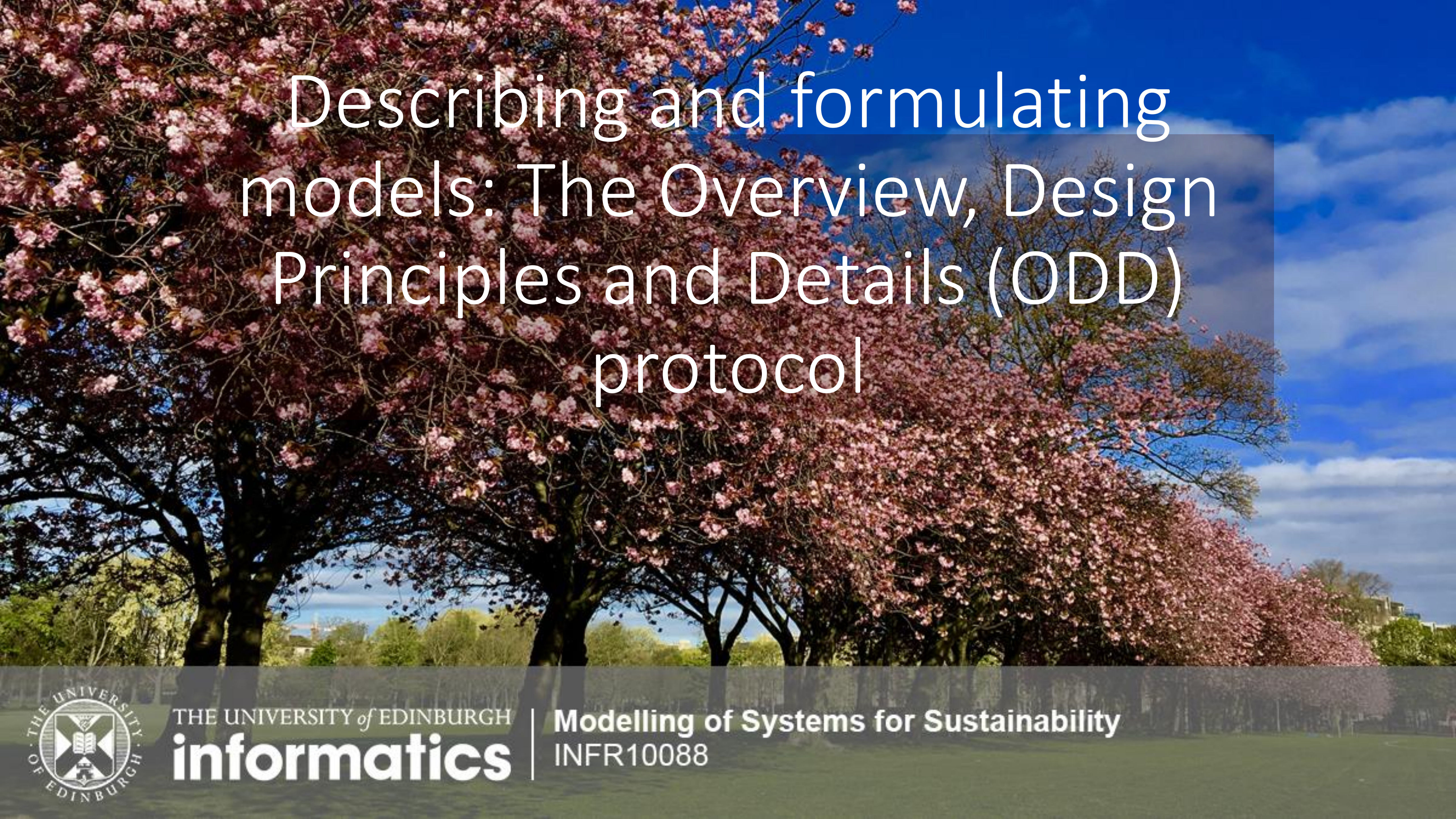
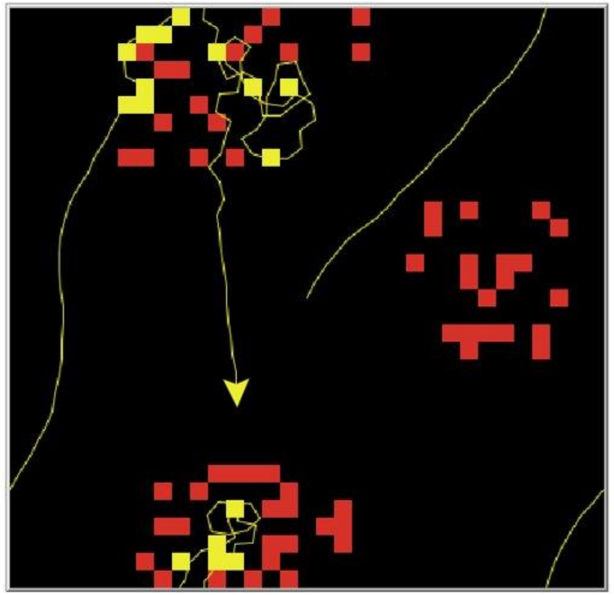


Describing and formulating models: The Overview, Design Principles and Details (ODD) protocol

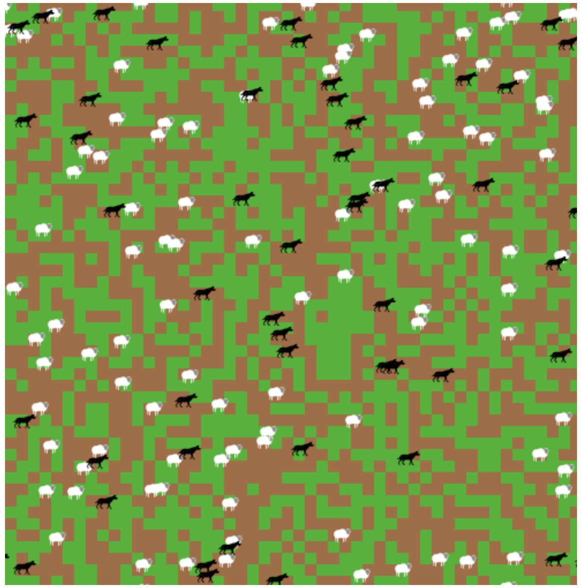
THE UNIVERSITY of EDINBURGH
informatics

Modelling of Systems for Sustainability
INFR10088

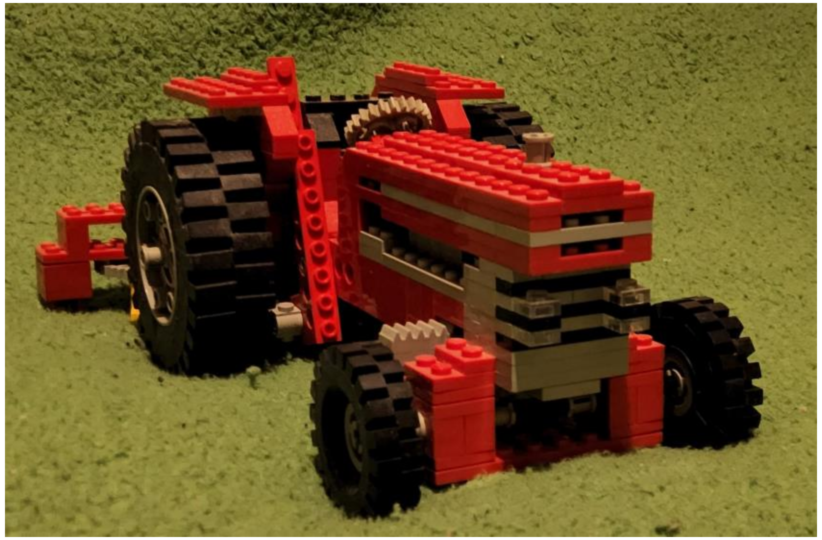
Last time...



