# Strengths and limitations of different types of declarative modelling languages

C. Maria Keet

Department of Computer Science, University of Cape Town, South Africa
https://sit.uct.ac.za
mkeet@cs.uct.ac.za

*DAMA Southern Africa, online*
*5 August 2024*

# Outline

# Outline

# Models galore

- Physical model; e.g., Lego brick house
- Mathematical model; e.g., climate change model, bacterial growth in cheese-making
- Machine learning & cs.; e.g., data-driven spellchecker, LLM
- Conceptual models; e.g., concept maps, declarative models of mathematical models, knowledge graphs, thesauri, UML diagrams, ontologies

# Models galore

- Physical model; e.g., Lego brick house
- Mathematical model; e.g., climate change model, bacterial growth in cheese-making
- Machine learning & cs.; e.g., data-driven spellchecker, LLM
- Conceptual models; e.g., concept maps, declarative models of mathematical models, knowledge graphs, thesauri, UML diagrams, ontologies

⇒ Gentle, mostly non-technical introduction to conceptual models in computing

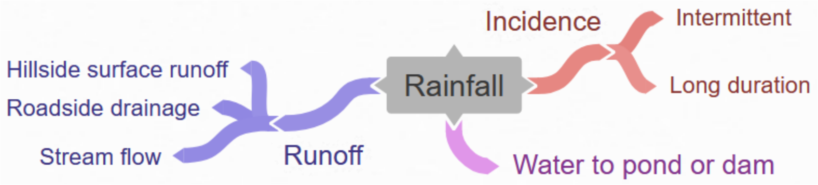
⇒ Running example about with rain (dance in the book)

# Beginnings: brainstorming about a topic with mind maps

- Mind maps are easy to draw
- and look pretty

# Beginnings: brainstorming about a topic with mind maps

- Mind maps are easy to draw
- and look pretty



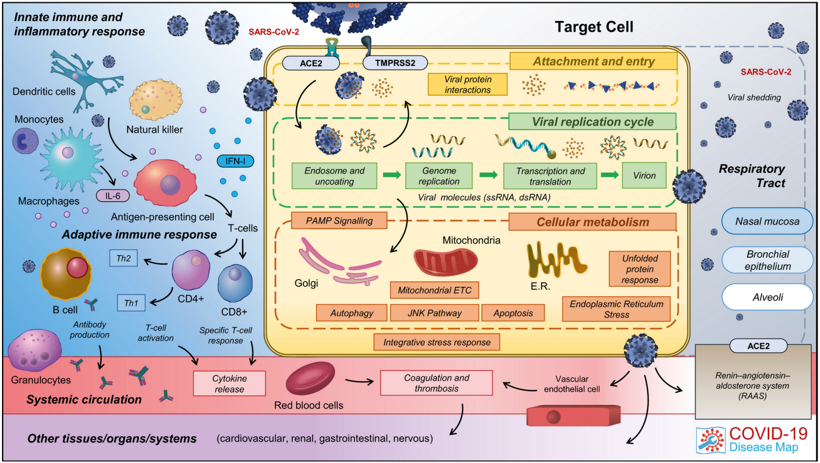
# Mind maps raise more questions than they answer

- What's a *good* mind map?
- What's the *right size* of a mind map?
- What's the *meaning* of a connecting line?
- How *useful* is it to make one (and for what purpose)?
- How to *develop* a good one?

# Partial solution: biological models

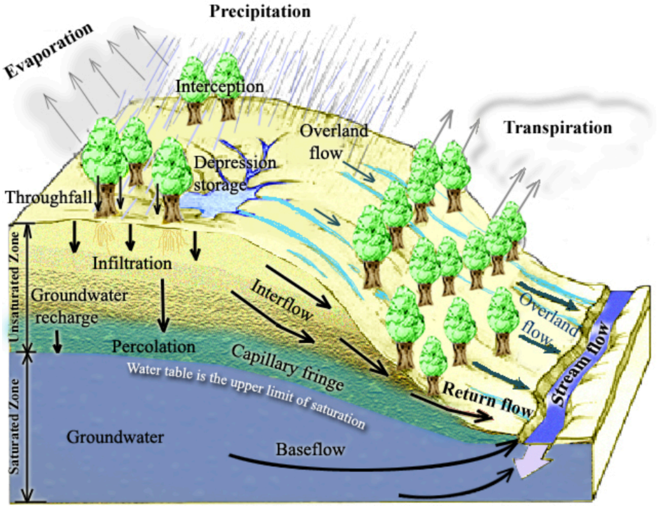
- De facto more or less (or un-)regulated domain specific language
- The lines and arrows have meaning
- There are icons to use, with meaning
- There are written or unwritten rules on how to put them together and how to read them

