Localization of selected FOSS4G desktop packages – state and some remarks

Milena Nowotarska
Poland OSGeo Chapter
`milena.nowotarska at gmail.com`

Robert Szczepanek
Poland OSGeo Chapter
`robert at szczepanek.pl`

Piotr Pachół
Poland OSGeo Chapter
`piotrpachol at gmail.com`

Keywords: localization, translation, QGIS, GRASS, gvSIG

Abstract

Free and open source software pretends to be easy to use for anyone. To accomplish that goal localization seems to be a must. Localization tools and state of localization for different FOSS4G packages are presented. Based on Polish GRASS, QGIS and gvSIG localization experiences, some proposals from the translator's point of view are outlined.

Introduction

Free and open source software (FOSS) is used all around the world, and as such it needs to be translated into different languages. A program which communicates with the user in one's own language is more friendly and intuitive. This is also the case in free and open source software for geospatial (FOSS4G). Usually it is the community which decides to take the burden of the localization of particular software. To make a program work in a given language, a couple of arrangements must be made. The workload is connected with two concepts: internationalization and localization.

Internationalization (i18n) is a process of such software design or adjusting, whereby it can potentially be adapted to different languages without making changes in the code. Internationalization is performed once embedded in the code by programmer. Instead of explicitly coding text strings, special functions for language conversion are implemented. This enables the proper display of fonts and characters, allows scripts to run from right-to-left, etc.

Localization (L10n) is a process of adding translations (and optionally other features or components) for a particular language. Localization can be efficiently implemented only if the program is internationalized. External resource files with translation, usually one per language, are used for an easy and scalable addition of new languages and modification of existing ones. Localization can be done simultaneously for many languages at the same time, even by non-programmers, and should be treated as continuous and never ending process related to program development.

Localization as process

In the localization process, several actors are involved – programmers, administrators and translators. We will focus mainly on translator's perspective.

As various FOSS4G projects consist of communities with their own regulations and internal rules, before any translation action, one should become familiar with those habits. A good starting point can be a project's web page which is related to the translation process [1] [2], a mailing list (general or dedicated) and direct contact with other translators. For larger projects, sometimes more than one person is involved in the translation for a particular language. In such cases, one of the translators coordinates the whole process. At this stage a person who is interested in localization should: subscribe to corresponding mailing lists, create an account on web-based systems, contact the main project translator and persons already involved in translation. As projects are still in developmental stages, there are just a few moments when there is nothing to do, so you can join a translation team any time.

If localization is web-based (online), from the technical point of view only an enabled web account is needed to start working. For desktop (offline) translations, the process is more complicated. We don't assume a situation when localization for a particular language starts from scratch, as this needs some additional actions. For any ongoing translation one must acquire the latest version of a language resources file from a central repository, using a web page or subversion control system (SVN). When talking about FOSS4G, those files are freely accessible by everyone without any restrictions. It is important to start working with the very latest version of a language file, taken just right before the translation.

The next step is the translation itself. At this stage, vocabularies, discussions with other translators and programmers, branch knowledge and the translator's experience are of main concern. It is a good practice to use already known branch terms, not to create new ones, unless we describe something new or original. For that reason, knowledge of similar programs with their terms is crucial. Good localization must consider target users and its main goal is to facilitate usage of the program. So sometimes, usefulness and simplicity of language is better than branch correctness. This refers to the program interface, but also to various forms of help including manuals. After changes in localization, the translator should check edited file(s) against a local copy of the program.

When making localization, sometimes the translator is not sure about some phrases. It is better to mark such translations, rather than just treat them as done. There are two possibilities in dealing with such situations, which are quite common. Every phrase can be in one of those three stages – translated, fuzzy (unfinished) or untranslated. In such cases, a fuzzy status can be assigned to a phrase translation. An additional option in some tools are comments, whereby the translator can describe her/his doubts in details.

As changes in language files are often made by several people, SVN or some other version control system is necessary. They help track translators' changes and merge uploads. If the translator has SVN write access, he can update language files directly, but usually the file is sent to one of developers who have such rights. Any changes in translation files should be submitted frequently.

Localization tools

Localization tools help in the development of internationalized versions of a program. They mainly include specialized editors for resource language files. The simplest versions just edit an appropriate file. More advanced editors include glossaries, translation memory, suggestions and spellcheckers. Traditional localization models require a local copy of the language file from the central repository and desktop program/editor. Such an offline model, as described in a previous chapter, requires several actions to update the translation file. At the moment, in the case of desktop FOSS4G, this is still the most popular model of localization.

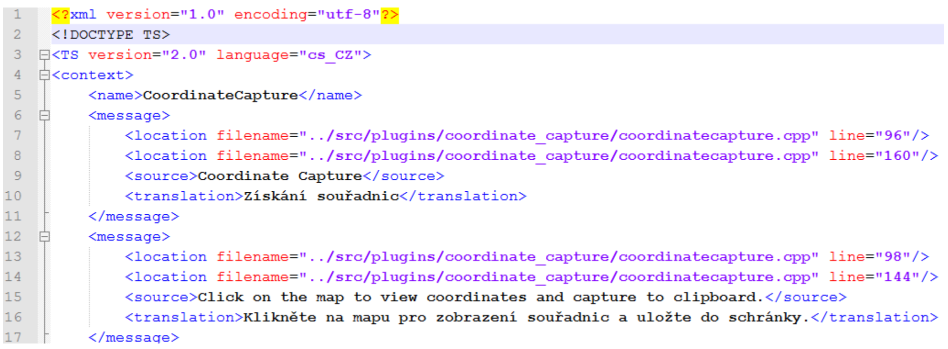
As software development tools take more and more advantage of Internet power, also on-line localization tools start to become used. In opposition to offline distributed translation, language files are localized directly through web pages. Functionality of this new method is still weaker compared to offline tools, but centralization of the localization process has many advantages.