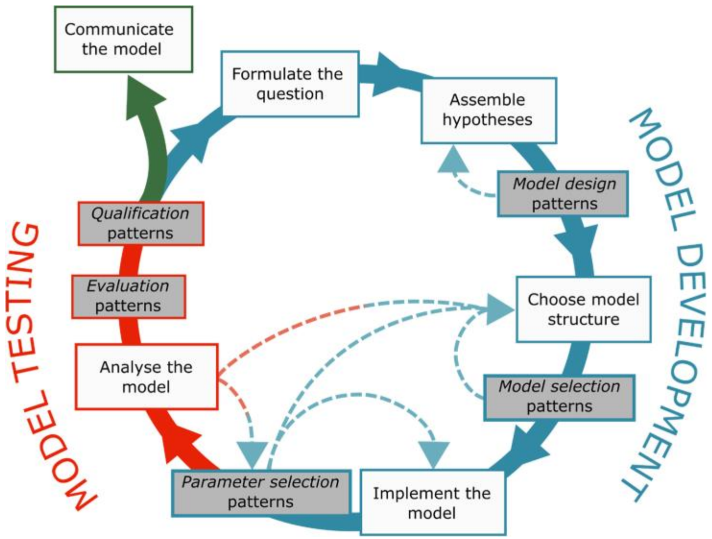
Mushroom hunt example



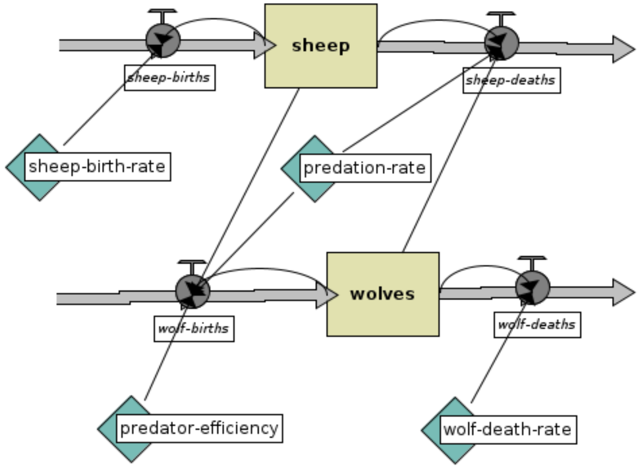
Intro to Agent-based modelling...



What and why of models



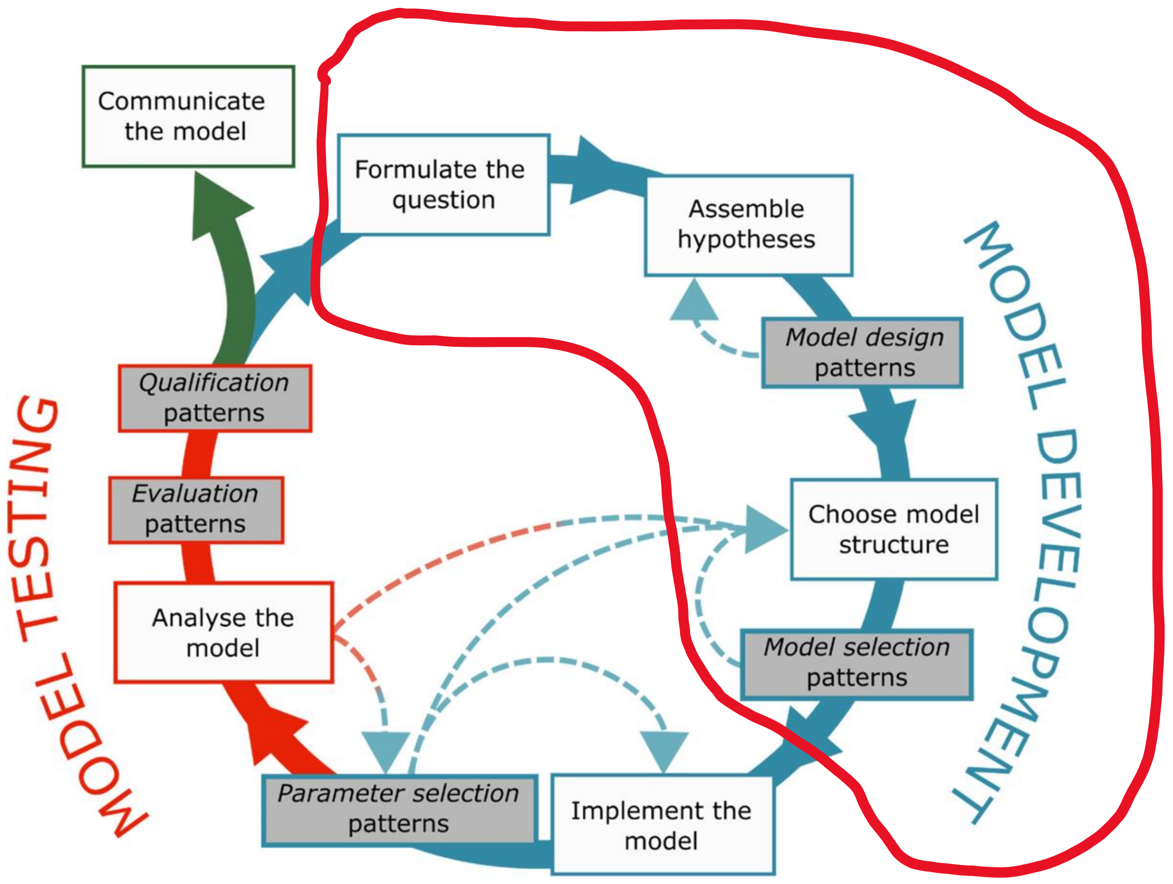
The modelling cycle



...and system dynamics

Today...

- Model formulation
  - *before* implementation
- Why?
  - So we think explicitly about the model
  - Communicate model
  - Basis for implementation
- "Overview, design concepts, details" (ODD) protocol
  - Invented for ABMs
  - Can be adapted for other types of model

