around them and metaphor becomes both the mental picture of reality (Lakoff, 1987; Halliday, 1994) and a trope that governs the thought (Forceville, 2016). Conceptual metaphors reveal the peculiarity of the way language is daily used as well as the way of cognition and perception (Gudavičius, 2009). Conceptual metaphors also represent the process of conceptualization that is used to depict and form new conceptions. The reasoning of human beings undergoes the process of categorisation when people apply specific schemes to systematise the world by means of metaphors. And metaphors are abundant in language just because they are abundant in the conceptual framework. The tropes are used as a kind of “basic templates” to represent and understand the world around (André, 2011). The background of linguistic communication is composed of the same schematic concepts on the basis of which a person thinks and acts. Metaphors, especially conceptual metaphors, represent not only the way a human being uses language every day, but they also pertain to thinking and cognition of a man (Lakoff, 1987; Gudavičius, 2009). A conceptual metaphor might be defined as a way of world conceptualisation and categorisation. It is a cognitive process that helps to form and reveal new concepts. When human beings aim at defining meanings of words, phenomena, concepts, entities or things, they usually invoke various metaphors. Thus, as Lakoff and Johnson claim, the perception and cognition of the world is conveyed and depicted through metaphors since human beings use metaphors to describe less familiar things and phenomena by means of things and phenomena that they are familiar with (Lakoff & Johnson, 1980; Lakoff, 1987). The essence of a metaphor is mapping of the link between two domains, i.e. the source domain/concept and the target domain/concept. The transfer of meaning is based on the similarity of things or phenomena.

Computer science, computer related language and texts exploit various metaphors as well. Software, programmes, menu items, elements, functions and computers are metaphorically compared to other things and phenomena. Examining different types of metaphors Boyd indicates that software metaphors might be related to the domains of computer elements, structure, software development cycles, components, data collection, processing, functions and other (Boyd, 2003/2020). Those metaphors have become so natural and deeply entrenched into conceptual schemes that people forget that some computer parts, components of software like *mouse, windows, desktop, home, workbench, tree, cookies, bug, virus, computer memory, key, manager* and other represent conceptual (dead) metaphors. They are defined as dead metaphors since due to their constant and daily usage the users of language do not longer recognise them as metaphors. Yet, dead metaphors are referred to as vital in the perception of the worldview (Lakoff & Johnson, 1980; Forceville, 2006; André, 2011). In addition to this, it must be mentioned that software texts also contain multimodal metaphors that might be perceived as tropes “whose target and source are each represented exclusively or predominantly in different modes” (Forceville, 2006 p. 384). Whereas multimodality might be perceived as “the use of several semiotic modes in the design of a semiotic product or event” (Kress & Van Leeuwen, 2001, p. 20) and has to be focused upon since much of the content that used to be presented and coded solely by language now is more frequently presented and communicated by means of visual representation (Kress, 2010). This is characteristic to software texts when information is presented by means of different pictures and icons.

Software frequently contains metaphors that are visually presented to demonstrate and explain to the users what specific functions, commands the software might perform (Fig. 1). They allow understanding the product faster since the visual mode is supported by the linguistic mode or vice versa.

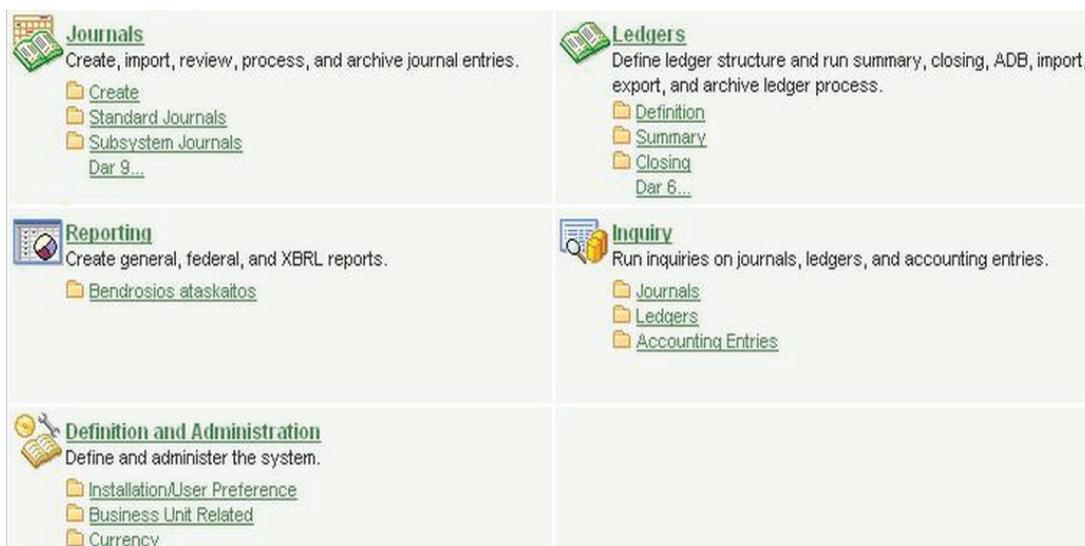


Fig. 1 Multimodal (visual) metaphors (Source: Oracle Database, 12c Release 1(12.1), 2015)

