Designing format-agnostic, robust metadata specifications

James Hester

ANSTO

April 12th, 2016

Setting the scene

Phil. Trans. R. Soc. Lond. 1796 86, 131-134

The direction of its motion seems to be towards the south preceding side, and it is about 5 or 6′ removed from its former place.

1ʰ 51′. The diameter of the comet is about 5′. It has no kind of nucleus, and has the appearance of an ill-defined haziness, which is rather strongest about the middle.

2ʰ 16′. The comet is about 2° 38′ *np* γ, in a line continued from 69 through 37 Cygni.

3ʰ 37′. The comet is about 2° 50′ *np* γ, in a line continued from 65 through 37 Cygni, or, perhaps more accurate, in a line from 70 continued through 37 Cygni.

It will probably pass between the head of the Swan and the constellation of the Lyre, in its descent towards the sun.

The direction of its motion is retrograde.

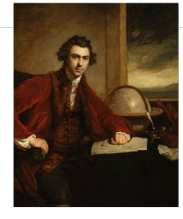
Place of the comet deduced from the above.

Nov. 7.	0 33	RA	20 3 48	PD	49 17 18
	3 37		20 0 58		49 37 18

As the appearance of one of these objects is almost become a novelty, I flatter myself that this intelligence will not be uninteresting to astronomers.

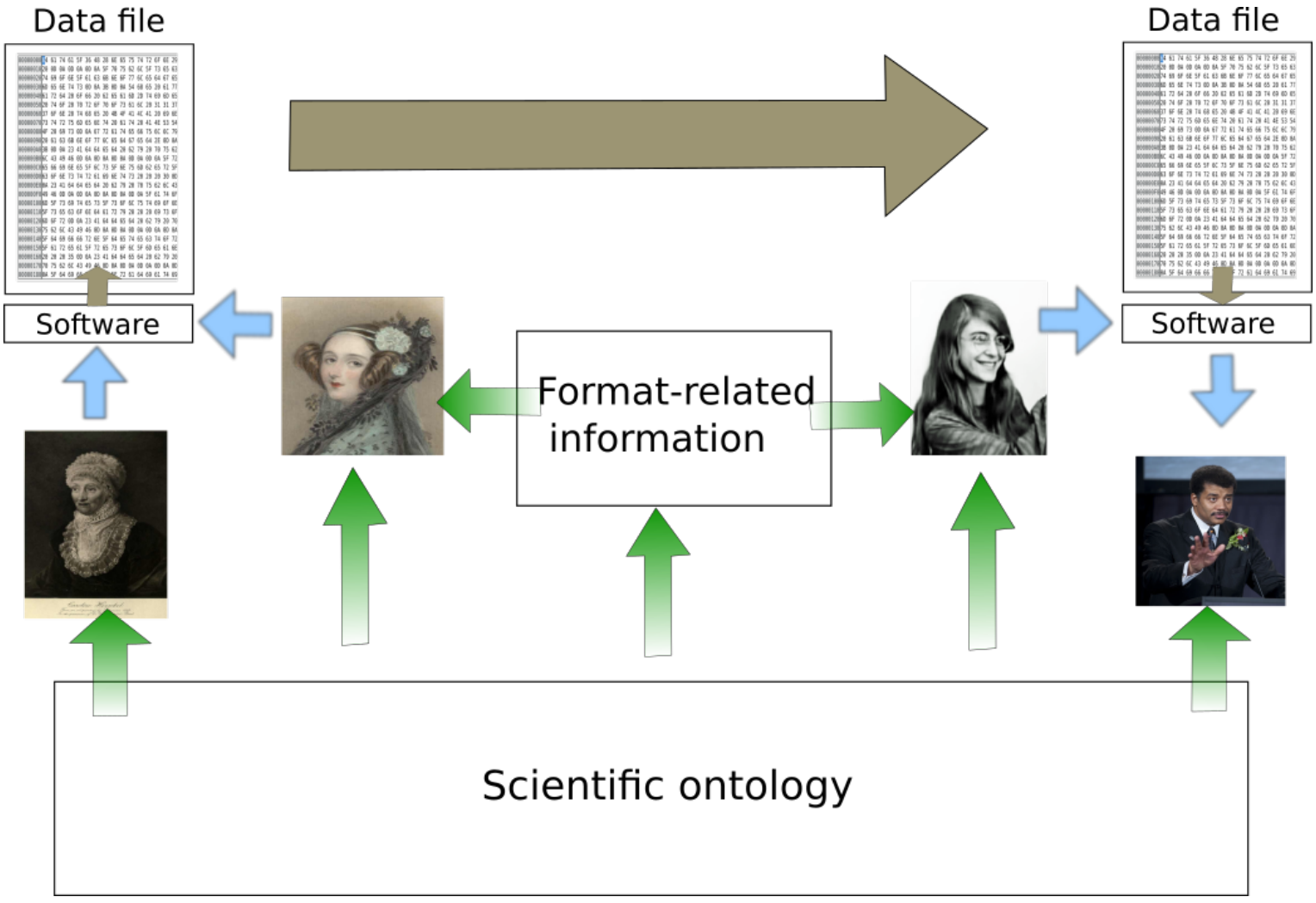
I have the honour to be, &c.