Offline translation

Localization tools selection depends on programming language and programmers tools. The oldest format for language files is the Portable Object File (*.po*) used by gettext with corresponding binary *.mo* format. This method is used in the GRASS project which is a very mature one. In the Quantum GIS project, Translation Source Files (*.ts*) with corresponding *.qm* formats are used for localization. When desktop application development language is Java (like gvSIG), usually *.properties* format is used for localization. The newest XML-based *XLIFF* format is not very popular yet. When working with offline translation, one must be aware of used resources file formatting (fig.1) in order to select the appropriate tool. There are editors or more advanced computer aided translation (CAT) tools, which can work with almost any format of translation files. It is the choice of the translator to select his favourite tool – simple or advanced, dedicated or general purpose.

Qt Linguist

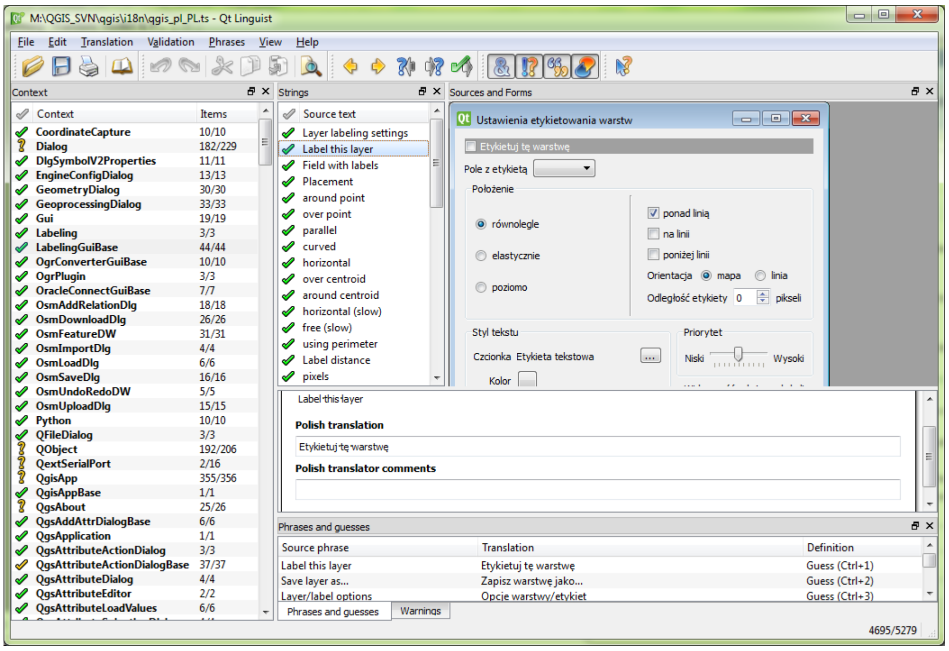
Qt Linguist is a tool integrated with the Qt development environment [3] by Nokia. One of the programs using this environment is Quantum GIS (QGIS). Qt Linguist is easy and user friendly yet very powerful. It is appropriate for beginners and advanced users with ergonomic



```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE TS>
<TS version="2.0" language="cs_CZ">
<context>
    <name>CoordinateCapture</name>
    <message>
        <location filename="../src/plugins/coordinate_capture/coordinatecapture.cpp" line="96"/>
        <location filename="../src/plugins/coordinate_capture/coordinatecapture.cpp" line="160"/>
        <source>Coordinate Capture</source>
        <translation>Získání souřadnic</translation>
    </message>
    <message>
        <location filename="../src/plugins/coordinate_capture/coordinatecapture.cpp" line="98"/>
        <location filename="../src/plugins/coordinate_capture/coordinatecapture.cpp" line="144"/>
        <source>Click on the map to view coordinates and capture to clipboard.</source>
        <translation>Klikněte na mapu pro zobrazení souřadnic a uložte do schránky.</translation>
    </message>
```

Figure 1. Sample qgis_cs_CZ.ts file for Czech localization of QGIS 1.5

interface. The translator can see the referring portion of the code or even see a corresponding widget in a preview (fig.2). This is a very unique and useful feature. Suggestions for translation are generated from existing strings and translation memory. Keyboard shortcuts are available, but without the possibility of personalization. The big advantage of Qt Linguist is the eventuality of the simultaneous display of translations for more than two languages. When translating from English to German, one may check the translation by looking, for example, at the Italian version.



Context	Items
CoordinateCapture	10/10
Dialog	182/229
DlgSymbolV2Properties	11/11
EngineConfigDialog	13/13
GeometryDialog	30/30
GeoprocessingDialog	33/33
Gui	19/19
Labeling	3/3
LabelingGuiBase	44/44
OgrConverterGuiBase	10/10
OgrPlugin	3/3
OracleConnectGuiBase	7/7
OsmAddRelationDlg	18/18
OsmDownloadDlg	26/26
OsmFeatureDW	31/31
OsmImportDlg	4/4
OsmLoadDlg	6/6
OsmSaveDlg	16/16
OsmUndoRedoDW	5/5
OsmUploadDlg	15/15
Python	10/10
QFileDialog	3/3
QObject	192/206
QextSerialPort	2/16
QgisApp	355/356
QgisAppBase	1/1
QgsAbout	25/26
QgsAddAttrDialogBase	6/6
QgsApplication	1/1
QgsAttributeActionDialog	3/3
QgsAttributeActionDialogBase	37/37
QgsAttributeDialog	4/4
QgsAttributeEditor	2/2
QgsAttributeLoadValues	6/6

Source text
Layer labeling settings
Label this layer
Field with labels
Placement
around point
over point
parallel
curved
horizontal
over centroid
around centroid
horizontal (slow)
free (slow)
using perimeter
Label distance
pixels

