The \texttt{clld toolkit}

Robert Forkel and Sebastian Bank

Max Planck Institute for Evolutionary Anthropology, Leipzig

October 7, 2014
The CLLD project

The clld toolkit
- The data model
- ROA, REST and …
  …Linked Data
- Versioning, updating, preservation

Towards a domain specific API
- Decoupling database and visualization
- Semantic interoperability
The CLLD project: Overview

Funded by the Max Planck Society for 4 years.

Creates infrastructure for publishing cross-linguistic datasets, including

- organization: a publication platform [http://clld.org](http://clld.org) supporting two publication models:
  - Standalone databases following an "edited series" model, like WALS, WOLD, . . .
  - Two journals for cross-linguistic datasets

- infrastructure: Glottolog, a language catalog and comprehensive bibliography

- technology: the clld toolkit powering our applications
The CLLD project: Datasets

Typological:

- **WALS** - the World Atlas of Language Structures - a database of structural properties of more than 2600 languages
- **APiCS** - the Atlas of Pidgin and Creole Language Structures
- **SAILS** - the South American Indigenous Language Structures
- **PHOIBLE** - a repository of cross-linguistic phonological inventory data

Lexical:

- **WOLD** - the World Loanword Database contains vocabularies of 41 languages from around the world annotated for loanword status
- **Tsammalex** - a multilingual lexical database on plants and animals
- **IDS** - the Intercontinental Dictionary Series (to be published in CLLD in 2014)
- **ASJP** - the Automated Similarity Judgement Project (to be published in 2014)

Encyclopedic:

- **Glottolog** - a language catalog and comprehensive bibliography
The CLLD project: WALS

Feature 22A: Inflectional Synthesis of the Verb

Values

- 0-1 category per word: 5
- 2-3 categories per word: 24
- 4-5 categories per word: 52
- 6-7 categories per word: 31
- 8-9 categories per word: 24
- 10-11 categories per word: 7
- 12-13 categories per word: 2

The Flagship
The CLLD project: WOLD

The vocabulary contains 1515 meaning-word pairs ("entries") corresponding to core LWT meanings from the recipient language English. The corresponding test chapter was published in the book Loanwords in the World's Languages. The language page English contains a list of all loanwords arranged by donor language.

<table>
<thead>
<tr>
<th>Word form</th>
<th>LWT code</th>
<th>Meaning</th>
<th>Core list</th>
<th>Borrowed status</th>
<th>Source words</th>
</tr>
</thead>
<tbody>
<tr>
<td>soil</td>
<td>1.212</td>
<td>the soil</td>
<td>True</td>
<td>1. clearly borrowed</td>
<td>aequal 'ground' French aequalum 'ground, floor, soil' Latin</td>
</tr>
<tr>
<td>dust</td>
<td>1.213</td>
<td>the dust</td>
<td>True</td>
<td>5. no evidence for borrowing</td>
<td>moile 'bog' Dutch</td>
</tr>
<tr>
<td>mud</td>
<td>1.214</td>
<td>the mud</td>
<td>True</td>
<td>3. perhaps borrowed</td>
<td>give 'to give' Old Norse</td>
</tr>
<tr>
<td>sand</td>
<td>1.215</td>
<td>the sand</td>
<td>True</td>
<td>5. no evidence for borrowing</td>
<td>planit/planifit French (Anglo-Norman) planchus 'lamentation' Latin</td>
</tr>
<tr>
<td>give</td>
<td>11.21</td>
<td>to give</td>
<td>True</td>
<td>1. clearly borrowed</td>
<td></td>
</tr>
<tr>
<td>plaintiff</td>
<td>21.21</td>
<td>the plaintiff</td>
<td>True</td>
<td>1. clearly borrowed</td>
<td></td>
</tr>
</tbody>
</table>
The CLLD project: APiCS

1 Order of subject, object, and verb

Description

This feature (based on WALS feature 81, by Matthew S. Dryer) concerns the ordering of subject, object and verb in non-contrastive, non-focussed transitive clauses without special topicalization, more specifically declarative clauses with both the subject and object realized as full nouns (not as pronouns).

