Interactions between aspects of Tetrapodal Mathematics on MathHub

Tom Wiesing
Computer Science, FAU Erlangen-Nürnberg

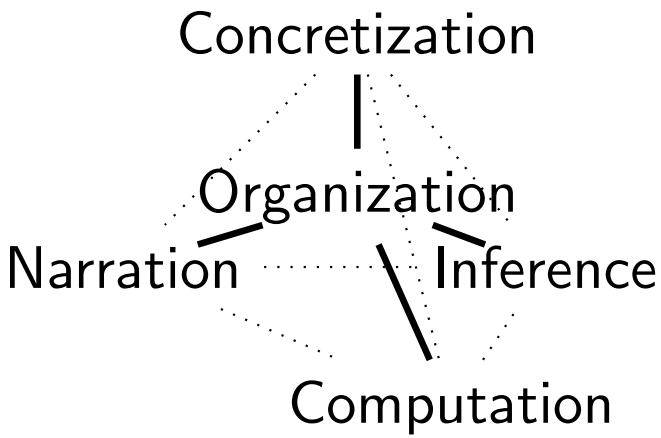
May 20, 2020
PhD Proposal
KWARC Group Seminar

Science, Technology, Engineering and Mathematics

- **STEM** = Science, Technology, Engineering and Mathematics
  - applicable to almost every topic in the modern world
  - applications become larger and larger
- Problem: Each experiment needs to be understood by the researcher performing it
  - due to the larger size this becomes very difficult
  - known as the **One-Brain-Barrier**
- Solution: Make use of computer support
  - use mathematics as a working example (it is **well-structured**)
  - expectation: results can be generalized to all of **STEM**

Aspects of Mathematics

- Mathematical practice usually consists of the following aspects
  - creating **narrative** text for readers guiding readers through specific topics (e.g. papers, textbooks)
  - making **inferences** based on existing knowledge (e.g. making new proofs)
  - **concretizing** mathematical knowledge (e.g. OEIS, LMFDB)
  - making (potentially large) **computations**
- these form a **tetrapodal** structure
  - are linked via a central organization (an **ontology**)



Research Questions

1. Can an information system provide infrastructure to jointly support all four tetrapodal aspects of mathematics?
2. Is it possible to support community-based workflows for authoring, maintaining, curating, and visualization of tetrapodal mathematics?
3. What kind of organization is needed for our model to work in practice?

Examples of Mathematical Software Systems (1)

- Inference
  - Proof Assistants (e.g. **HOL**, **Coq**)
  - Formal Programming Languages (e.g. λ**Prolog**)
- Computation
  - Symbolic Computation (e.g. **Mathematica**)
  - Numeric Computation (e.g. **Matlab**, **R**, **Python**, **Excel**)
  - Algebraic Computation (e.g. **Sage**)
  - Notebook model (e.g. **Jupyter**)
- Concretization
  - Record Data (e.g. **MySQL**, **Postgresql**, **MongoDB**)
  - Graph Data (e.g. **Neo4J**)
  - Array Data (e.g. **Rasdaman**)

Examples of Mathematical Software Systems (2)

- Narration
  - WYSIWYG Word Processors (e.g. **Word**, **LibreOffice**, **Google Docs**)
  - Document Preparation Systems (e.g. LaTeX, **HTML**)
- Organization
  - Encyclopedias (e.g. **Wikipedia**)
  - Graph Structured Collections (e.g. **Neo4J**, **GraphDB**, **MMT**)
  - Heterogeneous databases (e.g. **OEIS**)

Mathematical Services (1)

- computer support for mathematics = creating different mathematical services
- example services:
  - **navigate to definition** - clicking on a term jumps to the definition of it
  - **unit conversion** - converting quantity expressions from non-metric to metric
- services can be classified using two different mechanisms

Mathematical Services (2)

- some services operate on mathematical objects as a whole, some look inside of them
  - **navigate to definition** is a **shallow** service
  - **unit conversion** is a **deep** service
- some services depend only on the term itself, some on an entire corpus
  - **navigate to definition** is a **global** service
  - **unit conversion** is a **local** service
- **local** or **shallow** services are easy
  - they do not require much infrastructure
- **global** or **deep** services are hard
  - needs well-designed infrastructure to scale
  - I want to investigate the details of such an infrastructure