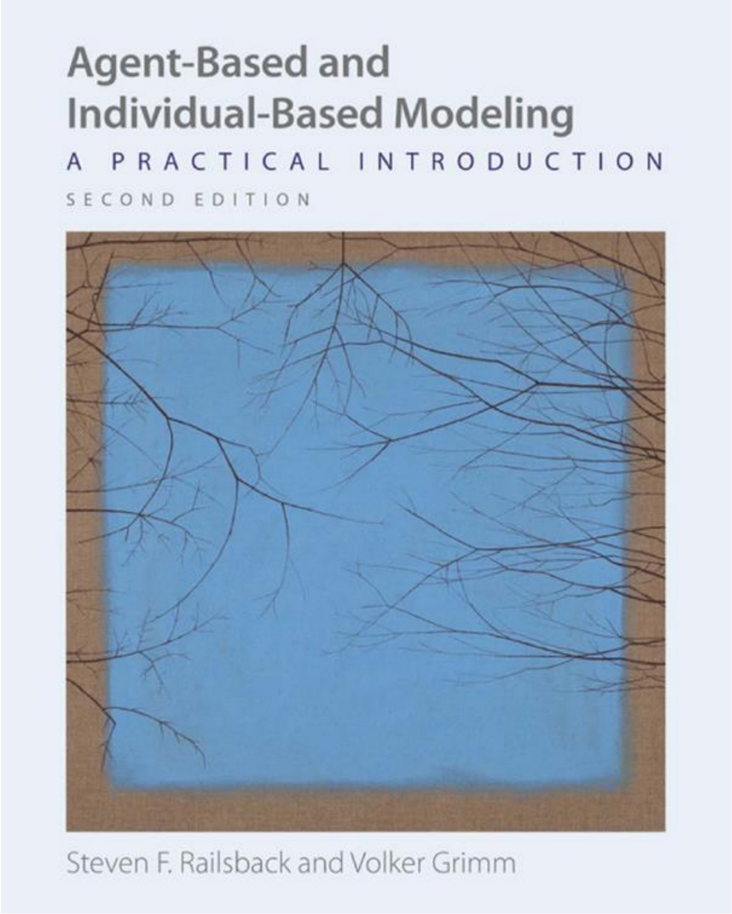


Overview - aims

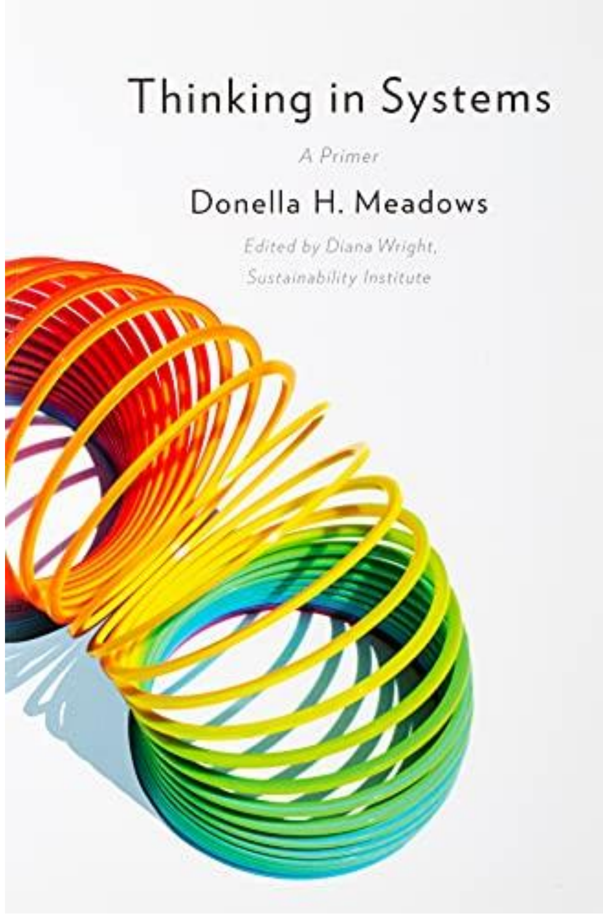
- Overall aim (Course Learning Outcome 2):
  "investigate a sustainability system question, identify system elements and their interactions, and codify a system model using an appropriate model description framework"
- This lecture:
  - Firm understanding of "Overview" and "Details" parts of ODD
  - Introductory understanding of "design concepts"
  - Example ODD description
- Complemented by Thursday's tutorial

Books

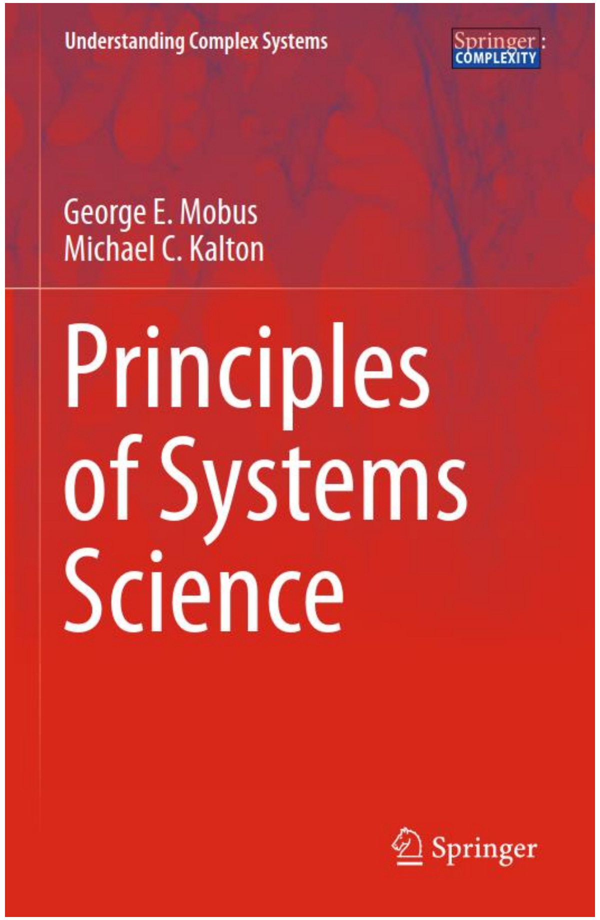
See "Resource List" on course website for links to online versions – chapters to read indicated in Schedule



Railsback and Grimm, Edition 2
For agent-based modelling and ODD
Now available from Library



Meadows
For philosophy of systems thinking and system dynamics approach



Mobus and Kalton
For principles of systems thinking

What is ODD?
Why do we use it?

Origin of ODD

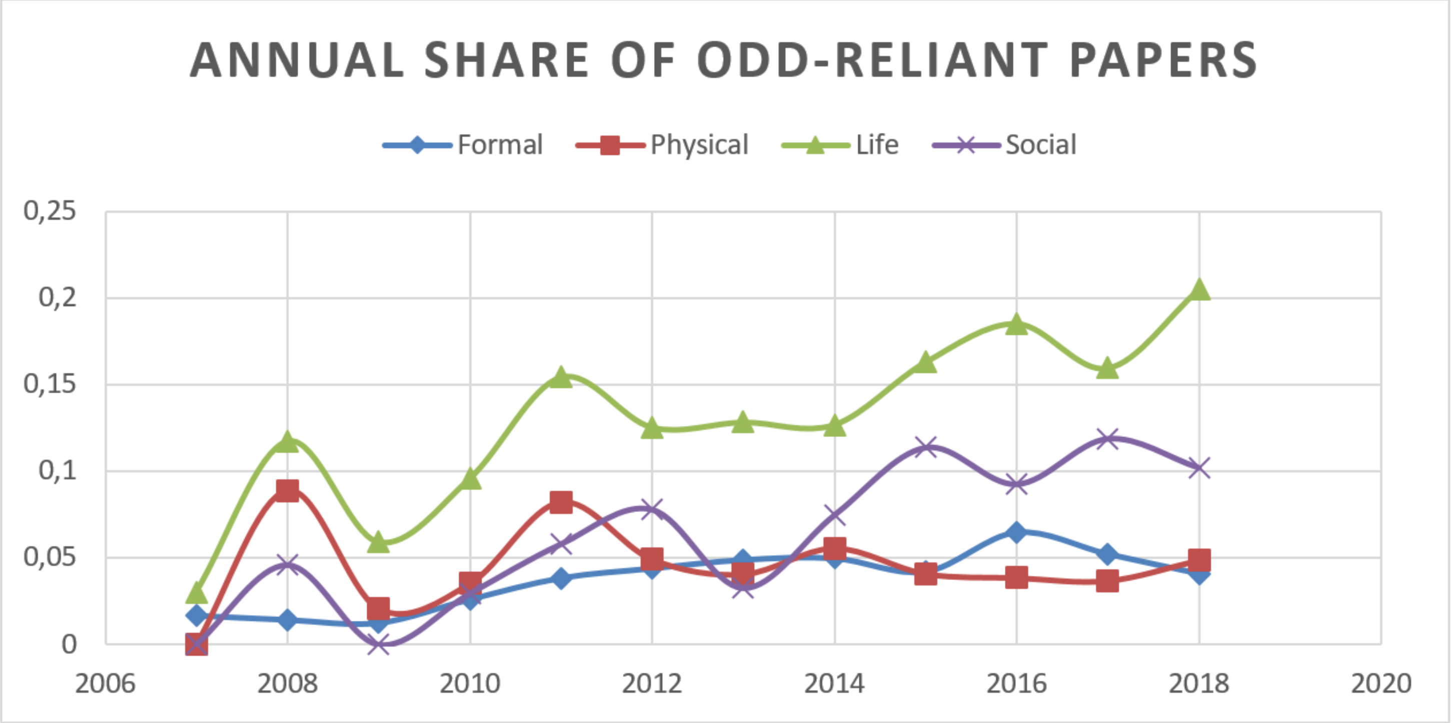
- The "Overview, Design Concepts and Details" (ODD) protocol is a standardised method of describing models
  - developed by the Agent-based modeling (ABM) community in 2006 to describe ABMs
  - Has been used for system dynamics models (Grimm et al. 2020)
  - Revised in light of experience in multiple fields (Grimm et al. 2020)
- Problem of unclear descriptions of models meant that
  - Models could not be replicated
  - And were not communicated clearly
- ODD protocol designed to be:
  - Complete and easy to understand by virtue of...
  - ...an organised description in a consistent order

Current use of ODD

- Now also used to **formulate** agent-based models as well as to **describe** them
  - A way of thinking about agent-based models
  - Analogous to differential equations in mathematical modelling
- Template available from course and book website
- Advantages: Rigorous method of formulating models
- Disadvantages:
  - Can seem a bit long-winded – but a "summary" version is also available
  - Could the code be the model description?