CAROLINA HERSCHEL.



Shared understanding: grammar, terms, symbols

Computer-mediated information transfer



Simplification

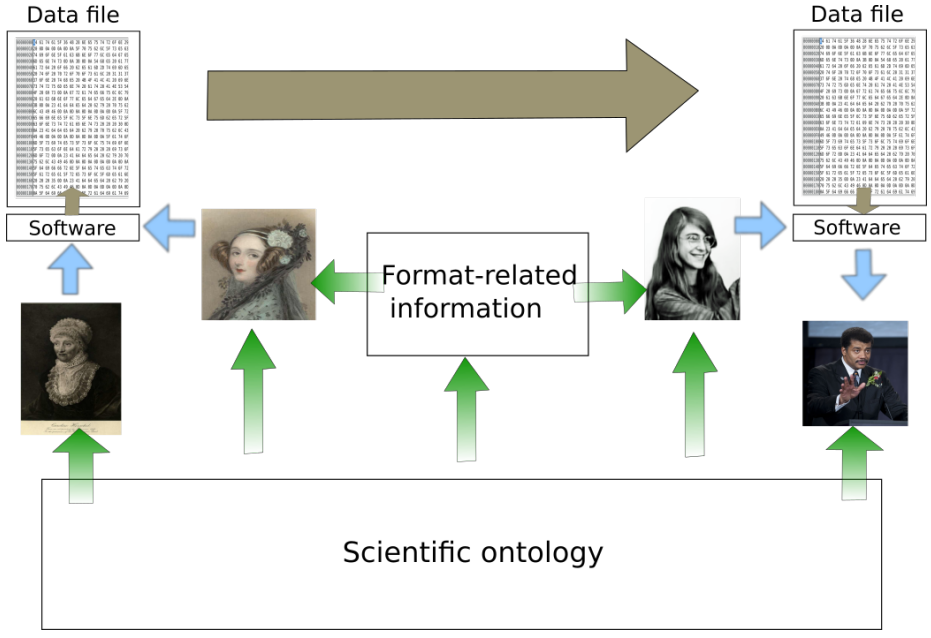
Principle: scientific and engineering knowledge does not depend on the file format.

- Therefore, metadata working groups can (and should) completely ignore file format discussion

Assumption: software authors have access to the common ontology

- So we are not trying to describe an ontology to a computer
- The primary audience will be humans
- There is no need to repeat the ontology in the file

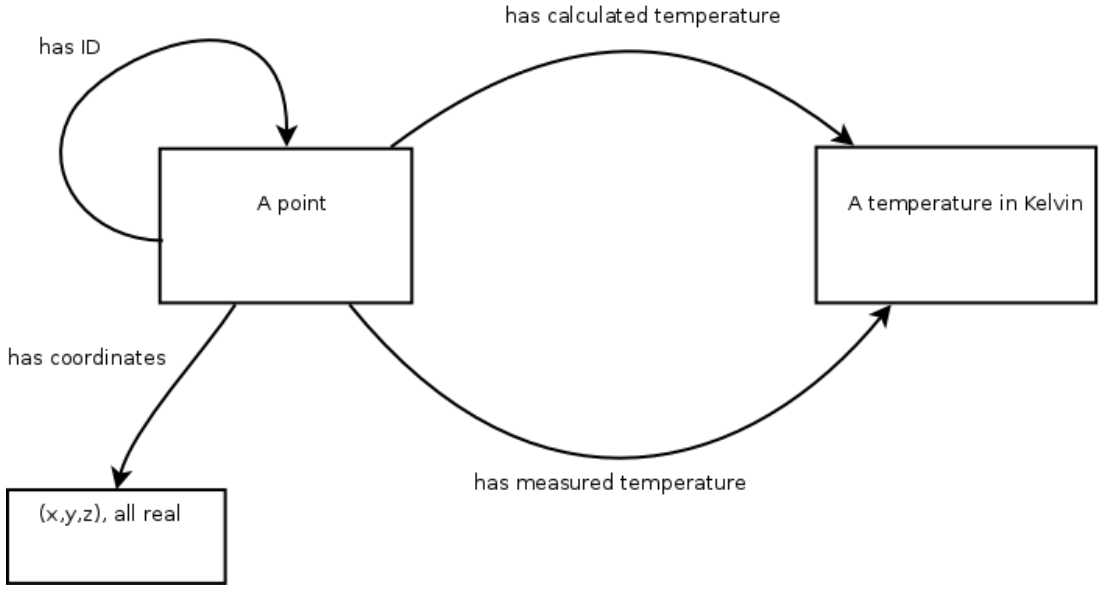
Questions



How do we describe our scientific ontology? What do we include?

How do we relate our scientific ontology to our format?