Etykietuj tę warstwę
Pole z etykietą
Położenie
[✓ równolegle] elastycznie poziomo
[✓ ponad linią] na linii poniżej linii
Orientacja [✓ mapa] linia
Odległość etykiety 0 pikseli
Styl tekstu
Czcionka Etykieta tekstowa
Kolor
Priorytet
Niski Wysoki

Label this layer

Polish translation

Etykietuj tę warstwę

Polish translator comments

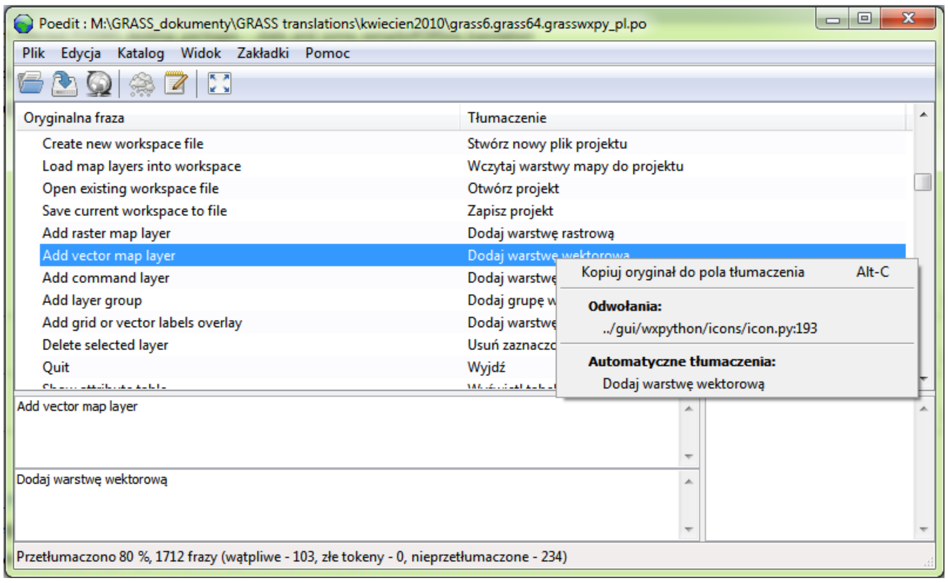
Source phrase	Translation	Definition
Label this layer	Etykietuj tę warstwę	Guess (Ctrl+1)
Save layer as...	Zapisz warstwę jako...	Guess (Ctrl+2)
Layer/label options	Opcje warstwy/etykiet	Guess (Ctrl+3)

4695/5279

Figure 2. Qt Linguist interface

Poedit

Poedit[4] is a simple, fast and easy tool for *.po* files editing. We use this program to localize the GRASS interface. It has a translation memory which can be generated from all of the *.po* files found on a hard drive and provides an automatic translation from the translation memory file. It is the user who decides how many words have to match during the automatic translation. Automatically translated strings result in a fuzzy status and are additionally marked with an icon to be easily found and checked by the translator.


Poedit : M:\GRASS_dokumenty\GRASS translations\kwiecien2010\grass6.grass64.grasswxpy_pl.po
Plik
Edycja
Katalog
Widok
Zakładki
Pomoc

Oryginalna fraza	Tłumaczenie
Create new workspace file	Stwórz nowy plik projektu
Load map layers into workspace	Wczytaj warstwy mapy do projektu
Open existing workspace file	Otwórz projekt
Save current workspace to file	Zapisz projekt
Add raster map layer	Dodaj warstwę rastrową
Add vector map layer	Dodaj warstwę wektorową
Add command layer	Dodaj warstwę
Add layer group	Dodaj grupę w
Add grid or vector labels overlay	Dodaj warstwę
Delete selected layer	Usuń zaznacz
Quit	Wyjdź

Kopiuj oryginał do pola tłumaczenia Alt-C

Odwołania:
../gui/wxpython/icons/icon.py:193

Automatyczne tłumaczenia:
Dodaj warstwę wektorową

Add vector map layer

Dodaj warstwę wektorową

Przetłumaczono 80 %, 1712 frazy (wątpliwe - 103, złe tokeny - 0, nieprzetłumaczone - 234)

Figure 3. Poedit interface.

Virtaal

Virtaal is a multiplatform tool, not related directly to any environment [5]. Virtaal can work with all translation file types mentioned in a previous chapter. The only problem is the stability and reliability of generated files. Sometimes this editor edits more than expected. The program is rather slow, but the general idea of an all-in-one toolset is excellent. Virtaal offers an online link to internet databases of translations. This is a very promising alternative for dedicated tools, but for the future, not now.

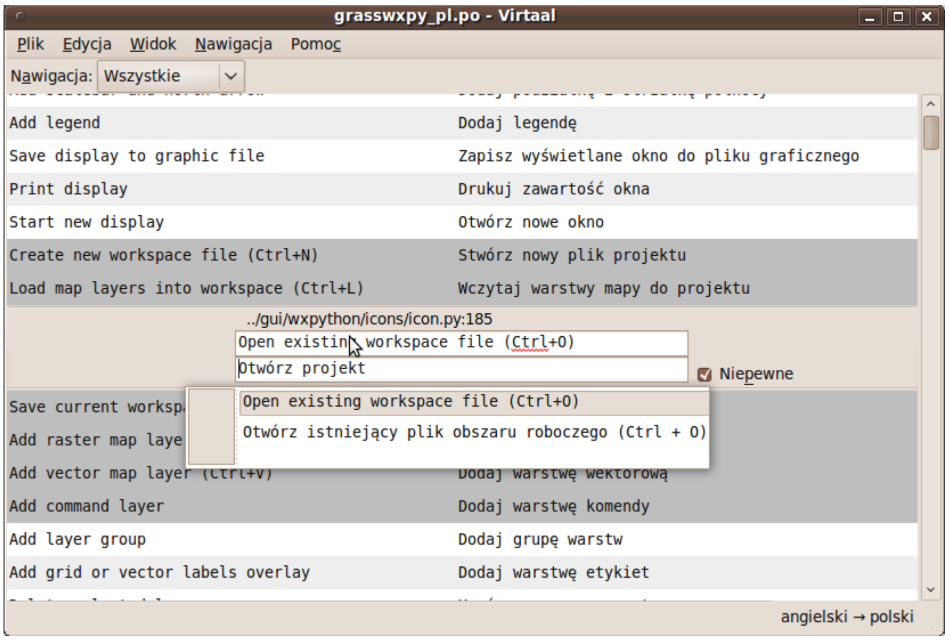
Online translation

Online tools use internet browsers as interfaces, thus they offer the same environment to all translators. When using online tools, no installation or manual update of programs is needed. After logging into the system, the translator just does his job without additional actions such as downloading files and sending changes to SVN administrators. Those actions are meant to be automated and beyond the translator's concern.

