Translating Drupal (the new way)

Gábor Hojtsy
September 19., 2007.

Sponsored by

All of my work explained in this session was sponsored by Google Summer of Code 2007.

Thanks Google!

About me

- Using and contributing to Drupal since 2003
- Multiple modules, Hungarian translations
- Drupal 6 core committer
- Locale module maintainer

About you

- Users of existing translations?
- Translating in-house modules?
- Occasional translation contributors?
- Members of official translation teams?

Setting the topic

Translating Drupal

- Translate the **built-in interface**
- User provided content translation is out of scope here (sessions about this topic were before lunch)

History

t() is born

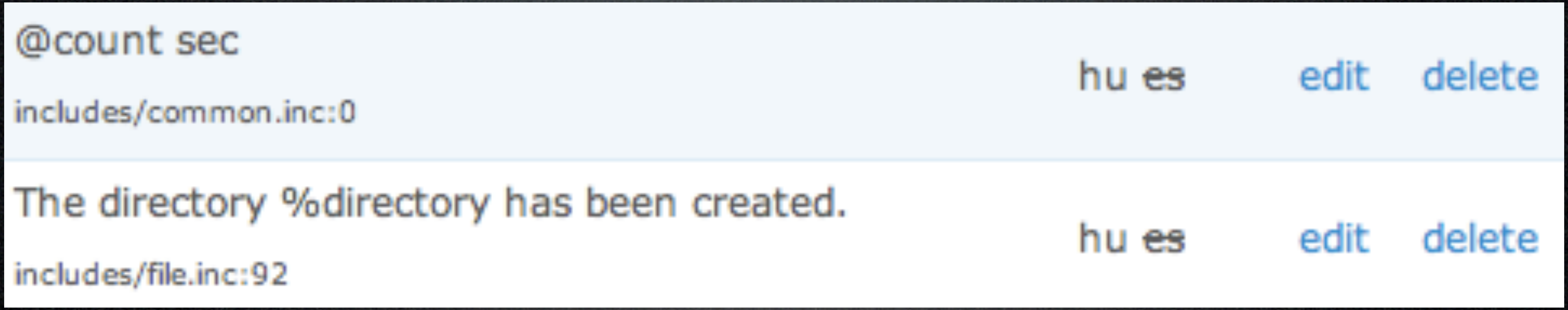
January 22, 2001

```
<?

function t($string) {
 return $string;
}

?>
```

(format_plural() was there earlier)

Timeline

- Jan 22, 2001 - t()
- Feb 12, 2001 - Locale module
- Feb 17, 2001 - Basic Translation interface, the strikethrough overview, languages added by SQL DDL



@count sec includes/common.inc:0	hu ~~es~~	edit	delete
The directory %directory has been created. includes/file.inc:92	hu ~~es~~	edit	delete

Timeline

> various smaller updates before I start watching Formula 1 with [a] a 0.5 liter box of Haagen-Dhaz (Belgian Choclate) and [b] my laptop on my lap for drupal hacking. Life hardly can get any better than this.

Commit comment by Dries for locale.inc 1.7

Timeline

- Jul 24, 2001 - Search form, even module filtering, thanks to:

```
"location LIKE '%module.php?mod=". check_input($edit[module]) ."%'"
```

- Jan 5, 2002 - Locale cache
- June 2, 2003 - Module filtering removed, due to URL restructuring

Timeline

- Sept 11, 2003 - Extractor.php is born
- Dec 28, 2003 - Short string cache
- Aug 11, 2004 - Gettext format based string sharing, web administration, plural formulas
- Aug 18, 2006 - @ and ! placeholders

Timeline

- Sep 1, 2006 - Translatable installer
- Mar 26, 2007 - New language system
- May 15-22, 2007 - Automated translation import
- Jun 8, 2007 - JavaScript translation

How does it work?

Translate in PHP

- **t**(‘Go to %page’, array(‘%page’ => $p));
- **format_plural**($count, ‘1 year’, ‘@count years’);
- Several special cases: permissions, watchdog type names, and so on

Original process

- t() stores new strings for translation, when called with unknown text
- Web interface to translate
- Different sharing methods through the years: SQL, Gettext