Describing an ontology



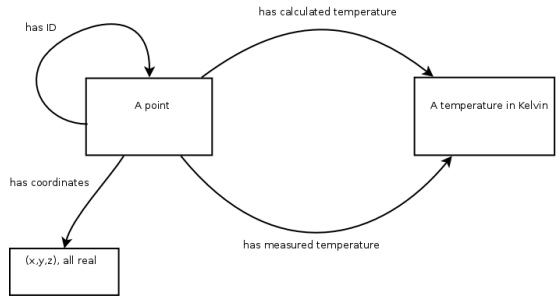
Spivak and Kent (PLoS ONE 2012): "ologs".

- "types" connected by "aspects"
  - "Identifier" aspects refer to the elements in types
- Types are sets and aspects are functions
- Lightly disguised categories from mathematical category theory
- Completely isomorphic to relational database schema

Contents of a metadata standard

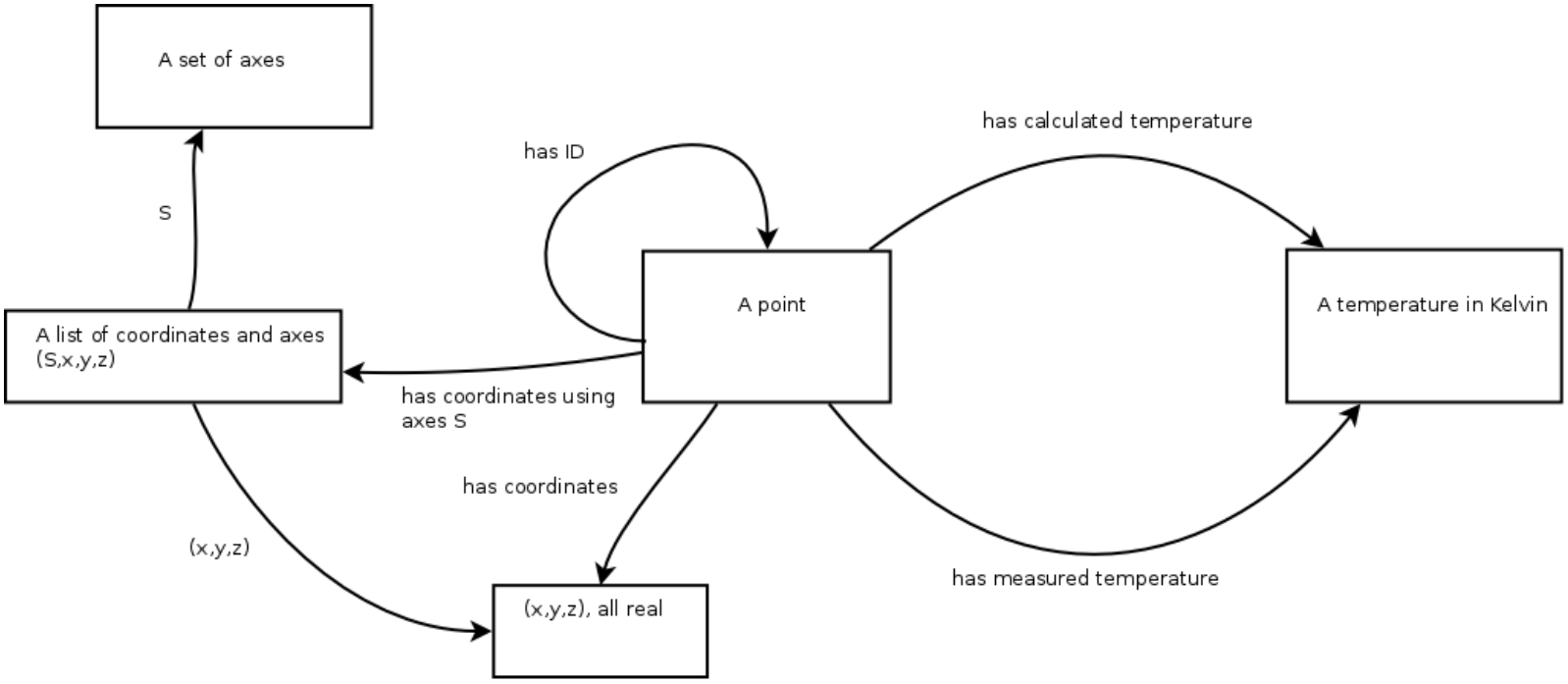
A collection of datanames

- The dataname is the aspect
- Provide a human-readable description that is sufficient to recreate the functional relationship
- Unambiguous (ie a function)
- Describe the domain and codomain
  - Often from a set of simple types (real, integer, finite set of strings, lists of these)
  - ID datanames can be used as shorthand for the domain



Point: an arbitrary identifier for a point in the mesh
Calc_temp: the calculated temperature at Point in Kelvin
Coord: coordinates of Point in mm

Expanding the standard



Always create new types for new dependencies

- New type includes coordinate system used
- But can clarify the old (human-readable) definition, e.g.