# COVID-19



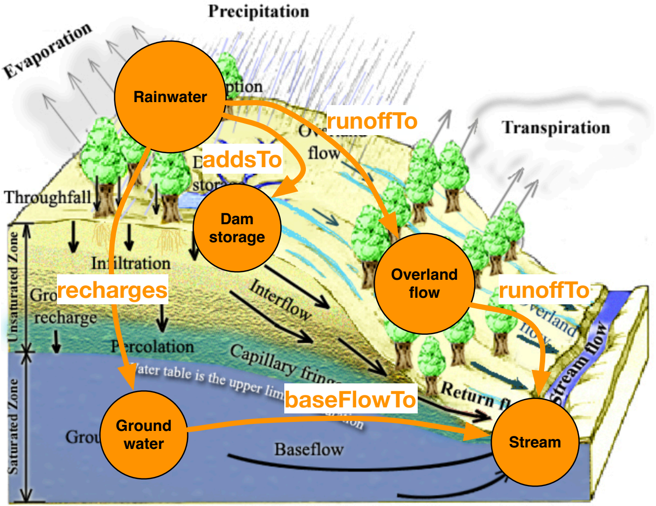

[Ostaszewski et al.(2021)]

# Physical Processes involved in Runoff Generation



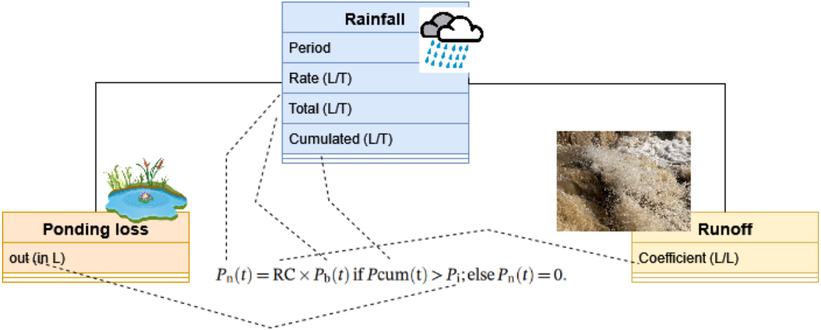

[Tarboton(2003)]

# Towards a knowledge graph




adapted from [Tarboton(2003)]

# Annotating hydrological models example

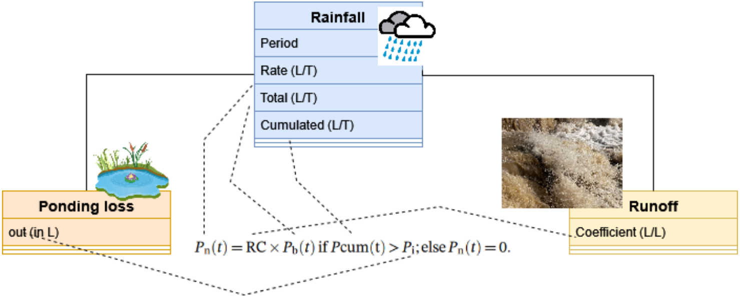



formula from [Chahinian et al.(2023)]

# Limitations of the domain models

- There are very many notations to learn
- Objects and arrows, but no way to specify how many outgoing arcs there may be
- Limited computational use among models (e.g.: do they overlap or contradict?)
- Proliferation of incompatible modelling tools that are cumbersome to maintain

# Rainfall questions – loose ends in the model




- What are the names of the relations? the constraints? The relevant attributes? And their data types?