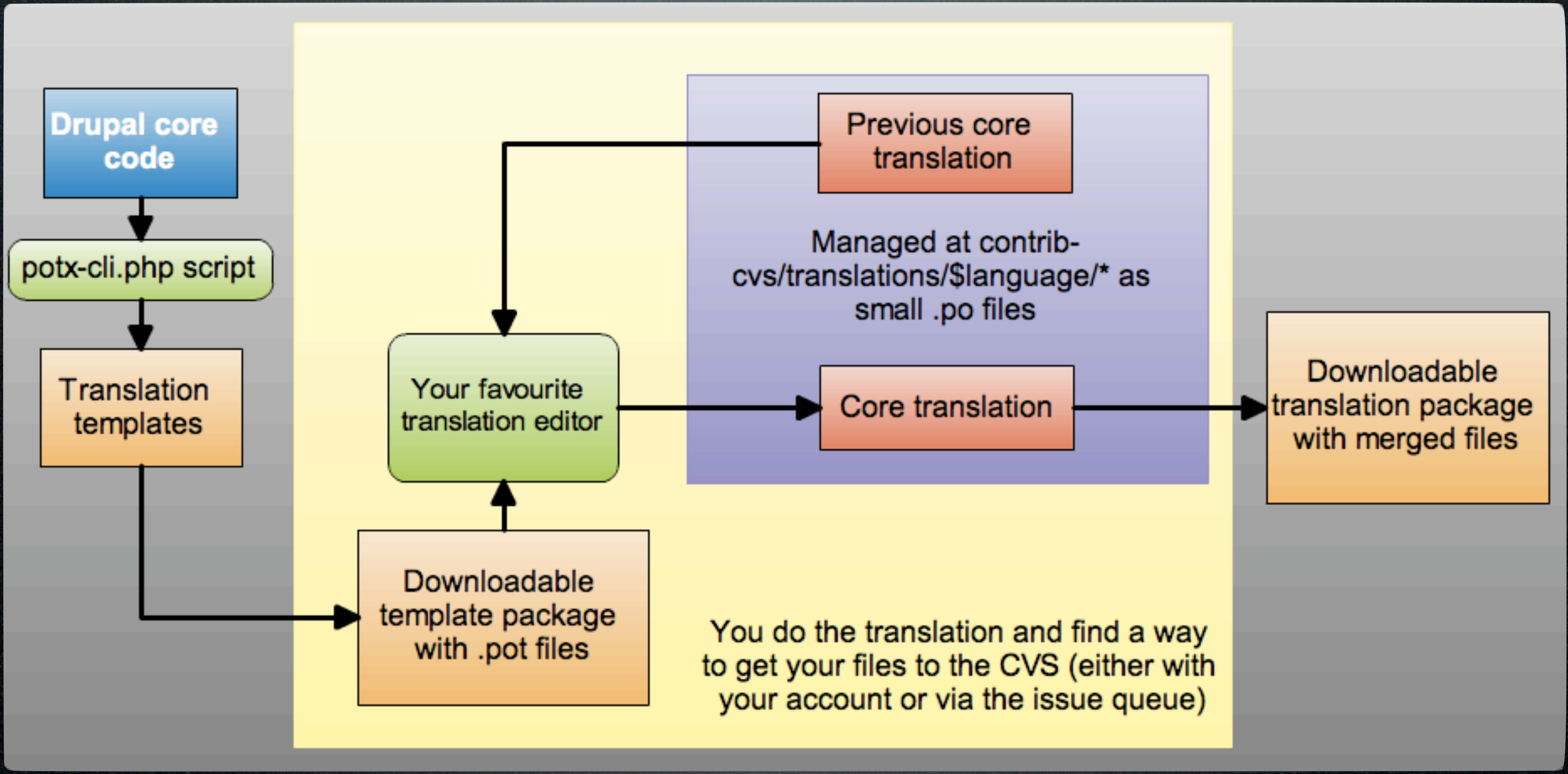
Collaborative process

- A preprocessor generates translation templates
- Translators grab templates and translate
- Package is distributed
- Users import the translations

