Designing an Alternative Approach to Mathematical Research

Kai Prince

2026-09-26

[Book Project]

[Check Journal-format Thesis requirements]

1 Introduction

Concept: Designing a Quarto Extension to generate an Open Education Website to maximise the accessibility of Open Research projects.

[Insert OpenMath motivations]

[Insert Disability Inclusion in Research motivations]

2 Open Research

The "Position Statement on Open Research" at The University of Manchester (2025c) covers what Open Research is, why it is important and its principles:

- Pre-registration of research;
- Transparency in research methodology;
- Public availability and reusability of research data and analysis code;
- Public accessibility and transparency of research communication;
- Data described according to FAIR Data Principles, ensuring that it is Findable, Accessible Interoperable and Reusable and publicly available where possible;
- Using web-based tools to facilitate collaboration.

2.1 UoM Policies

The University of Manchester (2025a) details state that researchers must be open and transparent when conducting and communicating their research in terms of:

1. the disclosure of any conflicts of interest;

2. the reporting of research data collection methods;
3. the analysis and interpretation of data;
4. making all research findings widely available (including sharing negative results as appropriate) by publishing as open access under a Creative Commons Attribution (CC BY) license;
5. disseminating research in a way that will maximise engagement and secondary use, with students obtaining prior permission from their supervisor; and
6. promoting public engagement/involvement in research.

The University of Manchester (2024) that research data must be stored in an "accurate, complete unadulterated and reliable manner" as well as meeting FAIR data principles, where the data must be made interoperable whenever possible, and "made available in a timely way with as few restrictions as possible".

The University of Manchester (2025b) endorse five principles of responsible metrics:

- Robustness
- Humility
- Transparency
- Diversity
- Reflexivity

3 Mathematical Research

3.1 Researcher Lifecycle

[Conceptual overlaps between Open Research & Open Education]

3.2 Mathematical Research Methodologies

3.3 Knowledge Exchange Initiatives

3.4 Research Outputs

[Papers]

[Presentations]

[Posters]

[Social Media Content]

[Public Outreach]

3.5 Proof Verification

[Rocq/Lean]

3.6 Open Research in Mathematics

[Accessibility]

[Findability]

[Interoperability]

[Reusability]

[Below is from Knowledge Graphs YouTube Lecture 1.7]

Linked Data Principles:

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (Resource Description Framework (RDF), SPARQL).
4. Include links to other URIs, so they can discover more things.

Tim Berners-Lee's 5-Star Criteria for Linked Open Data:

1. Available on the Web (whatever format) but with an **open license**, to be Open Data.
2. Available as **machine-readable structured data** (e.g., excel instead of image scan of a table).
3. As (2) plus **non-proprietary format** (e.g., CSV instead of Excel).
4. All of the above plus: use **open standards from W3C** (RDF and SPARQL) to identify things so that people can point at your stuff.
5. All the above plus: **link your data to other people's data** to provide context.

3.7 Ethical AI Research Practices

4 Authoring

[Format: What is it; What aspects they already cover; Summarise remaining actions]

4.1 Executable Papers & Open Education Resources

[Open Access & Open Education]

[Universal Design for Learning]

[Executable Papers]

4.2 Platforms

4.2.1 Overleaf

4.2.2 Obsidian

4.2.3 Quarto & Quarto Extensions

[Journal Article Templates]

5 Backend Development

[Approach - Algorithmic wherever possible]

5.1 Rendering Workflow & Knowledge Graphs

Data modelling process:

1. Understand the domain and define specific use cases (questions) for the application
2. Develop the initial graph data model:
    1. Model the nodes (entities).
        - Choose sematically orthogonal labels and avoid class hierarchies in labels or using more than 4 labels.
    2. Model the relationships between nodes.
3. Test the use cases against the initial data model.
4. Create the graph (instance model) with test data using Cypher.
5. Test the use cases, including performance against the graph.
6. Refactor (improve) the graph data model due to a change in the key use cases or for performance reasons.
    1. Design the new data model.
    2. Write Cypher code to transform the existing graph to implement the new data model.
    3. Retest all use cases, possibly with updated Cypher code.
7. Implement the refactoring on the graph and retest using Cypher.
    1. Identify which use cases are affected by the refactor.
    2. Rewrite any queries that can take advantage of the refactoring.
    3. Test all queries affected by the refactor to ensure they return the same results as before the refactoring.
    4. PROFILE queries to determine if refactoring is a benefit.

Use cases:

1. What are the **sources** `cited` for each **result**?
    1. Retrieve a result by its **embedding vector**.
    2. Trace the CITED relationships to the sources.
    3. Return the **citations** of the sources.
2. What is the **result** that `defines` a specific **object**?

    1. Retrieve the object by its **name**.
    2. Trace the DEFINE relationship to the result.
    3. Return the **description** and **href** of the result.
3. What **objects** are `required` for a **result**?
    1. Retrieve the result by its **embedding vector**.
    2. Trace the REQUIRE relationship to the objects.
    3. Return the **names** and **notation** of the objects.
4. What objects `compose` a specific object?
    1. Retrieve the specific object by its **name**.
    2. Trace the composed DEFINE and REQUIRE relationships to objects.
    3. Return the **names** of the objects.
5. What **properties** `apply` to a specific **object**?
    1. Retrieve the specific object by its **name**.
    2. Trace the composed DEFINE, REQUIRE and APPLY relationships to objects.
    3. Return the **names** of the objects.
6. What nodes `originated` from a specific **document**?
    1. Retrieve the document by its **path**.
    2. Trace the ORIGIN relationships to the nodes.
    3. Return the key values and labels of the nodes.

