Baroque AI

Programme Guide

by Simon Worthington

Version v1.0

Published by: Open Science Lab

Last updated: 2023-03-17

Created: 2023-03-16

Language: English (United States)

Created by: Simon Worthington

Contents

Baroque AI

Publication prototype

A prototype publication of a fictional 'exhibition catalogue' based on a Wikidata based collection of seventeenth century painting from the Bavarian State Painting Collections. The prototype shows how with a compuational publishing pipeline different distributed linked open data (LOD) sources can be brough together in a multi-format computational publication — allowing for asynchronous collaborative working. Distributed LOD sources include: Wikidata/base, Nextcloud, Thoth, Semantic Kompakkt, TIB AV Portal, and more.

Prototype series: Baroque TOC

Coordinated by Simon Worthington - NFDI4Culture @Open Science Lab, TIB, Hannover

March-April 2023

Venus und Cupido, Heinrich Bollandt, between circa 1620 and circa 1630. https://commons.wikimedia.org/wiki/File:Heinrich_Bollandt_-_Venus_und_Cupido.jpg This work is in the public domain.

Important links for the class

Coordination

- Class information and links: https://nfdi4culture.github.io/class-ADA-CP-pipeline/
- Project management and ticketing: https://github.com/orgs/NFDI4Culture/projects/2/views/1

Publication

- Demo publication: https://nfdi4culture.github.io/catalogue-003/

- Repo link: https://github.com/NFDI4Culture/catalogue-003

Activities

- Nextcloud Markdown document link: https://tib.eu/cloud/s/qBx8SbqiPBBedye
- Wikidata: collection query
- Jupyter Notebook - TIB AV Portal and Semantic Kompakkt: https://github.com/NFDI4Culture/video-and-3d-notebook
- GitHub: Fork and Clone publication repository: https://github.com/NFDI4Culture/catalogue-003

To edit this document - request access by emailing simon.worthington@tib.eu.

All software used is open-source OSI licence compliant. All content and other resources are open access with open licences.

To edit this document - request access by emailing simon.worthington@tib.eu.

Sample publications

Prototype exhibition catalogue: *<toc>Baroque</toc>*

A prototype framework publication for an exhibition catalogue.

<toc>
Baroque
Workshop Programme: Publishing
from Collections: Introducing
Computational Publishing for Culture
</toc>

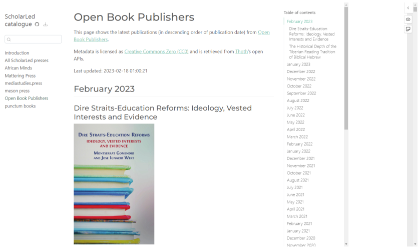
Cover

The catalogue uses a Wikidata based collection of Bavarian collections of Baroque paintings. See: 17C Bavarian painting

Publication URL: https://simonxix.github.io/Experimental_Books_workshop/

Prototype publication catalogue: *ScholarlyLed Catalogue*

Sample output of publishers titles from the Thoth single source book metadata service API.



Open Book Publishers

This page shows the latest publications (in descending order of publication date) from Open Book Publishers.

Metadata is licensed as Creative Commons Zero (CC0) and is retrieved from Thoth's open APIs.

Last updated: 2023-02-18 01:00:21

February 2023

Dire Straits-Education Reforms: Ideology, Vested Interests and Evidence

Website

Publication URL: https://simonxix.github.io/scholarled_catalogue/

Long list

Example publications:

- Exhibition catalogue demo: toc Baroque /toc from Experimental Books – Re-imagining Scholarly Publishing, COPIM. Workshop URL: https://experimentalbooks.pubpub.org/programme-overview
- Publishers catalogue demo: ScholarLed A catalogue of ScholarLed presses built on a Quarto / Jupyter Notebook model for computational publishing. The publication is automatically updated daily to reflect any new books added by the publishers.
- Proof of concept #1 - Computational Publication: Computational Publishing for Collections - ADA CP Prototype #1 - Nov 22
- Proof of concept #2 - To be confirmed, completion for end of April 2023. This contains all parts fully rendered: Cover, colophon, essay, collection, graph, TIB AV Portal, Semantic Kompakkt
- semanticClimate: To be confirmed - customised research papers readers made for regional climate change action plans based on IPCC reports and sourcing content from open research repositories.
- FSCI Summer School - publishing from collections class: To be confirmed, July 2023

