Localization

Jim Gettys, V.P. of Software
Overview

- Scope of Problem
- Current Status of Free Software
- Tools
- Localization Strategies
- Logistics & Issues
Problem Statement

- Technology for representing and composing the languages spoken, taught or used in your countries.
- Fonts, script layout, input methods, speech synthesis, musical instrumentation, collating order, dictionaries, spelling checkers.
- Linux is already more widely localized than Microsoft Windows since no cooperation required.
## Size of the Problem

<table>
<thead>
<tr>
<th>Population Range</th>
<th>Living Languages</th>
<th>Number of Speakers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>count</td>
<td>%</td>
</tr>
<tr>
<td>100,000,000 to 999,999,999</td>
<td>8</td>
<td>0.1</td>
</tr>
<tr>
<td>10,000,000 to 99,999,999</td>
<td>75</td>
<td>1.1</td>
</tr>
<tr>
<td>1,000,000 to 9,999,999</td>
<td>264</td>
<td>3.8</td>
</tr>
<tr>
<td>100,000 to 999,999</td>
<td>892</td>
<td>12.9</td>
</tr>
<tr>
<td>10,000 to 99,999</td>
<td>1779</td>
<td>25.7</td>
</tr>
<tr>
<td>1,000 to 9,999</td>
<td>1967</td>
<td>28.5</td>
</tr>
<tr>
<td>100 to 999</td>
<td>1071</td>
<td>15.5</td>
</tr>
<tr>
<td>10 to 99</td>
<td>344</td>
<td>5</td>
</tr>
<tr>
<td>1 to 9</td>
<td>204</td>
<td>3</td>
</tr>
<tr>
<td>Unknown</td>
<td>308</td>
<td>4.5</td>
</tr>
</tbody>
</table>

Localization of OLPC Applications

- Sugar environment is new
  - therefore needs localization work
  - but deliberately designed to minimize text.
- Other Linux applications have already been localized for many languages.
- Languages cross borders: share the work
Character Sets

- Unicode – fully supported in “modern” applications and toolkits.
- Legacy character set support also present, but modern applications are use Unicode.
- Collation order is generally well supported.
Script Layout

• Pango library – able to layout most “hard” languages, including: Arabic, Indic family, Hebrew, Persian, Amharic, Thai, etc.

• Modular layout engine and vertical text;

• BIDI Layout supported;

• Some issues remain – but overall in pretty good shape.
Fonts

- To share content and preserve cultural heritage, OLPC's goal is full coverage of all the world's languages – Linux has a better concept of language coverage of fonts than other systems.

- Formats: OpenType, TrueType and others...

- Quality screen fonts a problem for lo-res screens.

- XO-1's high screen resolution helps us: less “hinting” required for good results == more usable fonts: but...
Fonts (continued)

- the OLPC software environment may be used on existing systems at low resolution... therefore, we should work together on creating more “free” high-quality fonts.
Many Free Fonts Exist for Most Languages

- **SIL International** is another source of free fonts.
- But few free “high quality” screen fonts exist:
  - Hinting is boring and labor intensive – $10/glyph at U.S. rates; special tools used.
- Pooling of effort and/or expense makes great sense; donations?
- Coordinated buyout of fonts? If fonts are not MS metric compatible, cost is “reasonable”.
Speech Synthesis

- Size vs. fidelity vs. localization effort trade-offs: *festival, flite, espeak* are available.

- *Espeak* is small enough for us to bundle: ~10 languages currently supported tuned by native speakers – 10 more languages underway.

- Useful for accessibility, literacy training, GUI.

- Not a good guide for pronunciation – but may be better than a poor teacher.
Keyboards

- Currently: English (US international); Spanish (Latin America); Portuguese (Brazilian); Amharic (Ethiopic); Arabic; Nigerian (Igbo, Hausa, Yoruba); French (Rwanda); Thai; Urdu; Cyrillic (Russian); Turkish; Nepali; Mongolian; Kazakh; Devanagari; Uzbek; Pashto; Dari; Armenian; Khmer; Pulaar; Italian; Kreyòl; others are possible, but the lead time is 4-6 months for new layouts.
Input Methods

• Input method – how to type complex characters - Chinese, Japanese, Korean... some issues remain (example: Arabic ligatures; we avoid them).

• SCIM - Smart Common Input Method Platform - http://www.scim-im.org/projects/imengines

• SCIM is replacing older input method systems.

• Stroke/character recognizer localization is of some interest with the pen/tablet: in the future when we have a touch screen they will become essential.

• We need to know what languages are taught as “foreign” languages, as well as are native.
Current Shortcomings

• Non-Gregorian calendars
• Non-Latin digits (Roozbeh Pournader has patches, but these are not yet integrated).
• Sheer scale of the localization problem will eventually require changes in free software projects.
Sound Fonts (Music)

- We want much more than dead white male western instruments for dead white male composers!
- Clean samples of your musical instruments and music needed!
- Samples need appropriate licensing terms.
Dictionaries and Spelling Checkers

• Support exists for most major languages.

• Spelling, Hyphenation, Thesaurus dictionaries may be needed, check:
  - http://aspell.net/man-html/Supported.html
  - http://dictionaries.mozdev.org/installation.html
  - http://www.abiword.org/languages.phtml
  - http://wiki.services.openoffice.org/wiki/Dictionaries
Techniques

• It only takes a small team to localize for a language: e.g. Welsh, Icelandic:
  − Do it yourself, hire it out, find volunteers.

• Work in the projects whenever possible:
  − This makes your work available worldwide,
  − Lessens the ongoing work.

• Add to existing projects whenever possible.
Tools

• Examples: pootle, kbabel (offline), rosetta (on line)
  − tools to convert between systems,
  − most software uses “gettext” and standard .po files; Firefox and OpenOffice have their own systems for historical reasons.

• The cldr project - http://www.unicode.org/cldr

• Remember, contribute your translations to the “upstream” projects to minimize long term effort: share your work with the world.
Sugar Localization

• Most sugar applications are localized using Pootle which has been integrated with our source code repository. See: https://dev.laptop.org/translate/

• http://wiki.laptop.org/go/Localization

• More languages always welcome! Language coordinators greatly appreciated! We can't do it for you!

• OLPC'S Localization lead: Sayamindu Dasgupta
  – IRC: irc.freenode.net #olpc-pootle
Licensing

• Strings
  - Translated strings will often be useful among many projects, not just the project you are working on translating,
  - Therefore, since the MIT/BSD (3 clause) licenses are usable by all projects, these are the safest licenses to use for translation to enable widest sharing.

• Fonts: SIL OFL license recommended.
Next Steps

- Localization is by nature local: but language often crosses borders.
- Please come see me to identify issues.
- We need identified people/organizations responsible for:
  - Language, translation, keyboards, speech synthesis, effective free software community leaders.
Summary

- Do it yourself, hire it done, find volunteers
- Fonts – screen fonts for non-OLPC screens
- Speech Synthesis
- Musical Instrumentation Samples
- Dictionaries
- Work “Upstream”
Summary

• Supporting new languages varies from usually very easy (just translating strings), to a large amount of work and engineering taking many months.

• The sooner we know your needs, the sooner we can determine the amount of effort.