Graph-Based Archival Description

Aaron Hope, Amir Lavie, Jill Ruby, and Pavel Zhelnov

The Archives of Ontario's experience with Records in Contexts and Linked Open Data

Patrick Keilty - GLAM Incubator Director, University of Toronto
Anastasia Kuzminykh - Project Advisor, University of Toronto
Aaron Hope - Senior Archivist and Project Member, Archives of Ontario
Pavel Zhelnov - Research Fellow and Project Manager, University of Toronto
Cate Cleo Alexander - GLAM Incubator Coordinator, University of Toronto
Shion Guha - Project Advisor, University of Toronto

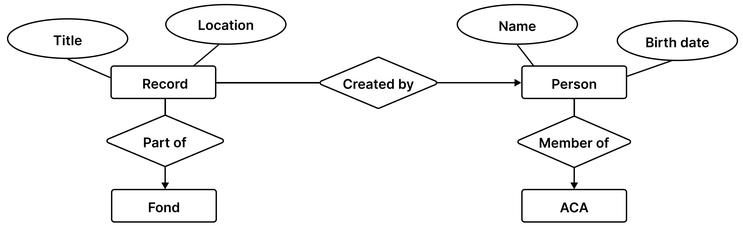
Peiwen Zhang - Research Assistant, University of Toronto
Jill Ruby - Archivist and Project Member, Archives of Ontario
Amir Lavie - Archivist and Project Member, Archives of Ontario
Thomas Fox - Student Designer, University of Toronto
Harrison Huang - Student Designer, University of Toronto
Russell Luchin - Student Designer, University of Toronto

Background

The ICA's Records in Contexts Conceptual Model aims to give archives the ability to link descriptive data to external resources, integrate descriptive entities and records across organizations, and capture multiple contextual dimensions of the same record. By allowing archivists to include various viewpoints and descriptions, as well as information about their own positionality, the ICA's standard aims to facilitate more equitable archival practices.

Objectives

This project explores a method for adopting the International Council on Archives' new Records in Contexts (RiC) standard and Linked Open Data (RDF data format) using the vast datasets of the Archives of Ontario. The team intends to establish a proof-of-concept allowing machine-readable encoding of meaning that enables complex logical inferences and search capabilities, and to connect currently siloed datasets with related data across the world.



Prepare Existing Data

Review the archive's existing data structure, identifying all column headers, and ensuring consistent naming practices.

Existing datasets (Authority, Descriptions, Listings, etc) should be converted to .csv format when following this project's procedure. The first row of each .csv should contain all possible headers found in dataset.

SQL databases can be used in place of .csv if available. Small modifications to the tool's code are necessary to enable this; interested individuals can reach out via contacts indicated on the project's GitHub page or open an issue at: https://github.com/gbad-project

Create Visual Schema

Model existing data fields and relationships by building visual graph templates in Draw.io using Richard Williamson's RiC-O Shape Library.

This project's Draw.io schema is freely available from the project's GitHub page (https://github.com/gbad-project), and can be referenced or modified to better suit specific organizational needs.

Once all relationships from existing data sets are specified move on to running parser scripts in order to convert .csv data to graph-based data types (.RML, .TTL, .TRIG).

Template | Draw.io File

Subject, Predicate, Object

RDF's conceptual model for representing relationships between resources and entities.

What is a Triple?

Implements the subject, predicate, object framework for graph-based data structures.

A fourth term, context, can be added to each triple to make them quads, enabling support for storing multiple named graphs within one database

.RML (mapping)
.TTL (Triple implementation)
.TRIG (Quad implementation)

Create Custom Triples

Apart from building visual graph schemas, custom mappings extending RiC-O can also be added by writing simple code in .ttl or .trig format, specifying required triples or quads for named graphs.

Records and entities can also be added individually by writing simple .ttl/.trig code, or visually by mapping relationships and specifying data fields within Draw.io.

Example's of both custom options can be found at the links below:

TTL Map | Visual Entry

Run Parser Scripts

Draw_io_parser.py
Parses draw.io file and generates .rml, an intermediary .ttl file representing visual schema.

Map_schema.py
Applies preprocessing to .csv if necessary (e.g., separate/merge columns or replace/clean some values according to pre-specified rules). Links the preprocessed dataset .csv to .rml.

Map_rml.py
Maps the .rml file to create the final .ttl file. Applies post-processing to .ttl if necessary (e.g., remove or add some triples based on a set of predefined SPARQL queries).

Graph-based Output

Mapped and validated data is returned by Map_rml.py in .ttl or .trig format, containing entities and relationships.

Load the generated .ttl or .trig file using triplestore software (e.g., Jena Fuseki, Virtuoso, GraphDB, etc.) to enable graph-based description, visualization, and SPARQL querying.

Both free and paid software options are available.

A small sample of this project's converted data in .ttl format can be found below:

Final Output

Resources

Faculty of Information, *A GLAM makeover for the Archives of Ontario [Video], 2024.*

GLAM Incubator, *Graph-Based Archival Description at the Archives of Ontario*, 2023.

ICA, *Records in Context Conceptual Model Version 1.0*, 2022.

ICA, *Introducing Records in Contexts: The New ICA Standard for Describing Records* [Video], 2022.

Richard Dancy, *Waiting for RiC*, Archivaria 98, 2024.

World Wide Web Consortium, RDF Primer 1.1, 2014.

Key Findings

- Graph-Based Archival Description can be implemented using freely available tools and resources.
- Visual schemas can help conceptualize relationships between records and various entities.
- Creating schemas that include all possible column headers allows for the conversion of an archive's entire dataset (over 2 million entries in this case), rather than only small samples.
- Data can be manually converted into TTL format by writing simple code or visually representing relationships using Draw.io.

Discussion

As the ICA's new standard is highly technical, there have been concerns surrounding varying executions between organizations, the level of expertise needed to develop an implementation, and the capital required to purchase RiC enabled information systems. This project's findings demonstrate the possibilities for implementing Graph-Based Archival Description using Linked Open-Data and freely available tools and resources. The team's implementation, while in its alpha stage, is openly available for any organization to utilize, and will soon be followed by a "release" version on the project's GitHub repository. Interested parties are encouraged to reach out to the contacts indicated on the project's GitHub page at: https://github.com/gbad-project