What is Language Converter?

C. Scott Ananian <cananian@wikimedia.org>
Content Transform Team,
Wikimedia Foundation (May 31, 2024)

Why do we use Language Converter?

Languages are fun!

Some example pairs for English speakers:

- English language variants:
  - color/colour
  - ten million/one crore
- Brazilian Portuguese differs to about the same extent:
  - berinjela / beringela

Lossless (reversible) -vs- Lossy conversions:

- Every elevator is a lift but not every lift is an elevator!

On wikipedia

See:
https://meta.wikimedia.org/wiki/Wikipedias_in_multiple_writing_systems

Chinese Wiki: two major writing systems, various dialects.

- zh-cn (Mainland China), zh-tw (Taiwan), zh-hk (Hong Kong), zh-mo (Macao) and zh-sg (Singapore)

Serbian Wiki: two writing systems, and two dialects.

- Latin alphabet Ekavian, Cyrillic alphabet Ekavian, Latin alphabet Ijekavian, Cyrillic alphabet Ijekavian

Kazakh Wiki: three writing systems.

- Cyrillic, Latin, and Perso-Arabic (Central Asian branch) alphabets. (Conversion to Arabic read only.)

On Wikipedia, continued

Kurdish Wiki: three writing systems

- Latin (Turkey/Syria) <-> Arabic (Iraq/Iran)
- no support for Cyrillic (ex USSR)

Inuktitut Wiki: two writing systems.

- Inuktitut syllabics (Nuvavut) <-> Latin
  - ᐃᓄᒃᑎᑐᑦ <=> Inuktitut
  - http://www.languagegeek.com/inu/inutext.html
- Syllabics lose case distinction from latin script

Anglo-Saxon Wiki: two writing systems

- Latin <-> Runic

On Wikipedia, even more

Tajik: (Cyrillic<->Latin, but not Arabic)

Uzbek: (Cyrillic<->Latin, but not Arabic)

Gan: (simplified<->traditional Gan Chinese, but not Romanized Gan)

Kyrgyz: (Cyrillic/Latin/Arabic, not yet deployed)

Uyghur: (Arabic/Cyrillic, not implemented?)

Chechen: (Cyrillic/Latin, not yet deployed)