# Outline

# Solutions to limitations of the domain models

- Devise one notation for all subject domains
- More expressive (more features) than only objects and arrows
- Computational support
- (Try to) Standardise to make tooling development 'economically' viable

# Solutions to limitations of the domain models

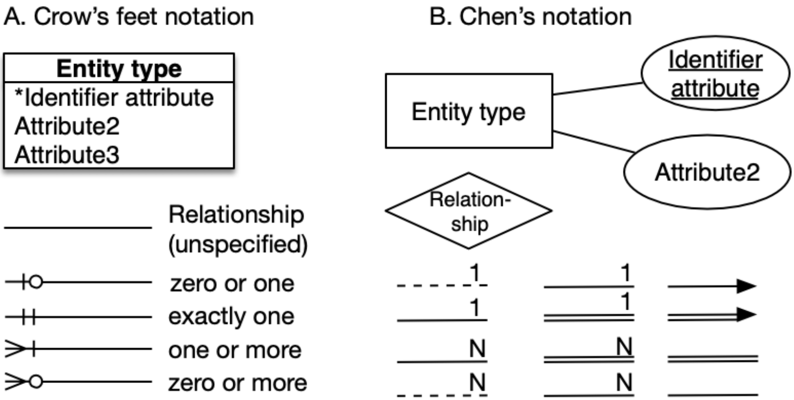
- Devise one notation for all subject domains
- More expressive (more features) than only objects and arrows
- Computational support
- (Try to) Standardise to make tooling development ‘economically’ viable

$\Rightarrow$ Conceptual data models

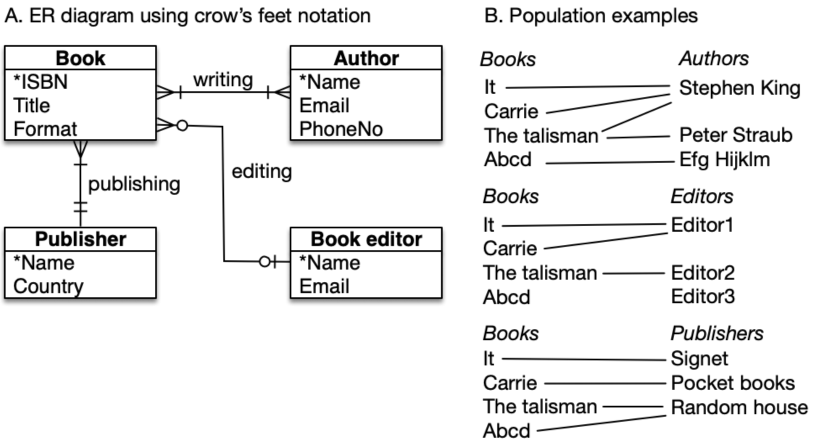
# Conceptual data models

- A class of models that capture the information about the data that are to be stored in the prospective software system (and possibly manipulated)
- There are several conceptual data modelling language families and notations

# Conceptual data models: Examples of language elements



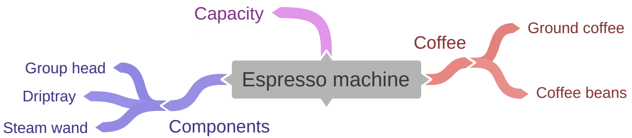
# Conceptual data models: An example



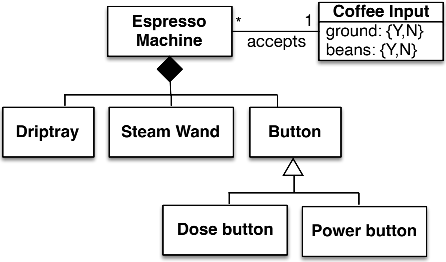

Convert automatically between EER, UML, ORM, choose preferred notation: Braun G, Fillottrani PR, Keet CM. A Framework for Interoperability Between Models with Hybrid Tools, J of Intelli. Inf. Sys. 2023.

# The espresso example of the talk's announcement

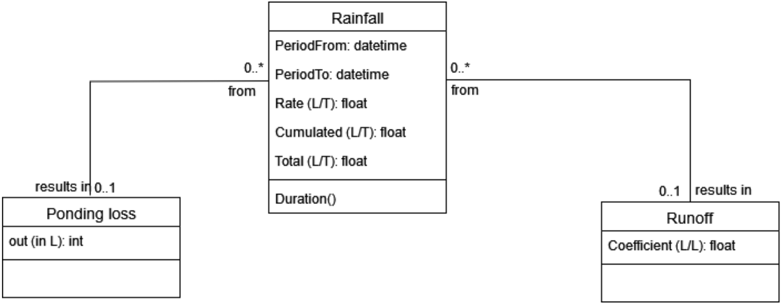
**Brainstorming about espresso machines...**




