Working with CERL data

Andreas Walker
CERL Annual General Meeting
2020-10-06
Preliminary notes

- The presentation is about half an hour, the rest of the time is for questions and discussion.
- We will be recording the presentation and discussion session.
- Slides are online here: gwdg.de/~walker5/docs/20201006_slides.pdf (useful if you want to click on the links).
How to get a burger

- **Raw ingredients**

- **Eat ready-made**, low effort but somewhat dependent on the cook

- **Make your own** from ingredients, packed for your convenience

Image credits: Burger Ingredients Vectors by Vecteezy | Cheese Burger by Jun Seita | Grocery Bag by Dawn Hudson
Two views on our data

**Internal format**
(JSON)

- **Aims at:**
  - Usability for human users
  - Generalistic tool

- **Restricted by:**
  - Available technologies
  - CERL development resources
  - Earlier design decisions

**Web interface**

- **Aims at:**
  - Machine-readable data
  - Building / using your own tools

- **Restricted by:**
  - Available technologies
  - Your own resources

**Serializations**
Capabilities of the web interface – what it does really well

- **Search** for records
  - Both full-text search and search in specific data fields
  - Fairly sophisticated search syntax (ElasticSearch)
    - See e.g. MEI Searching Guidelines or ElasticSearch documentation
  - Filter search results further by pre-defined facets

- **Display and edit** records in a document-like view
  - Shows *all* the information available in a record
  - Aims at human reader who intelligently extracts needed information
  - Assumes you are working with a small number of records and can still do the work of intelligent extraction for each one
Capabilities of the web interface – what it does really well

- Multilingual labels
- Information pulled from other databases for display (record only contains ID)
- Related information grouped visually
Limitations of the web interface – where it struggles

• Filter records based on more complex criteria
  • Search syntax is based on presence of information in particular data fields, but cannot usually describe more complex data structures
  • Example: Search in MEI can show you documents that have (a) a 16th century provenance and (b) information on purchase prices, but it cannot be used to know whether the price information is in the 16th century provenance block

• Display and edit records based on particular tasks
  • Because displaying a record is a non-trivial task that requires decisions about mapping the data model to a human-readable display, we can only provide generic solutions
  • Users and editors end up doing the cognitive labor of ignoring data they do not need for the task at hand, or adding data from external sources
Limitations of the web interface – where it struggles

It takes a lot of scrolling to edit this piece of information
Going beyond the web interface – cooking your own

- For some tasks, it can be useful to switch from the web interface to our second option: using and/or building **your own tools** for working with CERL data

- This can range from **integrating the data with your existing workflows** (e.g. working with CERL data in your favorite spreadsheet software) to **building specialized software** (e.g. a Python script that produces statistics on a set of CERL data), with different requirements in terms of technical expertise

- Going beyond the web interface will require input from both **domain experts** and **technology experts**, depending on the task and the available tools (and of course the two roles are **not mutually exclusive**).
Going beyond the web interface – what do you need?

- The web interface does not (and cannot) perfectly present the data model – it is always a translation. But it often informs our own mental models of the data.

- A first step in working with the “raw” data: developing a more precise mental model of how the data is structured, and mapping it to your own mental model of the domain.

- Example: You have a fairly complex mental representation of a book purchase, and as a human being you can easily manipulate it to ask questions centered on particular elements of that representation, like prices within a date range. How does this mental representation correspond to the data model?
Going beyond the web interface – reading the model

- You need to
  - find the *corresponding elements* in the data model,
  - diagnose how they are *related by structure*,
  - understand whether the combined information of data elements and structure can *answer your question* and
  - decide whether you need to *enrich or simplify* the data for the task at hand
Going beyond the web interface – reading the model

<table>
<thead>
<tr>
<th>timeperiod:</th>
</tr>
</thead>
<tbody>
<tr>
<td>start: 1550</td>
</tr>
<tr>
<td>end: 1600</td>
</tr>
</tbody>
</table>

- **priceCurrency:** "Lire: Soldi"
- **priceAmount:** "1:16"

All elements **grouped** in a single provenance block
Going beyond the web interface – simplification

• A standard case of simplification is representing the data in a format that is cognitively more accessible to humans but does not capture the full complexity of the data model’s structure:

• Example: You want to edit records in your favorite spreadsheet application, so you need to simplify the tree-like structure of a record into a tabular structure of columns and rows

• Challenges:
  • There is not a single mapping from a more complex to a simpler data structure, so you have to decide which one to use (which is exactly what happens if you use the CSV download functionality in AMPLE)
  • Simplification always means a loss of information. This is particularly important if you later want to manipulate the data and put it back in the database
Going beyond the web interface – simplification

Choosing the appropriate **simplification** for your task
Going beyond the web interface – enrichment

- A standard case of enrichment is adding data from an external source, e.g. another database or even your own research.