In some cases the difference between offline and online tools is hard to outline. The Open-Tran project [6], which is available online, is integrated with offline tools like Virtaal. Open-Tran provides recent translations from involved FOSS projects, and could be great for translations coordination within one domain/branch.

Transifex

This tool may be used for GRASS translation, as this project was recently added to the Transifex portal. Translation files can be downloaded/uploaded or edited online. The user

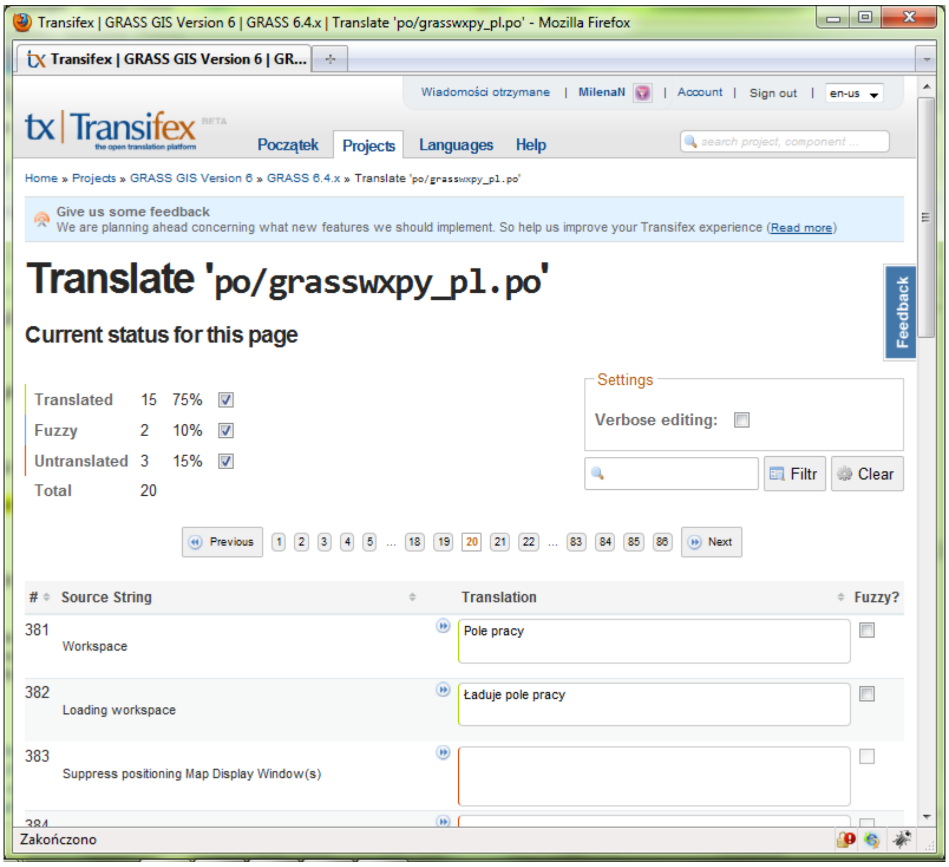


Add legend	Dodaj legendę
Save display to graphic file	Zapisz wyświetlane okno do pliku graficznego
Print display	Drukuj zawartość okna
Start new display	Otwórz nowe okno
Create new workspace file (Ctrl+N)	Stwórz nowy plik projektu
Load map layers into workspace (Ctrl+L)	Wczytaj warstwy mapy do projektu
Open existing workspace file (Ctrl+O)	Otwórz projekt
Save current worksp	
Add raster map laye	
Add vector map layer (Ctrl+V)	Dodaj warstwę wektorową
Add command layer	Dodaj warstwę komendy
Add layer group	Dodaj grupę warstw
Add grid or vector labels overlay	Dodaj warstwę etykiet

```
../gui/wxpython/icons/icon.py:185
Open existing workspace file (Ctrl+O)
Otwórz istniejący plik obszaru roboczego (Ctrl + O)
```

Figure 4. Virtaal interface.

can filter phrases by status (translated/fuzzy/untranslated).


Transifex | GRASS GIS Version 6 | GRASS 6.4.x | Translate 'po/grasswxpy_pl.po' - Mozilla Firefox
Wiadomości otrzymane | MilenaN | Account | Sign out | en-us
Transifex
Początek
Projects
Languages
Help
search project, component ...
Home » Projects » GRASS GIS Version 6 » GRASS 6.4.x » Translate 'po/grasswxpy_pl.po'
Give us some feedback
We are planning ahead concerning what new features we should implement. So help us improve your Transifex experience (Read more)
Feedback

Translate 'po/grasswxpy_pl.po'