Use of ODD in scientific work

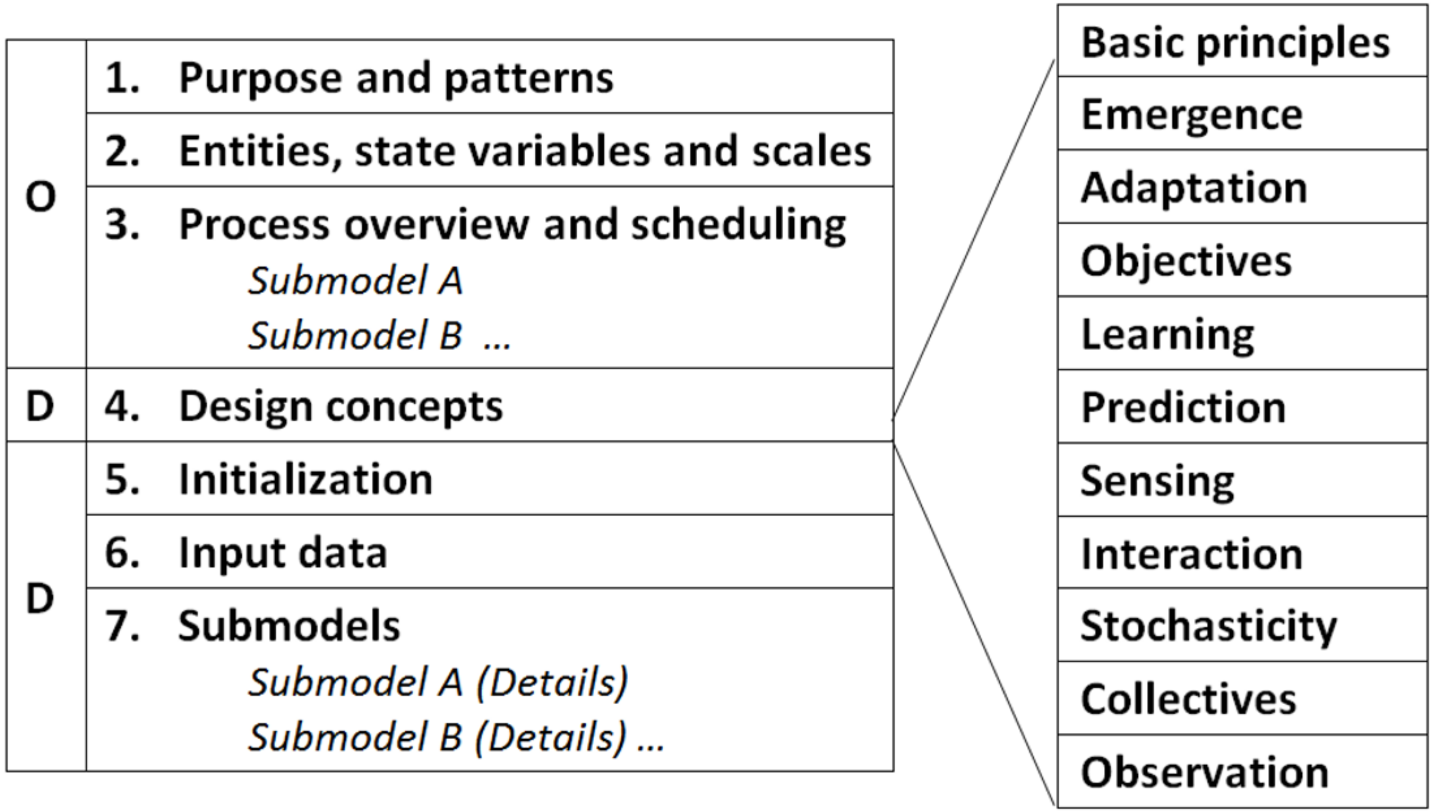
- Widely used by ecologists, and increasingly by social scientists
- Used in ~2000 publications



The ODD protocol

ODD elements

- **O**verview
- **D**esign concepts
- **D**etails



Grimm et al (2020) Journal of Artificial Societies and Social Simulation 23:7. CC BY-4.0

Word template

- Available from book website
- Delete *Questions* and *Explanation* after writing your text.

1. Purpose and patterns

Questions: What is the purpose of the model? What patterns are used to determine how useful the model is for its purpose?

Answer: ...

Explanation: Every model has to start from a clear question, problem, or hypothesis. Therefore, ODD starts with a concise but specific statement of the objective(s) for which the model was developed. Do not describe anything about how the model works here, only what it is to be used for. We encourage authors to use this paragraph independently of any presentation of the purpose in the introduction of their article, since the ODD protocol should be complete and understandable by itself and not only in connection with the whole publication (as it is also the case for figures, tables and their legends). The purpose statement should be specific enough to make it the primary "filter" for determining what should and should not be in the model. If one of the purposes of a model is to expand from basic principles to richer representation of real-world scenarios, this should be stated explicitly.

Making it real - Thursday's tutorial

- Homework: find a paper you're interested in with an ODD
- Tutorial:
  - Discuss an ODD with other students
  - Create your own ODD

1. Overview: Purpose and patterns

- **Purpose**
  - Statement of problem addressed by model
  - E.g. (mushroom hunter) "What search strategy maximises the rate of finding items if they are distributed in clusters?"
  - Not part of model, but important to guide construction of model
  - Important to be specific
- **Patterns**
  - What system we're modelling
  - And what patterns should a model reproduce, or not reproduce
  - Patterns may be quite specific and based on data from the real system

2. Overview: **Entities, state variables** and scales

- **Entities**
  - Agents (e.g. hunter)
  - "Local" Environment (e.g. forest/mushrooms) - "patches" in NetLOGO
  - "Global" environment (e.g. temperature, tax rates)
- **State variables**
  - Properties of an entity, e.g. agent's location, size, memory, speed
  - Some state variables are static, e.g. sex and species, or location of immobile objects – but still vary between agents
  - *Not* things that can be derived from entities, e.g. distance between two agents
  - *Not* a parameter that is constant over time and affects all agents equally, e.g. growth rate for all plants

2. Overview: Entities, state variables and **scales**

- **Temporal scales**
  - How long is simulated? (e.g. Years, minutes)
  - How long is a time step? (e.g. Days, milliseconds)
  - Changes on a timescale shorter than a timestep are effectively summarised
  - Driven by agent behaviour:
    - E.g. stock market model with time step of milliseconds can represent individual trades
    - But with time step of a day could only represent number of trades and mean sell and buy prices
- **Spatial scales (if space part of model)**
  - Spatial extent and size of grid cell (patch)
  - E.g. city 1km across might have 100x100 cells of 10mx10m (size of one building plot)

2. Overview: Entities, state variables and scales

- Mushroom hunt example
  - **Entities**: hunter and mushrooms
  - **State variables:** time hunted; number of mushrooms found; time since finding last mushroom
  - **Spatial scales:** Space represented by squares the size that the hunter can search in one time step.
    - Perhaps 1m x 1m patches, and 100m x 100m environment
  - **Temporal scales:** each timestep might be of the order of seconds

3. Overview: **Process overview** and scheduling

- Almost every **process** describes behaviour/dynamics of model entities
  - What are entities doing?
  - What changes happen in the environment in response to them?
  - If a process is complicated, it may get its own "submodel" description, which can be detailed in the final "Details" section of the ODD
- **"Observer" processes** do not describe behaviour/dynamics of model entities
  - Instead record statistics of system, e.g. number of mushrooms found

3. Overview: Process overview and **scheduling**

- Scheduling:
  - Which model entities
  - Execute which processes
  - In what order
- E.g. ask hunters (**entities**) to move (**process**), with hunters making their moves in a random **order**
  - This is asynchronous update – synchronous update or the same order on each time step is also possible
- **Pseudocode** can be useful for describing scheduling

4. Design concepts

- Template contains questions and explanations for the 9 design concepts – we will return to some of them.

