Geosimulation using fields and agents

JUDITH A. VERSTEGEN & OLIVER SCHMITZ

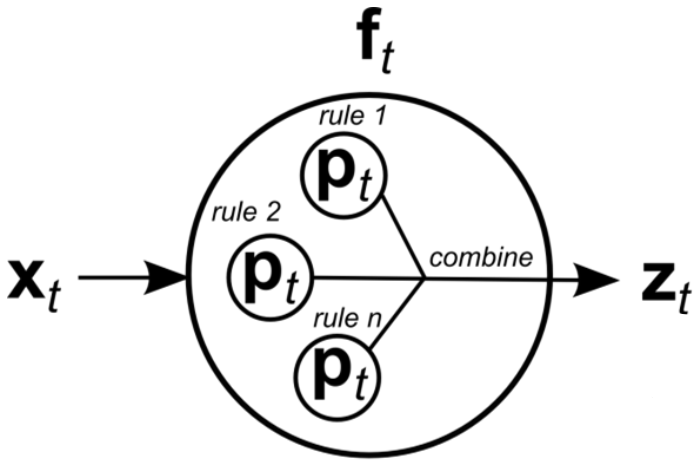
02-09-2021, OPENGEOHUB SUMMER SCHOOL

Motivation

coreymondello.com

Defining Geosimulation

variable	description
t	time step
$\mathbf{z}_t$	system state at t
$\mathbf{f}_t$	transition function
$\mathbf{x}_t$	inputs/drivers at t
$\mathbf{p}_t$	parameters at t

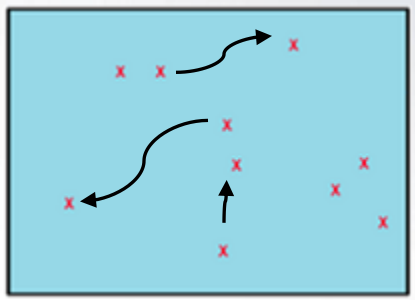


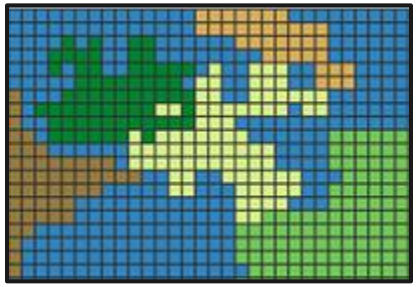
→ **impact**

Data-driven	Spatio-temporal models	**Process-driven**
Start with empirical data		Start with a theory (system description)
Based on correlations between drivers and the system state		Based on known/assumed cause-effect relations between drivers and system state
Also called: empirically-based model, statistical model, machine learning model		Also called: process-based model, physically-based model, **geosimulation model**

Modelling paradigms

	Agent-based modelling (multi-agent systems)	Field-based modelling (cellular automata)
system state	Set of discrete objects	Continuous or discrete
attribute(s)	Is linked to the agent	Has a value everywhere
processes	Behavior of a single agent, potentially moving	Behavior of cells that remain in their location