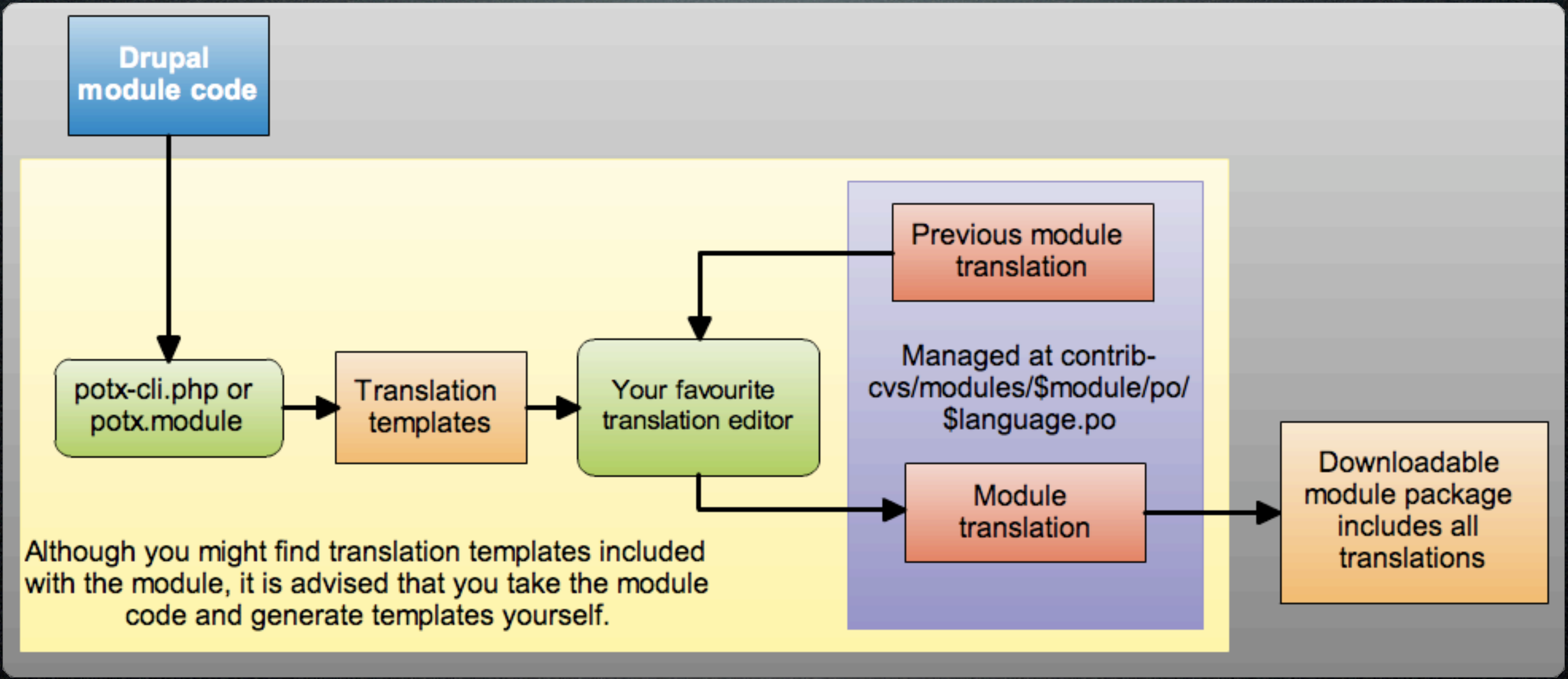
Translate in JavaScript

- **Drupal.t**() and **Drupal.formatPlural**() to translate strings in JavaScript
- Works just as well with on the fly string collection or preprocessing

Drupal core process


Drupal core code
potx-cli.php script
Translation templates
Downloadable template package with .pot files
Your favourite translation editor
Previous core translation
Managed at contrib-cvs/translations/$language/* as small .po files
Core translation
Downloadable translation package with merged files
You do the translation and find a way to get your files to the CVS (either with your account or via the issue queue)

Drupal contrib process


Drupal module code
potx-cli.php or potx.module
Translation templates
Your favourite translation editor
Previous module translation
Managed at contrib-cvs/modules/$module/po/$language.po
Module translation
Downloadable module package includes all translations

Although you might find translation templates included with the module, it is advised that you take the module code and generate templates yourself.

Still too hard

- Know CVS
- Have write permissions
- Use gettext tools to merge and translate
- Time releases properly
- Track project progress and goals
- Reuse translations as you can

Translation packages

- http://drupal.org/project/Translations
- Contributed project translations scattered all around the repository
- Some teams even have their own download locations

Usability problems

- Although users upload and install modules, they need to upload translations on the web separately
- Existing translations can be broken by new uploads

Executive summary:

Pain for translators

Pain for users

Better for users in Drupal 6

Locale automation

- Language selection at start of installer
- Fully localized installer, imports translations for all enabled modules
- Automated import when
  - new language is added
  - module is installed
  - theme is enabled