Fig. 1 above demonstrates the use of multimodal (visual and linguistic modes are employed) metaphors in Oracle software. For example, the visual icon of a book represents ledgers as a tool that is used in book keeping and accounting. The visual image of a diagram and an Excel spreadsheet is applied to represent the software component of reporting, whereas the visual use of the magnifying glass, table and stacked columns represent to tool of running different queries in the system. Since not all of the icons used might be perceived at first sight, definitions or names of the metaphors are attached next to the visual metaphors.

Translation strategies. Discussing how metaphors are translated and Lithuanised, it is necessary to define the term *translation strategy*. The term might be perceived as a set of actions and tactics that is employed to translate various texts as well as the linguistic corpus of the software. According to Pym (2011), a strategy is used with an intension to achieve some aim. It must be mentioned that the term *translation strategies* are sometimes synonymously applied to refer to actions, shifts, methods, techniques or procedures that translators undertake in the process of translation. Moreover, *translation strategies* as a term is used to define a way of solving certain translation issues and challenges (for example, new terms and concepts that exist in the source language but do not exist or have not been standardized in terms of their use in target language) that translators-localizers might face when localising (Chesterman, 1997, 2005). During the process of software localisation, decisions what and how specific language strings or software texts should be localised have to be made. Thus, the questions *what* and *how* become crucial since they also determine the level of localisation. For example, sometimes due to high costs, a decision to localise only the graphical user interface is made, therefore, localisation of such a product is partial and incomplete. During the process of localisation the question *what* is considered when the source text is examined, whereas the question *how* is addressed by means of dealing with the target text and specific translation strategies that are usually utilized to create meaning and tailor the software to the target language and its users.

As mentioned above, the terms – strategies, shifts, techniques, methods and similar are often used as synonyms, thus, Chesterman (2005) suggests applying the term *method* to denote a general way (free, literal, or other) of translating. Whereas, the term *strategy* is characterised as a cognitive effort that is necessary to deal with various translation issues and challenges that occur in the process of translation and localisation. The concept of *technique* describes micro and macro-level changes and procedures and the term *shift* indicates the result of

Table 1 Translation strategies by Chesterman (1997)

Syntactic/grammatical strategies	Semantic strategies	Pragmatic strategies
Literal translation	Synonymy	Cultural filtering
Loan, calque	Antonymy	Explicitness change (explicitation/implication)
Transposition	Hyponymy/hyperonymy	Information change (addition/omission)
Unit shift	Converses	Interpersonal change
Phrase structure change	Abstraction change	Illocutionary change (speech act change)
Clause structure change	Distribution change (expansion/compression)	Coherence change
Sentence structure change	Emphasis change	Partial translation
Cohesion change	Paraphrase	Visibility changes (notes, glosses, etc.)
Level shift	Trope change	Transediting
Rhetorical scheme change	Other semantic changes (modulation)	Other pragmatic changes (layout, choice of dialect)

a procedure (Chesterman, 2005). Translation strategies allow examining both translation and localisation as a set of action and a multi-level hierarchical process (Chesterman, 1997). The list of 30 strategies as proposed by Chesterman deals with translation and localisation issues within syntactic, semantic and pragmatic levels of language use. The list of translation strategies is indicated in the **table 1**.

Syntactic and grammatical strategies are mainly related to different form manipulations and changes, whereas semantic strategies deal with lexemes and semantic changes of meaning. Pragmatic strategies are used to create meaning in the communicative act.

The strategy of *trope change* falls under the set of ten semantic strategies of translation and is usually used to translate rhetorical tropes (metaphors, metonymies, personification and other). Following Chesterman, the strategy might be used in the following way:

- a trope of the source text is retained as a trope in the target text, i.e. $ST \text{ trope } X \rightarrow TT \text{ trope } X$;
- a trope of the source text is replaced by a different trope in the target text, i.e. $ST \text{ trope } X \rightarrow TT \text{ trope } Y$;
- a trope of the source text is removed in the target text, i.e. $ST \text{ trope } X \rightarrow TT \text{ trope } \emptyset$;
- a trope does not exist in the source text but is used in the target text, i.e. $ST \text{ trope } \emptyset \rightarrow TT \text{ trope } X$ (Chesterman, 1997, p. 105).