5. Details: Initialisation

- Initial conditions of model, i.e. values of state variables at t=0
- May be chosen randomly
- May want to choose initial conditions to model a situation
  - e.g. what would happen if all hunters start far from the forest

6. Details: Input data

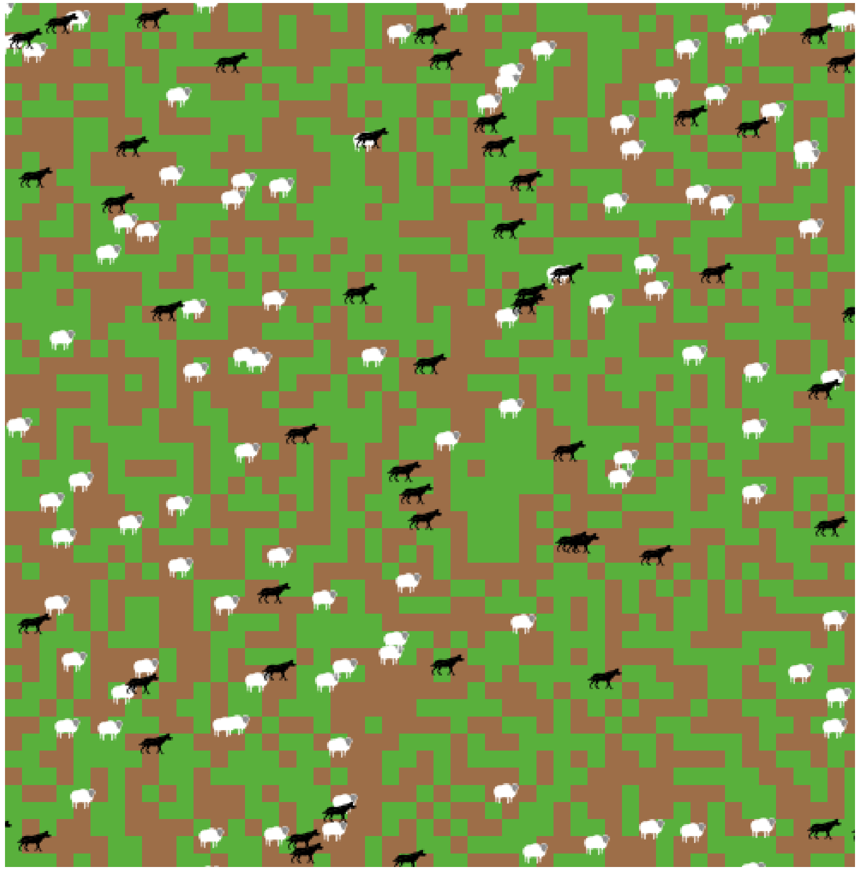
- Environmental variables like temperature or the stock market index
- May be read in from data files, or simulated
- Not affected by the model behaviour

7. Details: Submodels

- Details of the submodels listed in the process overview and scheduling section
- All equations, logical rules and algorithms need to be described
- Parameter meaning, units, default value, range of possible values
- Reasoning for parameters – e.g. justification from the literature

Exercise

- Identify subsection of Overview and Details of the Grass-Sheep-Wolves simulation



O	1. Purpose and patterns
	2. Entities, state variables and scales
	3. Process overview and scheduling *Submodel A* *Submodel B*
D	4. Design concepts
D	5. Initialization
	6. Input data
	7. Submodels *Submodel A (Details)* *Submodel B (Details) …*

Basic principles
Emergence
Adaptation
Objectives
Learning
Prediction
Sensing
Interaction
Stochasticity
Collectives
Observation

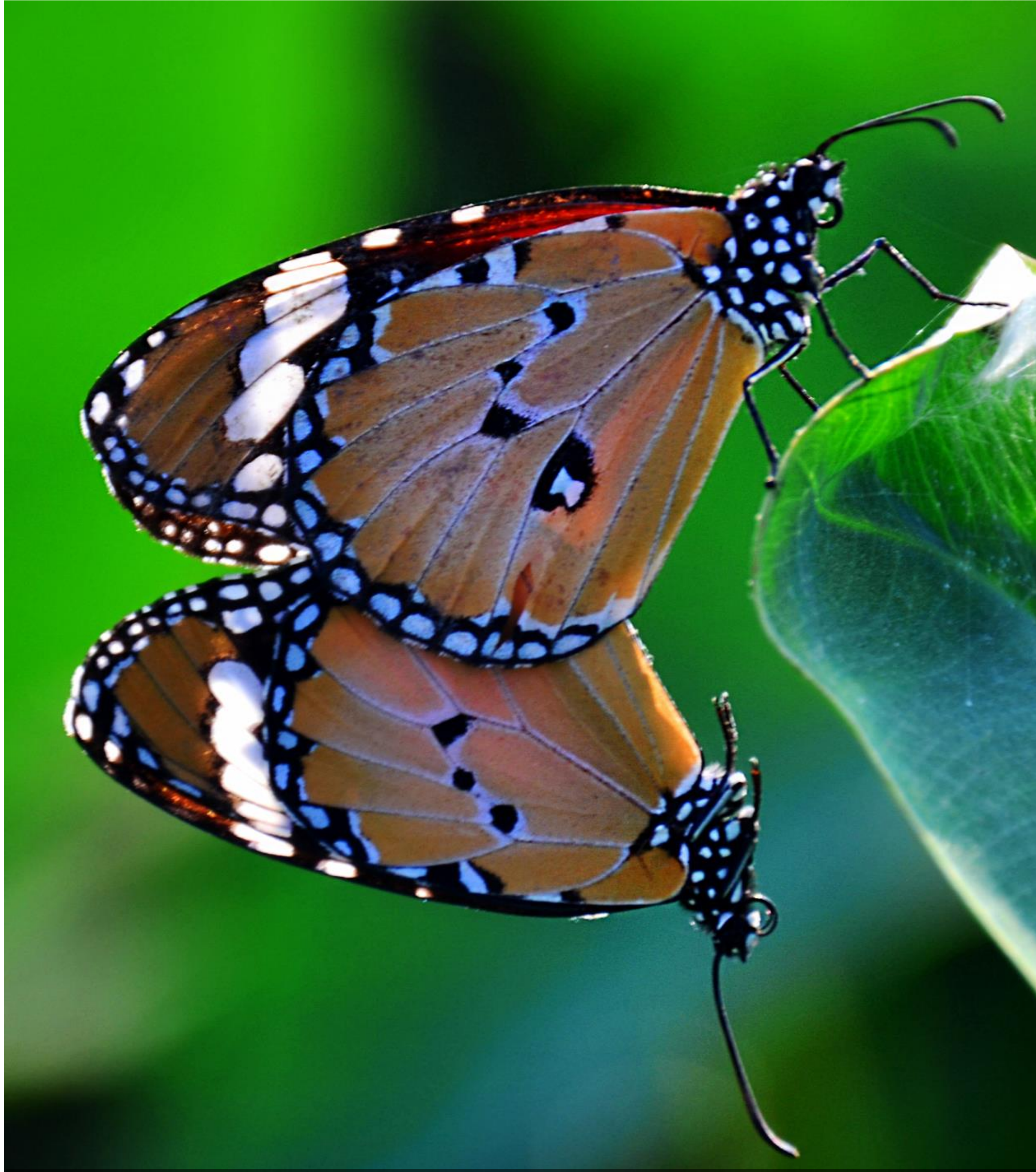
Example: Virtual corridors of butterflies

Wildlife corridors

- Linear elements in the landscape that facilitate movement of animals
  - Obvious to humans
- Could "virtual corridors" exist based on landscape features not obvious to the human eye?