...and 29 more, see the full list

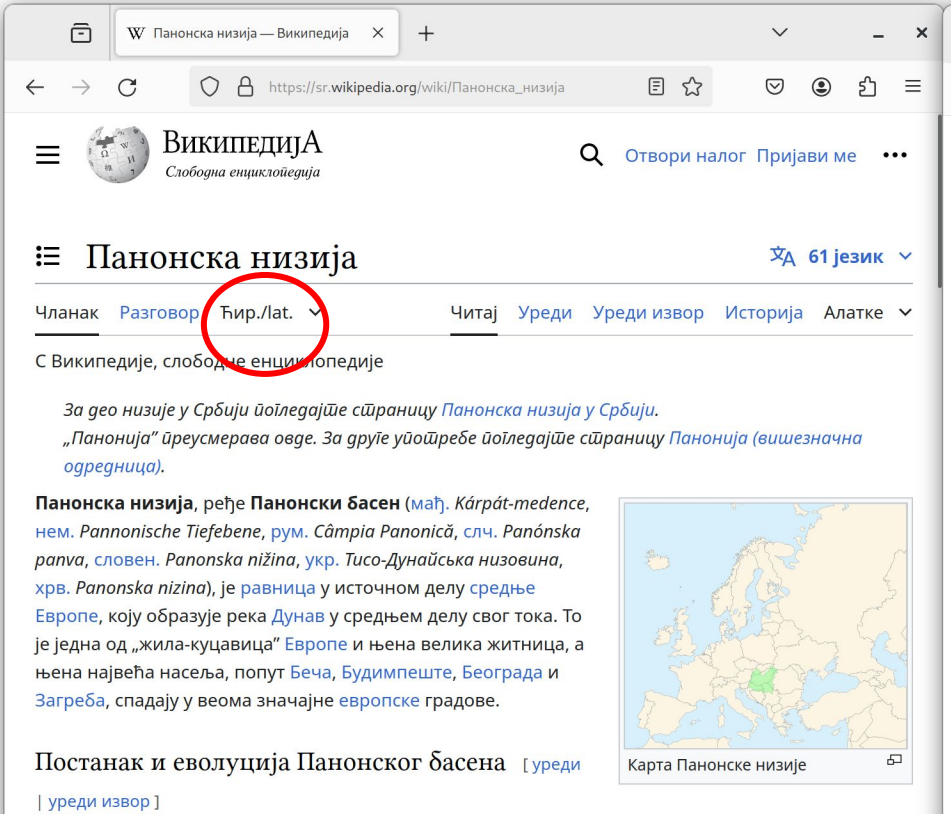
Languages are spoken by people

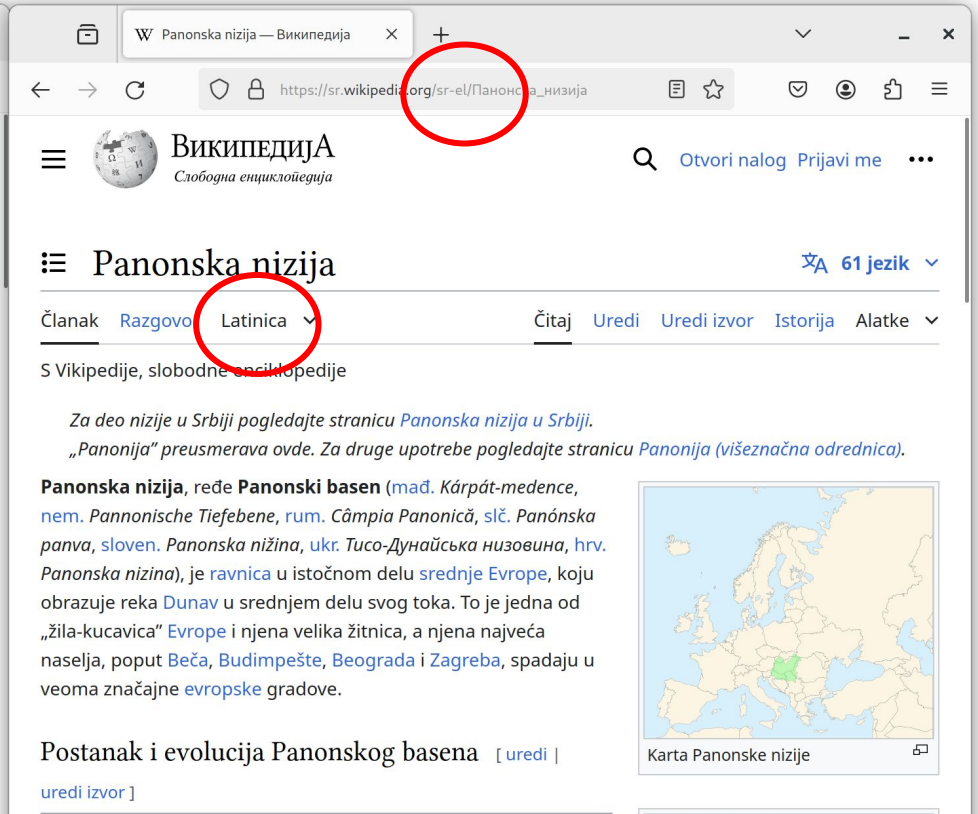
- Political issues with unifying/splitting wikis
  - And some folks who'd like to bury the hatchet
- Political issues with providing transliterations to banned scripts
- Biliteracy (or the lack thereof)

What does Language Converter do?

What does Language Converter do?

Allows content written in one (or more!) variants to be read in one (or more!) variants.





Some example markup

Marking up text which has variants:

- Bidirectional rules:

  -{en-uk: lift; en-us: elevator;}-

- Unidirectional rules:

  -{elevator => en-uk: lift;}-

- Disable conversion:

  -{R|lift}- or -{lift}-

Also a bunch of stateful options for adding/removing rules (more on this later), and for working around title limitations.