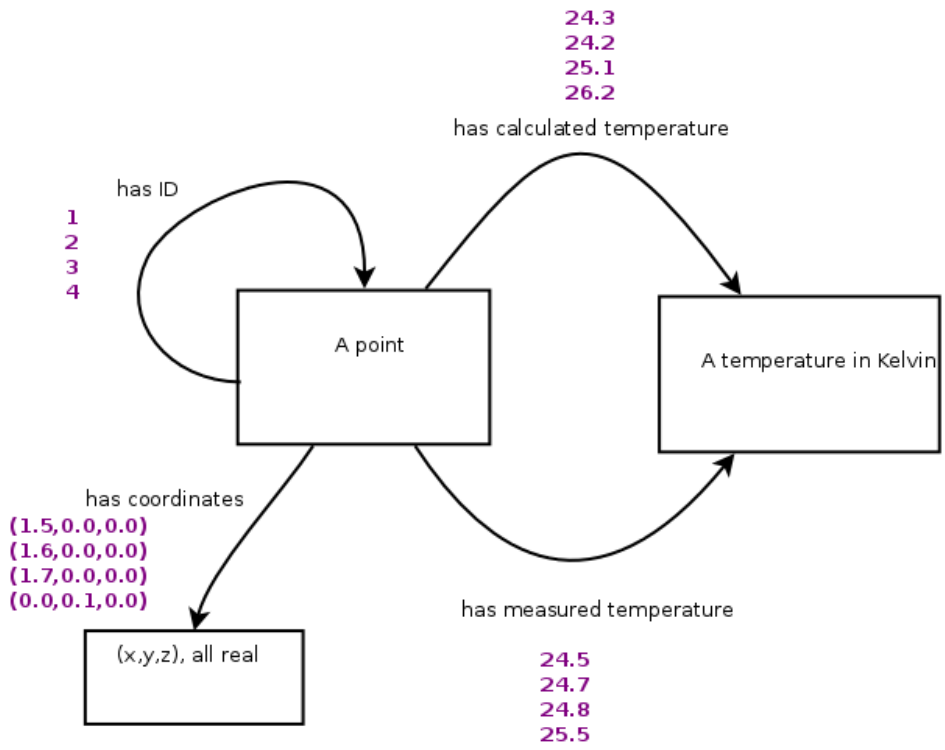
"coordinates in standard coordinate system"

Relationship to the datafile

A datafile documents an instance of the olog

For each file format specify:

- How ontological types are represented (e.g. Real ⇒ IEEE754)
- Which datanames are available
- Where values for those datanames are located in the file
  - No scientific calculations allowed, except for units