This paper, though focuses only on one semantic strategy of trope change, employs Chesterman's typology of strategies as a tool for the analysis for several reasons. First of all the subdivision of translation strategies is consistent and focuses on three linguistic levels of strategies, i.e., syntactic/grammatical, semantic and pragmatic. Moreover, Chesterman's classification of strategies is the only one to consider and examine trope changes (namely metaphors and metonymies) as a separate type of a semantic strategy of translation, indicating specific modifications, the trope might undergo when texts are translated or adapted to other target languages. Finally, this research also aims at checking if the typology of Chesterman's strategies can be employed together with the tool of multimodal transcription as proposed by Thibault (2001) and the model of different modes of meaning, composed linguistic, visual, audio, special and gestural modes, as suggested by the New London Group (1996).

Localisation and Lithuanisation. Localisation might be perceived in various ways due to the complexity and multi-disciplinary nature of both the phenomenon and process. The analysis of various scientific publications and research papers has disclosed that more than 20 different perceptions might be used to describe the phenomenon. For example, it might be characterised as adaptation (Schäffner, 2012), software translation (Dagienė et al., 2011), "as a distinct process or as a technological extension of translation-related phenomena" (Jiménez-Crespo, 2013, p. 15). Despite different perceptions available, this article does not intend to focus on the research on different perceptions and discussions of both the term and phenomenon of *localisation* and adheres to the point of view that localisation might be perceived as the process of adapting and tailoring software or some other product so that it would meet linguistic and cultural requirements of some target language, culture and market and would be both linguistically and culturally suitable to the target locale (Pym, 2006; Maumevičienė, 2018). The process

of localisation is rather complex and involves different stages of product, i.e., software preparation and development. This is related to software customisation, internationalisation (the process of generalizing and standardizing the product during the phases of product development and engineering when all culture specific information is neutralized), actual translation of linguistic content (menu items, language strings, graphic user interface, help files and documents and etc.), prototyping, testing and desktop publishing. Therefore, translation is a subset of all the tasks and activities the process of localisation involves. Localisation is usually some specific target language and culture or target languages and cultures related with some specific locales in mind. Taking this into consideration, instead of the term *localisation* synonyms, that would indicate any locale specific target languages and cultures, might be used to describe process of localisation. Due to this, within the Lithuanian discourse, the term *Lithuanisation* that means making the product or software Lithuanian, is frequently used when issues of software, webpage adaptation to the Lithuanian language and culture are discussed (Dagienė et al., 2011; Maumevičienė, 2012). The term *Lithuanisation* might be synonymously used to replace the term of *adapting* and is frequently applied within the linguistic context when toponyms, proper names and surnames and other words are Lithuanised by means of adding the Lithuanian language pertinent grammatical structures. However, the term of *Lithuanisation* is also employed when software is made Lithuanian when adding up various Lithuanian locale and language related settings as well as implementing the process of localisation for the Lithuanian language and culture.

Research Methodology

Though long and complete texts with figurative elements of speech are not characteristic and typical of the linguistic corpus of software, despite this, trope modifications, i.e. conceptual (visual and linguistic) metaphors, have been examined by means of applying a combined approach for the analysis: the semantic strategy of trope change as proposed by Chesterman (1997), the tool of multimodal transcription and the model of different modes of meaning construction that is composed linguistic, visual, audio, special and gestural modes (the New London Group, 1996). In addition to this, when trying to find out what changes the trope undergoes and how the Lithuanian language changes, Lithuanised language units of the localised version of *Oracle Business Suite* (further referred to as *Oracle*) software were compared with the original language strings. The software was developed in the United States by *PeopleSoft* enterprise that was established in 1987 and carried out its performance till 2005, when it was overtaken by *Oracle Corporation*. *Oracle* software is a fully integrated business software package that provides different functionalities and allows managing, coordinating and monitoring various organisation management related processes. Due to its features, software is used to increase the efficiency of the performance of various companies. The product was localised to the Lithuanian language and culture in 2008 after the purchase of the original software from the US company. The process of localisation took place in Lithuania and was organised as an in-house and in-context process with the representatives of both sides (the US and the Lithuania) working together as a multilingual and multicultural team. For the purpose of software localisation, a group of 6 Lithuanian translators was hired to deal with the linguistic content of the software after its actual customization and prototyping from the original version. Dealing with the linguistic content of the software translators had full access both to software, documents and other information and could meet with the developers of the software so as to discuss various issues that could have impact on the Lithuanisation of the linguistic content. Currently, the Lithuanised version of the software is used by different companies.

