




РАЗДЕЛ IV. ПРИКЛАДНАЯ ЛИНГВИСТИКА
SECTION IV. APPLIED LINGUISTICS

DOI: 10.18413/2313-8912-2021-7-4-0-6

Dmitry Yu. Gruzdev¹ 
Viktoriya V. Dyomochkina² 
Alexander S. Makarenko³ 

Leveraging VBA in translation

¹ Military University
14 B Sadovaya St., Moscow, 115432, Russia
E-mail: gru@inbox.ru
ORCID iD: 0000-0003-2874-6309

² Military University
14 B Sadovaya St., Moscow, 115432, Russia
E-mail: dyomochkina.viktoria@yandex.ru
ORCID iD: 0000-0002-4820-9693

³ Military University
14 B Sadovaya St., Moscow, 115432, Russia
E-mail: alex-makarenko@yandex.ru
ORCID iD: 0000-0002-3101-8014

Received 16 August 2021; accepted 13 December 2021; published 30 December 2021

Abstract. The article is focused on finding ways for the VBA application in translation. Most of the macro code solutions available online date back to the early 2000s. Given the fact, the researchers decided to look into the usefulness of the instrument in today's environment, which has changed significantly over the two decades. No matter how elaborate modern translating tools are, MS Word is still there, and time and again translators turn to it to convey various ideas to another language. Most specialists make do with what the designer built into the word processor, but there is a Developer tab they fail to notice. This is an opening, deliberately left by the developers to let users tap into the program and adjust it as needed. The authors go consistently from the analysis of existing codes, to writing their own based on the needs of today's translators and eventually to suggesting approaches for harnessing the tool to the translators' benefit. The ultimate proof of the value of translation-tailored VBA macros comes in the form of an experiment, built around the problem of converting measuring units in technical texts, a major translation challenge, particularly in the English-Russian pair of languages. The application of the code resulted in faster and more accurate calculations during the conversions, let alone better quality of the translated text. Not least important is that the translators now could focus on conveying the sense and idea, instead of switching between the MS Word window and numerous online/offline dictionaries, converters, corpora, etc., interrupting the string of their thoughts every time they came across numbers. This alone justifies taking the translator out of the loop when it comes to




standard processes. Other noteworthy results include: 1) list of tasks where macro solutions will relieve translators of extra workload, causing premature fatigue and degrading performance; 2) on-the-job path for consistent and gradual mastering of the tool, encompassing copying, modification, and combination of existing macros; 3) prove of the precedence of the code logic over its form and role of strategies in the use of macros.

Keywords: Visual basic for application; Translation automation; Macro; Code; Measurement unit conversion

How to cite: Gruzdev D. Yu., Dyomochkina V. V., Makarenko A. S. (2021). Leveraging VBA in translation. *Research Result. Theoretical and Applied Linguistics*, V.7 (4), 66-81, DOI: 10.18413/2313-8912-2021-7-4-0-6

УДК 81`33

DOI: 10.18413/2313-8912-2021-7-4-0-6

Груздев Д.Ю.¹ 
Дёмочкина В.В.² 
Макаренко А.С.³ 

**Объектно-ориентированный язык программирования
в практике письменного переводчика**

¹ Военный университет
Ул. . Б. Садовая, 14, Москва, 115432, Россия
E-mail: gru@inbox.ru
ORCID iD: 0000-0003-2874-6309

² Военный университет
Ул. . Б. Садовая, 14, Москва, 115432, Россия
E-mail: dyomochkina.viktoria@yandex.ru
ORCID iD: 0000-0002-4820-9693

³ Военный университет
Ул. . Б. Садовая, 14, Москва, 115432, Россия
E-mail: alex-makarenko@yandex.ru
ORCID iD: 0000-0002-3101-8014

*Статья поступила 16 августа 2021г.; принята 13 декабря 2021г.;
опубликована 30 декабря 2021г.*

Аннотация. Статья посвящена поиску путей применения объектно-ориентированного языка программирования VBA в переводе. Отправной точкой исследования послужили многочисленные решения в сети Интернет, адаптированные под нужды переводчиков в начале 2000-х годов. Цель заключается в том, чтобы изучить релевантность этого инструмента в современных условиях. Какими бы совершенными ни были новейшие инструменты в арсенале переводчика, редактор MS Word все еще активно используется в работе. Большинство специалистов довольствуются базовыми функциями, будучи не осведомленными о наличии вкладки «Разработчик», которую программисты намеренно сохранили для того, чтобы пользователи могли настроить ПО под свои нужды. Авторы последовательно переходят от

анализа существующих кодов к изучению способов написания собственных с учетом потребностей современных переводчиков, предлагают стратегии освоения ресурса. Для практической проверки основных выводов авторы проводят эксперимент, для которого была выбрана наиболее сложная проблема в паре английский-русский согласно опросу профессиональных переводчиков, а именно – конвертация единиц измерения в переводе технических текстов. Применение подготовленного кода привело к более быстрым и точным расчетам при конвертации, высокому качеству текста на ПЯ в результате того, что переводчики более не были обременены необходимостью переключаться между окном редактора и онлайн/офлайн словарями, конвертерами, корпусами, прерывая ход своих мыслей всякий раз, когда они сталкивались с цифрами. На основании полученных результатов авторы приходят к главному выводу о целесообразности использования специализированных макросов для выполнения стандартных операций, не требующих когнитивной работы переводчика-человека. Среди других значимых результатов следует отметить: 1) список переводческих задач, для которых подходят решения на основе макросов; 2) стратегию последовательного освоения инструмента без отрыва от работы на основе копирования, адаптации существующих макросов и составления новых кодов; 3) определение приоритета логики кода над его формой, а также роли стратегии применения макросов в глобальном контексте решения переводческих задач.