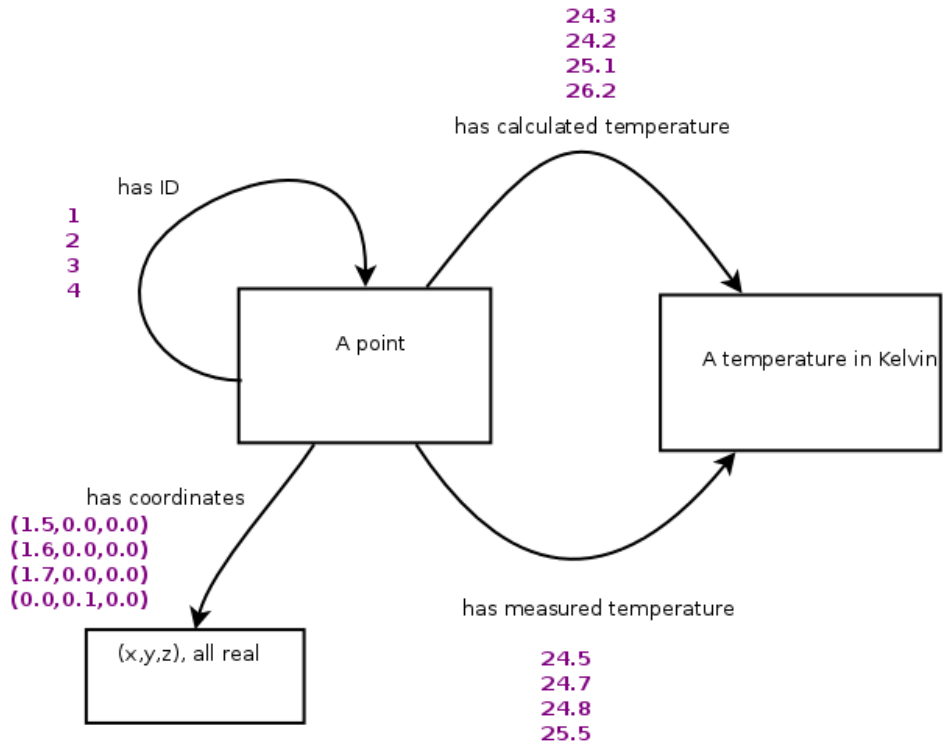


Requirements for a data format

Key requirement

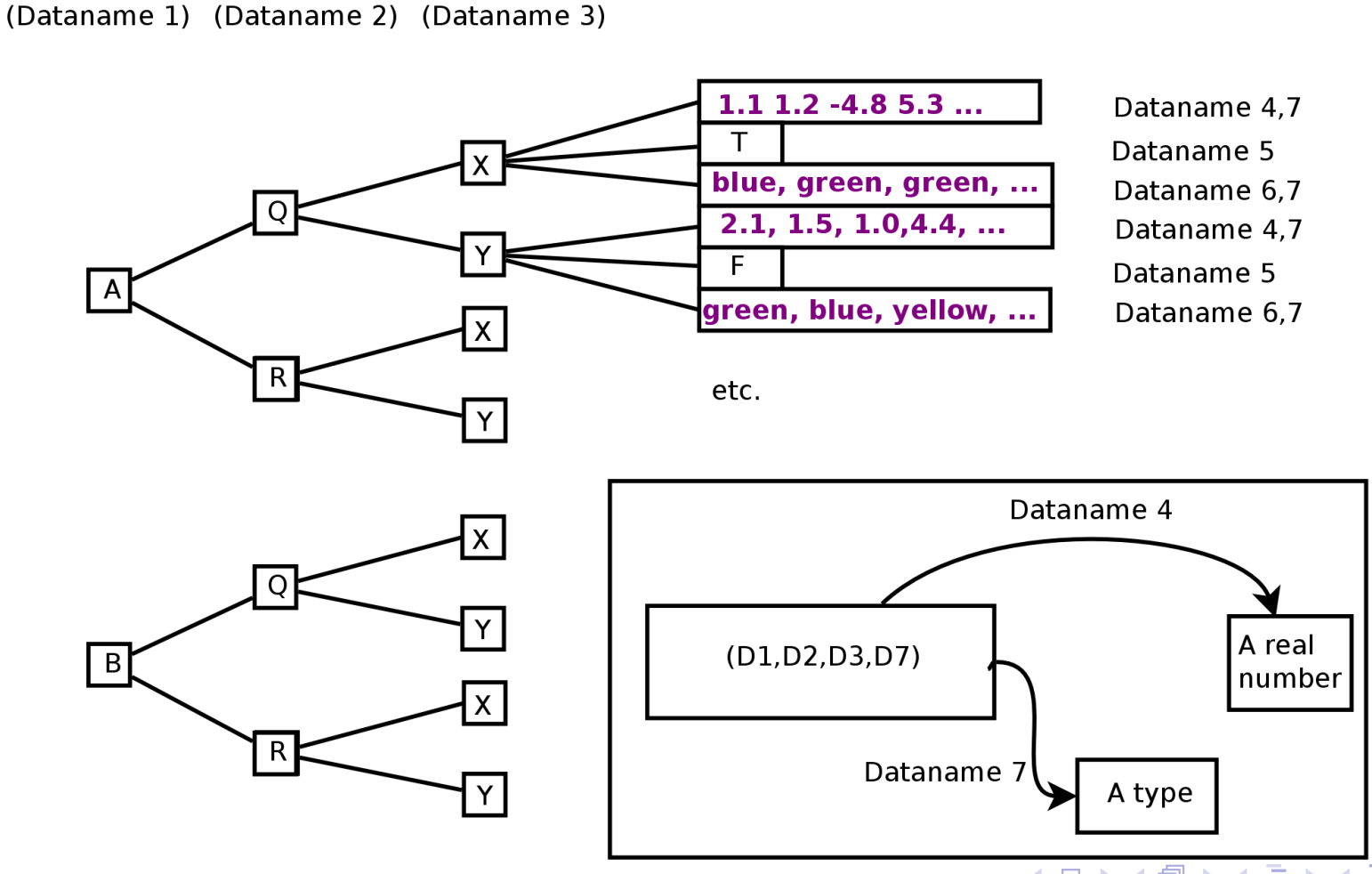
Must be able to associate multiple values of the appropriate type with an arbitrary number of datanames

- Any data format features beyond this should have non-metadata justifications (e.g. storage, transfer, access efficiency)



Example: Hierarchies with arrays (e.g. HDF5)