- **Example:** A spreadsheet of information about book purchases contains ISTC numbers; bibliographic information about the titles is then pulled from the database and put into the spreadsheet automatically.

- **Challenges:**
  - You need to match up the two data sources with one another, e.g. by using unique identifiers.
  - The two data sources may have incompatible structures, making it necessary to simplify both to a shared common core.
Going beyond the web interface – we at DCG

- We want to make life easier for both domain and technology experts interested in working with CERL data (that includes ourselves)

- We provide **various serialization formats** for our data that help communicate to technology experts what can be done computationally with the data

- We **choose helpful ontologies** for our RDF representation that connect our data model to well-known standards in the GLAM world
  
  - However, standards develop over time, so this needs to be revisited (see also my blog post on the topic)
  
  - I would suggest forming a small CERL working group for people interested in the question of how to best present our data using existing ontologies (as an on-going task rather than a one-time decision)
Going beyond the web interface – ontologies

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix rel: <http://purl.org/vocab/relationship/>.
@prefix gn: <http://www.geonames.org/ontology#>.
@prefix rdaGr3: <http://rdvocab.info/ElementsGr3/>.
@prefix skos: <http://www.w3.org/1999/02/skos/core#>.
@prefix gnd: <http://d-nb.info/gnd/>.
@prefix ct: <http://www.cerl.org/namespaces/thesaurus/>.
@prefix rdaGr2: <http://rdvocab.info/ElementsGr2/>.
@prefix rdaRelGr2: <http://metadataregistry.org>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix wgs84_pos: <http://www.w3.org/2003/01/gps/wgs84_pos#>.
@prefix edm: <http://europeana.eu/terms/>.
```

Status: Deprecated
Who is using our data, and how?

- See the DCG paper on Linked Open Data from May 2020 for a small bibliography of external projects making use of our data to, e.g.:
  - Normalize their own data based on the Thesaurus
  - Answer questions about large-scale bibliographic trends

- But we also have use cases inside CERL:
  - Providing researchers with tabular representations of our data sets
  - Enriching externally created spreadsheets with data from our databases
  - Answering questions that cannot be answered by search

- Coming up:
  - We are providing MEI data for Coding da Vinci, a regional Hackathon in Hannover this year *(What is a hackathon?)*
Going beyond the web interface: the DH community

• The DH community offers a lot of **ready-made tools** for secondary steps like visualization and analysis of data (still usually needed: data pre-processing, transformation, cleanup)

• For some possibilities of employing these tools for working with CERL data, see Alex Jahnke’s talk at *Printing Revolution & Society 1450-1500. Venice Conference, Palazzo Ducale, 19-21 Sept. 2018*: Watch on YouTube

• Also, don’t forget about the new **DH working group** at CERL (and join us?)

• But so far, we have seen **little uptake of our data** in that community
  
  • Is it there, but we don’t hear about it? (Please tell us!)
  
  • Are there **barriers** to using our data?
    
    • Necessary domain expertise may not be available easily
    • It may simply not be well-known enough
Towards a „GLAM workbench“ for CERL

- There is still a lot of untapped potential in our data

- Tim Sherratt presents the idea of the GLAM workbench, a collection of tools and building blocks for working with data from GLAM institutions

- We envision something similar for CERL:
  - Examples for making use of our data outside the interface
  - Explain our data from both perspectives (domain and technology)
  - Position as a “translation” between CERL community and DH community

- Thoughts on concrete implementation:
  - Collection of Jupyter Notebooks (heavily annotated Python code)
  - See also the National Library of Scotland’s newly launched notebooks
  - DCG to provide some initial contents (see next slides)
  - Aim to get more contributions from wider CERL community (and beyond)

Image credits: Project Jupyter Logo by Project Jupyter Contributors
The anatomy of a Jupyter Notebook

- Mixes Python code (or other languages) with detailed descriptions, making it possible to explain every step conceptually and technically.

- Can be loaded in various environments as an interactive tool, exported as a script, or exported to a static reading version (e.g. a website or PDF).
Some examples

- “What’s a good name for a printer from Göttingen?”
  - displaying **bar charts for categorical data** in the CERL Thesaurus

- Preview of the notebook
Some examples

- “Who came to Göttingen, and where did they go?”
  - displaying a **map view for geocoordinates** from the CERL Thesaurus

- Preview of the notebook
Some examples

• “How did these books come to Princeton?”
  - displaying a map view for geographic networks from MEI

• Preview of the notebook
Thank you

Contact: Andreas Walker (walker@sub.uni-goettingen.de)
Some more notebooks

- Biblioteca Virtual Miguel de Cervantes: http://data.cervantesvirtual.com/blog/notebooks/
- National Library of Scotland: https://data.nls.uk/tools/jupyter-notebooks/
- Tim Sherratt's presentation (LIBER Webinar)
- List of Jupyter notebooks beyond cultural data: https://github.com/jupyter/jupyter/wiki/A-gallery-of-interesting-Jupyter-Notebooks