Ключевые слова: Объектно-ориентированный язык программирования; Автоматизация процесса перевода; Макрос; Код; Конвертация единиц измерения

Информация для цитирования: Груздев Д.Ю., Дёмочкина В.В., Макаренко А.С. Объектно-ориентированный язык программирования в практике письменного переводчика // Научный результат. Вопросы теоретической и прикладной лингвистики. 2021. Т.7, №4. С. 66-81. DOI: 10.18413/2313-8912-2021-7-4-0-6

Introduction

In the modern world, the pace of life has become much higher in terms of operations and the volume of information processed. The trend has not spared translators, either. (Ubin, 2001: 64). Moreover, it is not uncommon for a translator to switch from one subject to another, with customers placing even higher demands on deadlines and quality. That is where electronic tools and software fit in (Shevchuk, 2013: 9). As a result of the rapid development of information technologies over the past two decades the professional image of the modern translator has sustained profound changes. This inevitably leads to a search for new reserves and resources to optimize and adapt the translation to new requirements.

Today, translators already have at their disposal a large number of software products from dictionaries to specialized software such as Translation Memory (TM) systems (Shevchuk, 2013: 70). However, with the emergence of the latter, the usual translation tools have by no means become a thing of the past. MS Word is still widely used. The potential for automating word processing, as well as adapting this software to translators' needs, is enormous because of the VBA programming language that the developers integrated into MS Office (Granell, 2015: 72). Meanwhile, many specialists, translators included, are not even aware of its existence (Blayney, Sun, 2019: 12). The purpose of this paper is to explore the potential of VBA in MS Word and to find ways to use it in translation.

Archives

The essence of translation automation is to minimize the human role and, with a single tap on a key or click of the mouse trigger an automatic chain of operations, such as searching or editing. This can be achieved, among other things, with the macros specifically created for frequently recurring tasks.

There are off-the-shelf solutions for translators on the Internet, compiled by translators. Most of them have become subjects of extensive discussions on professional forums such as Gorod Perevodchikov (Russian for *City of Translators*) and Proz (Gorod Perevodchikov, Proz). A further search will lead to the websites of the "programmers", containing scores of macros and templates. Each product was compiled by translators to meet their own needs. The developers do not claim them to be comprehensive solutions to global translation problems and the interface often falls short of user-friendly standards. However, there are complex products, full-fledged software complete with an installation file. For example, the website of translator T. Harvey Ciampi offers macros for some repetitive secondary tasks (Ciampi), combined into the Translator add-on for MS Word. This endows translators with assets to perform some text-editing operations in automatic mode, for example, replace commas with dots in decimals in English documents, or replace the space between a number and a unit of measurement with a non-breaking space.

Translation macros partially duplicate each other in functionality, which is not at all evidence of plagiarism, because in solving the same problem the developers have framed their ideas in different program codes. Meanwhile, this fact confirms the existence of

a translation issue and the need to find a solution through automation.

To understand the scope of translation tasks that can be shouldered by macros, a list of their major functions has been prepared (Gorod Perevodchikov, EnglisHelp, Pfeiffer, 2005):

- Searching the text for all fractional numbers separated by commas, and replacing them with dots (in the English format),
- Replacing the spaces between numbers and units with non-breaking spaces,
- Formatting the titles in the English style (the first letters of the words are in CAPs),
- Searching for untranslated text,
- Inserting text in MS Word with all formatting canceled,
- Calculating for a given text the number of so-called "translation" pages (the total number of characters, spaces included, divided by 1,800),
- Counting the number of paragraphs, images, tables, footnotes, etc.,
- Removing unnecessary spaces, including after and before opening parentheses and before punctuation marks,
- Removing extra tabs at the beginning of a paragraph,
- Extracting terms that are not in the Word spell-checker,
- Searching for a selected word or phrase in the MS Word document in Google, Yandex, BNC, Collins, Oxford, and other online resources,
- Searching for dictionaries in Google that contain certain words or phrases (in English, Italian, French, and German),
- Automation of searching for words and phrases in dictionaries, bypassing the copy-paste procedure.
- Searching for dictionaries and parallel texts online.

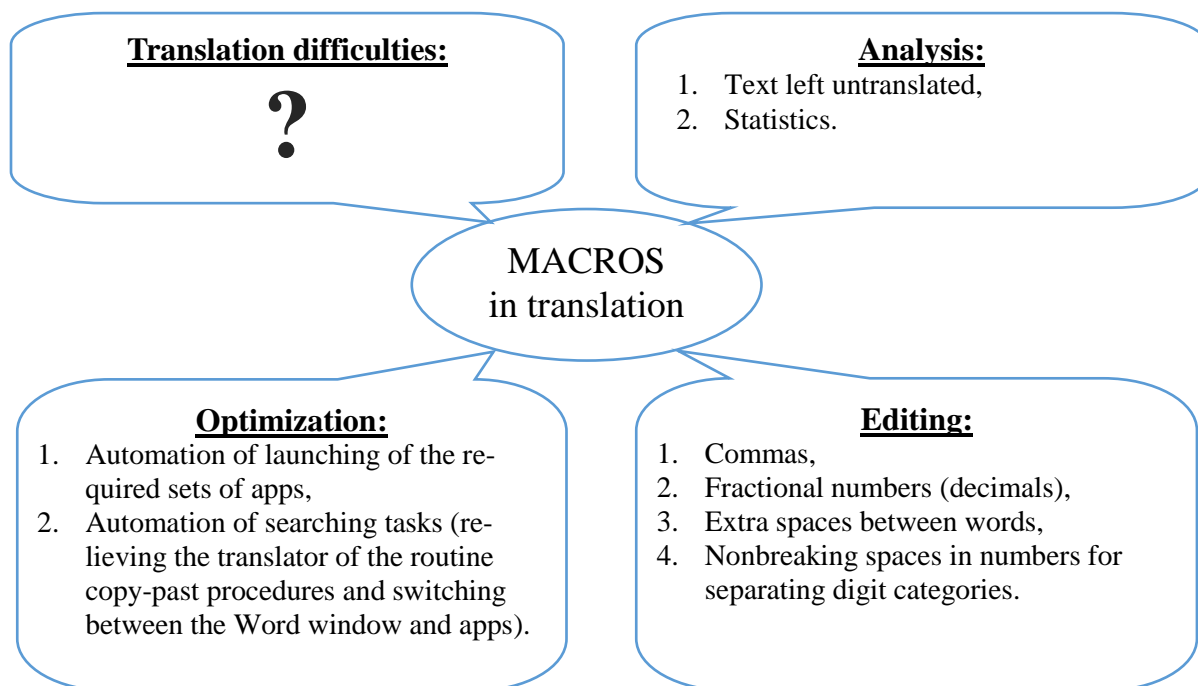


Figure 1. Application of macros in translation.