OMDOC

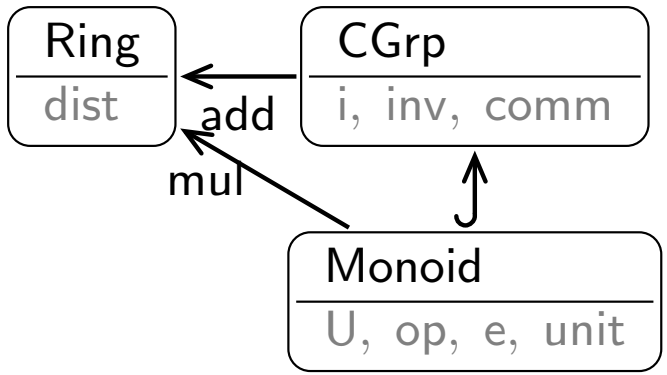
- Problem: Need a data model to be able to represent documents
- Solution: **OMDoc** = format for **O**pen **M**athematical **Doc**uments
  - format for encoding STEM documents and knowledge
  - developed mainly by *Michael Kohlhase*
  - can handle formal, informal and flexiformal content
    - formal = e.g. formalized proof, well-typed program, formal library
    - informal = e.g. paper, textbook, presentation
    - flexiformal = anything in-between, e.g. informal document with well-annotated formulae in between

MMT

- Problem: Need to be able to semantically operate on these documents
- Solution: Use **MMT** = Meta-Meta-Tool
  - framework for knowledge representation, implemented in Scala
  - developed mainly by *Florian Rabe*
  - avoids a specific representational paradigm and is language-independent
  - makes use of OMDOC, can thus handle formal, informal and flexi-formal documents

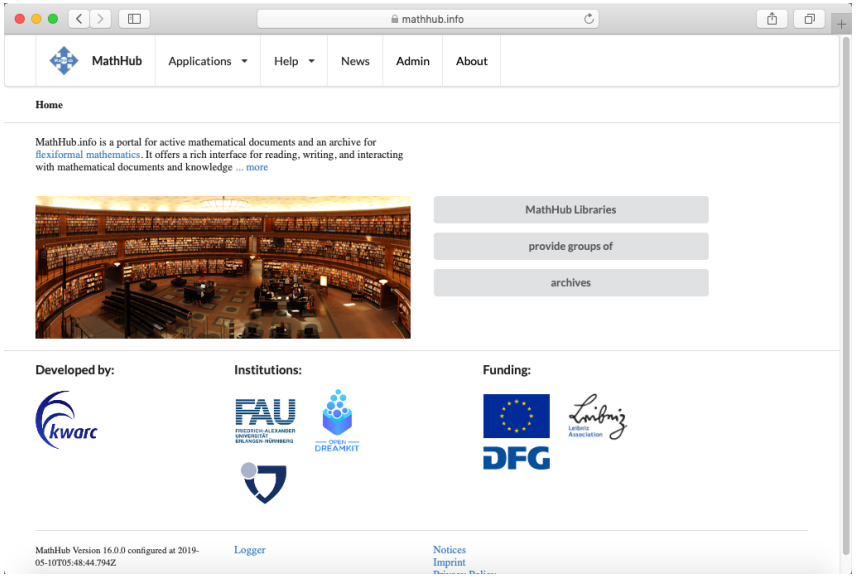
Knowledge in Theory Graphs

- Knowledge is modeled using theory graphs
  - theory = collection of declarations
  - declaration = represents a **symbol**, **statement** or **property**
  - theory graphs = theories and relations between them, e.g. truth-preserving mappings or extensions
- Theory Graphs scale well
  - very well-composable (each model only occurs once)
  - parts of them can be lazy-loaded
  - $\Rightarrow$ use them as a basis for a scalable ontology