Aiming to carry out the research *Oracle* software, both the original product and the localised (Lithuanised) version were examined to describe translation strategies that were applied to adapt the original product to the Lithuanian language and culture. During the analysis of the target and source texts of the software, 84 852 translation units / language strings of the original and the corresponding Lithuanian language strings, menu items and components of graphical user interface were manually collected to compose the linguistic corpus of the research. Due to the amount of the software strings only elements that fall into the scope of the graphical user interface (GUI) were investigated. All the other components of software, such as online help files, documentation and other resource files were not taken into consideration due to a large amount of the data. The screenshots of the software and its GUI were manually collected in Microsoft Excel spread-sheets. All in all, the research corpus contains 84 852 translation units / language strings where the length of a language string might range from 4 characters to 5000 characters without spaces. Metaphors composed around one third of the research sample,

i.e., 25 401 (one metaphor equals 1 unit) in total; whereas, multimodal metaphors where usually the visual and linguistic modes are present, represent 12 001 units.

Multimodal metaphors were examined by means of applying the adapted and simplified tool of multimodal transcription since the screenshots of software usually contain visual frames, composed of visual images that describe what is actually seen by the software user and what functions of the software are performed. Next to the image the translation of the text into the Lithuanian language is presented. When examining the linguistic mode, delivery, vocabulary and metaphors, information structure, coherence relations were focused upon, whereas the analysis of the visual mode focused on colours, perspective, foregrounding, backgrounding and similar elements of visual design. Audio and gestural modes are not characteristic to software; thus, they were not included in the analysis. **Table 2** below demonstrates a simplified example of the multimodal transcription with translation.

Table 2 Example of multimodal transcription with translation

No	Visual frame	Visual image and linguistic mode (text, structure, coherence)	Translation into the Lithuanian language
1		<p>The pictures of a book and a table are used to describe a menu item that is general ledger and journals related. The visual pictures are supported by a textual explanation and the title of the menu item that provides a message for the users of the component to carry out different tasks with journals, hierarchically presented below by means of separate folder icons and hypertexts create, review and other.</p> <p>All linguistic items and messages are structured as statements. Logical coherence is retained when the title of the menu component is first introduced in a nominal case and every constituent item is indicated below in a hyperlinked text of green colour.</p>	<p>Žurnalai Sukurti, importuoti, peržiūrėti, apdoroti ir archyvuoti žurnalus</p> <p>Žurnalų kūrimas Standartiniai žurnalai Sistemos žurnalai</p>

This research applies qualitative research method. Due to a large scope of materials and the error of calculation since all language strings were manually collected, the idea to provide quantitative data was rejected. As mentioned above, this paper presents findings and research results that are related just with the Lithuanisation of software metaphors with an aim to find out what changes the trope undergoes when trying to localise the worldview that metaphors stand for.

The trope change as a type of semantic translations strategies in *Oracle* software has been examined applying the model of trope changes as proposed by Chesterman (1997). To differentiate between the source and the target texts (Lithuanised language strings) the acronyms SL (source language) and TL (target language) were used. In this paper, up to 16 visual and linguistic metaphors are discussed.

Results and Discussion

The analysis of the Lithuanised version of *Oracle* software provides different cases when metaphors are either adapted and assimilated, eliminated or changed to some other metaphors that are more characteristic to the Lithuanian language and culture. The research

results indicate that cases when the metaphor of the source text is preserved in the target text are numerous. The following examples below demonstrate that the trope is applied and retained in the Lithuanised language strings:

- SL[1]: Administration home
- TL[1]: Administravimo pradinis puslapis
- SL[2]: System architecture
- TL[2]: Sistemos architektūra
- SL[3]: Event Workbench
- TL[3]: Įvykio darbastalis
- SL[4]: Tree manager
- TL[4]: Medžio tvarkytuvė
- SL[5]: The Process Category %1 is currently disabled for the Server %2.
- TL[5]: Šiuo metu serveris %2 negali naudoti proceso kategorijos %1.
- SL[6]: Parent record / node
- TL[6]: Aukštesnio lygmens įrašas / mazgas
- SL[7]: Sibling record / node
- TL[7]: Žemesnio lygmens mazgas
- SL[8]: System memory
- TL[8]: Sistemos atmintis
- SL[9]: Search in post
- TL[9]: Ieškoti pašte

For example, conceptual metaphors of the source language strings, i.e., *system memory*, *event workbench*, *system architecture*, *tree manager* or *search in post* are retained and assimilated in the target language strings, i.e., *sistemos atmintis*, *įvykio darbastalis*, *sistemos architektūra*, *medžio tvarkytuvė* and *ieškoti pašte*. This is related to the fact that the metaphors mentioned above when software is compared to a human being and human capacities to remember and store information (the metaphor of *system memory*), or when software is compared to a house/building or a part of a building and a workshop (the use of the metaphors such as *system architecture*, *event workbench*, *search in post*), gardening (the metaphor of a *tree*) are related with software programming and technical solutions or functions that are mainly performed by software engineers and developers. At the same time it must be mentioned that adapting the metaphors mentioned above the grammatical strategy of literal translation was used as well.