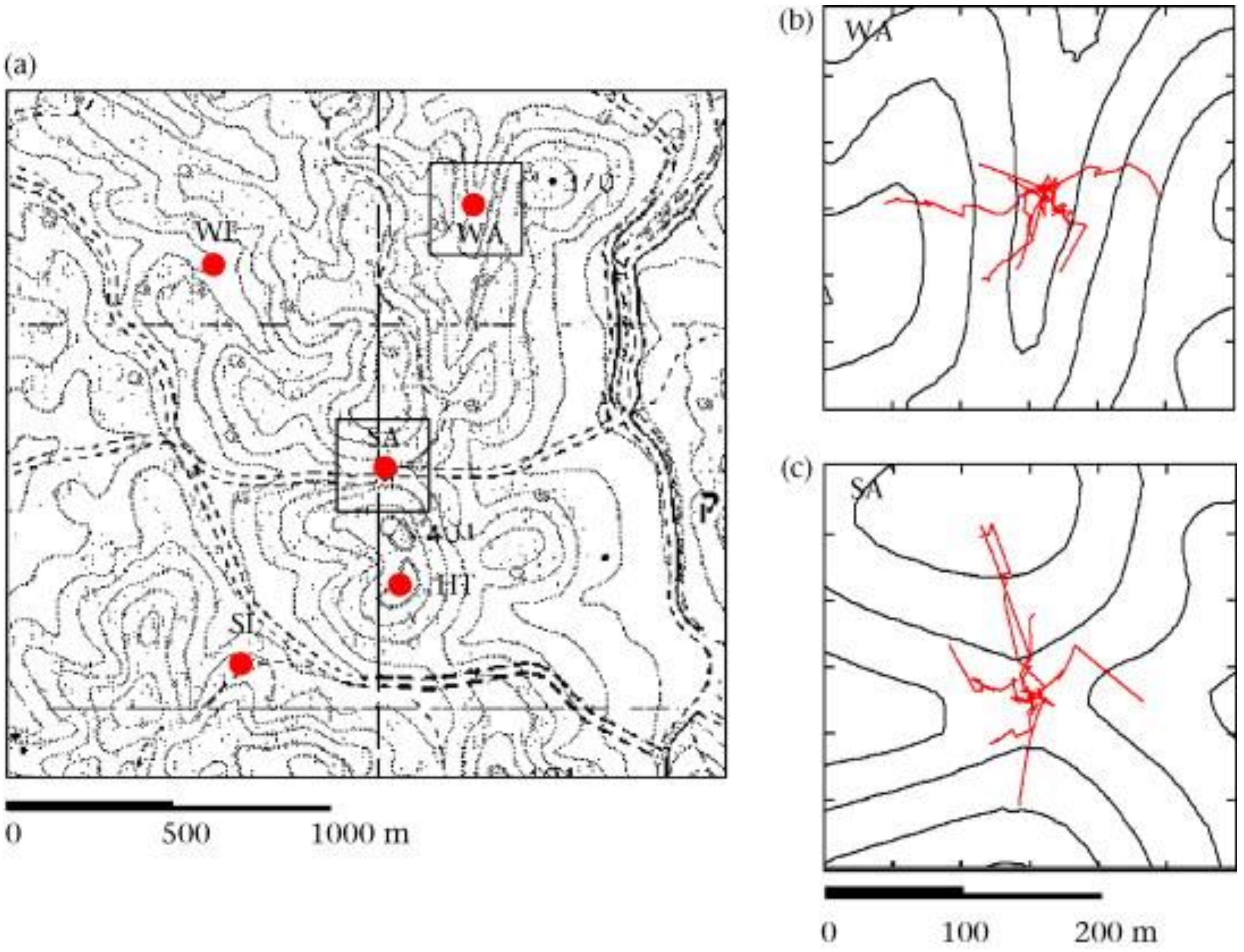
Hilltopping butterflies

- Hilltoping strategy: Male butterflies fly to top of hills and await females
  - Hilltops are good rendevous points



Butterfly virtual corridors

- Measured in the field

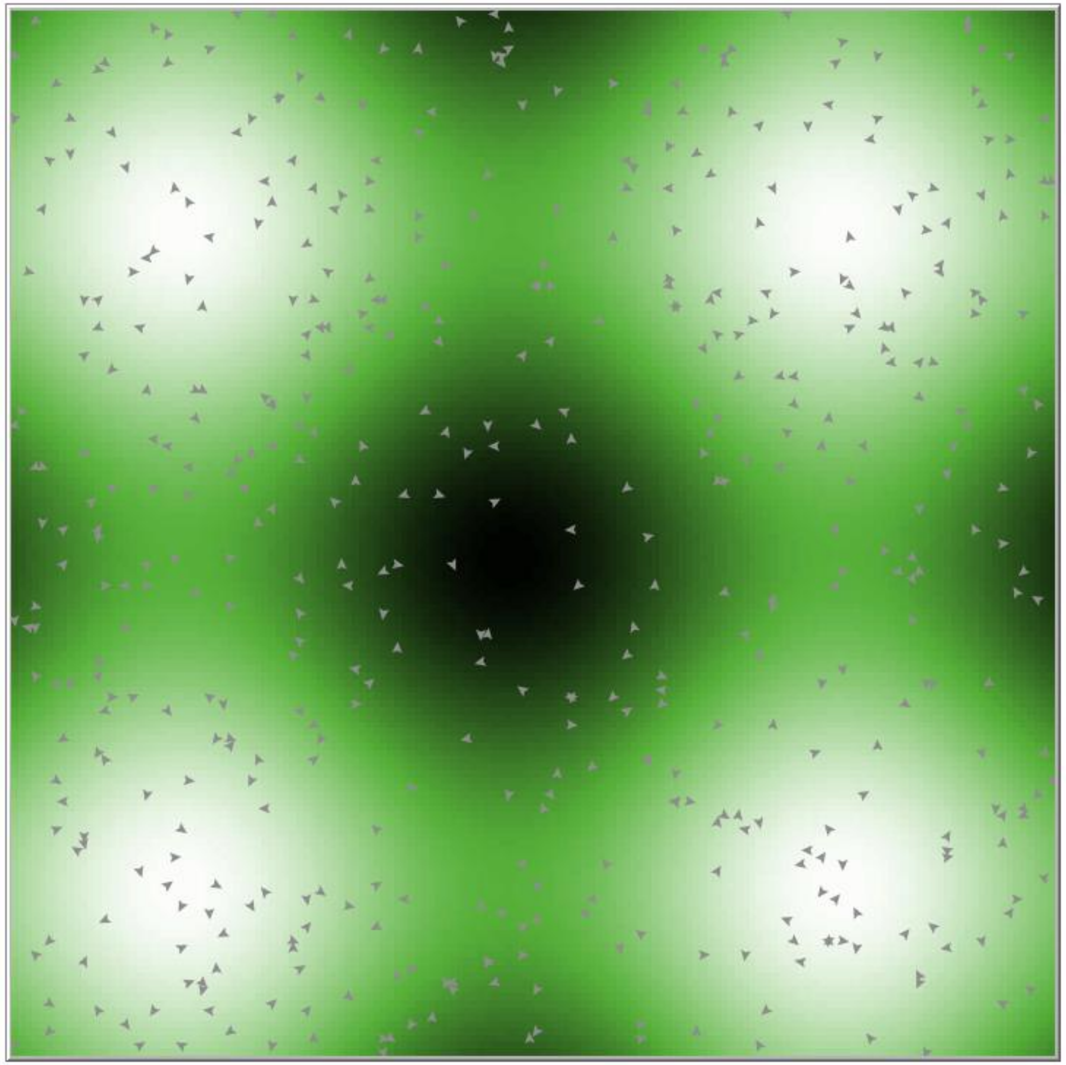


1. Purpose and patterns

- **Purpose:** Investigate questions:
  - Under what conditions do the interactions between butterfly hilltopping behaviour and the topography of the landscape lead to virtual corridors?
  - How does variability in butterfly movement affect the emergence of virtual corridors?
- **Patterns:**
  - Butterflies reach hilltops
  - Movement is stochastic