Current status for this page

Translated	15	75%	☑
Fuzzy	2	10%	☑
Untranslated	3	15%	☑
Total	20		

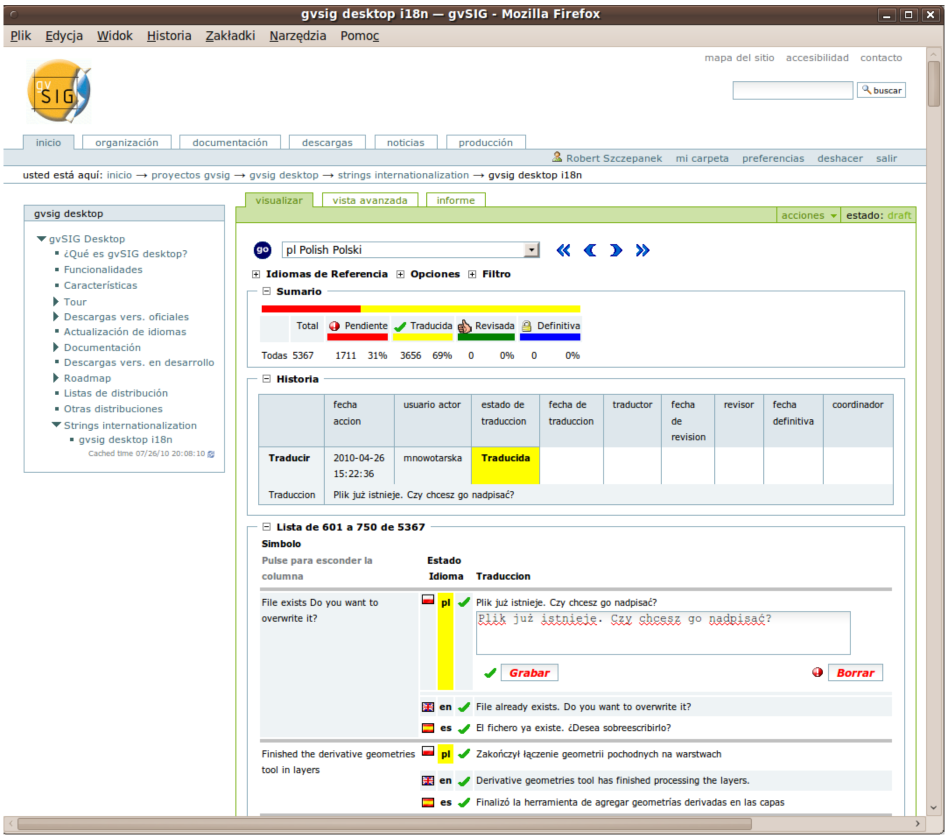
Settings
Verbose editing: ☐
Filtr
Clear
Previous 1 2 3 4 5 ... 18 19 20 21 22 ... 83 84 85 86 Next

#	Source String	Translation	Fuzzy?
381	Workspace	Pole pracy	☐
382	Loading workspace	Ładuje pole pracy	☐
383	Suppress positioning Map Display Window(s)		☐

Figure 5. Transifex interface

gvSIG i18n

Recently gvSIG developed an online translation portal [7] within its project portal. The translation interface is customizable and enables advanced searches. A useful feature is the definition of displayed languages (fig.6). The Web interface is simple, but commands and options are only in Spanish. This can be a problem for some translators. As no context information for phrase is available, translation for some phrases may be difficult. Translations are accepted in three steps: traducida, revisada, definitiva.


gvsig desktop i18n — gvSIG - Mozilla Firefox
Plik Edycja Widok Historia Zakładki Narzędzia Pomoc
mapa del sitio accesibilidad contacto
buscar
inicio organización documentación descargas noticias producción
Robert Szczepanek mi carpeta preferencias deshacer salir
usted está aquí: inicio → proyectos gvsig → gvsig desktop → strings internationalization → gvsig desktop i18n
visualizar vista avanzada informe
acciones estado: draft
gvsig desktop
gvSIG Desktop
¿Qué es gvSIG desktop?
Funcionalidades
Características
Tour
Descargas vers. oficiales
Actualización de idiomas
Documentación
Descargas vers. en desarrollo
Roadmap
Listas de distribución
Otras distribuciones
Strings internationalization
gvsig desktop i18n
Cached time 07/26/10 20:08:10
go
pl Polish Polski
Idiomas de Referencia
Opciones
Filtro
Grabar
Borrar

Sumario

	Total	Pendiente		Traducida		Revisada		Definitiva	
Todas	5367	1711	31%	3656	69%	0	0%	0	0%

Historia

	fecha accion	usuario actor	estado de traduccion	fecha de traduccion	traductor	fecha de revision	revisor	fecha definitiva	coordinador
Traducir	2010-04-26 15:22:36	mnowotarska	Traducida						
Traduccion	Plik już istnieje. Czy chcesz go nadpisać?								

Lista de 601 a 750 de 5367

Simbolo Pulse para esconder la columna	Estado Idioma		Traduccion
File exists Do you want to overwrite it?	pl	✓	Plik już istnieje. Czy chcesz go nadpisać? Plik już istnieje. Czy chcesz go nadpisać?
	en	✓	File already exists. Do you want to overwrite it?
	es	✓	El fichero ya existe. ¿Desea sobreescribirlo?
Finished the derivative geometries tool in layers	pl	✓	Zakończył łączenie geometrii pochodnych na warstwach
	en	✓	Derivative geometries tool has finished processing the layers.
	es	✓	Finalizó la herramienta de agregar geometrías derivadas en las capas

Figure 6. gvSIG i18n interface

Tools evaluation

Some of the localization tools are user friendly with nice and ergonomic interfaces (ex. QtLinguist). Some are simple with just basic functions (ex. Poedit). At the moment offline tools are much more sophisticated, but with new technologies implemented into web based tools things can change rapidly. A good example of such recent progress is gvSIG i18n.

Help on start-up is crucial for beginners. A very nice and easy tutorial is prepared by Virtaal. Also Qt Linguist provides a One Minute Guide.

Only one of the tested tools (Qt Linguist) is able to display the Graphical User Interface (GUI) thus simplifying the translation when working with a SVN copy of the repository. It is often hard to guess the correct translation without knowledge about the context. With a real interface preview it is even possible to check if the translation string fits in a GUI defined space.