About the prototype

Publication type: Use case - An exhibition catalogue

1. We are creating a *demonstration prototype*: **An exhibition catalogue about a baroque painting collection.**
2. Objectives:
    1. Write an exhibition catalogue essay using AI tools
    2. Review the 'catalogue essay and AI tools' as open peer review
    3. Create the parts of the the catalogue:
        1. Cover
        2. Colophon
        3. Essay
        4. Collection
3. What is the collection?
    1. The catalogue uses **part** of a Wikidata based collection of Bavarian collections of Baroque paintings. See: 17C Bavarian painting.
    2. We focus on the Baroque period: Bavarian Collections, 1590-1750 query link
    3. We make a small collection of paintings - 9 in this case.
4. How are we using computational publishing and what is the prototype experiment?
    1. Creating a publication from different **distributed** (federated) remote sources using linked open data.
    2. Showing how asyncronous work can be carried out by team working on a single publication - this is the power of the **TOC** part! Which in more advanced domains becomes **package management**.

Learning points

Workflow activities that will be covered to create the exhibition catalogue:

1. Real-time collaborative editing,
2. Creating a Wikidata query of a collection,
3. Displaying a painting catalogue sample collection from Wikidata LOD query for a multi-format publication.
4. Editing a Jupyter Notebook in MyBinder,
5. Embedding media objects: Video - TIB AV Portal, and; Semantic Kompakkt,
6. Using GitHub
7. Accessing API content for colophon
8. Editing Wikidata collection query in Juypter Notebooks
9. Asycrononous collective working and making a publication from multiple remote Linked Open Data (LOD) sources, and
10. Rendering a multi-format publication with CSS styling.

Software (open-source)

Over 2023/24 the computational components will be added to the ***ADA Semantic Publishing Pipeline*** as well as introducing **Vivliostyle Create Book** markdown renderer and swapping to **Jupyter Book** computational book platform away from Quarto – https://github.com/NFDI4Culture/ada

- Wikidata – https://www.wikidata.org/
- Jupyter Notebooks – https://jupyter.org/
- Jupyter Book – https://jupyterbook.org/
- Quarto – https://quarto.org/
- Semantic Kompakkt – https://semantic-kompakkt.de/
- TIB AV Portal – https://av.tib.eu/
- HedgeDoc – https://HedgeDoc.org/
- Thoth – https://thoth.pub/
- Vivliostyle – https://vivliostyle.org/
    - Create Book – Markdown renderer
- Wikibase – https://wikiba.se/
- Zenodo - https://zenodo.org/
- NextCloud - Tetx editor and Markdown editor - Text : https://github.com/nextcloud/text Markdown: https://apps.nextcloud.com/apps/files_markdown

AI Software

To be confirmed

https://openai.com/blog/chatgpt

https://www.perplexity.ai/

Activity: Nextcloud Markdown editing

Nextcloud Markdown document link: https://tib.eu/cloud/s/qBx8SbqiPBBedye

- Nextcloud has a markdown editor that allows real-time editing.
- The files generated here can simply be added to the publication **TOC** and then rendered in the final publication.

Catalogue essay

The markdown editor will be used to create the catalogue essay.

AI tools will be used to generate a sample text.

The tools and the text will be reviewed using Open Peer Review methods.

Sample AI tools

- https://openai.com/blog/chatgpt
- https://www.perplexity.ai/

About open peer review

Worthington, S. (2022). Designing an Open Peer Review Process for Open Access Guides. Community-Led Open Publication Infrastructures for Monographs (COPIM). https://doi.org/10.21428/785a6451.e0245b43

Activity: Create a Wikidata query

Objective: User builds a Wikidata query. See example query: paintings, Bavarian Collections, 1590 - 1750 - query link

External LOD and media used: Wikidata LOD, and Wiki Commons, Web Gallery of Art (images) - https://www.wga.hu/

Notes: Wikidata Query (help)

- Allows for non-expert query building with plain language
- View query as plain language and as code
- Experience of building a query
- Contact with some basic building blocks of Wikidata
- View and export SPARQL query

Steps

1. Go to https://query.wikidata.org/
2. Build a query around the 17C Bavarian painting collection to replicate the catalogue selection to be used in Activity B. Example:
    1. Code Repo: Current publication link
    2. Rendering: Current publication link
    3. Example: Paintings; in collection; Bavarian Collections; 1600 - 1700 - query link
3. ***Step-by-step instructions*** to replicate parts of this query link base on this collection 17C Bavarian painting:
    1. Go to https://query.wikidata.org/
    2. Enable split view with ***i info*** button top left.
    3. Filter: **instance of** P31, **painting** Q3305213 - wdt:P31 wd:Q3305213.
    4. Filter: **collection** P195, **Bavarian State Painting Collection** Q812285 - wdt:P195 wd:Q812285.