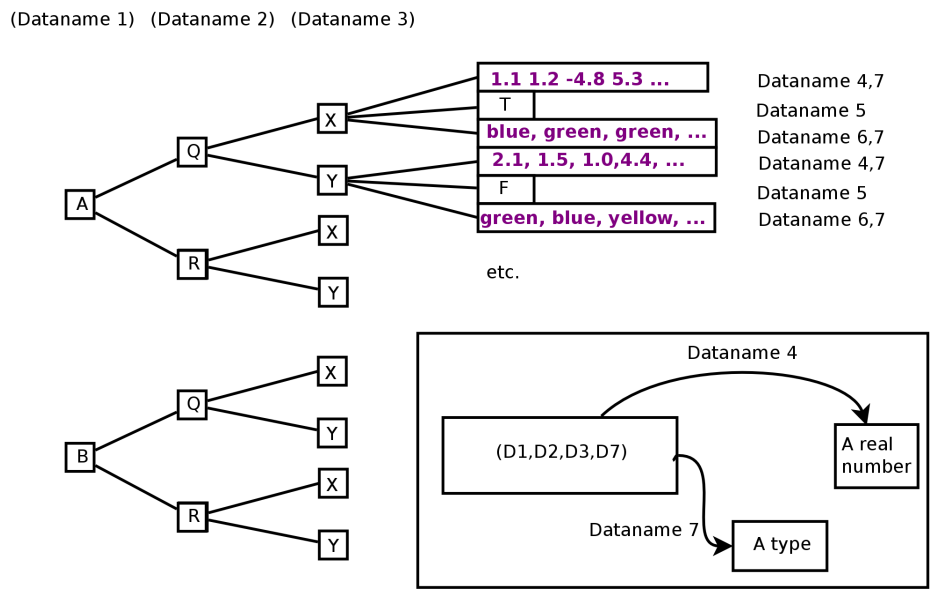
- A hierarchy can reduce repetition of identical values
  - Dataname 5 depends on the values of datanames 1-3
  - Datanames 4 and 6 depend on the values of datanames 1-3 and 7
- Array position can be used to link values



Applications: synthesising pre-existing specifications

Step 1: For each specification, identify all multiply-valued objects

- For example: lists, columns, group "types"
- These objects become the datanames
- Identify their domains (the values are the codomain)
  - The list position may be a 'hidden' domain
- Single-valued objects of interest also collected
- All other format-specific information is irrelevant.



Synthesis continued

Step 2: Identify identical datanames.

- Only possible if datanames act on identical domains.
  - If the dependencies are different, the datanames are different!

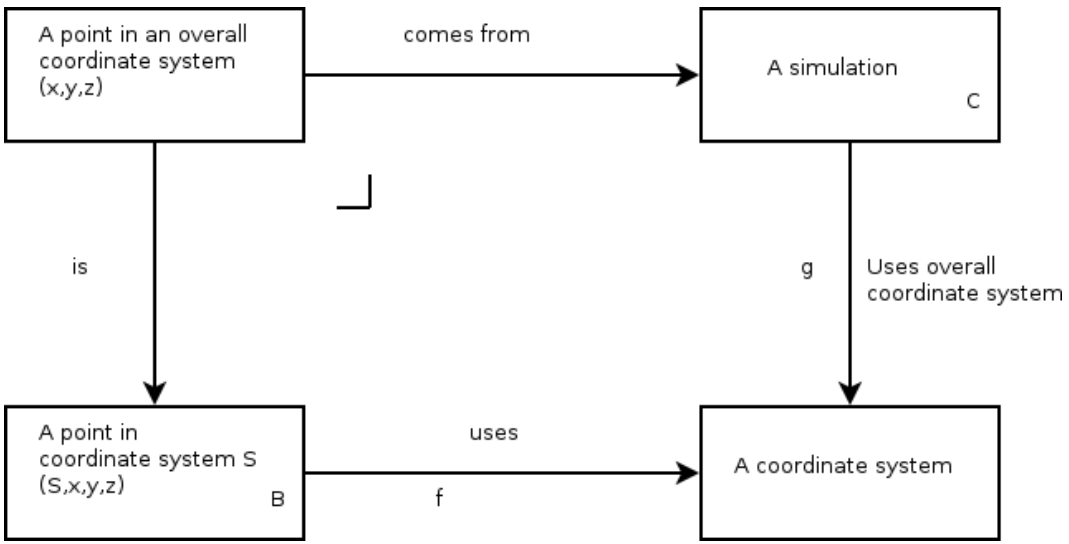
Step 3: Add transformations between domains from different standards

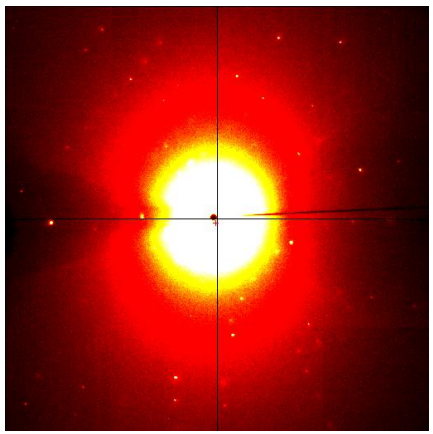
- New datanames
- How do we express relationships?