**... compared to software design for espresso machines**

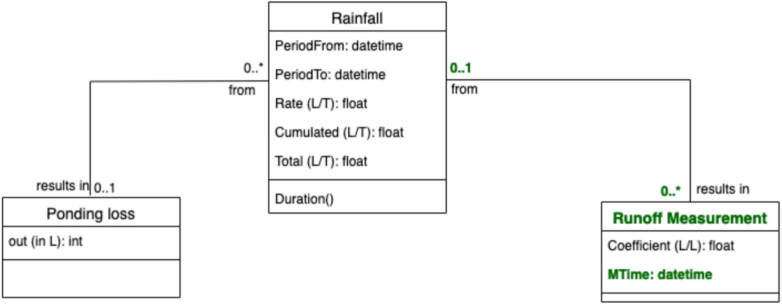


# Rainfall, again — UML style




- Convert UML class diagram (semi-)automatically into program code
- Easier to communicate with other domain experts and programmers what's in the code
- Easier to reuse with other math formula that use same entities

# Rainfall, again — UML style




- Convert UML class diagram (semi-)automatically into program code
- Easier to communicate with other domain experts and programmers what's in the code
- Easier to reuse with other math formula that use same entities

# Limitations of conceptual data model models in theory or practice

- For one specific application only – need to re-do it for each application, integration issues
- Solutions to recurring modelling issues re-invented time and again (and same mistakes made)
- Mostly informal diagrams that suffer from ambiguity (intentionally or not)
- Limited authoring guidelines[1]
- Some quality control mechanisms

[1] Mainly the CSDP for ORM [Halpin(2001)]; an example for EER: `http://www.meteck.org/modellingbook/DanceSchoolExample.html`; TDD proposal for UML [Tort and Olivé(2010)]

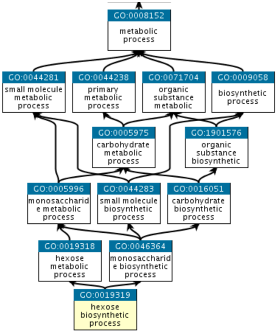
# Outline

# Solving limitations of conceptual data model models with ontologies

- ‘Model’ for a subject domain, of use across multiple applications for use, reuse, integration
- Provides solutions to recurring modelling issues, saves re-inventing the wheel
- Logic-based, as precise as permitted within the language
- Multiple quality control mechanisms (theories, methods, techniques, tools)

# An ontology...

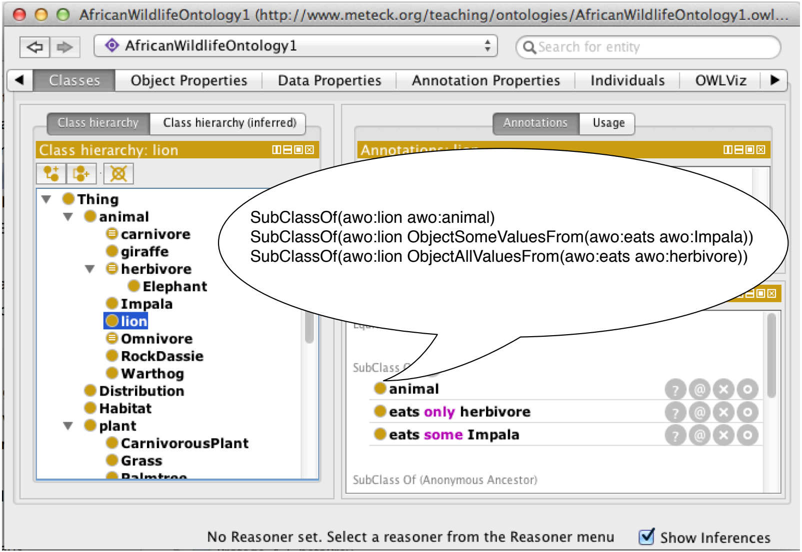
(informally) **an ontology** is an engineering artefact in *machine-processable format*, which contains the *entity types, their relationships, and constraints* that hold over them of a particular *subject domain*.



