LaTiS: Interoperability via a Universal Functional Data Model

Doug Lindholm, Anne Wilson

https://github.com/dlindhol/LaTiS

Laboratory for Atmospheric and Space Physics, University of Colorado Boulder
AGU FALL MEETING San Francisco | 15–19 December 2014 IN33B-3771

Data Model:
Standardized representation of data elements and how they are related to one another.

- http://en.wikipedia.org/wiki/Data_model

Abstraction:
Information representation that is similar to it's meaning (semantics) while hiding the implementation details.

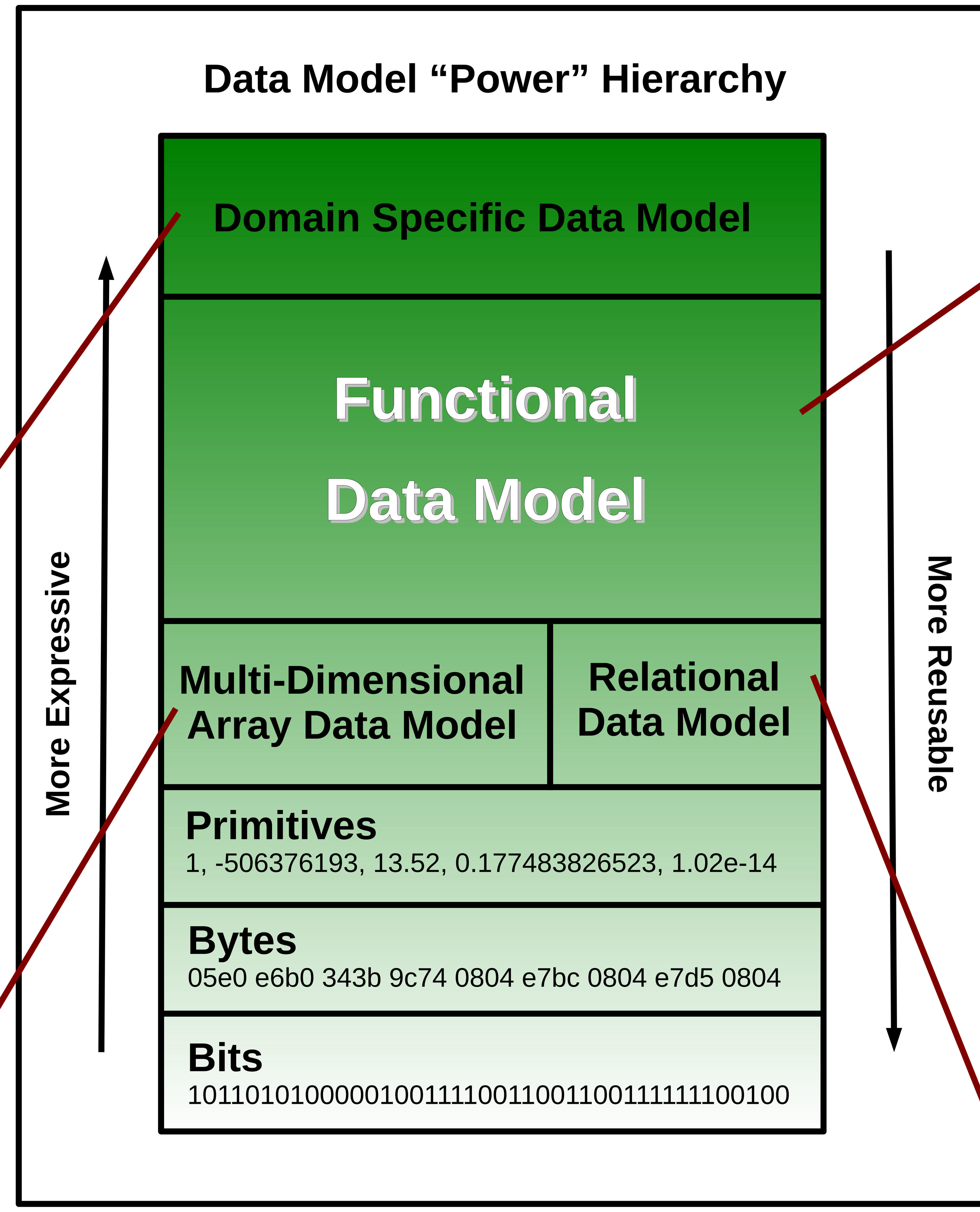
- http://en.wikipedia.org/wiki/Abstraction_(computer_science)

To be most useful to the data user, a data model should provide an abstraction that represents how the **user** thinks of the data, not how it is **stored**.

Rule of Least Power:
Use the least powerful language suitable for expressing information on the World Wide Web

- Tim Berners-Lee, World Wide Web Consortium

- Powerful languages inhibit information reuse.
- The less powerful the language, the more you can do with the data stored in that language.
- Use a *scalable* language so it can be extended with more powerful features.



