Formal Representation of Scientific Knowledge

Florian Rabe (Advisory Board Member)

Computer Science, University Erlangen-Nuremberg, Germany

Background

About Me

Theoretical Computer Science

- foundations — logic, programming languages, formal systems
- knowledge representation — specification, formalization, ontologies, programming
- scalable applications — module systems, libraries, system integration, data sharing

Methods

- survey, abstract, and transfer — unify, connect different research areas
- modularity and reuse — maximize sharing across languages, tools
- system development: language design – implementation – library building – applications

Formal Knowledge in Computer Science

- Early 20th century: vision of mechanizing mathematics

  birth of computer science
- Development of formal logic
  - competition between set theory, λ-calculus
  - today many different logics
- Development of programming languages
  - competition between imperative, functional languages
  - today many different languages
- Sophisticated automation support for
  - formal modeling
  - computing
  - proving
  - querying

all in different highly-optimized systems

Selected Flagship Projects

Software verification

- 2004–2010: Klein et al., L4 micro-kernel operating system
  390,000 lines of human-written formal logic
- since 2005: Leroy et al., C compiler (CompCert)
  almost complete, high performance

Mathematics

- 2006–2012: Gonthier et al., Feit-Thompson theorem
  170,000 lines of human-written formal logic
- 2003–2014: Hales et. al., Kepler conjecture (Flyspeck)
  $> 5,000$ processor hours needed to check proof

Selected Flagship Projects (2)

Artificial intelligence

- since 1984: Lenat et al., common knowledge (CyC)
  2 million facts in public version
- since 2000: Pease et. al., foundation ontology (SUMO)
  $25,000$ concepts

Other fields

- since 2001: OBO Foundry, collection of biomedical ontologies
  > 1000 ontologies, $> 10M$ classes
- since 2021: Wikidata, open data knowledge graph $100M$ data items

these are ontologies

Knowledge Sharing

Major Push for Sharing of Research Data

Major push towards open research data

- 2006: OECD Council recommendations
- 2016: FAIR principles for Findability, Accessibility, Interoperability, Reusability
- 2018: European Open Science Cloud, EU infrastructure
- 2018 (Germany): NFDI, 30 consortia, up $5M$ EUR each

similar initiatives in most countries

But existing services essentially shallow

- represent data set as a whole
- little support finding/. . . /reusing individual data items

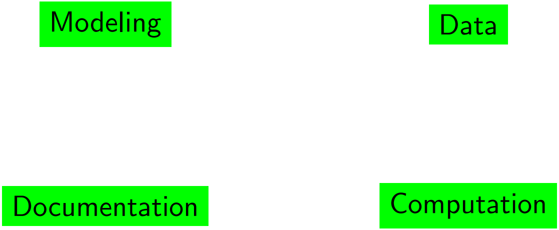
Shallow vs. Deep Services

Service	Shallow	Deep
Identification	DOI for a dataset	DOIs for each entry
Provenance	who created the dataset?	how was each entry computed?
Validation	is this a list of integers?	does it represent a 3×3-matrix?
Finding	find a dataset	find entries with certain properties
Access	download a dataset	download a specific record
Interoperability	only manually	automatable
Reuse	only manually	automatable

- Shallow services are generic — easy to build
- Deep services require formal ontology of background knowledge — much harder

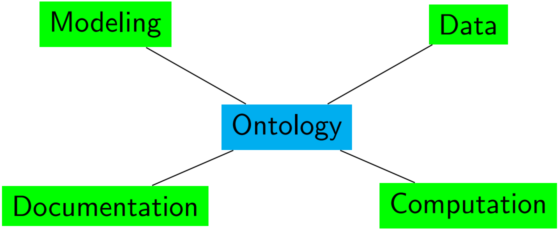
4 Aspects of Knowledge (Tetrapod model)

- Documentation: informal but rigorous, math-based
  needed for human consumption
- Modeling: formal mathematical/physical properties
  needed for machine understanding
- Computation: data structures and algorithms
  needed for practical applications
- Data: large sets of objects needed for exploration, analysis