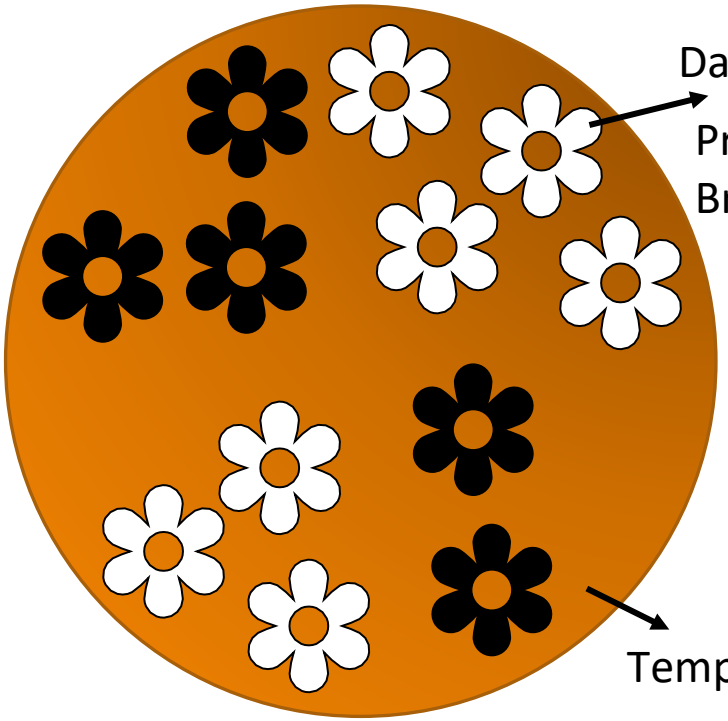


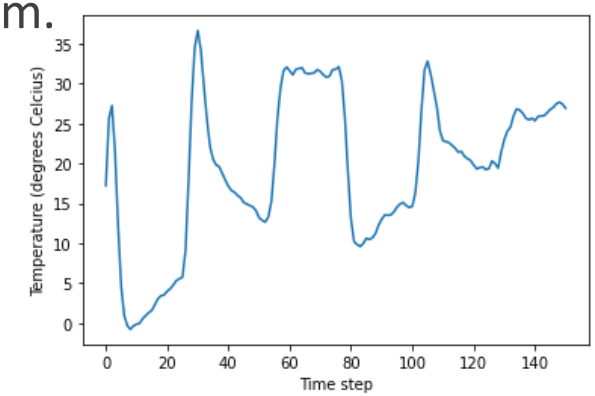


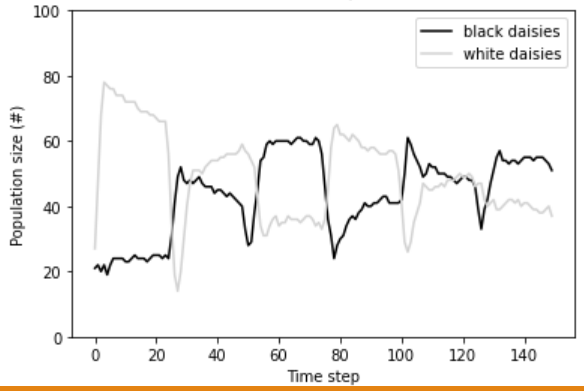
Systems have multiple phenomena that do not fit into one paradigm!

Example: Fields & Agents!

DaisyWorld was introduced by James Lovelock and Andrew Watson (1983), to illustrate the Gaia Hypothesis that organisms interact with their surroundings, creating a self-regulating system.