All operations belong to the (1) analysis, (2) editing, and (3) optimization phases (see Figure 1). The analysis part is directly related to statistics the translator needs to estimate the cost or generate an invoice for the translation done. In MS Word's basic functions this is accomplished in the form of counting typed characters with and without spaces, as well as words. A macro will process the text and divide the resulting digit by 1,800 to get the number of standard pages (Pfeiffer).

Editing macros are suitable for application both before and after translation, relieving the translator from mechanical, time-consuming operations with the text, including checking the text for repeated spaces, bringing decimal numbers to a common format as per the target language requirements (Katz, 2011: 45).

It is not fortuitous that macros for automating the search for linguistic data in reference software have been included as well. A good translator, no matter what "caliber", always keeps a large number of dictionaries and reference materials handy. The capacity of modern resources, packed into specialized programs featuring a simple and user-friendly

interface, has gone a long way in making the process of finding the required linguistic information simpler and faster. However, today's arsenal counts more than few programs, resulting in several windows which a translator has to juggle with, whenever he needs to check a word, its compatibility or find a synonym, pressing the copy-paste combination time and again. While requiring little cognitive resources, this chain of standard procedures takes its toll on the translator who often spends hours at the computer.

Most Russian translators will immediately remember the offline version of Multitran, where the developers partially simplified this task by making the program automatically paste the copied text into the query field. Of course, this feature dramatically simplified the navigation among open windows, although the Alt+Tab key combination was still needed to jump to the desired program.

Over time cloud and online technologies matured, providing around-the-clock access to up-to-date information. Inevitably it appealed to everyone, translators

included. As a result, preference was given to cloud solutions, relegating the classic, offline resources to a supporting role. While gaining in the data access field, translators obviously lost in making their digital workspace leaner and better organized. Now they need not only to switch to a web browser (which has become a single interface for all the web resources) but also select the required tab. This resulted in a growing number of keyboard and mouse manipulations, increasing the search time. The solution came in some dated macros written by Alexander Volkov (GiGatran.ru). They allow to assign key combinations to copy selected words to the search windows of Google or Yandex, online dictionaries Oxford, Collins, and the

Table 1. Code for automatic queries in Google

№	Code	Function
1.	<pre>Option Explicit Private Declare PtrSafe Function _ ShellExecute Lib "Shell32.dll" _ Alias "ShellExecuteA" (_ ByVal hWnd As Long, _ ByVal lpOperation As String, _ ByVal lpFile As String, _ ByVal lpParameters As String, _ ByVal lpDirectory As String, _ ByVal nShowCmd As Long) As Long Private Declare PtrSafe Function _ FindWindow Lib "User32.dll" _ Alias "FindWindowA" (_ ByVal lpclassname As Any, _ ByVal lpCaption As Any) As Long</pre>	The universal part of the code
2.	<pre>Public Sub goGoogle() goFind _ "http://www.google.ru/search?q=", 1</pre>	Request code

Another optimization idea was picked at K. Yeltsov's workshops (RosNOU), focused on tailoring and writing macros for problems arising during translation. The portfolio of the practicing interpreter and translator offers multiple solutions, from the simplest unit converters to a DIY TM system (Yeltsov). At one of his lectures, K. Yeltsov voiced the idea of controlling all the programs on the PC

British National Corpus (BNC). The codes provided a way to bypass the chain of tedious manual manipulations with the mouse and key board to get to the right window or tab and copy the query into the search line manually. Even for those who have not yet mastered VBA, understanding and cloning the code for expanding the search to other WEB-based resources is not at all a challenge.

Below is the code for automating the query-making process in Google (see Table 1). The code starts with a general part, i.e. a universal part, thus not subject to changes. Only the commands in the second line of the table need to be modified, namely the function name and web address.

through MS Word using VBA commands. Below is a code, located on the Internet, to run a third-party application:

```
'Dim RetVal
RetVal = Shell("CALC.EXE", 1) ' Run Calculator.
```

A very simple code, it can be easily modified to run any program on the PC. By replacing CALC. EXE with the path to any dictionary, Multitran, Oxfort, Collins, etc. the macro can be adjusted to run it from MS Word.

```
'Dim RetVal  
RetVal = Shell("C:\Program Files  
(x86)\mt\network\Multitran.EXE", 1) ' Run  
Multitran.
```

The next logical step will be to combine several application triggering macros in to a code for simultaneous launching:

```
Sub Launch()  
mt  
mx  
ox  
bnc
```

End Sub

This code, which can be reduced to a simple list of macro names for launching individual applications, can significantly speed up the preparation of the translator's workspace. Each professional has a different set of tools, which can vary from task to task. Most importantly, it is not one dictionary, but five or more. For different types of work, professionals tend to use a different set of tools, which can easily be combined into one macro, for example, according to the type or subject of translation. Thus, the next time the translator will not have to look through the list of his tools, clicking twice on the icons of the required applications. To get started, the only thing needed to be done is to open the text editor window and launch the required set of tools by clicking the icon on the ribbon or Quick Access Toolbar (see Figure 2).