The use of these metaphors is deeply entrenched in the Lithuanian language and have become dead metaphors that computer developers and software users hardly notice them as metaphors. Moreover, the metaphors were adapted as computer terms since representatives of computer science, software developers and providers were the first to use the terms in the Lithuanian language. It must be noted that there are numerous cases when computer and IT terms are adopted by the Lithuanian language from English due to the fact that the concepts and terms in the Lithuanian language are not created as fast as they should be. In many cases, language professionals are not as quick to react to create new words for the terms and computer engineers and developers start using those terms as professionalism and colloquial words. The terms or words become so popular among language users and this is how the items pass into the standardised language use and become approved by terminologists. The use of the same conceptual (structural or other) metaphors that describe the functioning of software with buildings, parts of buildings, institutions, workshop in the source and target texts indicate both the transfer of some conceptions and the influence of the product (software) and source language on the target language. Their adoption in the Lithuanian language reveals the fact that the categorisation of computer science and software related objects, the structure of some systems and perception in this particular case is retained with no modifications in the Lithuanian language and culture.

The analysis of various trope changes in both the original and Lithuanised software strings provides examples when metaphors of the source text are preserved in the target text. Examples 2, 3, 4, 8, 9 demonstrate cases of dead metaphors that are used in the Lithuanian language, i.e. *įvykio darbastalis* (*event workbench*), *sistemos architektūra* (*system architecture*), *medžio tvarkyklė* (*tree manager*), *sistemos atmintis* (*system memory*), *ieškoti pašte/paštas* (*search in post*). Some of the metaphors, such as *system memory*, *tree manager*, *post/mail* are rather natural, yet the metaphor *įvykio darbastalis* (*event workbench*) is rather unusual. Despite the fact that different tools, tables and tabs are used to create sourcing and public procurement events and the comparison of the process to work at the workshop and the *workbench* might be assumed to be logical, the Lithuanian compound *įvykio darbastalis*, though adopted, might sound unusual.

Moreover, it is interesting to note that some of the visual metaphors were also adopted with no changes. For instance, the icons  (a folder that represent software as a library of documents and files),  (a floppy disk that is not used any more but still represents the function of saving information or a file),  (the magnifying glass that represents the function of search),  (the icon of a letter that is used to represent the process of notification and information provision), and the like, need no explanations since they have become universally applied in various applications, software systems, search engines, such as *Outlook*, *Microsoft*, *Google*, and Lithuanian software users understand their meaning with no additional information or instructions. Yet, it is interesting to note that some visual modifications were still made. The original background colour of *Oracle* software was dark blue, whereas in the Lithuanised version the colour of green was selected for backgrounding colour with the icons foregrounded within. In addition to this, all the textual information (names of language strings, menu items and components as well as message texts) is presented in green font colour. It might be assumed that green colour was selected due to the fact that it is considered to be one of traditional colours in Lithuania that might be found in colour patterns of the traditional Lithuanian clothing, textile and hand-woven garments, belts.

However, there are numerous cases when visual metaphors as the following ones , , ,  and similar were modified by means of linguistic mode since they could be perceived ambiguously. For example, the first picture of people discussing together might be associated with meetings and calendars, job interviews and the process of hiring or firing or headhunting. The icon of the tool kit box and a screwdriver might be perceived as some setup part of the software and the last icon of a nice building might stand for the Parliament, the national library, some museum, theatre or some tax office. Nevertheless, all the visual metaphors are preserved in the Lithuanised version with no modifications due to the fact that specific names and descriptions follow the icons, i.e. Lithuanised field names are used to define and reveal the meaning of the icon and the picture. For instance, no additional explanations are needed to decode and understand the following software components:  **Pirkimo/pardavimo konkursai** that represents sourcing or public procurement that the item stands for. Linguistically, the English item *sourcing* was replaced by a Lithuanian word combination that is structured of three lexical items, *pirkimas* – *purchase/acquisition*, *pardavimas* – *sales* (here the nouns *pirkimai* / *pardavimai* – *purchase/sales* represent a converse) and *konkursai* – *public procurement*. The icon of tool kit with a screwdriver  **PeopleTools** is followed by a coined phrase *PeopleTools* that is a brand name of the software and indicates a part of software where help files and books are stored and kept for users. In terms of the linguistic item *PeopleTools*, the phrase is left untranslated and used as a loan/calque since no Lithuanisation is allowed due to the intellectual property rights of the brand name. Whereas the icon of a hierarchical structure and composition when followed by an explanation *medžio tvarkyklė* – *tree manager*  **Medžio tvarkyklė** and the icon of a palace followed by the name *bankai* – *banks*  **Bankai** are used to explain the functions of software components. The examples of visual metaphors above demonstrate that in the following cases localisation of the software was not carried out to its full extent when trying to replace the visual icons with the ones that would be meaningful to the Lithuanian software user.