All offline tools provide usage of translation memory. This mechanism attains use of already translated phrases. From available online tools, only Pootle provide this feature.

Very often within one translation file, strings are repeated, sometimes in the same context, but sometimes in different ones. If the context is similar, the translation should be uniform, at least within one program. In an ideal situation, this should be extended to all programs within one branch. Translation memory is one of the most useful features here. If connected with handy keyboard shortcuts, doing a translation becomes a pleasure. When working with larger translations, keyboard shortcuts also increase the effectiveness of the work.

Depending on a translator skills and the program to localize, different tools can be selected. Some of the tools provide only basic functions, which in some cases can be an advantage. From our experience, the following weaknesses of presented tools are the most significant:

- Qt Linguist – Sometimes broken links to the GUI preview on Windows.
- Poedit – Bad shortcuts and not handy, slow proposals.
- Virtaal – Slow response. Spoiling .ts files. Less control with the header in .po files.
- Transifex – After online translation sending back broken .po files with a notice not understandable for a common translator.
- gvSIGi18n – In Poland we received a slow response from the whole central gvSIG web system.
- Pootle – No keyboard shortcuts.

If we had to choose just one tool, our choice would be Qt Linguist. Unfortunately it is focused mainly on a Qt development environment.

Localization state of selected desktop FOSS4G

As a main area of our interest in FOSS4G we decided to choose three projects developed within Open Source Geospatial Foundation (OSGeo)[8] – GRASS, Quantum GIS (QGIS) and gvSIG. All of them are mature and well known GIS programs. Some of the projects (QGIS) release new versions every six months, while others do it not so often. To make it independent from release frequency, we made the measurement for a SVN developers trunk. Localization is a continuous process, but there are periods of increased activity, just before the release. This relatively short period of about two weeks is named "call for translation" and is dedicated mainly to translation improvements. A presented snapshot was made for the 23rd of April, 2010.

Translated phrases can be in one of three states: untranslated, fuzzy or translated. Only those with 'translated' status were counted. As we evaluated three different desktop programs, different tools were used for counting. GRASS *.po* files were analyzed with Poedit, QGIS *.ts* files with Qt Linguist and gvSIG *.properties* files with gvSIGi18n web statistics.

As a first, general approximation, we compared the localization progress with the total population of top languages of world speakers [9]. The results are presented in Table 1. For QGIS and gvSIG there is just one localization file, while for GRASS we decided to select one of four – GRASSwxpy version. From the potential users perspective, the worst situation is for Hindi, Bengali and Arabic languages. Assuming a threshold of 70% of localized phrases, we get a number of potential users between 891 000 000 for gvSIG and 1 191 000 000 users for QGIS.

Language	**Number of speakers (millions) [9]**	**GRASSwxpy**	**QGIS**	**gvSIG**
Chinese	1 213	1	31	33
Spanish	329	80	79	90
English	328	100	100	89
Arabic	221	–	3	-
Hindi	182	–	–	-
Bengali	181	–	3	-
Portuguese	178	48	74	37
Russian	144	70	76	83
Japanese	122	82	99	–
German	90	90	98	84
Indonesian	85	61	5	-

Table 1. Localization progress (%) of selected desktop FOSS4G against the top world languages.

Our goal was to estimate the extent of selected FOSS4G localization. In terms of space and population of end users. Where the top world's languages are spoken is shown in Figure 8. The next figures contain all localized languages, even tiny ones.

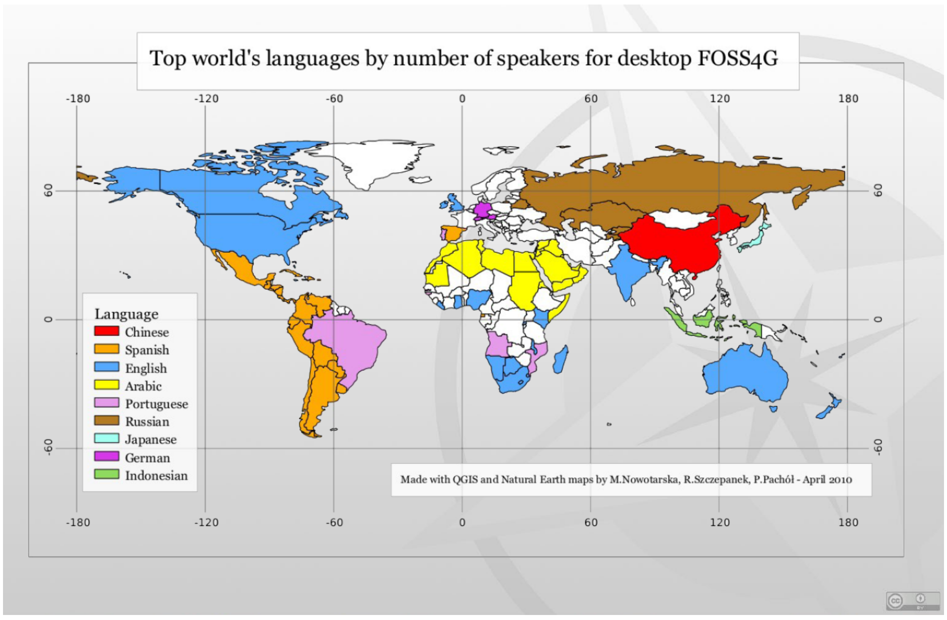

Top world's languages by number of speakers for desktop FOSS4G
-180
-120
-60
0
60
120
180
60
0
-60
Language
🟥 Chinese
🟧 Spanish
🟦 English
🟨 Arabic
🟪 Portuguese
🟫 Russian
Japanese
German
🟩 Indonesian
Made with QGIS and Natural Earth maps by M.Nowotarska, R.Szczepanek, P.Pachół - April 2010

Figure 8. Top world's languages by the number of speakers.

