Glottolog 3.0

A collaborative, versioned catalog of languages and dialects

Robert Forkel
Poznan, 2016-09-15, A new era for cross-linguistic databases

Max Planck Institute for the Science of Human History
Outline

1. Glottolog
2. Open Source and Open Data
3. Glottolog and collaboration
Glottolog is a comprehensive online catalog of languages.

- but also of dialects (less comprehensive, though)
- and a bibliography, linked to languages
- and a genealogical classification of languages
http://glottolog.org
Why not Ethnologue or ISO 639-3?

So Glottolog is very much like Ethnologue. Why another one?

- The editorial process of Ethnologue is not fully transparent, the change request process for ISO 639-3 is slow.
- Ethnologue is behind a paywall, ISO 639-3 not fully integrated in the web at large and the semantic web in particular.
- Ethnologue and ISO 639-3 are not really targeted at academia, they have a different business model.

Glottolog wants to provide data like Ethnologue, but curated in a more transparent, collaborative, community owned way.
Open Source and Open Data
So, why are we looking at Open Source software development best practices to improve management of research data like Glottolog?
Open Source collaboration

Open Source software in the age of GitHub is a tremendous success story for worldwide online collaboration.

This is exactly the kind of collaboration we want to enable for data sets like Glottolog, which clearly

• profit from more curators

  *given enough eyeballs, all bugs are shallow*  \textit{(Linus’ Law)}

• ”belong” to the academic community more than to any one institution, thus – given current funding schemes – will have to be transferred to a different owner at some point.
What spurred this surge in collaboration on Open Source software?

Remember: Licenses grant rights people wouldn’t usually have!

Open Licenses which allow derived works are the basis of Open Source:

*The ability to create derived works means that anyone can also modify the source or data as they see fit. In practice this means forking: creating a new custom version of some software, or a modified (corrected, reformatted) version of a dataset.*  

(Leigh Dodds)
The default practice in the open source world is that code will be:

- published in a public repository
- published with a complete version history [...] 
- published in an environment that supports transparent reporting of issues, bugs and suggestions 
- published in an environment that includes good documentation tools, such as a wiki 
- and, most importantly, published in an environment that allows forks and improvements to be folded back into the original project

I’d go as far as suggesting that each of these are as important to our modern experience and expectations of open source, as the basic rights granted by open licences. 

(Leigh Dodds)
Today, this infrastructure is GitHub.
Figure 1: The data behind the Open Tree of Life is curated in a series of GitHub repositories.
How do we turn Glottolog into Open Data?

We need to model Glottolog data in a way suitable for distributed version control systems.

- line-based text formats, i.e. text that can be meaningfully handled by **diff**
- BibTeX for bibliography files
- INI files for languoid metadata.
- A directory tree to model the classification.
- Some tools to simplify manipulation of the language tree.
- An API to access the data in the repository programmatically.
@book{94863,
  address    = {New York},
  author      = {Sapir, Edward},
  publisher   = {Harcourt and Brace},
  title       = {Language},
  year        = {1949},
  bibtexkey   = {sapir_language1949},
  inlg        = {English [eng]},
  macro_area  = {Africa},
  src         = {wals},
  srctrickle  = {wals#5298}
}

Listing 1: Bib\TeX is used for reference data.
• Well supported in many bibliography management tools like
  • Zotero
  • jabref
• Our workflow is already adapted to it
• The (missing) details in the data model – e.g. no splitting of authors – align well with our messy data.
• We only use BibTeX as container format – no \LaTeX{} in field values, but UTF-8 encoded text.
Listing 2: INI files are used for metadata on languoids.
• Good support (e.g. syntax highlighting) in many text editors.
• The programming language Python supports reading and writing **INI** files out-of-the-box.
• Format is extensible – new sections and options can be added any time without disrupting the processing pipeline.
Listing 3: A directory tree is used to model the language classification.
Glottolog and collaboration
The GitHub workflow

**fork**  Create your own copy of the data repository. The repository you forked from is also called **upstream**.

**edit**  Change the data in your copy.

**commit**  Register meaningful groups of changes in your copy.

**pull request**  Propose merging your changes into upstream, i.e. **cld/glottolog**.

**merge**  Incorporate changes from other forks of the repository.
Use cases: Transfer of ownership

Forks are essential for the open source software development model for another reason as well:

They allow for seamless transfer of ownership of codebases.

For Glottolog this means

- the data repository can be forked - any fork is as good as the original repos
- the code for the web application has an open license, can be run anywhere, and ingest data from any fork
- the only thing bound to an institution that has to be explicitly transferred (with consent of the owner) is the domain name glottolog.org
Functionality built on top of the repository – rather than on top of the web application

- reduces traffic at glottolog.org
- works off-line
- works for forks, too
- thus, local changes can be incorporated in workflows right away
Use cases: Add "your" language

Working on varieties which are not in Glottolog?

- mint Glottocodes (using functionality built on top of the repository)
- add languoids to your fork of the repository
- use "your" Glottocodes in your data ...
- ...while waiting for "upstream" to incorporate your changes.
What happens when your changes are not accepted and merged into upstream?

- You either discard your changes, revert back to the status before and keep in synch with upstream;
- or you keep your changes,
  - and keep merging changes from upstream, resolving any conflicts resulting from your changes locally
  - or try to convince the community that your fork should become the new upstream repository (the "traditional meaning of fork in Open Source software development").
Example: Changing a language name – fork

![GitHub repository](https://github.com/clld/glottolog/blob/master/clld/glottolog/Mozilla Firefox)

- **Fork** button highlighted
- `glottolog / languoids / tree / sino1245 / kuki1245 / naga1409 / zeme1241 / mara1379 / tkhu1238 / tkhu1238.ini`
Example: Changing a language name – edit

```plaintext
# -*- coding: utf-8 -*-

[core]
name = T. Khullen
glotcode = tkuh1238
level = dialect
macroareas = Eurasia
countries =

[altnames]
multitree = T. Khullen
```
Example: Changing a language name – commit
Example: Changing a language name – pull request
Example: Changing a language name – pull request

Comparing changes
Choose two branches to see what’s changed or to start a new pull request. If you need to, you can also compare across forks.

- base fork: clid/glotolog
- base: master
- head fork: shh-dlce/glotolog
- compare: master

*Able to merge.* These branches can be automatically merged.

Create pull request
Discuss and review the changes in this comparison with others.

Commits on Sep 09, 2016
- xrotwang

Showing 1 changed file with 1 addition and 1 deletion.

```plaintext
languoids/tree/sino1245/kuki1245/naga1409/zeme1241/mara1379/tkhu1238/tkhu1238.ini
```

```plaintext
- name = T Khullen
+ name = T Khullen
```
Example: Changing a language name – pull request
Example: Changing a language name – merge

Now we are in the "upstream" repository!

The "merge" button is only visible to editors of this repository.
Example: Reviewing pull requests

Update kxoe1243.ini #12

Merged
d97hah merged 1 commit into clld:master from afheh:afheh-patch-1 on 15 Jun

afheh commented on 16 May

Changed lat/long to those of Rundu in the West Caprivi, which is nowadays the main settlement area of Khwe-speakers; previous lat/long implied that Khwe is a language of Zambia, which is not actually the case

Write

Leave a comment
More complex changes – such as re-arranging the classification of a subgroup or whole language family – typically

- start out as issues
- which can be discussed
- and eventually may lead to pull requests
- issues can easily be referenced in
  - commit messages
  - pull request descriptions
Example: Reclassifying Dogon

Anyone with a GitHub account can open issues.
Collaborate!

https://github.com/clld/glottolog