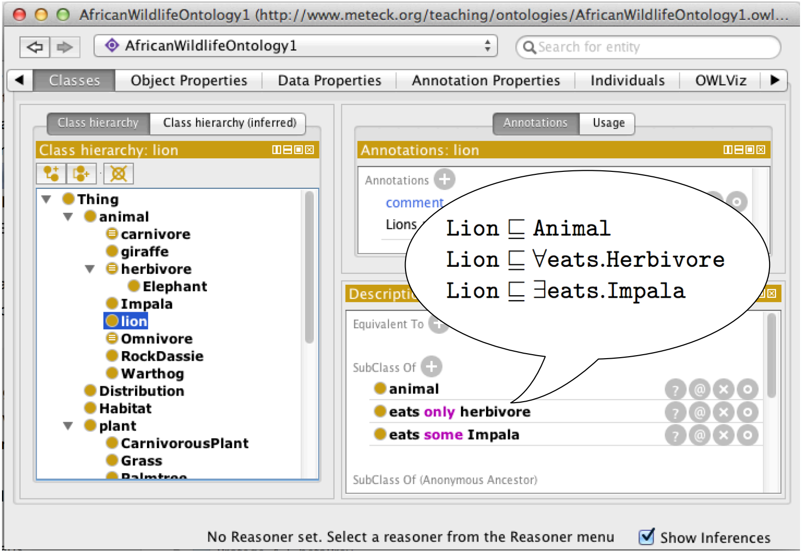

Fragment of the Gene Ontology, graphically

http://geneontology.org/docs/ontology-documentation/

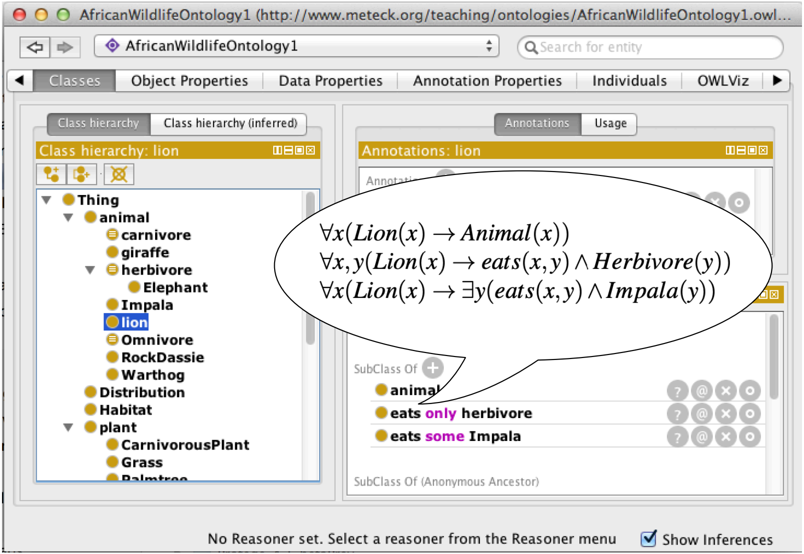
# Another rendering of an ontology & behind the GUI



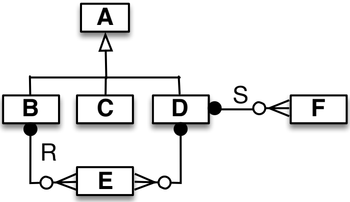
# … and underlying that serialisation



# … and underlying that serialisation



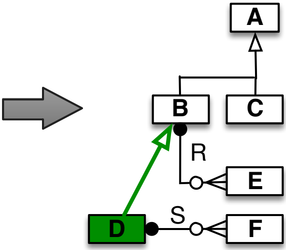
# A note on automated reasoning – Illustration