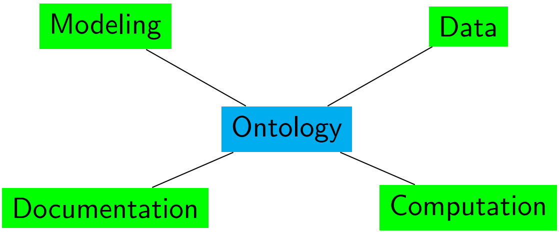
4 Aspects of Knowledge (Tetrapod model)

- Documentation: informal but rigorous, math-based
  needed for human consumption
- Modeling: formal mathematical/physical properties
  needed for machine understanding
- Computation: data structures and algorithms
  needed for practical applications
- Data: large sets of objects needed for exploration, analysis
- Central ontology key to knowledge sharing



4 Aspects of Knowledge (Tetrapod model)

- Documentation: informal but rigorous, math-based
  needed for human consumption
- Modeling: formal mathematical/physical properties
  needed for machine understanding
- Computation: data structures and algorithms
  needed for practical applications
- Data: large sets of objects needed for exploration, analysis



expressivity of ontology is bottleneck for knowledge sharing

Shallow vs. Deep Ontologies

Shallow Ontology Language

- high-level abstraction → knowledge graph structure
  scales well to large sets
- modeling focuses on
  - concepts — City, temperature
  - individuals — Totnes:City
  - relations — Totnes in England
  - properties — Totnes temperature $15^\circ C$

Deep Ontology Language

- fine-grained modeling — prerequisite for sharing complex knowledge
- supports mathematical/physical
  - objects and operations — temperature series
  - formulas and equations — differential equations for temperature

My MMT System

A universal framework for formal knowledge

Vision: cover

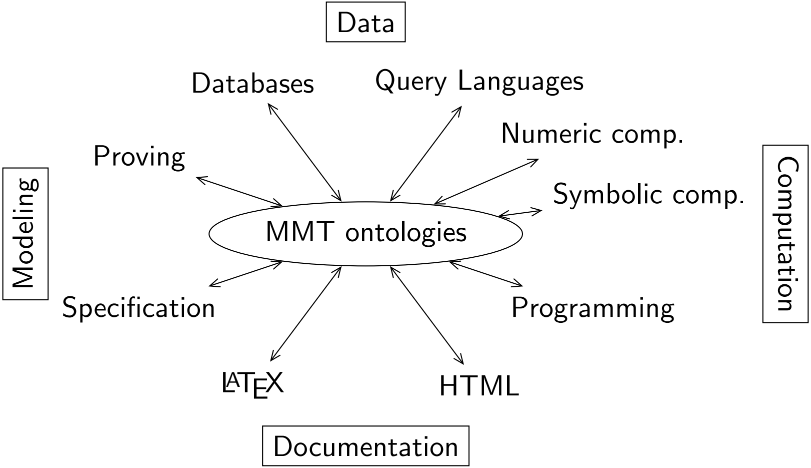
- all aspects: modeling, logic, computation, documentation, data, ...
- all domains: CS, math, logic, STEM, ...
- all tools: search, library managers, IDEs, wikis, ...

- evolving, partial solution
- developed since 2006 (with Michael Kohlhase)
- $> 100k$ loc, $> 1k$ pages of publications

`http://uniformal.github.io/`

MMT as Ontology-Based Mediator

Deep ontologies and tool interfaces formalized in MMT enables sharing knowledge across tools



Small Scale Example

```
theory Logic {
   prop   : type
   assert : prop → type
}

theory OntologyLanguage {
  include Logic
  conc       : type
  individual : type
  isA        : ind → conc → prop
}

theory MyOntology : OntologyLanguage {
  city    : conc
  Totnes  : individual
  assert (Totnes isA city)
}
```

Large Scale Example: The LATIN Atlas

- Highly modular network of formal systems and translations
  - formal logics
  - mathematical foundations
  - type systems
  - programming languages
- Written in MMT since 2008
- Originally with T. Mossakowski, M. Kohlhase, 20 contributors by now
- $\sim$ 1000 MMT modules