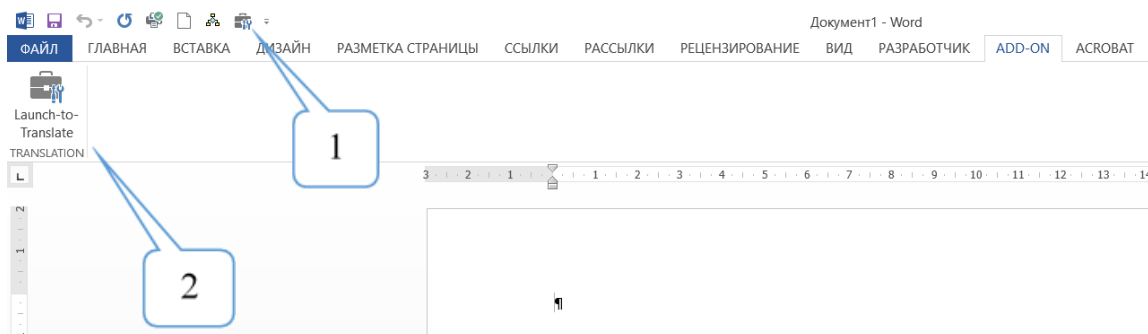


Figure 2: Macro icons for launching dictionaries and special translator software on (1) the Quick Access Toolbar and (2) the MS Word text editor ribbon.

Most of the solutions date back to the mid-2000s when the translator's toolkit got saturated with specialized software installed on the first truly personal machines. A search on professional forums will invariably lead to codes compiled during that period. Though all efforts to run these macros on modern PCs are doomed to failure as modern 64-bit systems do not accept them, the same forums offer a solution that boils down to adding **PtrSafe** (Table 1) to the code line Private Declare **PtrSafe** Function _.

This said, some functions of private VBA codes for automation of text editor

functions are already implemented in the basic MS Office features, for example, translation of words from the MS Word window in the online application. The idea remains the same, but the functionality does not fully meet the needs of a modern translator. True, it has become more convenient to have search results displayed in a right-hand field in the editor window. It does save the user an effort to go to the browser, although not so much as to abandon the old macros at all. The point is that the information is searched through the databases of the Microsoft Corporation. This is not

enough for a professional translator who understands that quality depends on, among other things, the plurality of reference resources at his disposal.

Case study

As much as all of the solutions discussed above make the burden of the modern translators easier, they are quasi-translation in nature. They increase the efficiency of analysis, text editing, and searching for linguistic information in third-party sources (Folaron, 2010: 424).

It is still required to establish the role of macros in overcoming translation difficulties. There is no better starting point than asking practicing translators. Of all the mentioned difficulties the need to convert units of measurement in the translation of military and technical texts proved to be the most suitable for the task for several reasons. First, the issue has already been partially solved in some old macros, so the work will boil down to modifying and refining the code. Second, this was the task prompted by K. Yeltsov to evaluate the VBA proficiency of those who

wanted to participate in his VBA workshops. The would-be attendees had to automate the conversion of all measurements from the Metric system into Imperial (Yeltsov).

Questionnaires done by military translators, particularly working in the Russian-English pair of languages, suggest that they often have to convert units of speed, temperature, volume, mass (weight), distance, area, force, power, density, etc. from the Anglo-American system to Metric and back when translating technical texts.

Most conversions that military translators have to deal with are limited to multiplication or division by a factor. Meanwhile, there are instances of inconsistent approaches to the measurement of certain indicators, which makes the task more of a challenge. For example, fuel consumption in Russia is commonly expressed in terms of the number of liters needed to cover 100km (l/100km), while the British will normally consider the number of miles a vehicle will travel on one gallon of fuel (MPG).

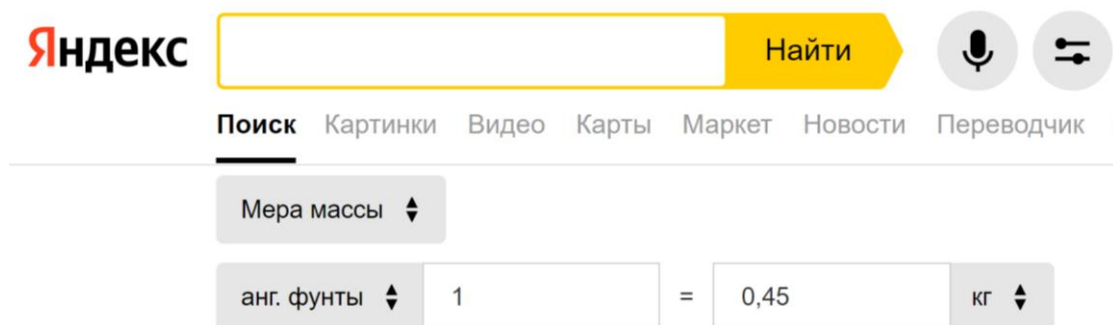


Figure 3. Unit converter in the Yandex search engine

The Internet is replete with various on-line converters, such as those based on the Yandex and Google search engines (see Figure 3), let alone scores of formulas websites have to offer. The built-in Windows calculator also features a converter function (see Figure 4). However, being yet another

application, it will make the not-so-short string of windows the translator has to juggle with even longer. This will mean an additional distraction from the translation, eventually adding to the fatigue of the specialist.

