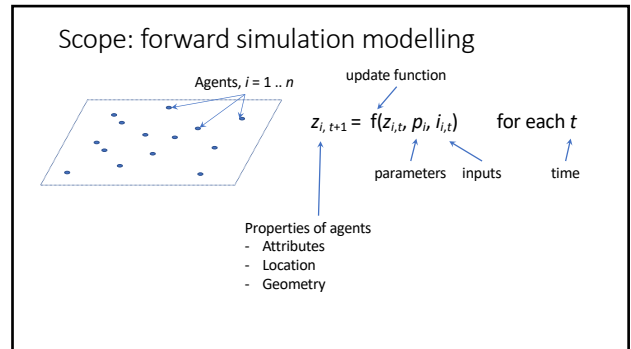


# Campo

## spatial agent-based modelling

Derek Karssenberg

1





2

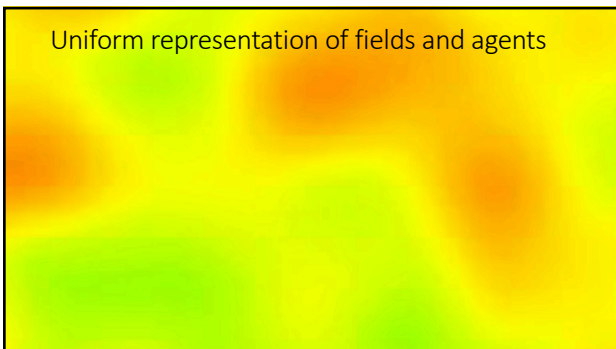
- ### Key concepts of Campo
- 1) System representation: Fields and Agents
  - 2) Model building: Algebra on Fields and Agents (no low-level programming)

3

### 1) Fields and Agents: perceived dichotomy

Fields	Agents
	
extensive & continuous one instance	bounded & mobile multiple instances

4



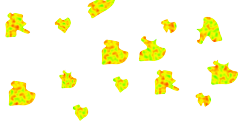
5

### Campo data model

6

### 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 (agent), areal coverage is 'modelling area'



7

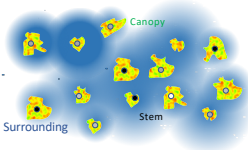
### Defining a phenomenon

```
foodenv = campo.Campo()
foodstores = foodenv.add_phenomenon("foodstores")
```

8

### Phenomenon contains Property Sets

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



Phenomenon Trees  
 Property Set Canopy  
 Property Set Surrounding  
 Property Set Stem

domain of each Object:  
 crown  
 circular centered at stem  
 point at stem

9

### Defining a property set

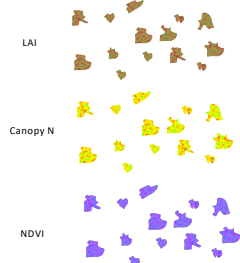
```
foodenv = campo.Campo()
foodstores = foodenv.add_phenomenon("foodstores")
foodstores.add_property_set("frontdoor", "foodstores_frontdoor.csv")
print(foodstores.frontdoor)
```

Prints:  
 Property set: frontdoor  
 Type: Point  
 Properties: 0

10

### Multiple Properties per Property Set

Phenomenon Trees  
 Property Set Canopy  
 Property LAI  
 Property Canopy N  
 Property NDVI  
 ...  
 Property Set Surrounding  
 ...  
 Property Set Stem  
 ...



11

### Defining a property

```
...
foodstores.add_property_set("frontdoor", "foodstores_frontdoor.csv")
foodstores.frontdoor.level = 12.1
print(foodstores.frontdoor)
print(foodstores.frontdoor.level)
```

Prints:  
 Property set: frontdoor  
 Type: Point  
 Property: level  
 Property: level

12

## Campo operations

13

## A single Algebra for Agents and Fields

$a = a\_function(b)$

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

Referring to phenomena, property set, for instance:

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

Diagram illustrating the structure of the code above:

- `trees` is labeled as **phenomenon**
- `canopy` is labeled as **property set**
- `lai` is labeled as **property**

14

## Operators

```
...
# addition of point or field properties
foodstores.frontdoor.x = foodstores.frontdoor.y + \
    foodstores.frontdoor.z

# comparison operator
foodstores.frontdoor.healthy = foodstores.frontdoor.z < \
    foodstores.frontdoor.threshold
```

15

## Functions

```
...
# assign a random value from a uniform distribution
foodstores.frontdoor.lower_inc = 0.0
foodstores.frontdoor.upper_inc = 0.1
foodstores.frontdoor.increment = \
    campo.uniform(foodstores.frontdoor.lower_inc, \
        foodstores.frontdoor.upper_inc)
```

16

## Campo control flow

17

## Framework for control flow

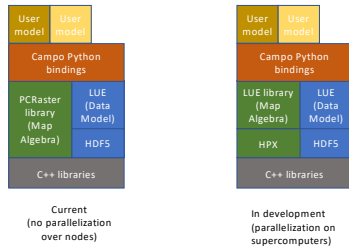
```
class MyFirstModel(DynamicModel):
    ...

    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
```

18

### Campo: software stack



19

### Info and downloads

<http://campo.computationalgeography.org>

20