$B \sqsubseteq A$

$C \sqsubseteq A$ $\quad B \sqsubseteq \exists R.E$

$D \sqsubseteq A$ $\quad D \sqsubseteq \exists R.E \sqcap \exists S.F$

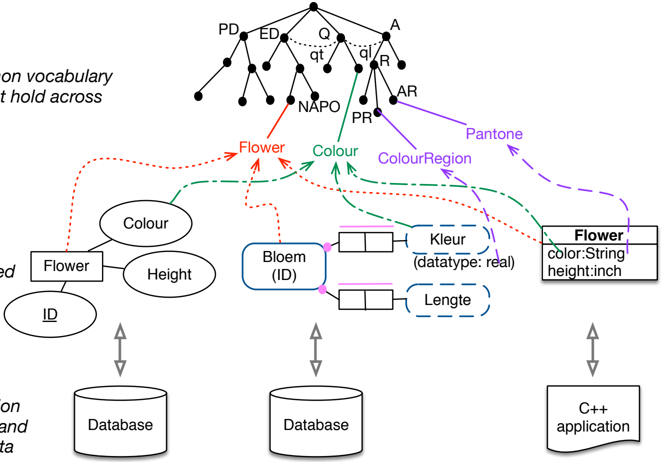


# Original idea of an ontology's use

**Ontology**
*provides the common vocabulary and constraints that hold across the applications*

**Conceptual model**
*shows what is stored in that particular application*

**Implementation**
*the actual information system that stores and manipulates the data*



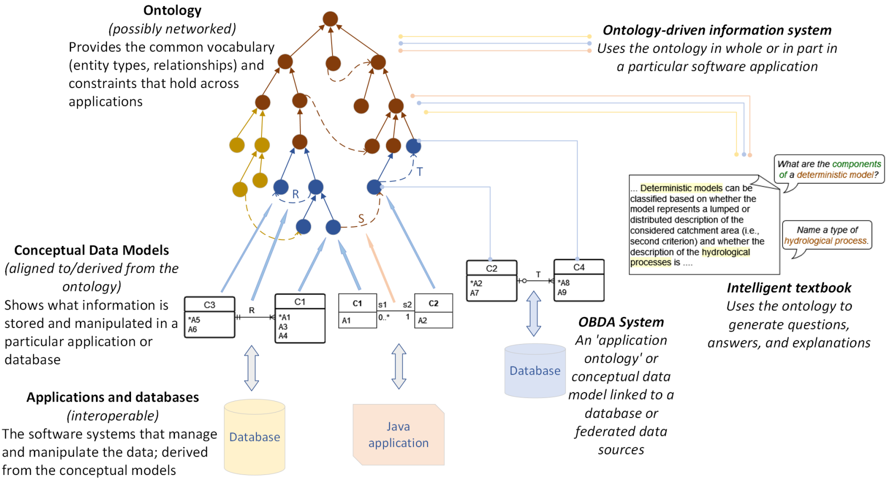
# Why ontologies?

- For their own sake, possible future use
- Representing a scientific theory precisely
- Facilitating communication among humans, between software applications or modules in a complex system
- Used for and in many different ontology-driven information systems: a.o., data integration, recommender systems, NLP, textbook enhancements, Q&A systems)

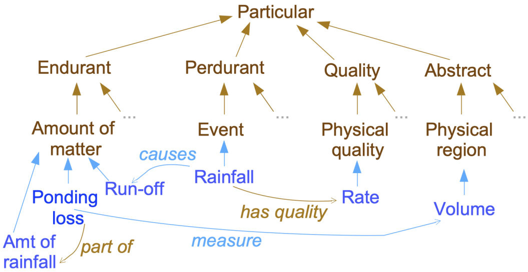
# Why ontologies?

- For their own sake, possible future use
- Representing a scientific theory precisely
- Facilitating communication among humans, between software applications or modules in a complex system
- Used for and in many different ontology-driven information systems: a.o., data integration, recommender systems, NLP, textbook enhancements, Q&A systems)
- Examples: finding 'new' knowledge (bio-chemistry) [Wolstencroft et al.(2007)] save research time (ecology) [Madin et al.(2008)] semantic comparisons of text (healthcare) [Reese et al.(2023)], energy-optimised building system control [Pruvost and Enge-Rosenblatt(2022)], digital humanities [Calvanese et al.(2016)] etc.

# Orchestration of ontologies and applications



# Rainfall: sample sketch and some axioms for an ontology




*Rainfall* $\sqsubseteq$ *Event* ('rainfall is an event'),
*Rainfall* $\sqsubseteq \forall$*hasQuality*.*Rate* ('each rainfall event has as quality only a Rate'),
*Rainfall* $\sqsubseteq \exists$*causes*.*Runoff* ('each rainfall event causes some amount of runoff' *),
*Rainfall* $\sqsubseteq \forall$*causes*.(*PondingLoss* $\sqcup$ *Runoff*) ('each rainfall event causes only an amount of ponding loss or runoff'),
*PondingLoss* $\sqsubseteq = 1$ *measure*.*Volume* ('each ponding loss has exactly one measure of volume' (of the loss)),

...

# Limitations of ontologies?

# Limitations of ontologies?

- Yes, there are...
- ... but not part of this talk (time constraints)

# Limitations of ontologies?

- Yes, there are...
- ... but not part of this talk (time constraints)
- A proposed solution is Ontology, a branch in philosophy
  - Investigates the nature of things, on paper mostly
  - e.g., general things: what is parthood, causality, role, function of an object, stuff?
  - Then use that to help figuring out domain entities; e.g., whether a software application is a whole, what a pandemic is, etc.