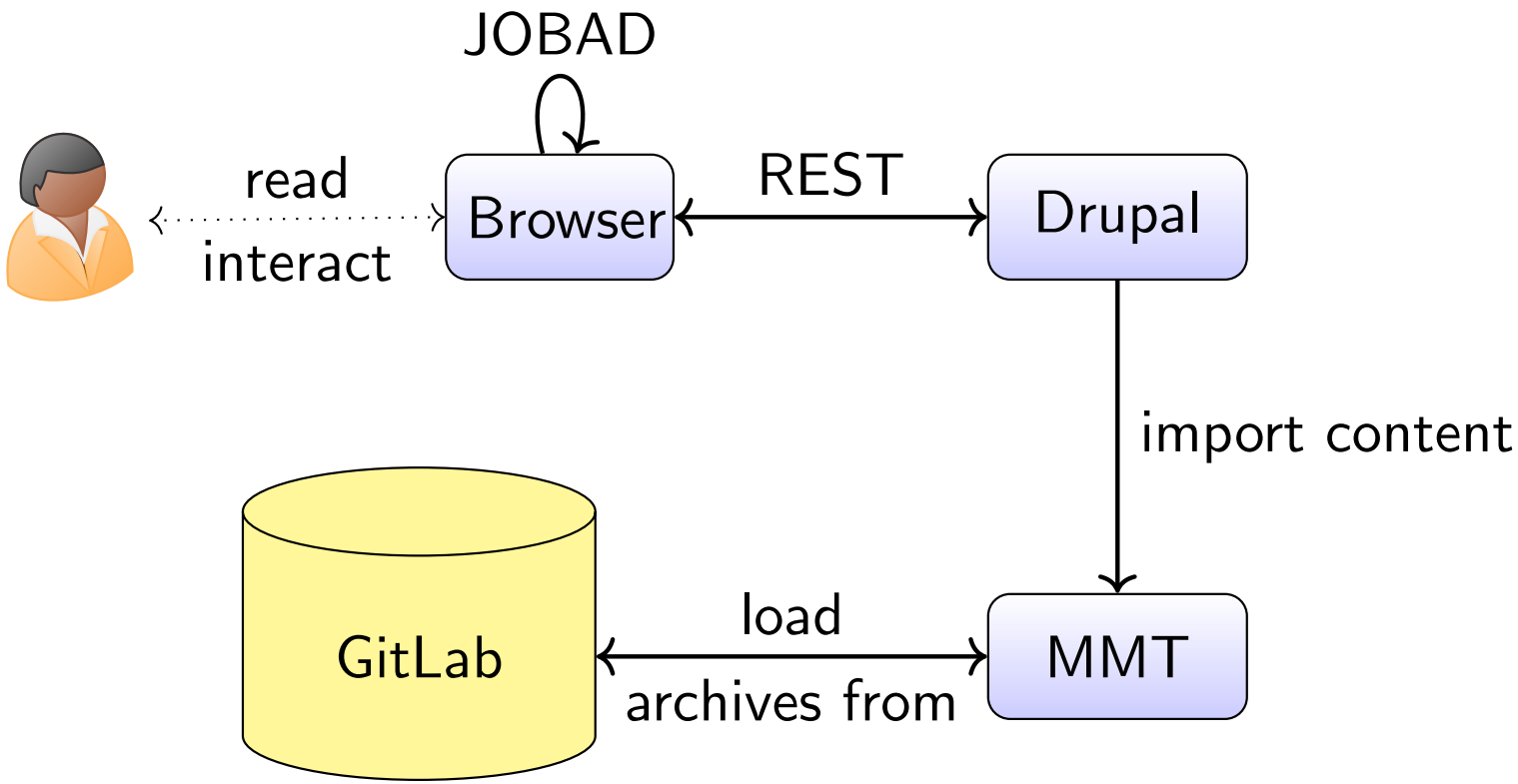


MathHub

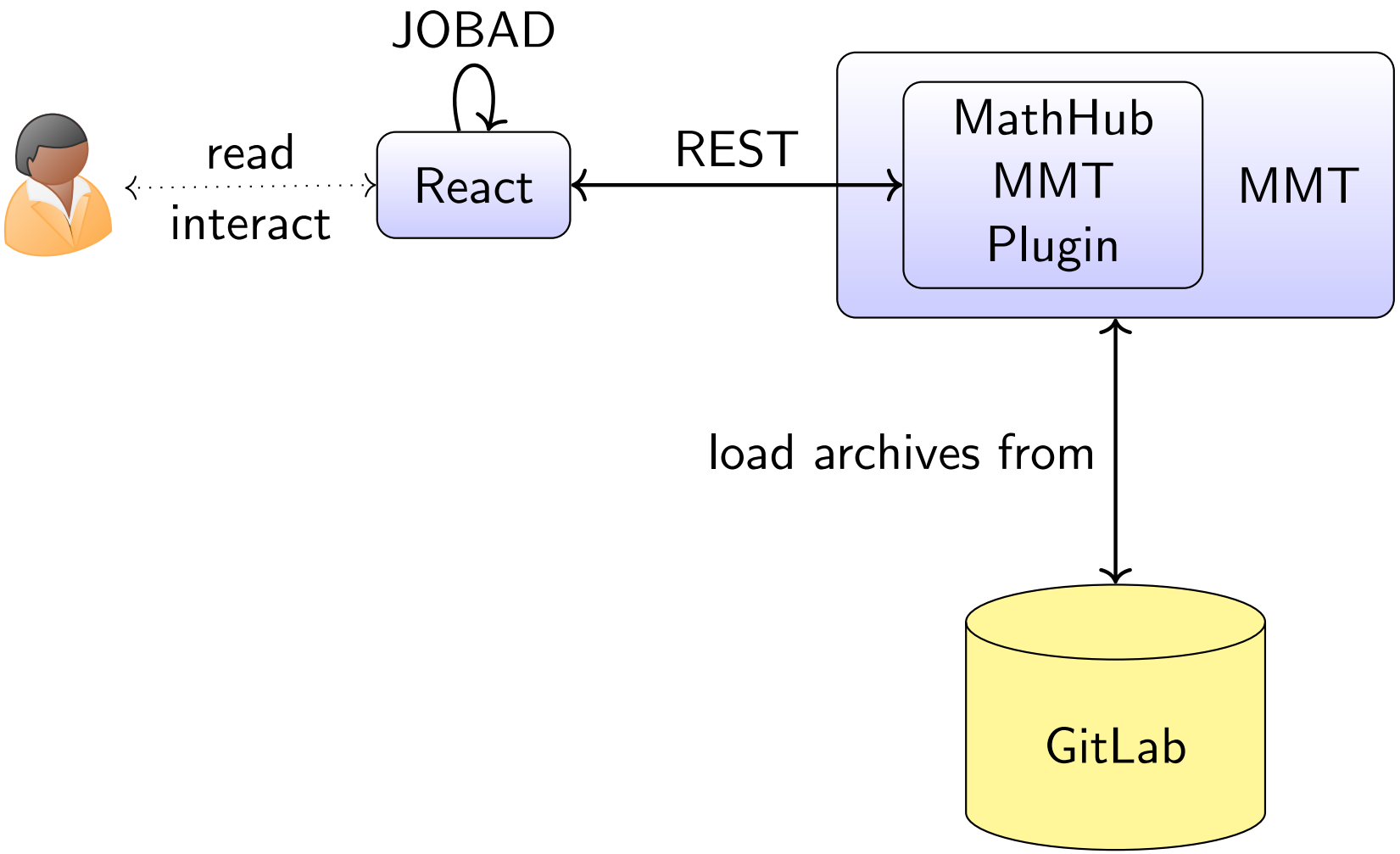
- **MathHub**
  `https://mathhub.info/`
  - portal for active mathematical documents and an archive for flexiformal mathematics
  - uses MMT as a backend and OMDoc as a representational format
  - this can use theory graphs



Original MathHub Architecture



Modified MathHub Architecture



WP1: Interaction-based Portals For Individual Aspects Of OMDoc-based Content

- Make "Portals" for the individual aspects
  - Focus on the services offered instead of the UI
- T1.1: **Narration**: Existing Interface
- T1.2: **Concretization**: MathDataHub (making mathematical data FAIR)
- T1.3: **Computation**: Active Documents using Jupyter
- T1.4: **Inference**: Not yet clear, perhaps interactive proof library or also Jupyter?

WP2: Organization Of Tetrapodal OMDoc-based Content

- Combine and link together the portal developed beforehand
- How tight does this integration have to be?
  - Are hyper-links enough? (work on this in T2.1)
  - Are deep embeddings needed? (work on this in T2.2)
- Compare and Evaluate in T2.3

WP3: Tetrapodal Search

- Mathematics contains a lot of knowledge
  - Searching is a key application to filter this knowledge for usefulness
- Need to support all aspects during Search
  - for instance search narration and formulae at the same time
- Investigate existing search engines (T3.1)
  - **SQL**, **ElasticSearch**, **MathWebSearch**, **Triple Stores**