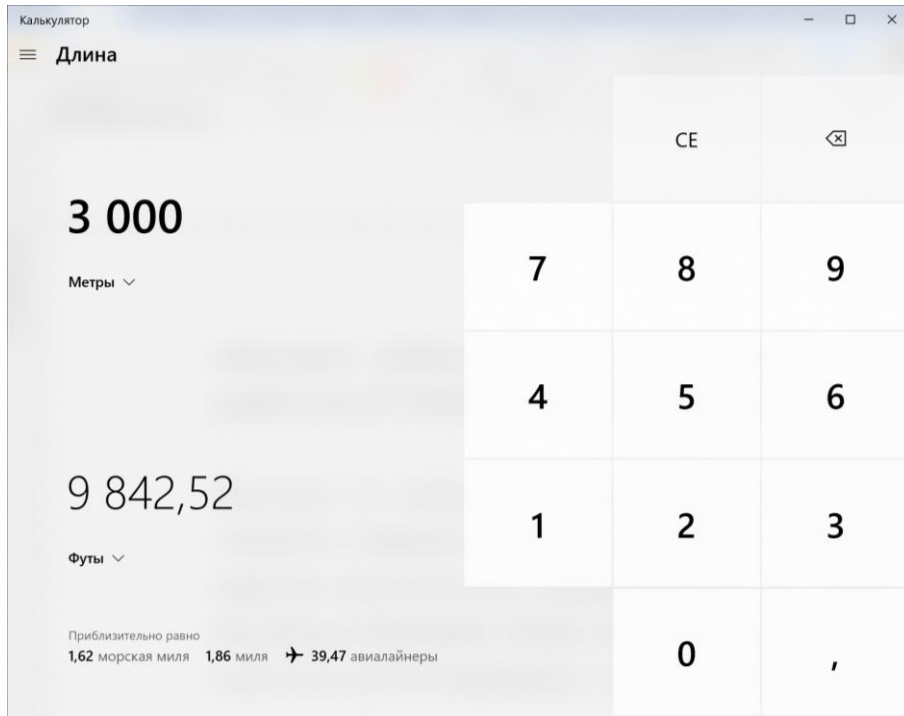


Figure 4. Unit Converter function in the Calculator App.
 Length measures.

Moreover, many units that military and technical texts abound with are not covered in standard software. These include ton-force (tf) or kilogram-force (kgf), in which the thrust of an aircraft engine is measured.

Based on these considerations of unit conversion challenges, a macro was prepared

(see Table 2). If there is a unit to be converted, the macro will be activated with a combination of keys. In this way, the translator will carry on with the translation while not distracted by extra manipulations to select appropriate units of measurement in the Calculator or online converter.

Table 2. Unit conversion code

№	Code	Function	Note
1.	<pre>Sub main() TranslSi "lbf-ft", "H-m", 1.35582 TranslSi "mph", "km/h", 1.60934 TranslSi "gallons", "l", 3.78541 TranslSi "MPG", "l/100km", 235.215 TranslSi "in3", "cm3", 16.3871 TranslSi "pound", "kg", 0.453592 TranslSi "inches", "cm", 2.54 TranslSi "feet", "m", 3.2808 End Sub</pre>	Unit conversion factors	
2.	<pre>Sub TranslSi(FindWord As String, ReplaceWord As String, sDelit As Double)</pre>	Code n executio pattern	The utility part of the code

№	Code	Function	Note
3.	<pre> Dim MyRange As Range Dim cnt As Long Dim iExit As Boolean Dim FindNumber As Double Dim ReplaceNumber As Double Dim slovo As String Dim sText As String Dim sNumber As String Selection.HomeKey unit:=wdStory FindWord = "lbf-ft" ReplaceWord = "N-m" sDelit = 1.35582 </pre>	Introduction to the conversion of round-forces to newton-meters	
4.	<pre> With Selection.Find .Text = FindWord . MatchWholeWord = True . Replacement.Text = ReplaceWord .Wrap = wdFindContinue If sDelit = 0 Then .Execute Replace:=wdReplaceAll Else .Execute Replace:=wdReplaceOne Do While .Found cnt = 0 iExit = False Parent.Collapse Direction:=wdCollapseStart Parent.MoveLeft Do While iExit = False Parent.MoveLeft If . Parent.Text Like "[0- 9\,\ \.]" Then cnt = cnt + 1 Else Parent.MoveRight Parent.MoveRight cnt = cnt - 1 </pre>	Search and conversion	The utility part of the code

№	Code	Function	Note
	<pre> iExit = True End If Loop Parent.MoveRight unit:=wdCharacter, Count:=cnt, Extend:=wdExtend If Trim(. Parent.Text) <> "" And Asc(. Parent.Text) <> 160 Then FindNumber = CDbI(Replace(. Parent.Text, ".", ",")) ReplaceNumber = FindNumber * sDelit Selection.Text = Format(ReplaceNumber, "##0.00") Selection.Find.Wrap = wdFindContinue Selection.Find.Execute Replace:=wdReplaceOne Else Selection.Find.Wrap = wdFindContinue Selection.Find.Execute End If Selection.Find.Wrap = wdFindStop Selection.Find.Execute Loop End If End With </pre>		

There are two obvious strategies for applying the unit conversion macro, which dictates the executive part of the code, i.e. (1) running the code in manual mode upon coming across values in the text or (2) applying the macro to the whole document (a) before or (b) after the translation.

Option 2a is of particular interest. In this case, the translator will be able to notice

different approaches to the measurement of individual characteristics of equipment and come up with a more interesting translation solution to make the target text sound more English or Russian, depending on the translation direction. Again, it calls for the example of the fuel consumption measurement in Russia and the UK.

Original	Translation
<p>Расход топлива танка Т-72 следующий:</p> <ul style="list-style-type: none"> • при движении на грунте – 260 л/100км; • на дороге с твердым покрытием – 240 л/100км. 	<p>Fuel consumption of the T-72 tank is as follows:</p> <ul style="list-style-type: none"> • when driving on the ground - 260 l/100 km; • on paved roads - 240 l/100km.

Those treading first steps in the trade tend to appeal to the lack of need to convert units from the international SI system, which is a fair remark. In the USA, for example, official technical and operational documents abandoned the Royal system in favor of the metric system of measurements (Clancy, 2001: 112). Nevertheless, Tom Clancy, the author of a documentary series of books focused on almost all services and arms of the U.S. Armed Forces, notes the "old guard" still sticking to the traditional for the U.S. approach.

Under the circumstances, the translator must always consider the pragmatic aspect. A specialist accustomed to measuring fuel in terms of miles per gallon will interpret the meaning in that particular way whenever coming across the "fuel consumption" characteristic, despite the presence of 1/100km after the numeral. This scenario of text perception is common when a quick decision is required, for example, by a logistician calculating the amount of fuel for friendly troops' equipment arriving for a joint exercise.