We use subject and object in a semantic sense, to refer to the agent-like and patient-like constituents in a monotransitive clause, as in e.g. French [Les soucs] mangent [le fromage] 'The mice eat the cheese'. As can be seen from this example, French has SVO order (Subject-Verb-Object), because the subject les soucs 'the mice' precedes the verb and the object le fromage 'the cheese' follows it. Since we only consider non-contrastive, non-focussed, non-topicalized clauses, cases like English It is the cheese that the mice eat (V=OSV) are disregarded here.

There are six logically possible orders of subject, object and verb, as shown in the list of feature values. Languages can have several word orders (e.g. German is SVO and VSO in main clauses and SOV in subordinate clauses), so several values can be true for this feature.
The CLLD project: AfBo

AfBo: A world-wide survey of affix borrowing

Latin affixes in Basque

Subjects are recipient/donor pairs rather than single languages

AfBo: A world-wide survey of affix borrowing by Seltart, Frank
is licensed under a Creative Commons Attribution 3.0 Unported License.
The CLLD project: eWAVE

2 He/him used for inanimate referents

- feature is pervasive or obligatory: 0
- feature is neither pervasive nor extremely rare: 12
- feature exists, but is extremely rare: 8
- attested absence of feature: 42
- feature is not applicable (given the structural make-up of the variety/PIC): 11
- no information on feature is available: 3

Feature area:
- Pronouns, pronoun exchange, nominal gender
- Typical example:
  I bet thee can’s climb he [s a tree]
- Example source:
  Southwest (Wagner 2008: 425)

Non-ISO languages

---
eWAVE edited by Kortmann, Bernd & Lunkemheiner, Kerstin
is licensed under a Creative Commons Attribution 3.0 Unported License.
The CLLD project: SAILS

SAILS: Macro-area specific
The CLLD project: PHOIBLE

PHOIBLE: more than maps...
The CLLD project: Tsammalex

Supplemental files
The CLLD project: Where’s my dataset?

Have a dataset in need of publication and presentation on the web?

- Submit to Harald’s Journal of Cross-Linguistic Databases or
- submit to Martin’s edited series of cross-linguistic databases clld.org or
- get a seasoned python programmer for a month to build your own app on top of the clld toolkit!

```
robert@astroman:/tmp/phoible$ cloc --exclude-dir=tests,data phoible/
 38 text files.
 36 unique files.
 28 files ignored.
```

<table>
<thead>
<tr>
<th>Language</th>
<th>files</th>
<th>blank</th>
<th>comment</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Python</td>
<td>17</td>
<td>230</td>
<td>173</td>
<td>954</td>
</tr>
<tr>
<td>CSS</td>
<td>1</td>
<td>25</td>
<td>49</td>
<td>159</td>
</tr>
<tr>
<td>Javascript</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SUM:</td>
<td>19</td>
<td>256</td>
<td>222</td>
<td>1113</td>
</tr>
</tbody>
</table>
The clld toolkit: Motivation

Survey databases are all alike.

Can we extract functionality needed to build WALS, WOLD, and APiCS into a reusable piece of software?

Design goals:

▶ There must be a core database model, which allows for as much shared functionality as possible.
▶ User interfaces of applications must be fully customizable.
▶ It must be easy to re-implement legacy applications using the framework.
▶ Optimize for maintainability, i.e. minimize lines-of-code for apps built with the framework.
▶ Find the right level of abstraction!
The `clld` toolkit is an open source Python package hosted on GitHub providing

- an extensible core data model
- a web application framework
  - powering all CLLD databases
  - providing a basic API built on Linked Data principles
  - "reference implementation" of a dataset browser
  - `clld` apps are web applications built as small layer of code on top of the `clld` framework.
- `clld` works with python 2.7 and 3.4 and has a test suite with 100% coverage.
Intermezzo: Disambiguation

- **CLLD**: The project.
- **clld.org**: The publisher/brand.
- **clld**: The software, aka toolkit, aka framework.
- **clld app**: A web application built using the clld framework.