Our base map was the world admin countries provided on a public domain license in a vector shapefile format by the Natural Earth portal [10]. Using QGIS we have added an attribute column with a language assigned to each country. For countries with many official languages (for examples in Africa), we have chosen just one of them, preferably a local one. In that sense, the presented research is underestimated.

For visualization purposes we divided localization progress into four groups (fig.9, fig.10, fig.11):

- no localization (0%)
- beginning of localization (1-30% translated phrases)
- partial localization (30-70%)
- (almost) complete localization (70-100%)

GRASS

The first analyzed desktop FOSS4G was GRASS, the version with wxpy interface (fig.9).

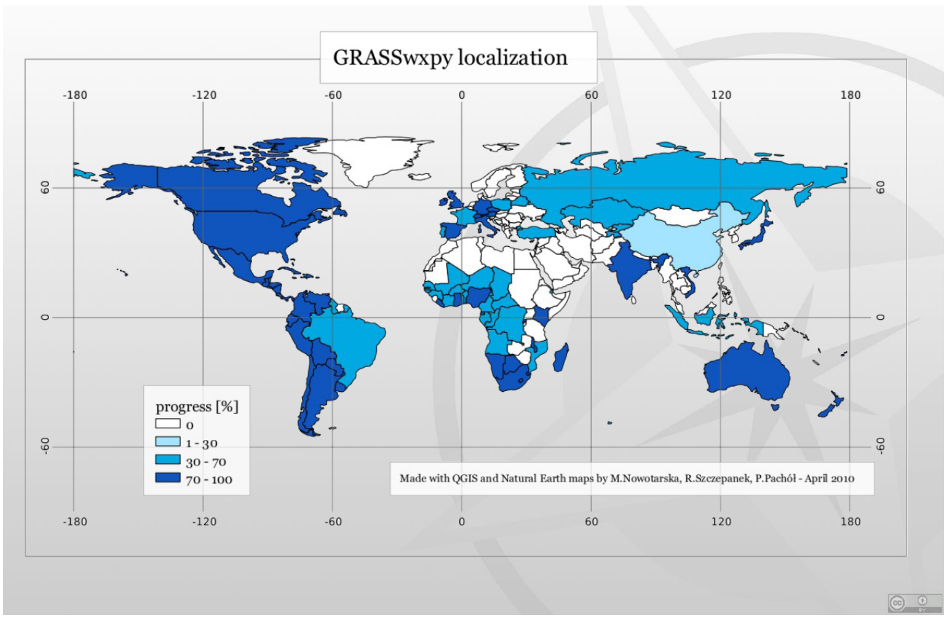

GRASSwxpy localization
-180
-120
-60
0
60
120
180
60
0
-60
progress [%]
0
1 - 30
30 - 70
70 - 100
Made with QGIS and Natural Earth maps by M.Nowotarska, R.Szczepanek, P.Pachół - April 2010

Figure 9. GRASSwxpy localization.

This mature and very powerful program has a very specific group of users. They are mainly scientists and more advanced users. Development is focused on new modules, not localization. Recent versions, which are better integrated with Windows, are changing this situation slowly. Both Americas, Australia and Europe have localized GRASS versions. The white spaces in the Scandinavian part of Europe (fig.9) are the effect of the high education level in those countries, where almost everyone speaks English. Northern Africa, Middle East and Southern Asia have no localization of GRASSwxpy.

gvSIG

When talking about gvSIG, it should be mentioned that its origin is from Spain and this project is developed in Spanish language. From a localization point of view, there are almost no partial localizations. A project is either localized or not. Compared with GRASSwxpy, gvSIG has partial Chinese, and complete Russian localization (fig.10).

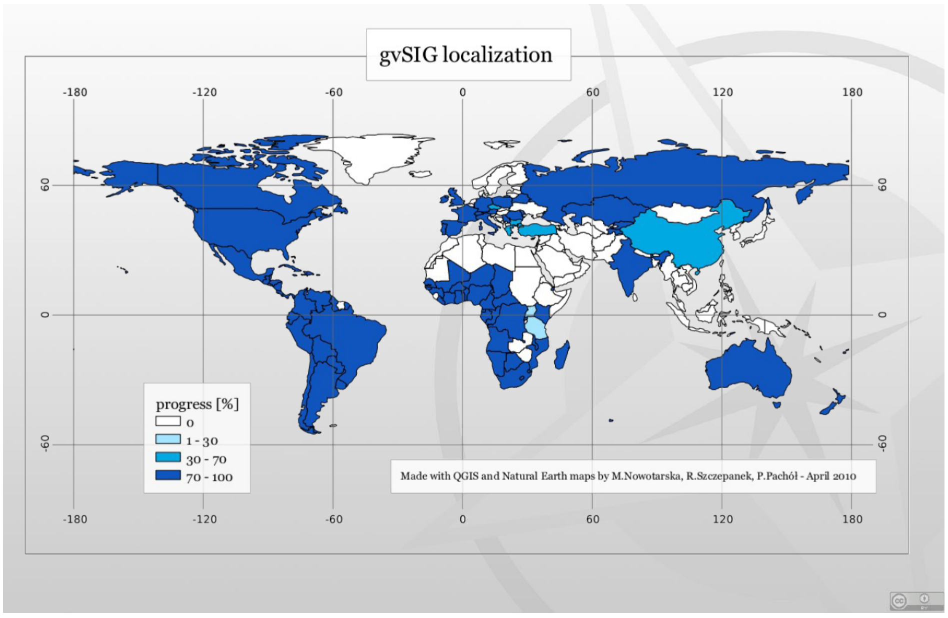

gvSIG localization
-180
-120
-60
0
60
120
180
60
0
-60
progress [%]
0
1 - 30
30 - 70
70 - 100
Made with QGIS and Natural Earth maps by M.Nowotarska, R.Szczepanek, P.Pachół - April 2010

Figure 10. gvSIG localization.