The stylistic aspect is also worth considering. Numerous figures with units of measurement "tear up" the text, forcing the reader to make an extra effort to digest and analyze the information. In Russian, it is the norm in scientific and technical texts, whereas in English it is acceptable only in tables and charts. In the text, units are preferably unfolded and integrated, for which understanding some measurement approaches is of the essence. The application of the converter before translation provides the result as follows:

Расход топлива танка Т-72
следующий:

- при движении на грунте – **0.9 mpg**;
- на дороге с твердым покрытием – **0.98 mpg**.

Based on the analysis of the measurement units and the above rationale, the following translation is suggested:

The T-72 will **travel** 0.9 m and 1 m on one gallon in cross-country and on paved roads respectively.

If the macro were used after the translation, the structure of the Russian sentence would interfere, with any chance to improve the stylistic aspect of the target text slipping away (Komissarov, 2011: 153). Moreover, in this scenario the numbers and units would have to be left untouched for the macro to pick them from the text:

The fuel consumption of the T-72 is as follows:

- in cross-country – **260 л/100км**;
- on paved roads – **240 л/100км**.

Should it be decided to work on the stylistic aspect in the process of translation without applying the macro beforehand, it will run contrary to the pragmatic adaptation logic, because the code will not be able to recognize the measurement unit afterwards:

The T-72 will **burn** 260l and 240l every 100km traveled in cross-country and on paved roads respectively.

To evaluate the effectiveness of macros in general and this code in particular in the work of translators, an experiment was conducted. Undergraduates of the Foreign Language Department of the Military University were asked to take part in it. All informants are characterized by the leading instructors positively, have good and excellent grades in the specialization subjects. The experiment was broken down into two phases. In Phase 1 the group was offered to translate a technical text, 1,100 characters long, replete with characteristics of military equipment on one of the passed topics. The task was to complete the translation as fast as possible without compromising the quality. All units had to be converted to the English system of measurements, for which the informants decided to use the standard calculator. There were no restrictions on the use of dictionaries, reference books, and DIY corpora, in particular. Being a large selection of texts in a specific area, sorted out and put together for linguistic researches, the latter proved to be

instrumental in overcoming multiple translation problems, though some found their way to the target texts (Shevchuk, 2009: 52).

The analysis of the translations revealed the following shortcomings:

- Inconsistency in the choice of units of measurement (feet and miles were found in the target texts without a good reason for the differentiation).

- Errors in the abbreviations of measurement units.
- Incorrect formatting of numerical data (no commas separating digits higher than 1,000 or, on the contrary, use of commas instead of dots in decimals in the English texts).
- Errors due to inaccurate converter settings.

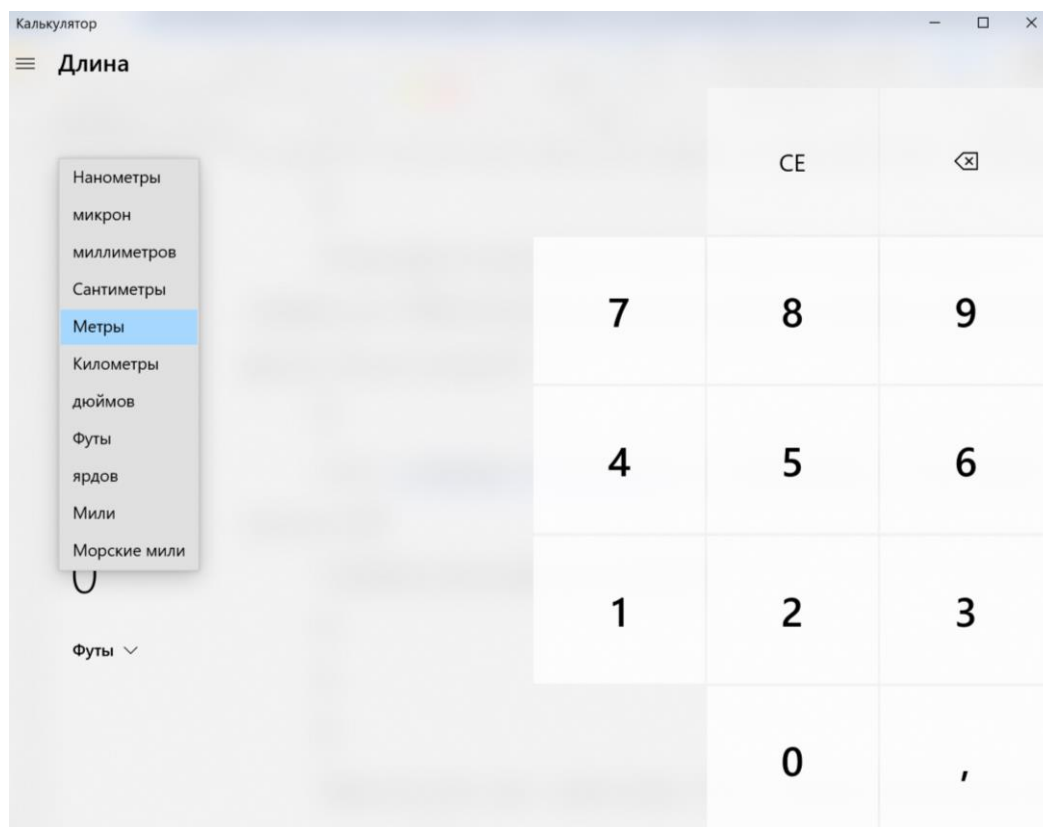
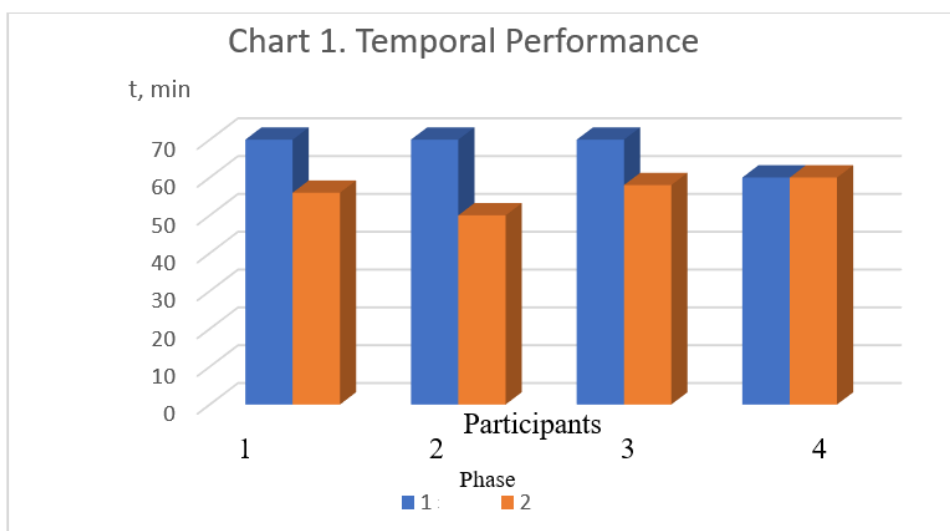