In the remainder of this presentation we will talk about the latter two.
The design of the data model was guided by three principles:

- All the target datasets have to “fit in” without loss.
- The data model must be as abstract as necessary, as concrete as possible.
- The data model must be extensible.
**clld data model: Entities**

- **Dataset** holds metadata about a dataset like license and publisher information.
- **Language** may be a languoid (Glottolog) or doculect (ASJP).
- **Parameter** a feature that can be determined and coded for a language – e.g. a word meaning, or a typological feature.
- **ValueSet** set of values measured/observed/recorded for one language and one parameter, i.e. the points in the Language-Parameter-matrix.
- **Value** a single measurement (different types of scales can be modeled using custom attributes).
- **Unit** parts of a language system that are annotated, such as sounds, words or constructions.
- **UnitParameter** a feature that can be determined for a unit.
- **UnitValue** measurement for one unit and one unitparameter.
- **Contribution** ValueSets can be partitioned into separate contributions sharing provenance.
**clld data model: Relationships**

**Figure 1:** The default clld data model. Note: Modelling constructions as Units and features as UnitParameters the case mentioned by Harald fits in.
**clld data model: Extensibility**

*clld* uses *joined table inheritance* as implemented in SQLAlchemy to provide extensibility of the core data model:

- Each core model can be specialised/customized in a *clld* app, adding columns or relationships.

```python
@implementer(ILanguage)
class Languoid(Language, CustomModelMixin):
    ...
```

- The ORM (Object Relational Mapper) transparently joins the two corresponding tables when querying, retrieving the specialized object, i.e. the full set of columns.

- Additional models can be added freely, reusing *clld* functionality to enable functionality like versioning, etc.
**clld data model: Lexical data**

Figure 2: The WOLD instantiation of the data model.

```python
@implementer(interfaces.IValue)
class Counterpart(Value, CustomModelMixin):

    ...  
    word_pk = Column(Integer, ForeignKey('word.pk'))
    word = relationship(Word, backref='counterparts')
    ...
```
Figure 3: Many-to-many relation between words and meanings in WOLD.
**clld data model: Glottolog**

Figure 4: In Glottolog genealogy is implemented via a self-referential *father* relation on *Language*.

```python
@implementer(ILanguage)
class Languoid(Language, CustomModelMixin):
    ...
    father_pk = Column(Integer, ForeignKey('languoid.pk'))
    children = relationship(
        'Languoid',
        foreign_keys=[father_pk],
        backref=backref('father', remote_side=[pk]))
    ...
```
**cldd resources: Overview**

*Data done the Web way.*

cldd implements a Resource Oriented Architecture.

- Data model is good basis to support shared behaviour across apps.
- Resource concept makes model entities actionable.
  - Resources are the things we describe and publish.
  - Resources define the level of granularity that is of interest.
- cldd knows how to display filtered lists of resources of the same type
- and detail views of single resources.
clld resources: Adaption

- ZCA (Zope Component Architecture) provides machinery to register behaviour tied to interfaces, e.g. to resources.
- Resources can be adapted to representations:
  - Glottolog: Language represented as family tree in newick format.
  - ASJP: Contribution serialized in ASJP wordlist format.
  - All lists can be represented as feeds.
- The web pages created by a clld app are just resources adapted to HTML.
- These registry entries can be overridden by clld apps, e.g. providing custom DataTables, custom map markers, custom maps.
- Again it’s about the right level of abstraction: Writing a clld app as declarative as possible, just implement adapters.
### Figure 5: Two adaptations of a Language object in ASJP.
Figure 6: Tsammalex defines a new resource type EcoRegion. EcoRegions behave just like other resources, i.e. they can be listed, bookmarked and associated with maps.
We regard Linked Data principles as rules of best practice for publishing data on the web.

How do clld apps fare with respect to the five-star rating for Linked Data?

* Make your stuff available on the web (whatever format).
** Make it available as structured data (e.g. excel instead of image scan of a table).
*** Non-proprietary format (e.g. csv instead of excel).
**** Use URLs to identify things, so that people can point at your stuff.
***** Link your data to other people’s data to provide context.
clld and Linked Data: three stars

Make your stuff available on the web, as structured data in non-proprietary formats.

- clld apps do just that.
- Most CLLD datasets are published under CC-BY, i.e. open, licenses.
Figure 7: The data of a WALS feature is available in various formats. Note that the map on the page is created by calling the WALS API to retrieve the GeoJSON representation.
Use URLs to identify things, so that people can point at your stuff.