2. Entities, state variables, and scales

- Entities:
  - Butterfly agents
  - Patches of land
- State variables
  - Butterly location (x, y coordinates)
  - Patch elevation
- Scales
  - Spatial: Each patch is 25m x 25m; 150 x 150 patches in grid
  - Temporal: 1000 time steps. Each time step represents time butterfly takes to travel 25-35m (i.e. to next patch)

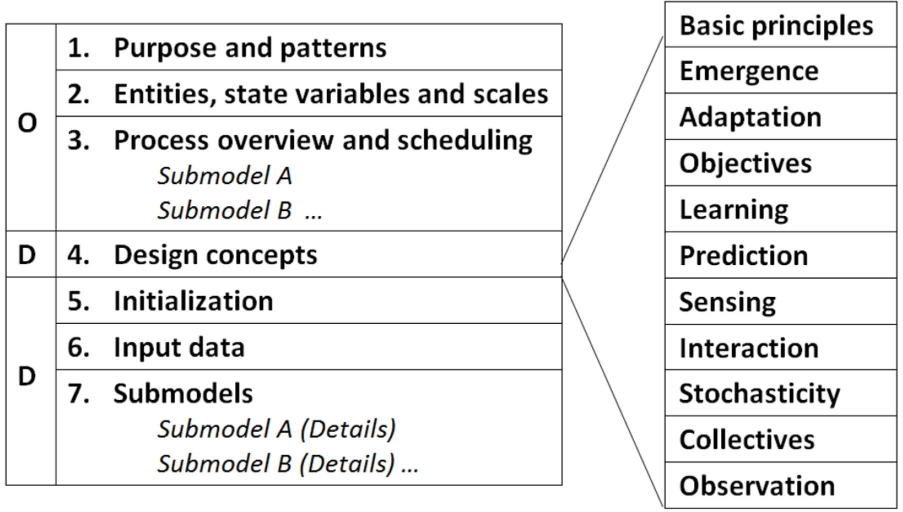


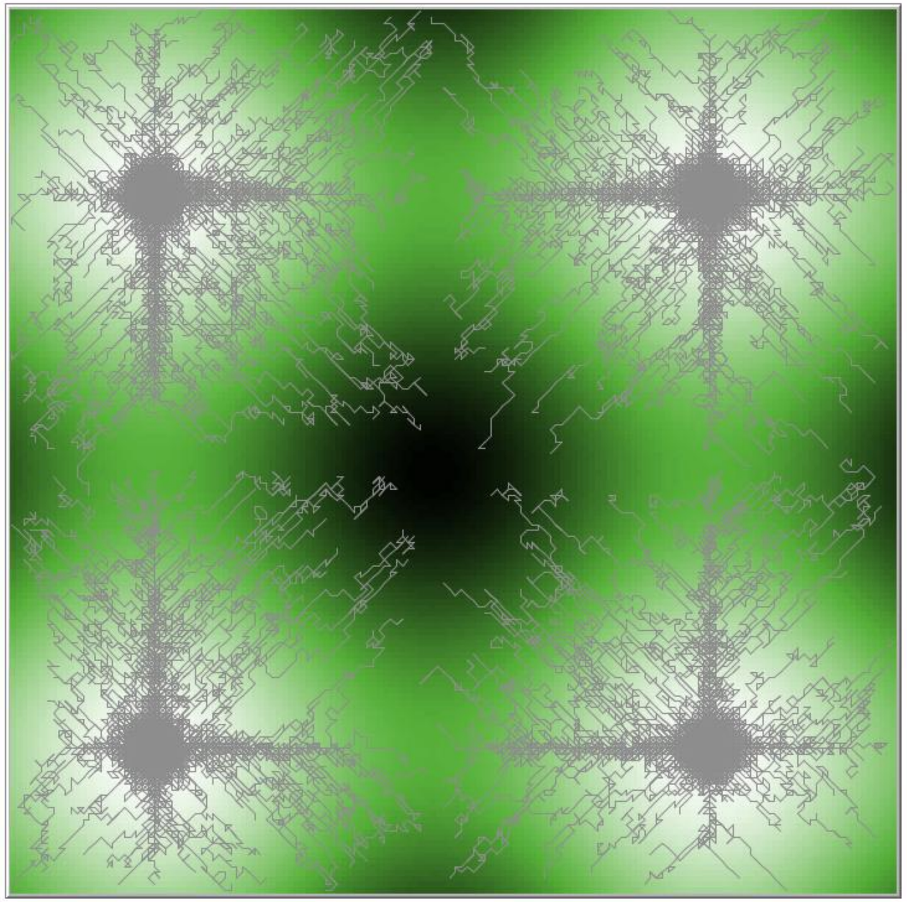
3. Process overview and scheduling

- One process: butterfly movement
- On each time step each butterfly moves once – details are in the "Submodels" section
- Order in which butterflies move does not matter, since there are no interactions between butterflies

4. Design concepts

- **Basic principle:** Virtual corridors
- Corridors **emerge** from **adaptive behaviour** of moving uphill
- The **objective** of hilltopping and the associated **prediction** that butterflies will be present are assumed. No **learning.**
- **Sensing** of uphill direction is important and **interaction** of butterflies is ignored, even though it is observed in field studies (Pe'er 2003)



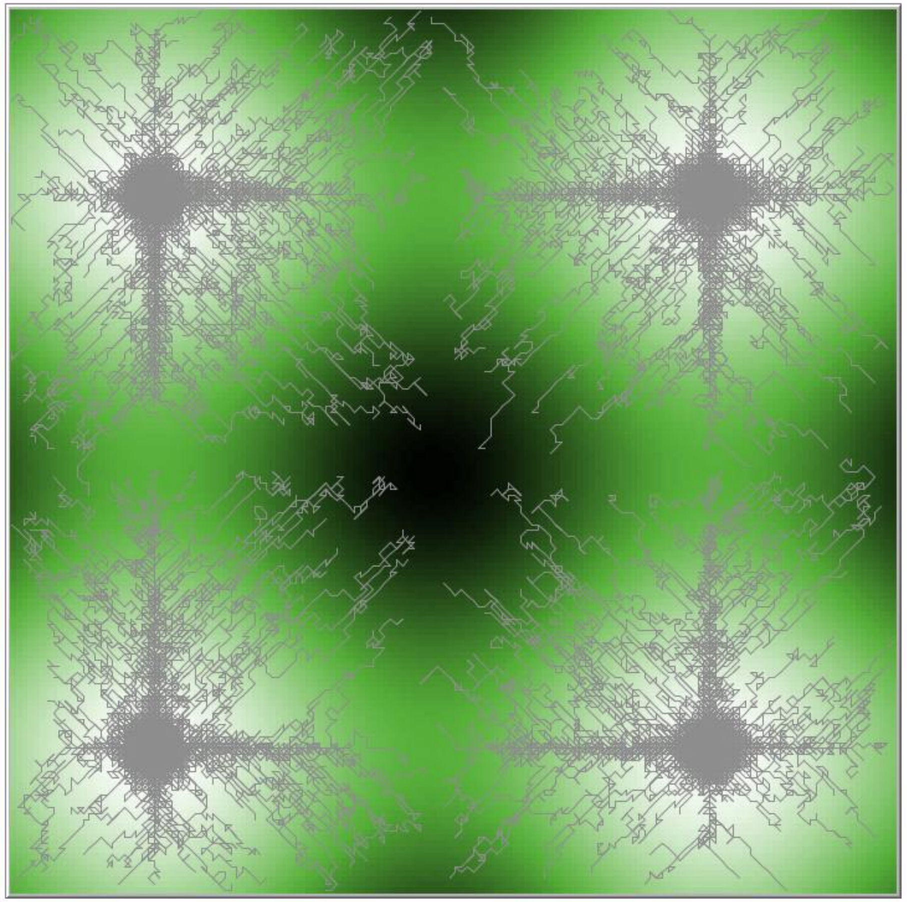


4. Design concepts (ctd)

- **Stochasticity** (quantified by **parameter**) used to represent complex sources of variability:
  - Limits in sensing ability
  - Factors other than topography, e.g. Flowers
- **Observation** of corridors visually
  - Will be desirable to introduce a "corridor width" **measure** for quantification