Nodes:

- SOURCE
    - [[referenceID]]
    - citation
- DOCUMENT
    - [[path]]
- OBJECT:
    - [[identifier]]
    - notation
- RESULT
    - [[embeddingVec]]
    - [abstractionVec]
    - desc
    - href

Intermediate Nodes:

- RESULT
    - create applied result with abstraction relationship to the original result unless the original result is sufficiently abstract

Relationships:

- CITE
- PROPERTY
- ABSTRACTION
- APPLIED

[SQ4R]

Quarto processing:

0. Source Documents
1. Pre-render
2. Render Filters
    - Pandoc Structure:
        - Metadata
        - Blocks
            * Blocks
            * Inlines
3. Post-render
4. Output Documents

Embedding RAG:

0. Source Documents
1. Data Ingestion
2. (Unstructured) Text Chunks
3. Embeddings (using an Embedding Model)
4. Vector Database
    - Semantic-aware

GraphRAG:

0. Source Documents
1. Chunks
2. Knowledge Graph Generation
    - Entity Extraction (using an LLM)
    - Relationship Extraction (using an LLM)
3. Community Detection (using an LLM)
4. Hierarchical Community Structures (using an LLM)
5. Community Levels - Root, Low, or High (using an LLM)
6. Generate Community Summaries (using an LLM)
7. Vector Database (using an Embedding Model)
    - Context-aware

Real-Time Knowledge Graphs (Graphiti for dynamic data management):

0. Source Documents
1. Chunks
2. Knowledge Graphs Construction

Stage	Quarto	Embedding RAG	GraphRAG	Temporal RAG
Input	Markdown Documents			

Stage	Quarto	Embedding RAG	GraphRAG	Temporal RAG
Output	Formatted Documents	Semantic-aware Vector Database	Context-aware Vector Database	Temporal-aware Vector Database

[Directory specific]

5.2 Data Processing

5.2.1 Data Cleaning

[Automatic Abbreviation Detection]

[Override/Example processing]

5.2.2 Mathematical Data

5.2.3 Mathematical Proofs

5.2.4 Terminology, Definitions & Conclusions

[Definition/Div/Paragraph processing, e.g., emph text = defined term]

Basic Definition Schema:

- name: emph text
- container_notation
- element_notation

Definition Relationships:

- REQUIRES -> Definition
- HAS_PROPERTY -> Definition
- SUBCATEGORY_OF -> Definition
- CONTAINS -> Definition

[Synonym Sets]

[Language-Independent URIs for Designator and Designatum, e.g., Wikidata Identifier]

Want to get information about "the apple" from Wikidata:

```
HTTP GET request
Accept Header: text/html
https://wikidata.org/entity/Q89
```

URI represents Designatum

```
HTTP/2 303 See Other
https://wikidata.org/wiki/Q89
```

URI represents Designator

```
HTTP GET request
Accept Header: text/html
https://wikidata.org/wiki/Q89
```

Gets HTML Document

Want to get machine understandable information about "the apple" from Wikidata:

```
HTTP GET request
Accept Header: text/turtle
https://wikidata.org/entity/Q89
```

URI represents Designatum

```
HTTP/2 303 See Other
https://wikidata.org/wiki/Special:EntityData/Q89.ttl
```

URI represents Designator

```
HTTP GET request
Accept Header: text/turtle
https://wikidata.org/wiki/Special:EntityData/Q89.ttl
```

Gets HTML Document RDF Document (Turtle is a way to encode RDF)

5.2.4.1 Anticipating International Translations

5.2.4.2 Addressing Biases

[Decolonising the Curriculum]

5.3 Data Management

5.3.1 Data Processing

5.3.1.1 Dependency Identification & Sorting

5.3.1.2 Internal Reference Management

5.3.1.3 External Reference Management

5.3.1.4 Cross Referencing & Embedding

5.3.1.5 Version Tracking

5.3.2 Error Handling

5.4 Document Rendering, Progressive JSON & CSS

5.4.0.1 LaTeX, MathJax & Presentation MathML

[Math Environment HTML Promises]

5.4.0.2 Alt-Text, MathSpeak & Context MathML

6 Frontend Development

6.1 PDF Output

6.2 Web Output

6.2.1 Progressive Web App

6.2.2 Accessibility Options and Features

6.2.3 Linked Knowledge

6.2.4 Semantic Search

[Jina.AI?]

6.2.5 Customisable Notation & Terminology

6.2.6 Interactive Proofs

6.2.7 Personalised & Varied Learning

6.2.7.1 Goal-Setting & Progress Monitoring

6.2.7.2 Assisted Note-Making

6.2.7.3 Playful Learning

7 Discussion

7.1 Copyright Considerations & the Importance of Individuality

7.2 Research Implications

[Research Progression through Blog writing]

7.3 Teaching Implications

7.4 Interdisciplinary Contexts

The University of Manchester (2024). *Research data management policy*. Available at: https://documents.manchester.ac.uk/DocuInfo.aspx?DocID=33802

The University of Manchester (2025a). *Publications policy*. Available at: https://documents.manchester.ac.uk/DocuInfo.aspx?DocID=76067

The University of Manchester (2025b). *Responsible metrics statement*. Available at: https://documents.manchester.ac.uk/DocuInfo.aspx?DocID=75142

The University of Manchester (2025c). *University of manchester position statement on open research*. Available at: https://documents.manchester.ac.uk/DocuInfo.aspx?DocID=55136