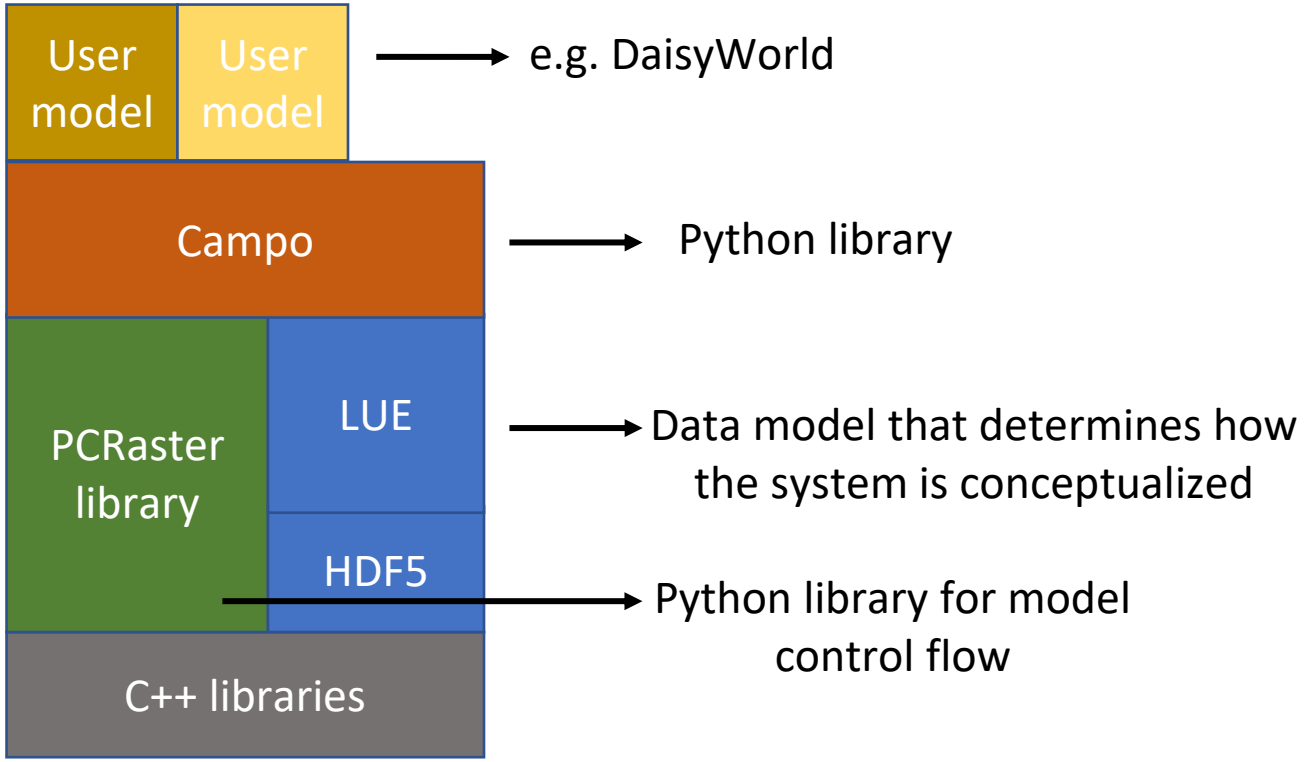
Problem statement

Current software:

- Often departs from ABM, and approaches fields as sets of square objects → computationally sub-optimal
- Has separate sets of functions for fields and agents

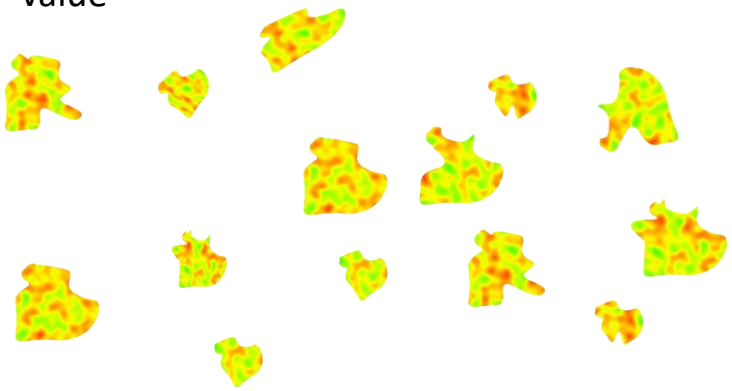
→ No full integration between fields and agents

Campo


User model
User model
e.g. DaisyWorld
Campo
Python library
PCRaster library
LUE
Data model that determines how the system is conceptualized
HDF5
Python library for model control flow
C++ libraries

Phenomenon: agents or field

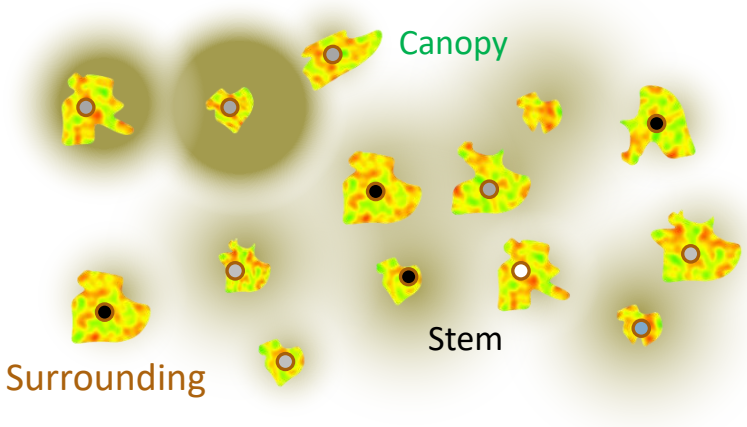
‘Agents’: Phenomenon containing >1 **Objects**, areal coverage of each Object is where it has a value



‘Field’: Phenomenon containing 1 **Object**, areal coverage is ‘modelling area’

Phenomenon has Property Sets

Forest system: trees, stems, tree canopy, seed dispersal..



Phenomenon **Trees**	spatial *domain of each **Object***:
Property Set **Canopy**	*crown*
Property Set **Surrounding**	*circular centered at stem*
Property Set **Stem**	*point at stem*

Single Algebra for Agents & Fields

Syntax:

```
a = a_function(b)
```

Calculates for each Object its property `a` as a function of property `b`

Referring to phenomena, property set, for instance:

```
trees.canopy.lai = a_function(trees.canopy.ndvi)
```

Framework for control flow

```
class MyFirstModel(DynamicModel):

   ...


   def initial(self):

      # functions here are run once at start

      # create/modify Phenomena for initial state of system

      # I/O using framework functions


   def dynamic(self):

      # functions are run for each time step

      # program time transition function

      # I/O using framework functions
```

Future work

- Parallelization on supercomputers via HPX
- Add more functionality to Campo
- We're interested in your case studies with fields and agents!

Contact us via:

https://campo.computationalgeography.org/

judith.verstegen@wur.nl

o.schmitz@uu.nl