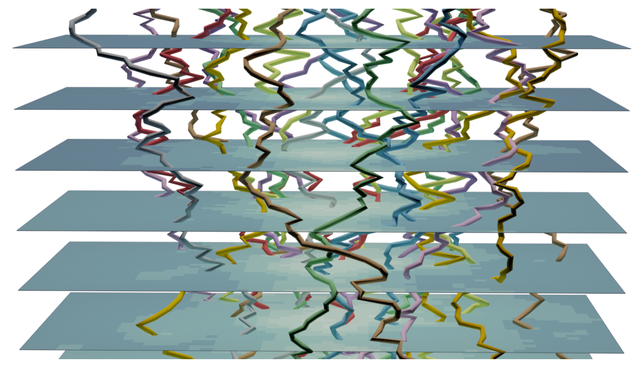
Building spatial simulation models that run out-of-the-box on compute clusters: the LUE model building framework

Kor de Jong (k.dejong1@uu.nl), Oliver Schmitz, Derek Karssenberg

Department of Physical Geography, Faculty of Geosciences, Utrecht University



Possible applications

- Global scale high-resolution hydrological modelling
- Continental scale land use / land cover change modelling
- Real-time forecasting of hazards (e.g. floods, droughts)
- Agent-based modelling of extremely large systems (e.g. disease modelling, energy networks)
- Representing spatial agents including their spatial contexts or spatial properties in GIS analysis or numerical simulation models
- Analysis of remote sensing imagery (e.g. object-based image analysis)

Challenge: design a modelling framework that:

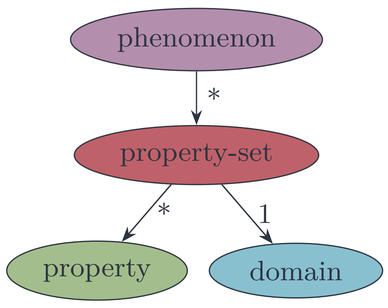
- **Can represent systems of discrete agents and continuous fields**
  $\Longrightarrow$ Requires integration of agent- and field-based modelling
- **Allows models to be executed on laptops and compute clusters**
  $\Longrightarrow$ Requires resulting models to be scalable
- **Can be used by domain experts**
  $\Longrightarrow$ Requires an API with a high abstraction level

LUE modelling framework for testing hypotheses:

- **Data model** $\Longrightarrow$ for representing state
- **Framework** $\Longrightarrow$ for representing and evaluating processes
- **High level APIs** $\Longrightarrow$ for defining processes

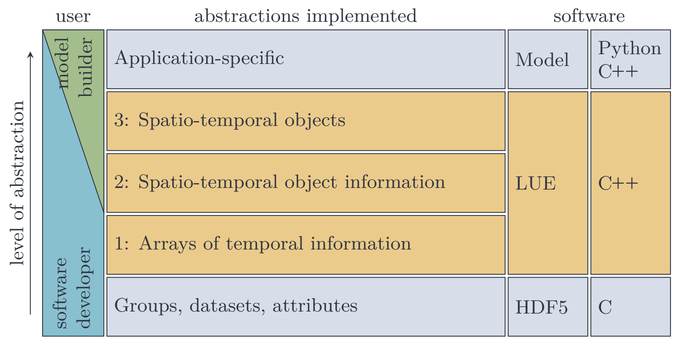
The software is / can be used in modelling studies (next box).

Conceptual data model



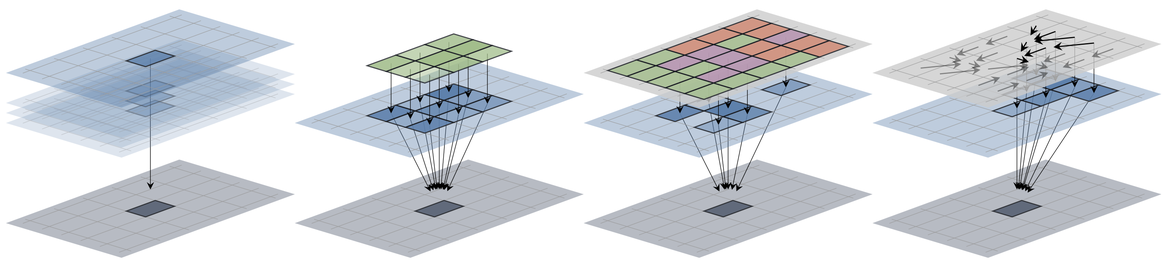
- Abstraction level above raster and vector data models
- Allows discrete agents and continuous fields to be treated as variations of a more general approach to organize data