# Outline

# Feature-based comparison

**Table 7.1** Comparison of types of models along a set of properties

| | Feature | | | | | |
|---|---|---|---|---|---|---|
| Model type | Main aim or function | Where used (mainly) | Development methodologies | Software assistance | Language freedom | Precision |
| Mind Maps | Basic structuring of a topic | Education, business | A little | Yes, many drawing tools | Limited | Low |
| Biology models | Visualise biology knowledge (structures and processes) | Biology research, textbooks | No | Drawing tools, some runtime usage (simulations) | Ranges from self-imposed to complete freedom | Low/ medium |
| Conceptual data models | Capture characteristics of data to be stored and processed in an program | Analysis and design stage of database and program development | Yes | Drawing tools, limited runtime usage | Ranges from standardised languages to partial freedom to design one | Medium |
| Ontologies | Represent knowledge of a subject domain precisely and in a computer processable way | Computing and IT (Data integration, Enterprise systems, Web search, etc.) | Yes, many | Editors (diagram, text), runtime usage | Ranges from standardised languages to partial freedom to design one | High (but medium/ low is possible) |
| Ontology | Characterise one small aspect of interest precisely and in much detail | Research | No | No | Yes, can define as one goes along | High/ Very high |

# Task-based comparison

- Which model about rainfall was the most useful?
- For what?

# Task-based comparison

- Which model about rainfall was the most useful?
- For what?
- Comparison in the book, on learning a textbook page about labour relations:
  - ⇒ Conceptual data model as 'winner', unless critical inquiry (higher up in Bloom's taxonomy) is the aim

# Model quality, for all types

- Syntax checking is technically straightforward to implement
- For each type: what is a good model can be answered only partially, typically only one aspect at a time (scientific experiments with humans are limited)
- Most theories, methods, and techniques are available for ontologies only
- Certain things we don't know yet how to check for

# Bias in modelling?

- Bias is a problem in data-driven AI technologies; does declarative modelling have the same problems?
- No, but it's also not necessarily free of it
- It's currently not part of the modelling guidelines, methods, and methodologies

# Bias in modelling?

- Bias is a problem in data-driven AI technologies; does declarative modelling have the same problems?
- No, but it's also not necessarily free of it
- It's currently not part of the modelling guidelines, methods, and methodologies
- Mostly conscious modelling decisions during the modelling processes
- Albeit (likely?) not with malicious intent – acts of omissions cf. acts of commission

⇒ Where did it happen? Where could it slip in?

# Good candidates for incorporating bias

- Property manipulations
  - Which properties record, which one to exclude (AC/heating calculations to include computers in a room, insulation, ceiling height (EU), or not (USA))
  - Setting the permissible value range (e.g., hypertension, alcohol use disorder)

# Good candidates for incorporating bias

- Property manipulations
  - Which properties record, which one to exclude (AC/heating calculations to include computers in a room, insulation, ceiling height (EU), or not (USA))
  - Setting the permissible value range (e.g., hypertension, alcohol use disorder)
- Aggregation or granularity
  - Terrorism databases (bombing targets: just Government Building or subclassing with, say, Hospital, State's Generic Medicine Manufacturing Plant, Water Purification Plant, Military base, and Homeland security torture bunker)

# Good candidates for incorporating bias

- Property manipulations
  - Which properties record, which one to exclude (AC/heating calculations to include computers in a room, insulation, ceiling height (EU), or not (USA))
  - Setting the permissible value range (e.g., hypertension, alcohol use disorder)
- Aggregation or granularity
  - Terrorism databases (bombing targets: just Government Building or subclassing with, say, Hospital, State's Generic Medicine Manufacturing Plant, Water Purification Plant, Military base, and Homeland security torture bunker)
- Familiarity bias, own culture's narrow assumptions (e.g., composition of a person's name, family vs household)