Other examples, i.e., 1, 6, 7, entail a case when the trope of the source text is replaced with another trope in the target text. The original English syntagma in example 1, i.e., *Administration home*, is composed of a metaphor when parts of software, its subsystems or webpages are being compared to *a building*, *house* or *home* and the private environment. Despite the current COVID experience when private houses, rooms and private spaces were transformed into offices and workplaces, Lithuanians are not likely to associate their office with their home or house. This is related to the worldview and the mythical attitude shaped mind-set of Lithuanians when the

conception of *house* and *home* is related to a private and sacred space and ambience and fortress of soul that has to be preserved, protected and cherished (Vyšniauskaitė, 1999/2004). It is also conceived as the centre of the world where all the generations lived, live and will continue living (Gaudiešiūtė, 2007). Due to that the conception *namas* (a house), *namai* (home) determines the change of the trope in the Lithuanised language string where the metaphor of *home* is replaced with a Lithuanised metaphor *pradinis puslapis* (*the home page / first page*). In this case, the administration of software components is compared to a journal or a book and the action of turning, returning, opening or closing of the first page of the book, i.e. the main page of the software.

Examples 6 and 7 indicate the perception of software functioning as a family where certain hierarchical relationship and links are exist. The illustrations present cases when metaphors that exist in the source language are omitted in the Lithuanised versions. Various entries, lists, records and nodes of the software are metaphorically linked by means of *parent record/node*; *sibling record/node*. Such a comparison is used to define hierarchical display, order and levels of records and nodes. Yet, the analysis of the Lithuanised strings revealed the fact that the trope is dropped out in the Lithuanian version and is Lithuanised as follows, i.e. *aukštesnio lygmens įrašas / mazgas* (*record / node of a higher level*). Direct and literal translation of the English metaphor into the Lithuanian language would produce a strange word combination; therefore, the strategy of paraphrasing is applied. Instead of rendering the metaphors mentioned above as *tėvinis įrašas / mazgas* (*parent/father related record/node*), *brolinis įrašas / mazgas* (*sibling/brother related*) or *pirminis įrašas / mazgas* (*primary or the first record/node*), *antrinis įrašas / mazgas* (*secondary or the second record/node*) the hierarchical levels of nodes and records are explained by means of a word combination *aukštesnio/žemesnio lygmens mazgas* (*higher level/lower level node*). Due to the omission of the trope in the Lithuanised software string, the message becomes comprehensible.

Example 5 presents a case when a metaphor is introduced in the target text though it is not used in the original software text. The Lithuanian version contains a personification when the *server* is compared to a person, office worker or administrator who is or is not able or cannot run, complete and perform certain processes, i.e. *Šiuo metu serveris %2 negali naudoti proceso kategorijos %1* (*the Server %2 cannot currently use the Process Category %1*). The use of the trope in the Lithuanised sentence is determined by the application of both the semantic strategy of the trope change when the server is being conceived as an animate (human) being, capable of completing various tasks, and the syntactic strategy of clause structure change, which modifies the passive voice of the original language string to the active voice in the Lithuanian language string. Active voice in the Lithuanian version of the string is used to change the subject of the sentence. If in the source language string the lexeme *server* is used as the object, i.e., the doer of the action, it becomes the subject that carries out and performs the action in the target language string. The clause structure change also transforms the perspective of the original and the Lithuanian sentences, since agents of the sentences change.

Conclusions

The analysis carried out identified that the typology of translation strategies as proposed by Chesterman (1997) that is usually used to analyse translations of literary texts can be successfully applied to examine the linguistic adaptation of the linguistic corpus of the software. In addition to this, the typology can be combined with other instruments of analysis, such as the tool of multimodal transcription and a set of different modes might be applied to study software texts that express meaning and communicate with the software users by means of different modes.

The examination of the occurrences of trope change as a semantic strategy of translation and Lithuanisation in the target (Lithuanian) language strings evidences the fact that conceptual (dead) metaphors that might be represented by means of linguistic and/or visual modes and their changes are associated with micro and macro-level structural changes of the software, i.e. information, explicitness, layout change and the like. Information, explicitness and illocutionary changes as types of pragmatic strategies often follow the translation strategy of trope change, especially in those cases when the trope, i.e., metaphor, is changed by another metaphor in the target language and culture and introduced, when no metaphor is used in the source language. In cases when metaphors are assimilated and adapted in the target language with no changes, the semantic strategy of trope change is accompanied by the grammatical strategies of translation, i.e., loan, calque, literal translation or other. The analytical insights also evidence that trying to localise different software metaphors cultural filtering and information are applied and this demonstrates how the Lithuanian language modifies the product localised.