- “People” includes yourself
- forces you to think about the things you want to describe and at which level of granularity
- enables distributed development of data and the basis for merging via globally unique identifiers
- puts coarse provenance information in each identifier
Figure 8: The level of granularity of the WALS data allows to link comments, history and examples to datapoints.
clld and Linked Data: 4-out-of-5 stars

Generally, the usefulness of “4-out-of-5 stars” Linked Data has to be stressed:

- Linked Data as uniform data access API (following the “crawler” paradigm)
- enables distributed databases,
- allows follow-your-nose API discovery (cf. REST),
- plays well with the web at large (Internet archive, bookmarking, google, etc.),
- allows easy hosting (thus helps with sustainability, and is attractive for developers/administrators as well) – which cannot be said about SPARQL endpoints.
Publishing Linked Data can be as easy as putting a bunch of files on a web server.

- **clld** apps will be able to fall back to that, i.e. dumping the resources they serve as static files by enumerating their URL space.
- This allows for a graceful degradation of service:
  - When served from the app, resources will point to a canonical URI using the appropriate HTTP Link header.
  - These URIs will still resolve in the static-files-on-webserver scenario.
  - So when served as static files from a plain HTTP server, most things will still work.
Link your data to other people’s data to provide context.

While HTML provides the prime example of embedding links to provide context, for structured data and common domains RDF models are more useful.

- Again “other people” includes yourself.
- VoID is used to convey basic provenance and license information.
- Typically all statements of linguistic interest (i.e. value assignments) are linked to sources.
clld and Linked Data: the 5th star

- Our publication platform does spit out RDF.
- The RDF model for a particular clld app can be completely customized.
- But should it?
- Balance between
  - uniform access across CLLD apps and
  - semantic interoperability with existing infrastructure.
  - Is it more useful to model resources as having multiple types or provide mappings?
- Example: Model lexical data using lemon?
- Generally, in terms of user-friendliness, the problem is not a choice of RDF models but consumable formats (csv, Newick, . . . )
Glottolog as hub in the CLLD Linked Data cloud:
  - language catalog (linking in turn to lexvo, dbpedia, etc.), iso639-3 is often not sufficient.
  - shared bibliography

WOLD as catalog for comparison meanings (cf. Leipzig-Jakarta list) – a concepticon, or an ontology.

PHOIBLE may play such a role for phonological segments, e.g. as reference for transcriptions.

filling in blanks: Identify phonological descriptions for languages missing in PHOIBLE by inspecting Glottolog.

fill in missing values in WALS for phonological features by looking up PHOIBLE.
clld and Linked Data: A workflow for research based on CLLD data

1. Identify suitable datasets.
2. Aggregate the data in a triple store (crawling/importing dumps).
3. Filter data in the triple store (using provenance information, etc.).
4. Export data to suitable format for analysis.

- CLLD and Linked Data will mainly play a role during aggregation of raw data.
Several models are possible:

- versioned data in database
- only current data in database, archived older versions (ZENODO)
- updates via database migration scripts (versioned together with the software)
Figure 9: Archiving SAILS with ZENODO means longterm preservation and better citeability via DOI.
Standardization the Microsoft way?

- As demonstrated above, a standard software stack is useful.
- But software has a half-life of less than 10 years.
- Next step is essential: extract a **domain specific** API which can become standard.
  - Linked Data is still lacking in domain specificity.
  - Domain specific means semantic interoperability of linguistic concepts.
Towards a domain specific API: Decoupling database from visualization/analysis

▶ for OLAC there’s OAI-PMH
▶ for mapping (i.e. leaflet, tilemill) there’s GeoJSON
▶ but then there’s RefLex
▶ and http://phonotactics.anu.edu.au/
▶ and the WALS Sunburst explorer
▶ ...
**clld databases on OLAC**

Figure 10: 3 out of the top-ten of OLAC archives by number of distinct languages are based on CLLD datasets.