# Good candidates for incorporating bias

- Property manipulations
  - Which properties record, which one to exclude (AC/heating calculations to include computers in a room, insulation, ceiling height (EU), or not (USA))
  - Setting the permissible value range (e.g., hypertension, alcohol use disorder)
- Aggregation or granularity
  - Terrorism databases (bombing targets: just Government Building or subclassing with, say, Hospital, State's Generic Medicine Manufacturing Plant, Water Purification Plant, Military base, and Homeland security torture bunker)
- Familiarity bias, own culture's narrow assumptions (e.g., composition of a person's name, family vs household)
- Optimism/pessimism bias, or misunderstanding subsumption ('COVID-19 drug in clinical trial' is a 'COVID-19 drug') and others

# Outline

# Recap and final remarks

- Different types of declarative models: mind maps, diagrams, conceptual data models, ontologies
- Modelling languages' respective expressiveness strengths and weaknesses
- Which one suits best depends on the task at hand

# Recap and final remarks

- Different types of declarative models: mind maps, diagrams, conceptual data models, ontologies
- Modelling languages' respective expressiveness strengths and weaknesses
- Which one suits best depends on the task at hand
- Professionalism: knowing which type to use when
- Resurgence in interest in modelling thanks to limitations of only data-driven AI
- Plenty of opportunities for tools, methods, methodologies, quality control, evaluation, bottom-up development from diagrams that depict models

# Acknowledgments


- Collaborators and students over the years (see book for the list)
- Funding sources over the years, including, mainly NRF, DST&MINCyT, UCT, HPI
- Rainfall example extended from secondment to HSM:

STARWARS

Funded by the European Union

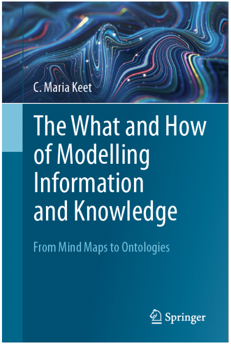
- Springer for the 20% discount on the What and How book

Springer

# Thank you!

# Questions?

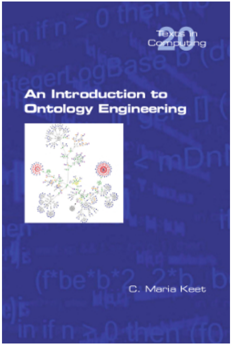
- My book on modelling,
- aimed at a broader audience, and
- available in hardcopy and eBook
- https://link.springer.com/book/10.1007/978-3-031-39695-3



# Thank you!

# Questions?

- My textbook on ontology engineering (aimed at computer scientists)
- Free pdf (and slides and exercises) at `https://people.cs.uct.ac.za/~mkeet/OEbook/`
- Also available in paperback (College Publications):



# References I


D. Calvanese, P. Liuzzo, A. Mosca, J. Remesal, M. Rezk, and G. Rull.
Ontology-based data integration in epnet: Production and distribution of food during the roman empire.
*Engineering Applications of Artificial Intelligence*, 51:212–229, 2016.

N. Chahinian et al.
Evaluation of an early flood warning system in bamako (mali): Lessons learned from the flood of may 2019.
*J Flood Risk Mgmt.*, 2023.

T.A. Halpin.
*Information Modeling and Relational Databases*.
San Francisco: Morgan Kaufmann Publishers, 2001.

Joshua S. Madin, Shawn Bowers, Mark P. Schildhauer, and Matthew B. Jones.
Advancing ecological research with ontologies.
*Trends in Ecology & Evolution*, 23(3):159–168, 2008.

M. Ostaszewski et al.
Covid19 disease map, a computational knowledge repository of virus–host interaction mechanisms.
*Molecular Systems Biology*, 17(e1038), 2021.

Hervé Pruvost and Olaf Enge-Rosenblatt.
*Using Ontologies for Knowledge-Based Monitoring of Building Energy Systems*, pages 762–770.
2022.
doi: 10.1061/9780784483893.094.

# References II


J. Reese et al.
Generalisable long covid subtypes: findings from the nih n3c and recover programmes.
*eBioMedicine*, 87(104413), 2023.

David G. Tarboton.
Rainfall-runoff processes.
2003.
URL `https://hydrology.usu.edu/rrp`.
Utah State University, USA.

Albert Tort and Antoni Olivé.
An approach to testing conceptual schemas.
*Data & Knowledge Engineering*, 69:598–618, 2010.

K. Wolstencroft, R. Stevens, and V. Haarslev.
Applying OWL reasoning to genomic data.
In C.J.O. Baker and H. Cheung, editors, *Semantic Web: revolutionizing knowledge discovery in the life sciences*, pages 225–248. Springer: New York, 2007.