Figure 5. Unit Converter function in the Calculator app.
Drop-down list of measurement units.

One informant typed 11,482.94 instead of 11,482,940 when converting 3,500 kilometers into feet because the original unit of measurement were incorrectly set in the calculator. It follows from the explanation that the error is blamed on the proximity of meters and kilometers in the drop-down menu. Due to the lack of the necessary experience in converting units the inconsistency in the resulting value and unit never flagged the attention of the translator.

Follow-on conversations confirmed the original assumptions of the experiment. The

interpreters noted that the need to switch between windows distracted them from the translation process, and the calculator in particular called for numerous additional adjustments. Moreover, built-in programs and some online converters did not provide for certain categories of complex units, as a result of which the informants either chose not to convert at all or proceeded “blindfolded”. Some adopted the approach to deal with kgf, resorting to the algorithm for converting the weight and ignoring the “force” part.



The only difference in Phase 2 was that all numbers and units were converted by the macro. The translators ran it before they took up working on the text, taking the advantage of the results to choose better structures, while adding final touches during the translation. This proved to be instrumental in improving the quality of the target text and reducing time (see Chart 1).

VBA Workshop

To further pique interest among translators in the VBA capabilities of MS Word, it would be in order not only to demonstrate the simplicity of writing codes, but also to highlight several options of pursuing individual projects.

It has been repeatedly noted in some papers that logic always supersedes the elegance of DIY solutions (Gruzdev, Makarenko, 2019: 151). When the sequence of functions put together is observed, the cumbersomeness of the code, which may affect only the time of its execution, is of no particular significance.

MS Office developers retain this feature in their product only because they are cognizant of the fact that it is impossible to provide for all the needs of prospective users. Who better than an expert to understand how to render a string of standard commands into a leaner, more straightforward code, personalizing the word processor.

What would be the best approach to mastering VBA then? Given the simplicity of the tool, it will not be an exaggeration to say that the systematic and thorough approach based on reference books and manuals is an overreaction, threatening to kill the initiative (Mirzaxmedova, 2020: 9). The issue calls for a different approach. The nature of the VBA programming language and the developers' foresight in integrating hints and cues endows any Basic-savvy user with the ability to write the simplest codes (Zeng, Hrubes, 2017: 9). Thus, it is prudent to adopt an on-the-job training approach. Under the circumstances, the role of reference books and manuals will only be limited to the elaboration of particular functions in the process of programming. The rationale for choosing this particular format of mastering VBA is the availability of a few off-the-shelf solutions on the Internet. At this stage it is reasonable to indulge in reverse engineering, adapting codes to one's needs by, for example, changing variables. Further steps will be to refine and optimize the code based on alternative commands in reference sources (Stoter, Small, Rubin 2020).

No less promising within the chosen strategy of mastering VBA is the Record Macro function (see Figure 6), which captures the sequence of standard functions of the word processor and combines it into a single hot-key solution (Amvrossiev, 2018: 23). The potential of this resource is mainly applicable

when creating formatting macros, for example, to bring texts to conformity with the

customer's requirements.

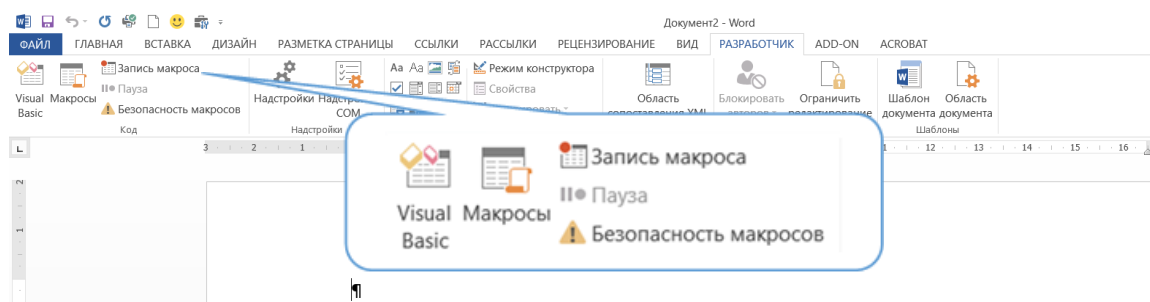


Figure 6. The Record Macro function on the MS Word Ribbon.

The concept remains the same: (1) code analysis in the VBA editor to increase awareness and programming literacy, followed by (2) code editing through the integration of non-standard functions.

Numerous solutions for automating processes in MS Word on forums will come in handy in mastering VBA and honing skills in programming new functions to meet professional needs on a personal computer. Understanding the VBA capacity will foster the search for new ways to harness its potential in translation, which will inevitably lead to writing more complex codes for automating some aspects of translation.