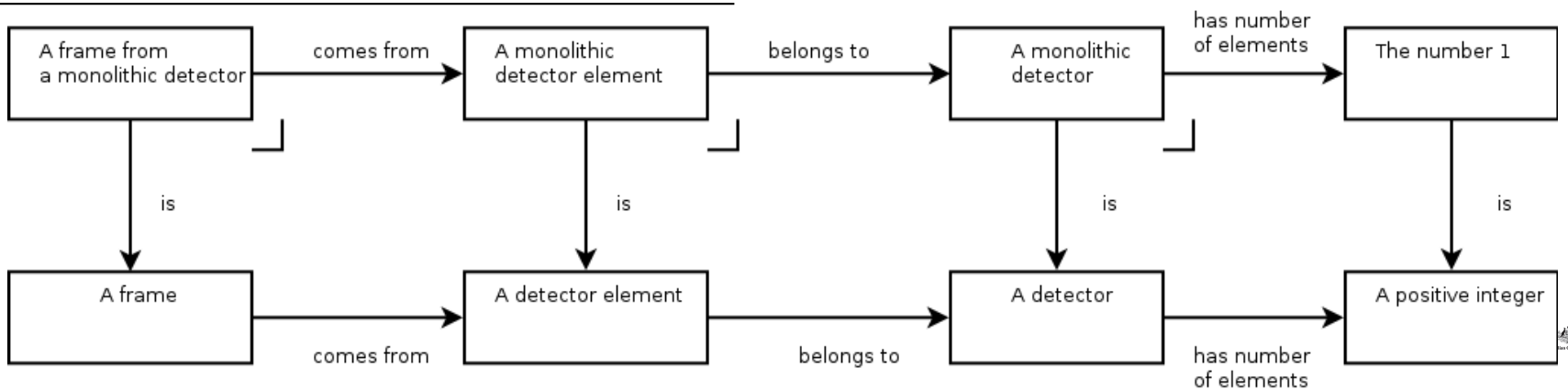
Expressing relationships: pullbacks

- Pullback (fibre product):

