Universiteit Utrecht

Faculty of Geosciences
Department of Physical Geography

A physical data model for fields and agents

Kor de Jong, Merijn de Bakker, Derek Karssenberg

Introduction

We are developing a conceptual data model (called LUE) for representing both fields and agents. One of the goals of this work is to be able to build a high level field and agent based modeling environment with a single data type that is capable of representing all state variables that are manipulated by the model.

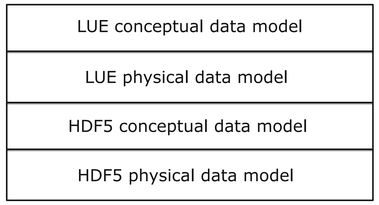
To be able to store the information represented by the conceptual data model we need a physical data model (a dataset format). Here we present some of the early results of developing such a data model using the HDF5 conceptual data model and software library.

Main requirements

- Represent fields and agents
- Efficient: compact, capable of parallel I/O
- {1,2,3}D space, time
- Optimal file layout for each individual data type
- User-defined coordinate types
- User-defined value types
- Open for addition of new kinds of data
- Portable

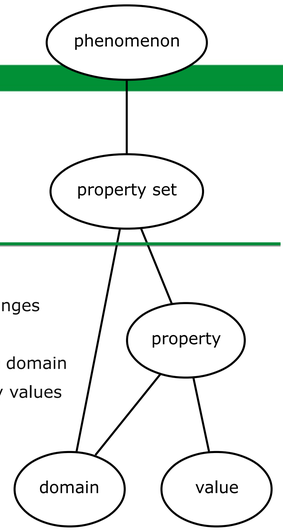
Data model stack

Various data models, layered on top of each other.



Conceptual data model

Phenomenon: something that changes through space and time
Property set: properties sharing a domain
Domain: when and where property values are valid
Property: characteristic
Value: magnitude
Item: agent, individual, object, …



Taxonomies

There are many aspects in which data types differ from each other. To be able to represent all data types and optimize storage layout, these aspects must be identified. For all permutations, the optimal physical data model must be implemented.

Collection of items
- Constant through time
- Variable through time

Time domain
Sharing
- Shared between items
- Unique per item

Location
- Omnipresent
- Located in time: point, period, cell, …

Space domain
Location
- Omnipresent
- Located in space: point, line, region, cell, ...
- One or multiple geometries per item

Indexed or not
Topological or not

Value
Discretization
- Not discretized
- Discretized: region, cell, ...

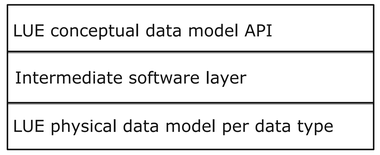
No-data
- Not masked
- Masked: by value, by bitmask, ...

...

By permuting the aspects of the terminals of the taxonomies, we end up with many different data types we can represent, eg:

- Constants
- GPS tracks
- Spatio-temporal rasters
- TINs
- Networks
- ...

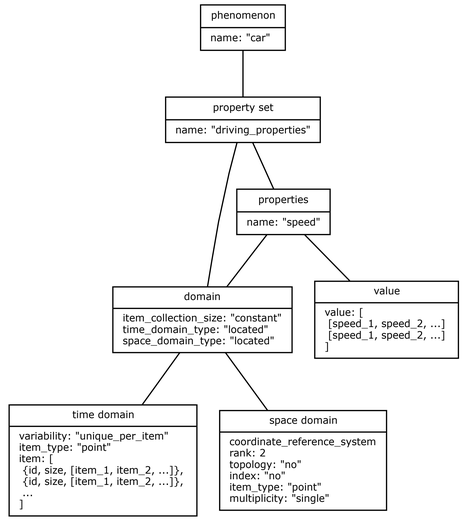
Our approach is to implement an optimal physical data model for each data type and add a software layer to bridge this low level API to the higher level conceptual model. End users only have to use the high level API.



Physical data model

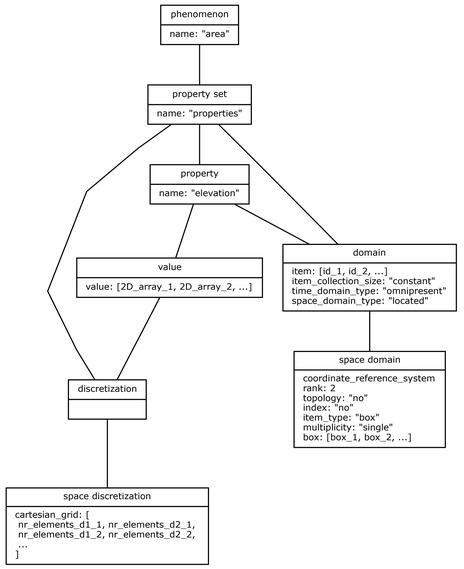
In the following examples, boxes and information between accolades are stored as HDF5 groups. Groups contain either HDF5 attributes, like names, and HDF5 datasets.

Agents in HDF5



The example shows the physical data model layout for storing the location and speed of a fixed collection of cars.
Time domain: for each car, an id and a growing collection of time points
Space domain: for each car, a location in space
Value: for each car, a growing collection of speeds

Fields in HDF5



The example shows the physical data model layout for storing elevation fields as a raster per area.
Time domain: not needed, the elevation doesn't change over time
Space domain: for each area a box
Discretization: for each area, for each space dimension, the number of elements
Value: for each area, a 2D array of elevations

Future work

- Support and tune parallel I/O
- Support more data types (including relations and networks)
- Support topological space domain
- Support spatial indices
- Implement support for uncertain data

References

De Bakker, M, K. de Jong, O. Schmitz, D. Karssenberg. 2016. A conceptual data model and modelling language for fields and agents. EGU poster A.449
The HDF Group. Hierarchical Data Format, version 5, 2016. http://www.hdfgroup.org/HDF5/

Kor de Jong: k.dejong1@uu.nl, **PCRaster development team**: http://www.pcraster.eu