Large Scale Example: The LATIN Atlas (2)

It's big — that's me pointing at logic 101

Very Large Scale Example: The MathHub Portal

GitHub-like but for MMT projects https://gl.mathhub.info

- 251 Repositories
- 187 Users
- 28.5 GB in 2021, probably doubled by now

Example: proof assistant libraries in MathMub

System	# Modules	# Declarations
PVS	$1k$	$20k$
Isabelle	$10k$	$1M$
HOL Light	200	$20k$
Coq	$2k$	$150K$
Mizar	$1k$	$70k$

Case Study: Concrete Datasets

Problem

Mathematical datasets are getting huge

- dozens of datasets of $> 10^6$ objects
- generated programmatically

  akin to measurements in experimental sciences
- ad hoc maintenance, no systematic FAIRness

Example:

- file "ec.csv" with $3M$ lines
- column headers: "label", "isogMat"
- some line: 11a1,"1,5,25,5,1,5,25,5,1"
- background knowledge needed to interpret: isogeny is a property of elliptic curves and

$$\mathrm{isogeny}(X_0(11)) = \begin{pmatrix} 1 & 5 & 25 \\ 5 & 1 & 5 \\ 25 & 5 & 1 \end{pmatrix}$$

Solution: The MathDataHub System

MathDataHub = SQL+MMT

- SQL database for mathematical datasets
- semantic schemata defined in MMT

collaboration with K. Bercic

Semantic database schema

- ontology of background knowledge in MMT   definition of isogeny
- database table formalized as MMT record type
  isogMat : $\mathbb{Z}^{3\times 3}$ and isogeny assertions
- metadata annotation for database encodings
  3×3 matrix encoded as 9-element list
- MMT generates SQL database schema and encode/decode functions

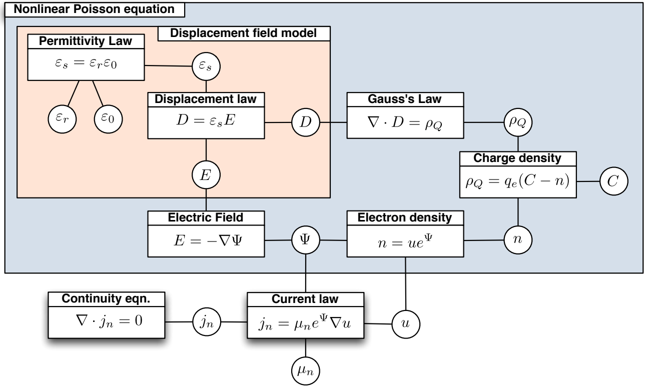
Other systems can now Find/.../Reuse each record via its mathematical representation.

Case study: Mathematical Modeling and Simulation of Physical Systems

Problem

Hard to Solve Differential Equations

- systems of differential equations without closed solutions
- numerical solutions found by discretization, fixed point iteration
- feedback loops between iterations



Experience and Solution

Collaboration with T. Koprucki, K. Tabelow (WIAS Berlin)

- 2 days to understand each other
- 1 week to design ontology
- 3 student months to adapt MMT into useful system

visualize and design iteration strategies

Observation:

- Domain experts tend not to separate ontology-relevant from other knowledge
- Ontology modeling in MMT helps design interfaces for tool integration

knowledge	domain expert concerns	needed in deep ontology
geometry	exact shape, discretization	set of parameters
physical quantities	measurement, initial conditions	existence
equations	derivation, using	formal statement
iteration	pros/cons of strategies	concept of a strategy

Conclusion

Take-Home Messages

- Knowledge often spread over many optimized systems
  applies to any scientific domain
- Sharing demanded by researchers and political bodies
  but practical details separate research problem
- Formal ontologies can mediate knowledge sharing
  depth of ontology limits complexity of shared knowledge
- MMT is a system for
  - developing ontology languages and deep ontologies
    no commitment to a particular domain or logic
  - building knowledge management applications
    general or domain-specific
- Collaboration of knowledge management experts and domain experts fosters knowledge sharing