$$B \times_D C = \{(b, c) | b \in B, c \in C, f(b) = g(c)\}$$

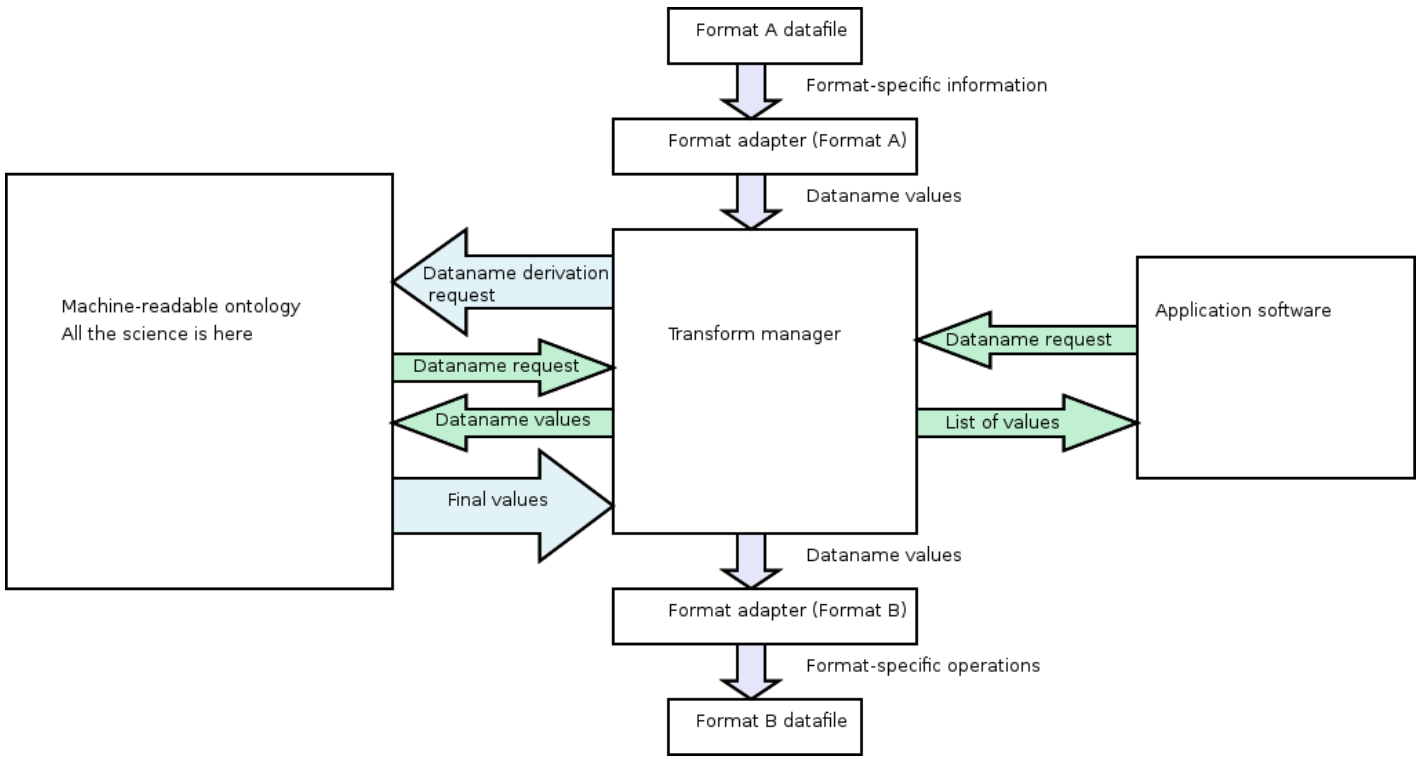






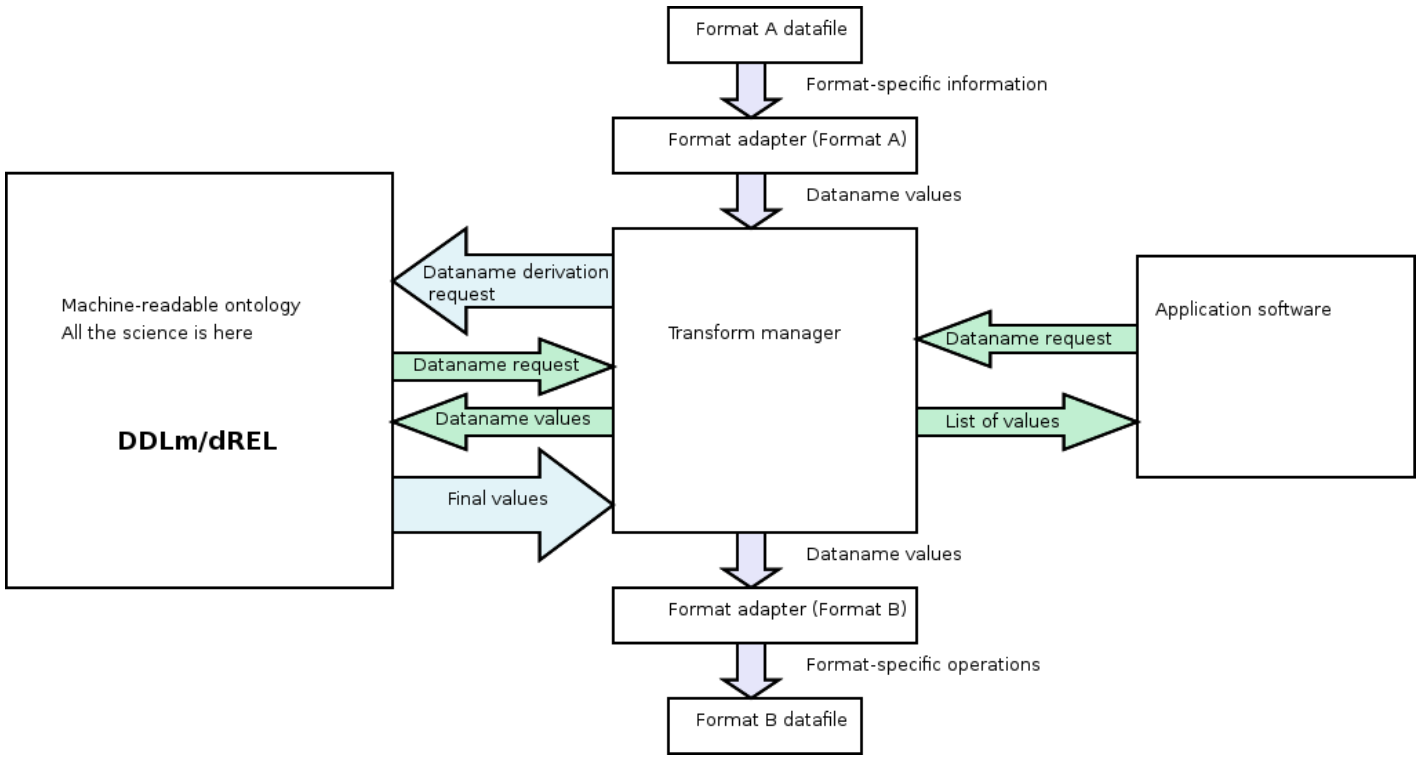
Computer translation between formats

- If mathematical relationships are encoded in a machine-readable ontology, software can automatically transform data.
  - Or a 'universal front end' for input software
  - The ontology becomes a universal interchange format



Computer translation: tools

- Building blocks already exist:
  - DDLm: machine-readable ontology description language
    - Additional attributes allow pullbacks to be specified
  - dREL: machine-readable algorithmic transformations
- There is a working demonstration for NeXus, CIF image files (see final slide)



Summary

- Metadata working groups can ignore file format issues
- The olog framework:
  - permits clear assessment of format suitability
  - leads to a simple API for machine translation between arbitrary file formats using machine-readable ontologies
  - allows synthesis of old standards into new standards
  - enables trouble-free growth of the standard

Resources:

- Demonstration format transformation software (NeXus/HDF5, CIF):
  - `https://github.com/jamesrhester/PyFormatTransformer`
- DDLm:
  - `http://www.iucr.org/resources/cif/ddl/ddlm`
  - Spadaccini, N. and Hall, S. R., (2012) J. Chem. Inf. Model, 52(8) p 1907
- dREL:
  - Spadaccini, N, Castleden, I. R., du Boulay D. and Hall, S. R. (2012) J. Chem. Inf. Model 52(8) p 1917
- PyCifRW:
  - Implements DDLm and dREL in Python (and CIF support)
  - `https://bitbucket.org/jamesrhester/pycifrw`
- More detail in paper submitted to The Data Science Journal: see me for a copy