Introduction to Frictionless Data

Marek Szuba / GSI

2020-07-06, ESCAPE WP5 Tech Talk

Outline

The Problem of Data Integration

- Data integration is very important in modern science
- However:
  - different file/stream formats
  - different field syntax (*e.g.* dates, numbers)
  - different units
  - full 1:1 mapping rare: gaps, mistakes, ...
  - "special" values
  - unclear or varied QA criteria
  - licensing issues
  - ...
- "**Friction**" in data exchange

Open Knowledge Foundation

- Global, non-profit network
- Launched in Cambridge, UK in May 2004
- Chapters and groups in more than 40 countries
- Aim: advocate for open knowledge, including support for related projects
- `https://okfn.org/`

Frictionless Data

- OKN initiative
- Aim: "shorten the path from data to insight"
- Minimalist specifications for data and metadata
  - CSV and JSON
- *Open Source software libraries*
- Best-practice guides for data management
- Outreach activities
- Fellows Programme
- Adopted by French and UK governments, *BCO-DMO*, Kaggle, and more

BCO-DMO

- US Biological & Chemical Oceanography Data Management Office
- Collects oceanographic data from a multitude of sources
- *Very* messy input data:
  - numerous observables
  - input formats: from images to manually filled spreadsheets
  - no common convention for date format, temperature or depth units, *etc.*
  - metadata incomplete or implied
  - occasional corruption (or plain typos)
  - ...and so on
- Data managers clean the input and make it ready for hosting, using FD-based tools and pipelines

BCO-DMO

Lessons learned from using Frictionless Data:

- Faster generation of metadata
- Less custom scripting
- Fewer tedious, repetitive tasks
- Less dependent on programming skills
- More extensible

Work ongoing (as of mid-2019) on open-sourcing the whole pipeline.

Frictionless Data Toolset

- Table Schema Tools — declare and apply schemas for your data tables
- Data Package Tools — metadata I/O
- Tabulator — reading and writing of tabular data
- GoodTables — higher-level validator of tabular data
- Data Package Pipelines — declarative framework for data-processing pipelines
- DataHub — SaaS platform for publishing and sharing data

Table Schema Tools

- Libraries for Clojure, Go, Java, Javascript, Julia, PHP, Python, R, Ruby
- Optional support (mostly in Python) for data stores other than CSV files, *e.g.*: Pandas, SQL DBs
- Schema data: JSON Schema
- Supports remote links, streaming
- Inference, editing, validation and saving of schemas

Data Package Tools

- Libraries for Clojure, Go, Java, Javascript, Julia, MATLAB, PHP, Python, R, Ruby
- Data Package — a JSON file:
  - link to data file
  - schema
  - format, encoding
  - other metadata as needed
- Pass-through to TableSchema
- Supports grouping of resources
- Inference, editing, validation and saving of package files

Tabulator

- Python with a CLI interface
- Read-write support for CSV, MS Excel (both legacy and XML), SQL, JSON
- Read-only support for inline data, OpenDocument spreadsheets, Google Spreadsheets, Data Package, HTML tables, Newline Delimited JSON, TSV
- Supports multiple access schemes off the bat: inline data, Python file streams, local files, HTTP(S), FTP(S), AWS S3
- Transparent handling of ZIP- and GZIP-compressed files
- Extensible with custom file formats and loaders

GoodTables

- Python with a CLI interface
- More than just schema checking — can be used to validate data itself
- Not dependent on Table Schema
- Comes with a set of rules for handling common tabular-data errors
- API for creating custom checks

Data Package Pipelines

- Python with a CLI interface
  - `Dataflows` recommended for Python integration
- Aims:
  - capable of all ETL operations
  - suitable for heterogeneous data
  - not requiring programming skills
- Configuration: YAML file
- Streaming via stdout-stdin chaining of processors
- Standard library of processors
  - mostly for tabular text data
- Plug-in API
  - AWS S3, ElasticSearch, GitHub, GoodTables, ...
- Celery and webhook support
- Built-in Web dashboard

Data Package Pipelines

Example pipeline configuration

```yaml
worldbank-co2-emissions:
  title: CO2 emission data from the World Bank
  description: Data per year, provided in metric tons per capita.
  environment:
    DEBUG: true
  pipeline:
    - run: update_package
      parameters:
        name: 'co2-emissions'
        title: 'CO2 emissions (metric tons per capita)'
        homepage: 'http://worldbank.org/'
    - run: load
      parameters:
        from: "http://api.worldbank.org/v2/en/indicator/EN.ATM.CO2E.PC?downloadformat=excel"
        name: 'global-data'
        format: xls
        headers: 4
    - run: set_types
      parameters:
         resources: global-data
         types:
           "[12][0-9]{3}":
              type: number
    - run: dump_to_zip
      parameters:
          out-file: co2-emissions-wb.zip
  schedule:
    crontab: '0 * * * *'
```

DataHub

- Software-as-a-service platform built on Open Source and Frictionless Data
- Provided by Datopian at `https://datahub.io/`
- Access plans:
  - Core Data – free, infrequently updated, limited access
  - Premium Data — updates with notifications, workflow integrations, custom requests
- Publishing plans:
  - basic — free, max 1 GB of storage and bandwidth, no private data
  - several premium tiers, including “pay as you go”

DEMO

(nb. all Python modules are available on PyPI)

Frictionless Data and ESAP

Potential benefits to ESAP:

- Standardised metadata via Data Package
  - usable with non-tabular data too
- Table Schema, Tabulator — simplified, unified handling of tabular data
  - ...including the ubiquitous spreadsheet
- GoodTables — extended validation of input Possible implementation:
  - Python → Jupyter Notebook → trivial for most tools?
  - Data Package Pipelines — would have to consider use cases *vs* available modules

Further Reading

- Official how-to guide: `https://frictionlessdata.io/guide/`
- Specs, code, Web sites, slides, ...: `https://github.com/frictionlessdata/`
- FD at csv,conf,v5: `https://frictionlessdata.io/blog/2020/06/26/csvconf-frictionless-recap/`
- BCO-DMO pilot at csv,conv,v4: `https://doi.org/10.5281/zenodo.2687557`

THANK YOU