The GNOME™ Conference

GUADEC

Translating software using related languages

Rūdolfs Mazurs (rudolfs.mazurs@gmail.com)

Whoami: Rūdolfs Mazurs

- Leading Latvian GNOME translation team
- 10 years translating software, started out on Ubuntu 6.06
- Translation is my hobby

Outline

This talk is about software localization and how to translate large projects, like GNOME, if it is already translated to a similar language.

If you are a translator or wish to create a new language team, this talk is for you.

This talk is mostly informed by my attempts to figure out, how to translate Ubuntu into Latgalian (0.2M speakers), even though I don't speak it, but I do know Latvian (1.75M speakers).

Why should you care about localization (l10n)?

- Localization is a form of accessability
- Localized software is easier to learn
- Users are more confident in new situations
- People are more comfortable with the familiar
- Strengthens the language

Status quo

Challenges

Translating software is much harder than translating books:

- strings are not plain text (e.g. “Welcome, %s!”, “ by ”),
- no or cryptic context (e.g. “Exporting”, “Close”),
- terminology must be consistent,
- style must be consistent.

Tooling

What tools translators currently have:

- Spellchecker
- Glossary
- Guidelines
- Translation memory

Shameless plug

- Lokalize
- Translate toolkit
- Microsoft localization style guides
- GNOME i18n wiki

Automation

Can the computer translate?

Rule based:

- Dictionary
- Transfer
- Interlingual

Statistical:

- Large parallel texts
- Huge non-parallel texts + dictionary

Is any of this useful?

Dictionary method gives useful results for related languages. E.g. Česílko system translates correctly 90% of sentences from Czech to Slovak and Czech to Russian.

90% is not perfect, but is useful. Minor edits are far more easier than translating from scratch.

Machine translation word-by-word

What is mt-words

Script, based on translate-toolkit, that takes an existing translation in po format and a dictionary, giving an output for the desired language.

```
./po_dictum.py -i eog.lv.po -o eog.ltg.lv \
               --dictionary dictionary.csv

./po_dictum.py -i gnome-lv/ -o gnome-ltg/ \
               --dictionary dictionary.csv
```

Example: happy case

```
EN: "Close window"
LV: "Aizvērt logu"
LV: "Aizvērt| |logu"
    " word  | |word"
LG: "Aizdareit lūgu"
```

Example: duplicates

```
EN: "Open in Firefox"
LV: "Atvērt ar Firefox"
LV: "Atvērt| |ar| |Firefox"
    "      | |  | | dupl  "
LG: "Attaiseit ai Firefox"
```

Example: tags and URLs

```
EN: "Go to <a href='host'>http://ej.uz/a</a>"
LV: "Ej uz <a href='host'>http://ej.uz/a</a>"
LV: "Ej| |uz| |<a href='host'>|http://ej.uz/a|</a>"
    "  | |  | |   literal     |   literal    |literal"
LG: "Eima iz <a href='host'>http://ej.uz/a</a>"
```

Example: variables

```
EN: "Error: %(error)s in %(module)s"
LV: "Kļūda modulī %(module)s: %(error)s"
LV: "Kļūda| |modulī| |%(module)s|: |%(error)s"
    "     | |      | |  literal |  | literal "
LG: "Klaida īkš moduļš %(module)s: %(error)s"
```

Example: accelerators

```
EN: "Close _anyway"
LV: "Tomēr _aizvērt"
LV: "Tomēr| |aizvērt"
    "     | | accel "
LG: "Tūmār| |aiztaiseit"
LG: "Tūmār _aiztaiseit"
```

Example: not-really-accelerator

```
EN: "Failed to read time_t"
LV: "Neizdevās nolasīt time_t"
```

What format is the dictionary

Spreadsheet, CSV format. This is an example of dictionary, that contains words for window, potato, view and radio.

From	To	Problematic
logs	lūgs	
kartupelis	buļbe	yes
skats	skots	
radio	radeja	yes

Example: creating a dictionary

Create dictionary from a single file:

```
./po_dictum.py -i [path-to-po-file] \
               --new_words dictionary.csv
```

Create dictionary from a folder:

```
./po_dictum.py -i [path] -o [dummy] \
               --new_words dictionary.csv
```

Example: updating the dictionary

Update dictionary from a single file:

```
./po_dictum.py -i [path-to-po-file] \
               --all_words dictionary.csv
```

Update dictionary from a folder:

```
./po_dictum.py -i [path] -o [dummy] \
               --all_words dictionary.csv
```

Case study: Latvian to Latgalian

Effectiveness

Translate *evince* (then 359 strings, 1116 words) from Latvian to Latgalian

- Substantively wrong: 0 strings
- Word reordering required: 0 strings
- Failed to translate: 1 word
- Wrong form of a word: 16 strings, 19 words
- Error in the original: 7 strings

About 95% success rate.

Translating GNOME desktop

How much work is it to translate GNOME?

Source language	Dictionary size	Translated
Latvian	15 600	96%
Spanish	08 000	99%
German	13 800	100%
Chinese	32 000	92%
Hindi	01 800	63%
Scots Gaelic	04 700	48%

What is a word?

Whatever the “Magic 7” regex `^\W_0-9` says it is.

The script would fail on “don’t” and “can’t”.

Issues discovered

- Errors in strings accumulate
- Grammar of the related language gets copied
- Have to follow the schedule of other translation team
- Team will have to create terminology, and that one is **hard**

Positive side effects:

- Gives translator insight for free
- Dictionary *is* the spellchecker
- Spellchecking for the original language
- Checking can be done by less qualified language speakers

The future

Translation maintenance

Can the translations be maintained with this script? Not yet. How to detect changes:

- English string is changed — the string is now fuzzy
- source translation is updated — compare it with previous translation
- dictionary has changed — compare with the old dictionary or check if the word is newer than previous translation

How does it work with translation memory?

Dictionary maintenance

CSV (one big table):

- is a dead simple format, easy to use,
- is easy to abuse,
- has no integrity checks, storing metadata can be difficult.

Possible future web app:

- can provide integrity checks,
- gives options to store metadata,
- can have more advanced search for similar words,
- is expensive to build and maintain,
- but internet is slow and unreliable.

Interested?

Repo at `https://github.com/Mazurs/mt-words/`

Mailto: `rudolfs.mazurs@gmail.com`

I'll be at "Localization and Documentation BoF", afternoon of 9th July, Room 6