Physical data model



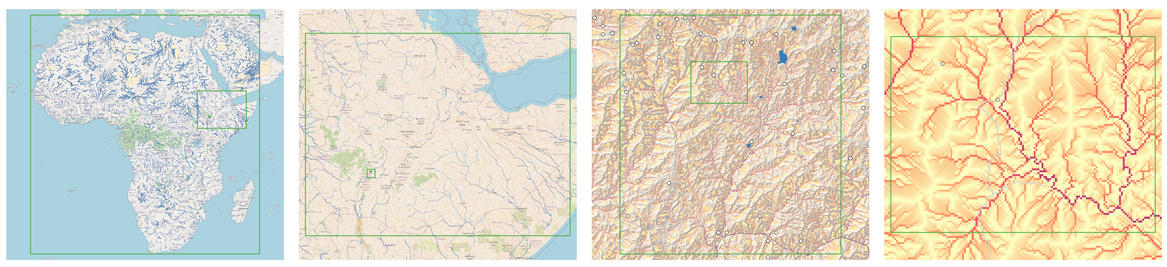
- Implementation of the conceptual data model
- Allows the storage of very diverse kinds of state variables in a single data set

Framework operations



- Current focus is on field-based modelling
- Modelling operations inspired by map algebra

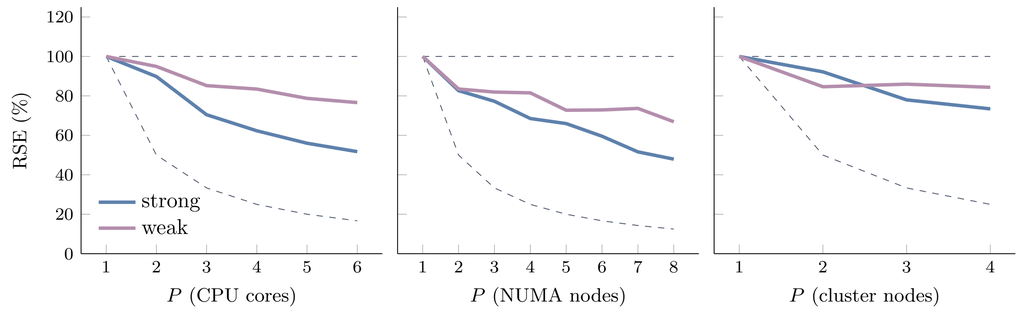
High resolution, continental scale modelling



PyCatch + MERIT Hydro data:

- 1 h time steps, 3 arc-second ($\approx$ 100 m) raster cells (87.600 $\times$ 84.000 cells)
- Model used 12 nodes (1152 CPU cores)

Scalable models



Good performance and scalability

- **Data orientation**
- **Asynchronous many-tasks** (AMT): HPX C++ library
- Case-study model: flow accumulation + 57 local operations

High level APIs

Runs on all CPU cores on all available compute nodes!

```
flow_direction = d8_flow_direction(elevation)
outflow, residue = accu_threshold(
    flow_direction, precipitation, infiltration_capacity)
```

Python API:

- Thin language binding around C++ API
- Works with regular Python interpreter

- Python or C++ LUE API
  $\Longrightarrow$ Convenient for model development
- AMT execution model
  $\Longrightarrow$ Good performance and scalability

[1] Merijn P. de Bakker, Kor de Jong, Oliver Schmitz, and Derek Karssenberg. Design and demonstration of a data model to integrate agent-based and field-based modelling. *Environmental Modelling & Software*, 89:172–189, 2017.

[2] Kor de Jong and Derek Karssenberg. A physical data model for spatio-temporal objects. *Environmental Modelling & Software*, 122:104553, 2019.

[3] Kor de Jong, Debabrata Panja, Derek Karssenberg, and Marc van Kreveld. Scalability and composability of flow accumulation algorithms based on asynchronous many-tasks. *Computers & Geosciences*, 162:105083, 2022.

[4] Kor de Jong, Debabrata Panja, Marc van Kreveld, and Derek Karssenberg. An environmental modelling framework based on asynchronous many-tasks: scalability and usability. *Environmental Modelling & Software*, 139:104998, 2021.

[5] LUE contributors. LUE scientific database and environmental modelling framework. https://lue.computationalgeography.org.