The analytical insights obtained during the research of translation strategies demonstrate that the language and culture of the product affect the target language and culture. The trope change as strategy indicates the impact of the American culture and language. However, at the same time, the Lithuanian language and culture adopts IT-related terminology, metaphors and new realia that reflect US-based thinking. The assimilation of new business models to handle processes in organisations that are related to software use confirms the influence of the product and American culture on the Lithuanian language and culture.

However, at the same time, the analysis of the trope change as a strategy indicates also the fact that not all conceptual models are assimilated and certain cases provide instances when software elements undergo Lithuanisation by means of replacing one conceptual system of the source language and culture with another conceptual system of the target language and culture. In such cases software elements and parts undergo culture specific changes, triggered by both the internal features of the Lithuanian language (prevailing grammatical constructions and syntagmatic and paradigmatic word relations) and the Lithuanian culture and worldview.

References

- 1 André, J. St. (2011). Metaphors for Translation. In: Y. Gambier, & L. Van Doorslaer (Eds.), *Handbook of Translation Studies Vol. 2* (pp. 84-87), Amsterdam and Philadelphia: Benjamins. <https://doi.org/10.1075/hts.2.met1>
- 2 Baldry, A., & Thibault, P. (2006). *Multimodal Transcription and Text Analysis*. London: Equinox.
- 3 Boyd, N. S. (2003/2013). Software metaphors. Retrieved February 16, 2022, from <http://www.educery.com/papers/rhetoric/metaphors/>
- 4 Chesterman, A. (1997). *Memes of Translation*. Amsterdam and Philadelphia: Benjamins. <https://doi.org/10.1075/btl.22>
- 5 Chesterman, A. (2005). Problems with Strategies. In: A. Károly, & A., Fóris (Eds.), *New Trends in TS. In honour of K. Klaudy* (pp. 17-28). Budapest: Akadémiai Kiadó.
- 6 Dagienė, V., Grigas, G., & Jevsikova, T. (2011). *Programinės įrangos lokalizavimo pagrindai*. Vilnius: Vilniaus universiteto leidykla.
- 7 Forceville, Ch. (2016). Pictorial and Multimodal Metaphor. In: N. M. Klug, & H. Stöckl (Eds.), *Handbuch Sprache im multimodalen Kontext//The Language in Multimodal Contexts Handbook*. Linguistic Knowledge series (pp. 130-145). Berlin. <https://doi.org/10.1515/9783110296099-011>
- 8 Forceville, Ch. (2006). Non-verbal and multimodal metaphor in a cognitivist framework: Agendas for research. In: Kr. Gitte, A. Michel, R. Dirven, & F. Ruiz de Mendoza Ibáñez (Eds.), *Cognitive linguistics: Current applications and future perspectives* (pp. 379-402). Berlin.
- 9 Gaudiešiūtė, V. (2007). *Namas - kaip pasaulio centras*. Retrieved January 16, 2022, from http://samogitia.mch.mii.lt/Zurnalas/2007_03/ZZ_2007_3_49_55.pdf
- 10 Gudavičius, A. (2009). *Etnolingvistika (Tauta kalboje)*. Šiauliai: Šiaulių universiteto leidykla.
- 11 Halliday, M. A. K. (1994). *An Introduction to Functional Grammar*. London: Edward Arnold.
- 12 Jiménez-Crespo, M. A. (2013). *Translation and Web Localisation*. London & New York: Routledge. <https://doi.org/10.4324/9780203520208>
- 13 Kress, G. (2010). *Multimodality: A Social Semiotic Approach to Contemporary Communication*. London: Routledge.
- 14 Kress, G., & Leeuwen, Th. Van. (2001). *Multimodal Discourse: The Modes and Media of Contemporary Communication*. London: Edward Arnold.
- 15 Lakoff, G., & Johnson, M. (1980). *Metaphors We Live By*. Chicago: The University of Chicago Press.
- 16 Lakoff, G. (1987). *Women, Fire and Dangerous Things: What Categories Reveal about Mind*. Chicago: The University of Chicago Press. <https://doi.org/10.7208/chicago/9780226471013.001.0001>
- 17 Maumevičienė, D. (2012). Linguistic insights into localisation. In: B. Fisher, & M. Nisbeth Jensen (Eds.), *Translation and the reconfiguration of power relations: revisiting role and context of translation and interpreting*. Münster: LIT Verlag (pp. 109-126).
- 18 Maumevičienė, D. (2018). Communication of cultures by means of localisation. In: O. Andreica, & A. Olteanu (Eds.), *Readings in Humanities - Arts and Humanities in Progress*, vol 3. Springer, Cham. https://doi.org/10.1007/978-3-319-66914-4_14
- 19 Oracle.com. (2015). *Database 12c | Oracle*. Retrieved March 5, 2017, from <https://www.oracle.com/database/index.html>