Functional Data Model
Abstraction that represents data as any combination of only three *variable* types:

- **Scalar**: A single variable
- **Tuple**: A group of variables
- **Function**: A mapping from one variable to another

- Captures the functional relationship semantics that are inherent in scientific data.
- Mathematical model that is scientific domain agnostic and lends itself to mathematical operations:
  - arithmetic that can be used directly to perform computations with datasets
  - algebra that can be used to symbolically manipulate datasets
  - Function "evaluation" that supports: sampling, resampling/regridding, currying
- Ideally situated in the power hierarchy to be expressive enough to be useful yet simple enough to be highly reusable across scientific disciplines.
- Can be as expressive as needed with complex composition of the three variable types (at the risk of reducing reusability).
- Can scale by extending base types to add semantics. Basic tools can still use the data in terms of the base types.
- **Universal**: Can be used to model (describe) any data.
- Enables unified interface to access and operate on disparate data sources.

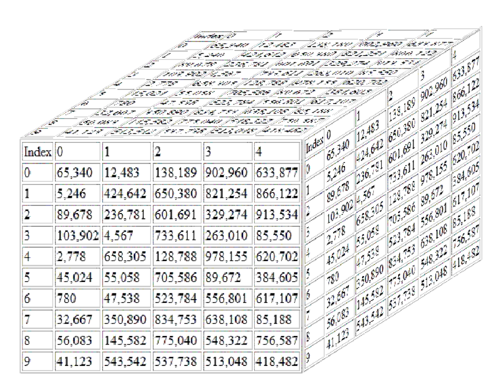
Domain Specific Data Model
Standard data representation designed to serve the users of a particular scientific community.

- Tend to be very powerful/expressive so they can adequately represent the details needed by all users in that scientific domain.
- Often too complex to be used interoperably by others outside of that domain.

Multi-Dimensional Array Data Model
Data represented as rigid grids of primitive values.

- Largely influenced by the storage implementation as opposed to data semantics.
- Higher level abstractions require "conventions' or additional metadata.

(*ix, iy, iz*) → value
ix → x, *iy* → y, *iz* → z
⇩
(x, y, z) → value

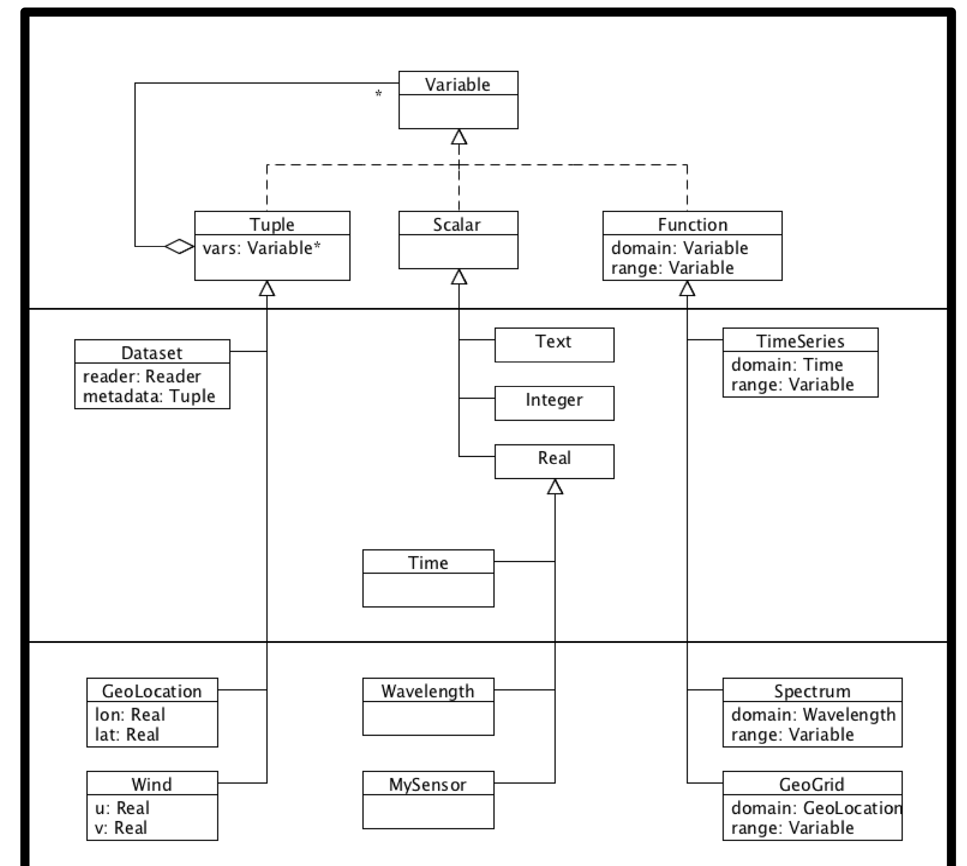


Relational Data Model
Data represented as tables (relations) of primitive values, as used by relational databases.

- Largely influenced by the storage implementation as opposed to data semantics.
- Higher level abstractions require multiple tables and complex join logic.

Sequence of triples:
i → (time, flux, class)
⇩
Time Series:
time → (flux, class)

	time	flux	class
0		3.5	B
1		4.6	A
2		4.7	A
3		4.1	A
4		3.2	B



Functional Data Model (as UML) plus extensions demonstrating more powerful abstractions.