Conclusions

The analysis of some features of translation-oriented VBA programming resulted in several conclusions. First, the Internet is replete with off-the-shelf macro solutions. Although focused on overcoming common difficulties spanning beyond the job of translation, these can be harnessed for ad-hoc translation solutions. Better utilization of VBA functions is contingent on the understanding of the programming language capabilities and translation expertise. Secondly, VBA is easy to master within the scope of MS Office. The best approach is on-the-job learning through (1) reverse engineering of ready-made codes, most of which are open-source, (2) their adaptation to the task at hand, followed by (3) the compiling of a macro from several ready codes; (4) automatic recording of macros

using the built-in Record Macro function, its analysis, and improvement. Third, when working on a macro it is essential to zero in on the strategy for its future application. Automatically applying a code to a document before or after translation, or calling it on demand by clicking hotkeys while working on a text may directly affect the speed of translation and quality of the target text.

References

- Amvrossiev, G. (2018). Automation of the Microsoft's Office programs Espoo, Metropolia University of Applied Sciences, Helsinki, Finland, available at: <https://www.theseus.fi/bitstream/handle/10024/150553/AutomationOfTheMicrosoftOfficePrograms.pdf?sequence> (Accessed 11 September 2021). (In English)
- Blayney, P. and Sun, Zh. (2019). Using Excel and Excel VBA for Preliminary Analysis in Big Data Research, available at: https://www.researchgate.net/publication/331386736_Using_Excel_and_Excel_VBA_for_Preliminary_Analysis_in_Big_Data_Research (Accessed 11 September 2021). (In English)
- Ciampi, T. Internet Search Techniques for Translators, available at: <http://web.ticino.com/multilingual/internet/internet-course-for-translators.htm> (Accessed 7 August 2021). (In English)
- Clancy, T. (2001). *Special Forces. A Guided Tour of U.S. Army Special Forces*, Berkley Book, New York, US. (In English)
- Folaron, D. (2010). Translation tools, in Y. Gambier and L. van Doorslaer (eds.), *Handbook of Translation Studies*, John Benjamins Publishing, Amsterdam, The Netherlands. (In English)

Granell, X. (2015). *Multilingual Information Management: Information, Technology and Translators*, Chandos Publishing, UK. (In English)

Gruzdev, D. and Makarenko, A. (2019). VBA and Regular Expressions in Translation, *Modern Humanities Success*, 8, Makhachlala, Russia. (In Russian)

Katz, A. (2011). Automating Your Work with Macros, *Excel 2010 Made Simple*, Apress, Berkeley, CA, US. (In English)

Komissarov, V. (2010). *Sovremennoe perevodovedenie* [Modern Translation Studies], R. Valent, Moscow, Russia. (In Russian)

Mirzaxmedova, N.D. (2020). Use of Macros in VBA Programming Language in Power Point, *International Journal of Progressive Sciences and Technologies (IJPSAT)*, 23 (1), Madrid, Spain. (In English)

Pfeiffer, K. (2005). ATA Conference 2005: Session TAC-7 Introduction to Word Macros and Their Application, available at: https://www.ata-divisions.org/GLD/docs/46_konferenz/Word_Macros.pdf (Accessed 1 September 2021). (In English)

Shevchuk, V. (2009). *Jelektronnye resursy perevodchika* [Electronic resources of translators], ABBYY, Moscow, Russia. (In Russian)

Shevchuk, V. (2013). *Informacionnye tehnologii v perevode. Jelektronnye resursy perevodchika – 2* [Information technologies in translation, Electronic resources of translators – 2], Zebra E, Moscow, Russia. (In Russian)

Stoter, B., Small, M. and Rubin, L. (2020). VBA might be dying but it ain't dead, available at: <https://www.linkedin.com/pulse/vba-might-dying-aint-dead-lance-rubin> (Accessed 11 September 2021). (In English)

Ubin, I. (2001). *Sovremennye sredstva avtomatizacii perevoda: nadezhdy, razocharovaniya i real'nost'* [Modern means of translation automation: hopes, frustrations and reality, Translation in the modern world], VTCP, Moscow, Russia. (In Russian)

Yeltsov, K. VBA master-class, Homework, 1 level of complexity, available at: <https://www.youtube.com/watch?v=q5lJpyRzQG> U (Accessed 17 June 2021). (In Russian)

Yeltsov, K. About E-Stylebook technology, available at: <http://kirilleltsov.narod.ru> (Accessed 4 December 2019). (In Russian)

Zeng, H. and Klose Hrubes, A. (2017). *Using Excel and VBA with APIs to wow your colleagues and patrons*, Ex Libris Northeast Users Group Conference, Jersey City, New Jersey, U.S. (In English)

Forum EnglishHelp (2021). Translator - macro for translator, available at: <https://www.englishhelp.ru/component/content/article/92.html> (Accessed 28 August 2021). (In Russian)

Forum GiGatran.ru (2007-2014). Methods of work with the on-line dictionary Multitran, available at: <https://gigatran.ru/online-multitran-multytran-how-to> (Accessed 4 August 2021). (In Russian)

Forum of Professional Linguists ProZ (1999-2001). Available at: <https://www.proz.com> (Accessed 4 August 2021). (In English)

Professional forum of Interpreters Gorod Perevodchikov (2001-2021). Macros for translators, available at: <http://www.trworkshop.net/forum/viewtopic.php?f=16&t=4391> (Accessed 4 May 2021). (In Russian)

Winter School of Translation – 2019 (2019). RosNOU, available at <https://archive.rosnou.ru/important/transschool010219/> (Accessed 4 May 2021). (In Russian)

Конфликты интересов: у авторов нет конфликта интересов для декларации.

Conflicts of Interest: the authors have no conflict of interest to declare.

Gruzdev Dmitry Yuryevich, Deputy Head of the English Department, PhD in Linguistics, Associate Professor, Military University.

Dyomochkina Viktoriya Viktorovna, PhD Student at the English Department, Military University.

Makarenko Alexander Sergeevich, Head of the English Department, PhD in Linguistics, Military University.