<table>
<thead>
<tr>
<th>Archive</th>
<th>Overall Rating</th>
<th>Number of Resources</th>
<th>Number of Resources Online</th>
<th>Distinct Languages</th>
<th>Distinct Linguistic Subfields</th>
<th>Distinct Linguistic Types</th>
<th>Distinct DCMI Types</th>
<th>Average Elements Per Record</th>
<th>Average Encoding Schemes Per Record</th>
<th>Average Metadata Quality Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Glottolog 2.3</td>
<td>⭐⭐⭐⭐⭐</td>
<td>7684</td>
<td>7684</td>
<td>7664</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>10.0</td>
<td>7.0</td>
<td>9.3</td>
</tr>
<tr>
<td>Ethnologue: Languages of the World</td>
<td>⭐⭐⭐⭐⭐</td>
<td>7480</td>
<td>7480</td>
<td>7479</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>10.0</td>
<td>7.0</td>
<td>8.3</td>
</tr>
<tr>
<td>SIL Language and Culture Archives</td>
<td>⭐⭐⭐⭐⭐</td>
<td>28448</td>
<td>5467</td>
<td>3080</td>
<td>0</td>
<td>3</td>
<td>5</td>
<td>13.2</td>
<td>8.3</td>
<td>8.9</td>
</tr>
<tr>
<td>The LINGUIST List Language Resources</td>
<td>⭐⭐⭐⭐⭐</td>
<td>2440</td>
<td>0</td>
<td>2430</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>11.0</td>
<td>7.0</td>
<td>8.4</td>
</tr>
<tr>
<td>WALS Online</td>
<td>⭐⭐⭐⭐⭐⭐</td>
<td>2621</td>
<td>2621</td>
<td>2420</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>10.0</td>
<td>7.0</td>
<td>9.3</td>
</tr>
<tr>
<td>The Rosetta Project: A Long Now Foundation Library of Human Language</td>
<td>⭐⭐⭐⭐⭐</td>
<td>6571</td>
<td>6571</td>
<td>2365</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>18.4</td>
<td>7.5</td>
<td>8.9</td>
</tr>
<tr>
<td>WALS Online ReIDB</td>
<td>⭐⭐⭐⭐⭐</td>
<td>7157</td>
<td>7157</td>
<td>2341</td>
<td>7</td>
<td>0</td>
<td>1</td>
<td>11.5</td>
<td>8.3</td>
<td>7.1</td>
</tr>
<tr>
<td>PHOIBLE Online</td>
<td>⭐⭐⭐⭐⭐⭐</td>
<td>1672</td>
<td>1672</td>
<td>1668</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>11.0</td>
<td>8.0</td>
<td>9.5</td>
</tr>
<tr>
<td>Graduate Institute of Applied Linguistics Library</td>
<td>⭐⭐⭐⭐⭐</td>
<td>8176</td>
<td>394</td>
<td>1335</td>
<td>23</td>
<td>3</td>
<td>5</td>
<td>14.3</td>
<td>7.2</td>
<td>7.8</td>
</tr>
<tr>
<td>Pacific And Regional Archive for Digital Sources in Endangered Cultures (PARADISEC)</td>
<td>⭐⭐⭐⭐⭐⭐</td>
<td>9266</td>
<td>9189</td>
<td>839</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>26.7</td>
<td>12.3</td>
<td>9.0</td>
</tr>
</tbody>
</table>
Figure 11: Configurable visualization of phonotactic features of the world’s languages.
Visualization: WALS Sunburst Explorer

Figure 12: Combined visualization of geolocation, genealogy and coding for a WALS feature.
Semantic interoperability

- Being able to evaluate provenance data during the aggregation of a dataset is useful (e.g. in the ASJP project, some sources of wordlists are regarded as less trustworthy than others).
- Unambiguous identification of languages is required; Glottolog will help with that.
  - Being able to answer the question “which data do we have on a selected sample of languages?” as well as
  - “what sample of languages can we investigate given we need a certain selection of data (lexical, structural, etc.)?”
- For lexical data lemon can help to interpret the raw data, i.e. matching senses across languages (cf. Moran and Brümmer 2013).
- The requirements of statistical methods may lead to a standardisation of structural language parameters (features in the WALS sense), but we are not there yet.
Semantic interoperability: Language identification

The languages described in APiCS and eWAVE show that iso639-3 is insufficient for language identification.
Semantic interoperability: Limitations

- Generally, useful data formats will be dictated by the needs of the analysis tools (e.g. phylogenetic software),
- so doing analyses directly on the RDF model can not be expected.
- Example APiCS: Interoperability of typological resources is hampered by the difficulty of cross-linguistic categories.
Semantic interoperability: APiCS and WALS

Figure 13: APiCS feature *Polar questions* – original and WALSified.
Towards a domain specific API

Roadmap:

1. "standardize" on software
2. determine what a proper API would look like (right now!)
   - collect use cases,
   - implement prototypes,
3. specify API – maybe ontologies, maybe RDF models, maybe ling-JSON . . .
http://clld.org

Thank you!