- 20 Pym, A. (2006). Localisation, training and the threat of fragmentation. Retrieved March 16, 2022, from <https://studylib.net/doc/7420545/localization--training--and-the-threat-of-fragmentation>
- 21 Pym, A. (2011). Translation research terms - a tentative glossary for moments of perplexity and dispute. In A. Pym (Ed.), *Translation research projects 3* (pp. 75-110). Tarragona: Intercultural Studies Group. Retrieved January 12, 2022, from http://isg.urv.es/publicity/isg/publications/trp_3_2011/pym.pdf Visited November 2021
- 22 Pym, A. (2014). *Exploring Translation Theories*. 2nd edition. London: Routledge.
- 23 Schöffner, Chr. (2012). Rethinking translating. *Meta: Journal des traducteurs*, 57 <https://doi.org/10.7202/1021222ar>
- (4) (pp. 866-883). Retrieved November 15, 2021, from <https://www.erudit.org/fr/revues/meta/2012-v57-n4-meta01064/1021222ar.pdf>
- 24 The New London Group. (1996). *A pedagogy of multiliteracies: Designing social futures*. Harvard Educational Review, 66(1), 60-92. <https://doi.org/10.17763/haer.66.1.17370n67v22j160u>
- 25 Thibault, P. (2000). The multimodal transcription of a television advertisement: Theory and practice. In: A. Baldry (Ed.), *Multimodality and multimediality in the distance learning age* (pp. 311-385). Campobasso: Palladino Editore.
- 26 Vyšniauskaitė, A. (1999/2004). *Lietuvio namai*. Vilnius: Lietuvos Liaudies kultūros centras.

Summary

Dainora Maumevičienė. Programinės įrangos metaforų lietuvinimas

Programinė įranga, kuri yra lokalizuojama, t. y. lietuvinama ir pritaikoma lietuvių kalbai, kultūrai ir jos naudotojams dažniausiai yra sukurta ne Lietuvoje, o Amerikoje įsikūrusiose programinės įrangos diegimo įmonėse. Todėl natūralu, kad originaliam produktui būdingi tam tikri požymiai (pvz., kultūrinės realijos, metaforos, teisiniai reikalavimai ir kt.), signalizuojantys ir atspindintys amerikietišką kultūrą. Programinės įrangos, kompiuterijos ir kituose su informacinėmis technologijomis susijusiuose tekstuose gausu metaforų. Programinės įrangos dalys, kompiuterio sugedimas, kompiuterio dalys, funkcijos, menu pavadinimai, komponentai ir pan. yra dažnai apibūdinami (konceptualiomis) metaforomis, kurios atspindi pasaulėžiūrą, pasaulio kategorizaciją ir suvokimą, būdingą tik tos kalbos vartotojams. Lokalizacijos esmė yra pritaikyti produktą tiksliniams naudotojams, jų rinkai, kalbai ir kultūrai ir kartu ir sukurti iliuziją, kad produktas tarsi buvo sukurtas naudotojui, naudotojo šalyje. Todėl atsižvelgiant į tai, kad metaforos, kurios yra naudojamos programinės įrangos ar programų dalyse ir atspindi tam tikrą pasaulio suvokimą, yra įdomu tirti, kaip jos yra keičiamos, kai produktas yra lokalizuojamas ir pritaikomas tam tikrai rinkai, kalbai ir kultūrai. Taip pat kartu kyla klausimas, ar galima lokalizuoti per metaforas išreiškiamą pasaulėžiūrą. Šio straipsnio tikslas yra ne tik ištirti ir apžvelgti, kaip yra lokalizuojamos programinės įrangos metaforos, kai programinė įranga yra lietuvinama, ar jos yra išlaikomos, asimiliuojamos, ir atsakyti į klausimą, ar galima „lokalizuoti pasaulėžiūrą“, bandant sukurti iliuziją, kad produktas tarsi buvo sukurtas tai tikslinei kalbai ir rinkai, t. y. lietuviams ir Lietuvai.

About the Author

DAINORA MAUMEVIČIENĖ

Assoc. prof. dr., Faculty of Social Sciences, Arts and Humanities, Kaunas University of Technology, Lithuania

Research interests

Localisation, Lithuanisation of software, translation studies, translation history, translation technologies, modern didactics in translation and interpreting teaching

Address

Mickevičiaus g. 37, 44249 Kaunas, Lithuania

E-mail dainora.maumeviciene@ktu.lt