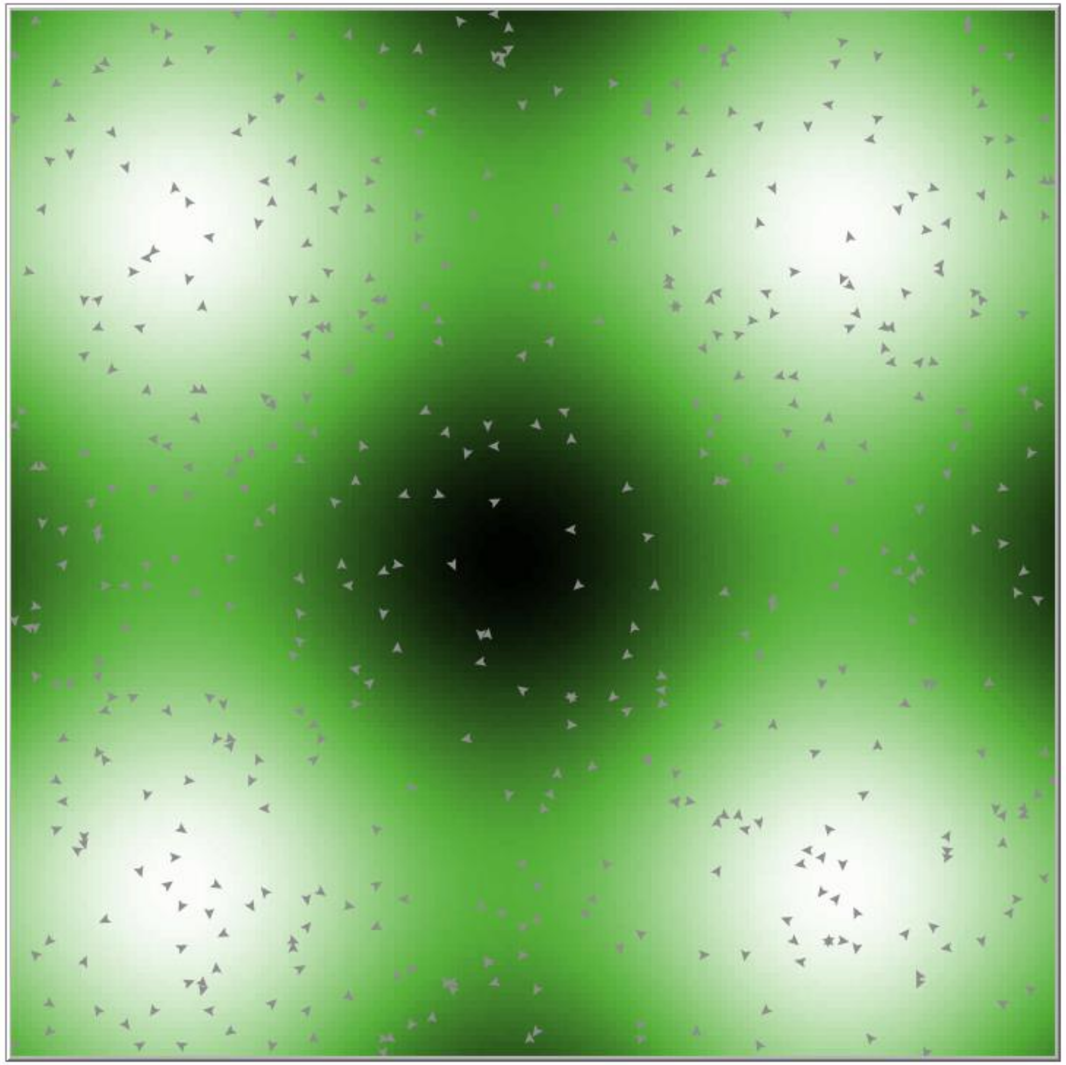
O	**1. Purpose and patterns**
	2. Entities, state variables and scales
	3. Process overview and scheduling *Submodel A* *Submodel B ...*
D	**4. Design concepts**
D	**5. Initialization**
	6. Input data
	7. Submodels *Submodel A (Details)* *Submodel B (Details) ...*

Basic principles
Emergence
Adaptation
Objectives
Learning
Prediction
Sensing
Interaction
Stochasticity
Collectives
Observation



5. Initialisation

- Environment. Patch elevation from either
  - Artificial topography
  - Real landscape
- Agents
  - 500, placed randomly



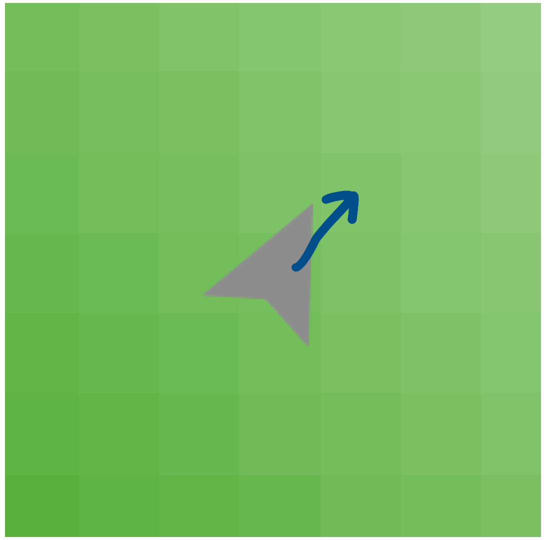
6. Input data

- Environment constant, so no input data

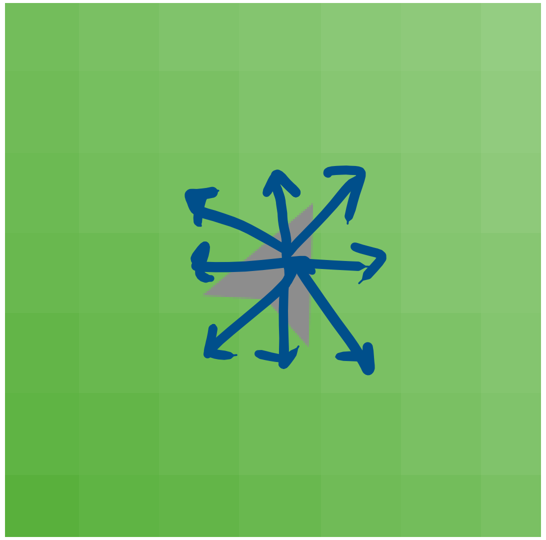
7. Submodels

- Submodel A: "Move"
  - Randomly decide if to move uphill with probability *q* or to move randomly with probability
    1-*q*
    - If uphill, move to neighbour with highest elevation
    - If random, move to random neighbour
  - *q* is a global **parameter**, the same for all agents at all times

Uphill



Random



Summary

- Rationale for the ODD protocol
  - Describing ABMs … but can be adapted to other model types
  - Rigorous formulation of models
- The ODD protocol
  - Generic and hierarchical
- Example application
- Next:
  - Lecture (Wednesday): Agent-Based Modelling framework and examples
  - Tutorial (Thursday): Reading and writing ODD descriptions
  - Lab (Thursday): Getting familiar with NetLOGO
  - Lab (Next Thursday): Butterfly hilltopping model