Quantum GIS

From the perspective of localization, the QGIS project is the most promising one. A very open community and easy to use tools encourages localization. Probably that is why this program, from desktop FOSS4G, has the best world coverage. There are only a few places where even partial localization is not available. But still Africa, the Middle East and part of Asia are places where more localization should be done (fig.11).

Discussion

Localization of FOSS4G is no longer a computer guru's task. With user-friendly desktop tools like Qt Linguist or online portals like Transifex, almost everyone with language skills and GIS knowledge can help in localization.

Diversity is very important and the big number of languages in the world is our heritage. But why do all three of the most spread out FOSS4G desktop programs use incompatible translation coding? Why when translating three programs we must use three different tools? A step ahead can be a tool like Virtaal, but it is a young project and still under development. A better solution seems to be a common, dedicated file format, like XLIFF. It provides many more features than any other current format. If every software would spit with XLIFF we could use one common translation glossary for many projects.

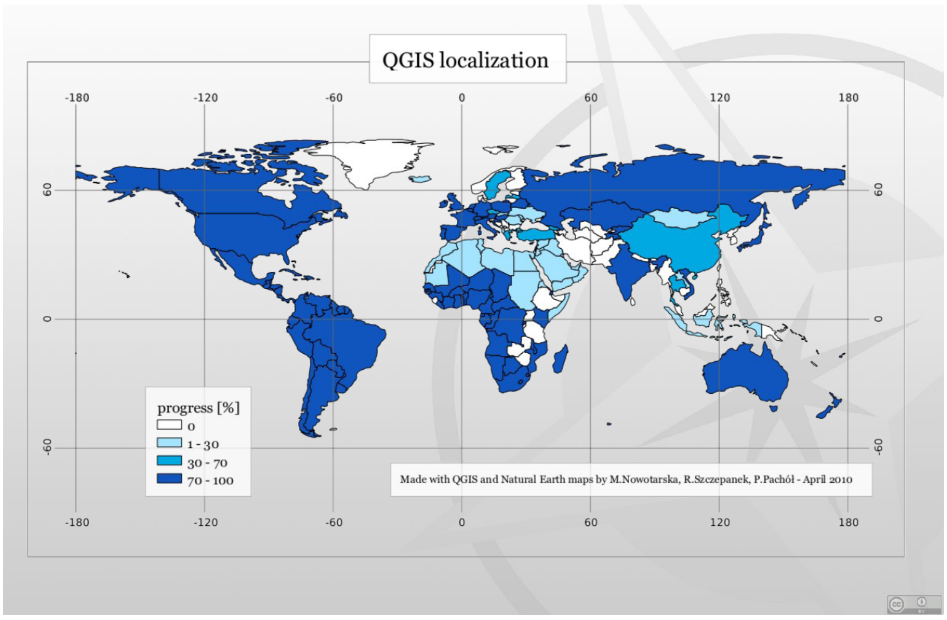

QGIS localization
progress [%]
0
1 - 30
30 - 70
70 - 100
Made with QGIS and Natural Earth maps by M.Nowotarska, R.Szczepanek, P.Pachół - April 2010

Figure 11. QGIS localization.

Anyway, we see a need for a common online OSGeo dictionary. It could be integrated later with any computer aided translation (CAT) tool. A centralized vocabulary for FOSS4G as the main source for translation with a user-friendly suggestion tool, would improve localization quality and efficiency. With fast internet access and new web technologies, online localization tools look simpler to coordinate work and merge outcomes. But the development of such tools will take time. In the meantime, an unification of translation tools is proposed, at least at the level of keyboard shortcuts.

QGIS appears to be the most multilingual desktop application, but GRASS and gvSIG also have made important achievements in localization. With the present state of localization, most of the users can even select which of those similar GIS desktop tools to use.

Desktop FOSS4G localization focuses mainly in developed regions of the world. Efforts should be focused on regions with low desktop FOSS4G localization – Africa, Middle East, Asia. Even more important than the localization of the GUI is for those users more detailed explanation of the tools itself. It can be done by context help, manuals and tutorials, of course all in their home language. The question is – how to attract translators from those countries? Without localization, good OSGeo products still will remain unknown. One of possible solutions can be cooperation with local universities and localization to be done as a part of students' master theses.

Conclusion

Offline translation tools like Qt Linguist provide advanced functions, but their weakness lies in unnecessary translation files traveling and merging.

Online tools still lack friendly and functional interfaces. Sometimes slow server response is also a problem.

The localization of FOSS4G desktop packages follows, with some exceptions, developmental patterns of the world. Desktop FOSS4G localization is much more advanced in developed countries than in developing ones. This is partially due to the big number of languages in developing countries, but also to less interest in such tools.

Available statistics on translation progress can't be compared directly. We assumed many simplifications in the presented research. One of them was – one language assigned to one country.

Even full GUI translation is not enough. It must be followed by help files, manuals and tutorials in local languages.

References

1. GRASS messages translation, July 2010.
   Available at: http://grass.osgeo.org/wiki/GRASS_messages_translation
2. Quantum GIS Wiki, Translation category, July 2010.
   Available at: http://www.qgis.org/wiki/Category:Translation
3. Qt Development Tools, July 2010.
   Available at: http://qt.nokia.com/products/developer-tools/
4. Poedit project, July 2010.
   Available at: http://www.poedit.net
5. Virtaal project, July 2010.
   Available at: http://translate.sourceforge.net/wiki/virtaal/index
6. Open-Tran project, July 2010.
   Available at: http://open-tran.eu
7. gvSIG localization portal, July 2010.
   Available at: https://gvsig.org
8. Open Source Geospatial Foundation, July 2010.
   Available at: http://www.osgeo.org
9. Lewis, M. Paul (ed.), 2009. Ethnologue: Languages of the World, Sixteenth edition. Dallas, Tex.: SIL International.
   Available at: http://www.ethnologue.com
10. Natural Earth – free raster and vector map project, July 2010.
    Available at: http://www.naturalearthdata.com