- First: Design a query language to express tetrapodal queries (T3.2)
  - **QMT** and **SPARQL** as a base?
- Then: Build a system to implement the queries (T3.3)

WP4: MathHub Tooling And Deployment

- having a system is not enough
  - need well-designed architecture
  - so that we can be sure it actually works
  - can use it to make general statements about any tetrapodal information system
- develop a component-based architecture (T4.1)
  - to easily to add / remove components where needed
- Produce tooling and documentation for starting new MathHub instances (T4.2)
  - to enable actually using the system reliably

WP5: Import And Export Of Non-OMDoc Content In MathHub

- not all mathematical content will be OMDoc
  - authors will want to keep working in their favorite system
  - we need to be able to import / export content into MathHub
- this is already done
  - using an explicit conversion approach
  - using a lazy-loading "Virtual Theories" approach
- need to document these processes (T5.1)
- make MathHub content available directly in external UIs (T5.2)
- need to make import/export scalable (T5.3)

Conclusion

- Summary
  - **STEM** is moving to bigger and bigger developments
  - computer support is needed to overcome the **one-brain-barrier**
  - I want to use **mathematics** as an example
  - Mathematics has a **tetrapodal** structure
  - Use **MathHub** to figure out the strucure of tetrapodal systems
- Questions, Comments, Concerns?
- Thank You For Listening!