See https://www.mediawiki.org/wiki/Writing_systems/Syntax

Some example code (transliteration)

```
class SrConverter extends LanguageConverter {
    public $mToLatin = array(
        'а' => 'a', 'б' => 'b', 'в' => 'v', 'г' => 'g', 'д' => 'd',
        'ђ' => 'đ', 'е' => 'e', 'ж' => 'ž', 'з' => 'z', 'и' => 'i',
        [...]
        'Х' => 'H', 'Ц' => 'C', 'Ч' => 'Č', 'Џ' => 'Dž', 'Ш' => 'Š',
    );
    public $mToCyrillics = array(
        'a' => 'а', 'b' => 'б', 'c' => 'ц', 'č' => 'ч', 'ć' => 'ћ',
        'd' => 'д', 'dž' => 'џ', 'đ' => 'ђ', 'e' => 'е', 'f' => 'ф',
        [...]
        'Nj' => 'Њ', 'n!j' => 'нј', 'N!j' => 'Нј', 'N!J' => 'НЈ'
    );
```

Some example code (hant/hans)

```
$zh2Hant = array(
'佩' => '佩',
'佟' => '佟',
[...9,623 lines...]
'；克制' => '；剋制',
'？克制' => '？剋制',
);
$zh2Hans = array(
'偵' => '侦',
'偵' => '侦',
[...4,651 lines...]
```

Some example code (FST)

These are “simple” table-based examples; actual converters can get much more complicated, involving special rules for word boundaries, long lists of ordered regular expressions, etc.

I made an effort (wikimedia/langconv) to use a more formal Finite-State Transducer mechanism to implement language converters, which has uncovered bugs and corner cases in existing converters.

Some example code (templates!)

The zhwiki as a gadget installed named 'NoteTA'. See zh:Module:NoteTA and zh:MediaWiki:Gadget-noteTA.js.

This is used to display the current set of word conversions for a page.

A template with NoteTA is used to define a set of conversions specific to the page.

Some example code (templates!)

From [[:zh:鋼鐵人3]] ([[:en:Iron Man 3]]):

```
{{noteTA
|G1=Movie
|G2=Show
|G3=美国漫画
|1=zh-hans:罗伯特; zh-tw:勞勃; zh-hk:羅拔;
|2=zh-hans:奥德利奇·齊連安; zh-tw:奧德奇·齊禮安; zh-hk:奧德奇·齊禮安;
|3=zh-hans:羅德斯; zh-tw:羅德; zh-hk:羅德;
}}
```

Pros and cons

- For variants with lossless conversions the process seems to work well, expanding the set of readers for our content
- In some wikis, a dominant variant is used for authoring content
- But… in some cases editors are not typically fluent in both variants. Content is written in both, interspersed.

Other caveats

- Source language is not tagged, and not always obvious
  - Serbian and Roman numerals!
- LanguageConverter doesn't especially care about word boundaries
  - Not important for Mandarin!
- Link resolution
  - Try every possible variant title, see what works!
- Glossaries
  - In many cases, the right conversion is context-sensitive, ie there are conversions for films, for physics, etc.

User Interface Localization

Messages used for the user interface are *not* handled by LanguageConverter.

Separate manually-created translations for (eg) zh-hans and zh-hant

Is this good?

- High quality localization for small # of messages

Or bad?

- The # of messages is not really small!

What are some alternatives?

Other ways to handle variants

- Null option.
  - Pick a single preferred variant.
- Read-only language converter.
  - Content authored in a single preferred variant.
- Bidirectional language converter.
- Content Translation Tools.
- Split the wikis.
  - With better tools to maintain forked wikis?
  - Content Translation Tools one option here?
  - One-click synchronization between wikis?

Are they other good ideas here?

Scaling translation

We have two scaling axes in localization:

- Message size
  - Wikidata tags
  - Interface strings
  - …
  - Full articles
- Language divergence
  - American/British English
  - Brazillian/Portuguese
  - Romance languages
  - “Eastern Punjabi” (ISO PAN, in India)/“Western Punjabi” (ISO PNB, in Pakistan)
  - English/Mandarin

Questions

Can we make tools which scale well along both axes?

Should we?

How many tools should we have?

What are the sweet spots?

Ok, let’s discuss!

(thanks)