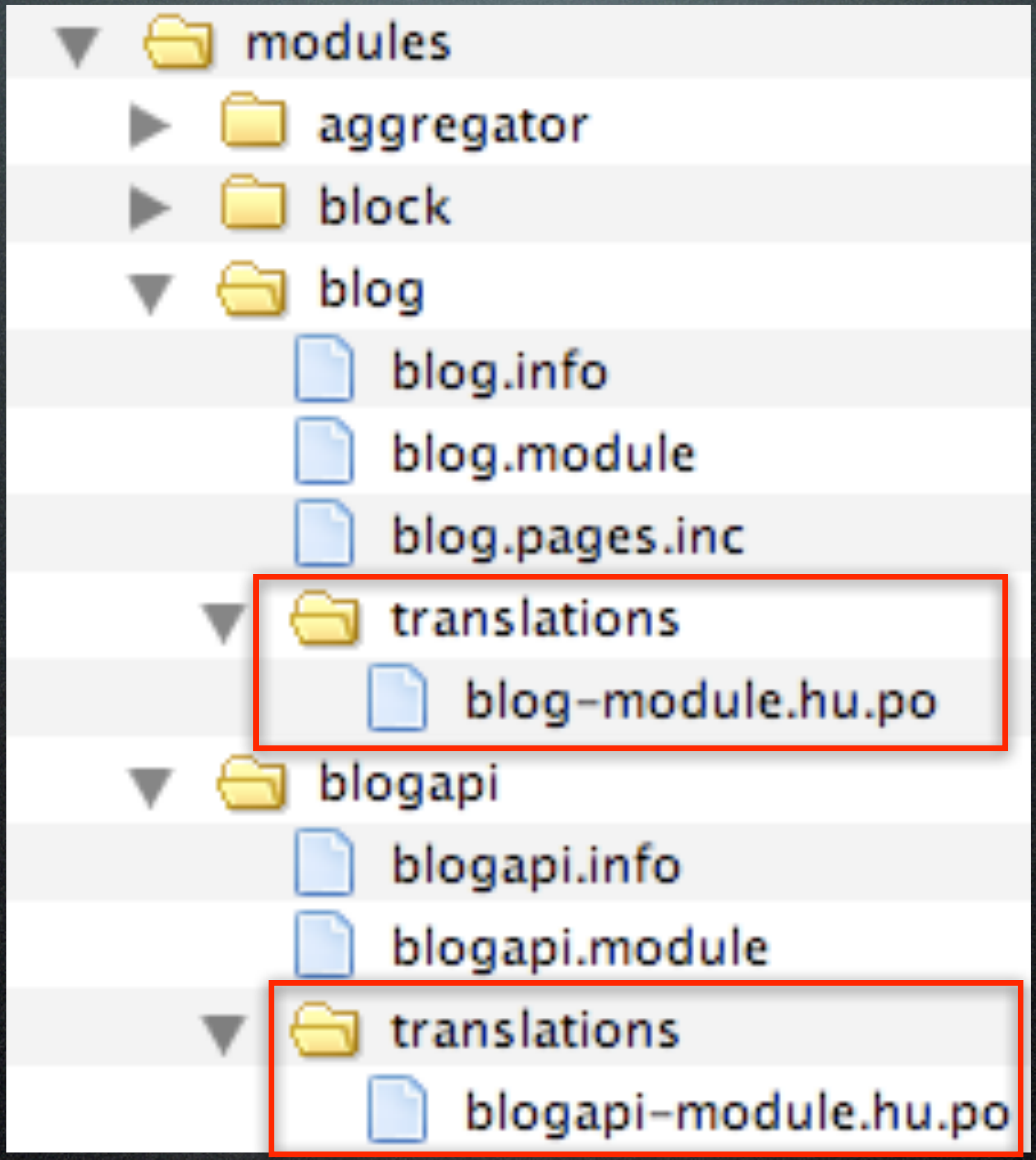
In need of a cleanup

- Unfortunately we don’t know what to remove when a module is disabled or uninstalled
- This is solved in part by version tracking in locale module

New packaging format

- All these changes required a new packaging format
- Based on more smaller files in a well defined directory structure
- ‘po’ renamed to **‘translations’**
- Now also used with core modules

New packaging format



```
modules
  aggregator
  block
  blog
    blog.info
    blog.module
    blog.pages.inc
    translations
      blog-module.hu.po
  blogapi
    blogapi.info
    blogapi.module
    translations
      blogapi-module.hu.po
```

Also better for
onsite translators
in Drupal 6

On-page translation

- Drupl 6 changes made on-page translation possible
- Served with the **l10n_client** contributed module
- Interface still to be refined (a lot)

Demo

What about collaboration?

Simplifying our lives

- Drupal.org knows about all the projects and releases available
- Their source code can be parsed for translatables
- Can be automated - we can even inform module developers about errors found

Localization server

- In part based on ideas from Bruno Massa
- Maintains a list of projects and releases
- Parses source packages for translatables

Localization community

- Translation community interface, based on organic groups
- Collaborate on project translations
- Different permission levels
- Import/export with Gettext support

New process

- Have seen groups.drupal.org?
- Translation groups, two permission levels: **open** and **controlled**
- Can host discussions, debates, “vocabularies”, “dictionaries”

New process

- Projects and releases are automatically available
- Translations can be imported from PO file, exported in PO packages
- Web based editing on custom interface
- TODO: backwards connection with projects

Demo

So when will this go live?

- Hopefully before or around the Drupal 6 release
- There are still features to implement and perfect
- It needs to fit nicely into the drupal.org infrastructure

Thanks!