    5. Play button - bottom left - renders query below
    6. Show: **creator** P170; **image** P18; **copyright status** P6216; **inception** P571.
    7. Play button - bottom left - renders query below
    8. Image grid view :-)
    9. Limit
    10. Dates from to 1590-1750 (code only) BIND(YEAR(?inception) AS ?inceptionyear) FILTER((1590 <= ?inceptionyear) && (?inceptionyear < 1750))
    11. Link query: https://w.wiki/6MGX - results: https://w.wiki/6MGY

4. Participants can change the selection criteria around the available criteria: artists, dates, etc., as in collection 17C Bavarian painting
5. Completion: Paste your query link into the HedgeDoc link provided. https://demo.HedgeDoc.org/s/4gr9JvUS7 - END of activity.

Activity: Editing a Jupyter Notebooks and accessing video

Objective: Running and editing Juypter Notebooks in MyBinder and retrieving video and 3D models as embeds.

External LOD and media used: TIB AV Portal, and Semantic Kompakkt

Notes: Jupyter Notebooks editing in MyBinder

- Run a Jupyter Notebook in MyBinder
- Edit a Jupyter Notebook
- Render a Jupyter Notebooks

Links:

- Sample Jupyter Notebook: Video and 3D Notebook embeds
- TIB AV Portal: https://av.tib.eu/
- Semantic Kompakkt demo site: https://kompakkt.wbworkshop.tibwiki.io/explore
- View a model, copy the iframe embed from the folder icon, top right. In the Notebook paste in the complete iframe cover replacing the existing iframe: <iframe name="Doric Column" src="https://kompakkt.wbworkshop.tibwiki.io/viewer/?entity=63e8c22910e4f555d1f656ca&mode=open" allowfullscreen loading="lazy" ></iframe>

Steps

1. Open Notebook in the browser using MyBinder - Video and 3D Notebook embeds - click the 'launch binder' button to run the Notebook in MyBinder.
2. Add new videos and 3D models to the Notebook from TIB AV Portal and Semantic Kompakkt.
    1. Open a second browser tab and load TIB AV Portal
    2. Choose a video and copy across the video ID from the URL https://av.tib.eu/media/60729
    3. Paste the video ID into the video iframe field and run the cell to render
    4. Open Semantic Kompakkt demo site: https://kompakkt.wbworkshop.tibwiki.io/explore
    5. View a model, copy the iframe embed from the folder icon, top right. In the Notebook paste in the complete iframe cover replacing the existing iframe: <iframe name="Doric Column" src="https://kompakkt.wbworkshop.tibwiki.io/viewer/?entity=63e8c22910e4f555d1f656ca&mode=open" allowfullscreen loading="lazy" ></iframe>
3. Run the Notebook
4. 3D view size, we can make the initial view bigger, add: <iframe width="1200" height="630"
5. Download Notebook
6. Render some videos and 3D models in the Quarto book. Pass along **video id codes** and **3d models** using a hedge doc and chat to the Quarto render. The rendering and final display will take less than 10 minutes (hopefully): a. The code needs to be added to the main repo; b. Rendered locally; c. Uploaded to GitHub; d. Time for GitHub Pages to finish loading.
    1. Code: https://github.com/SimonXIX/Experimental_Books_workshop/blob/main/paintings.ipynb
    2. Rendering: https://simonxix.github.io/Experimental_Books_workshop/paintings.html

Activity: GitHub on boarding

Objective: On boarding and familiarisation with using GitHub for publishing and asynchronous working.

Publication repository: https://github.com/NFDI4Culture/catalogue-003

- Creating an account
- Joining an organisation
- Forking a repository
- Cloning a repository
- Turning on Github Pages
- Enabling a local editor: Visual Code Editor
- Attribution and citation