Designing an Agent-Based Model (ABM) - Summary

Introduction

Agent-based modeling (ABM) is a powerful simulation technique used to model complex systems composed of interacting agents.

Topics of Discussion

Various aspects essential to designing and implementing ABMs:

- Model Design
- Model Implementation
- Calibration and Validation
- Running Experiments

Model Design

Conceptual Modeling

1. **Pattern-Oriented Modeling (POM)**
   - **Hierarchical Structure**: POM models systems across multiple scales and levels.
   - **Multi-Criteria Design**: Used for design, selection, calibration, and validation of models.
   - **Pattern Identification**: Identify patterns from empirical observations, literature, or other models.
   - **Model Structure**: Define a model structure using only necessary components to achieve the model's purpose.
2. **Hypothesis Testing in POM**
   - Identify and implement alternative submodels.
   - Compare and contrast these alternatives.
   - Iteratively refine the model to match observed patterns.

Participatory Modeling

1. **Cognitive Mapping**
   - Visual representation of mental models showing relationships between concepts.
   - Relationships are often binary (positive or negative associations).
2. **Fuzzy Cognitive Mapping (FCM)**
   - Extends cognitive mapping with fuzzy logic to represent varying degrees of association.
   - Relationships are assigned fuzzy values between 0 and 1.

3. **Rule-Based FCM**
    - Integrates rule-based reasoning to enhance modeling capabilities.
    - Uses if-then statements to define system behavior based on fuzzy relationships.

Model Implementation

1. **Start Simple, Add Complexity**
    - Begin with the simplest version of the model and gradually increase complexity to reach the desired level of detail (Medawar Zone).
2. **Components of a Complex System**
    - Hierarchical and modular design with submodels (e.g., mobility model, synthetic population, fire/smoke diffusion model, evacuation model).
3. **Practical Implementation Examples**
    - Use of procedures and primitives in NetLogo for defining agent behaviors.
    - Different types of variables (global and local) and agents (breeds).

Calibration and Validation

- **Calibration**: Adjust model parameters to ensure the model behaves as expected based on real-world data.
- **Validation**: Ensure the model accurately represents the real-world system it is intended to simulate.

Running Experiments

- Iteratively test and refine the model.
- Use the model to conduct experiments and explore different scenarios.

Example: Mushroom Search in Forest

- **Concept**: Model of human behavior in searching for mushrooms.
- **Behavior**: Agents scan the neighborhood and switch to detailed search upon identifying mushrooms.

Example: Wolf-Sheep Predation Model

- **Components**: Agents (wolves, sheep), environment (grass patches).
- **Interactions**: Predation (wolves eat sheep), grazing (sheep eat grass).
- **Emergent Behavior**: Patterns and structures arising from agent interactions.

Practical Exercises

- Engage in team-based learning activities using NetLogo models.
- Explore model dynamics, feedback loops, non-linear behavior, and emergent properties.

Conclusion

Designing an ABM involves a structured approach from conceptual modeling to practical implementation, calibration, and experimentation. Pattern-oriented and participatory modeling techniques, along with tools like NetLogo, provide a robust framework for developing and analyzing complex systems.

Designing an ABM

Multiple approaches

Groups (4)

Group 1 — 3 / 4 students

- Francisco Alves Camello Ne...
- Ahmed Hanif
- Otobong Nse

Group 2 — Full — 4 / 4 students

- Pitchaporn Likitpanjamanon
- Surya Naganathan
- Jay Pandya
- Helina Shrestha

Group 3 — Full — 4 / 4 students

- Lana Annisa Rizqilana
- Irene Irene Caroline Sihom...
- Kiki Winda Kiki Winda Ver...
- William William Arthurius

Group 4 — 3 / 4 students

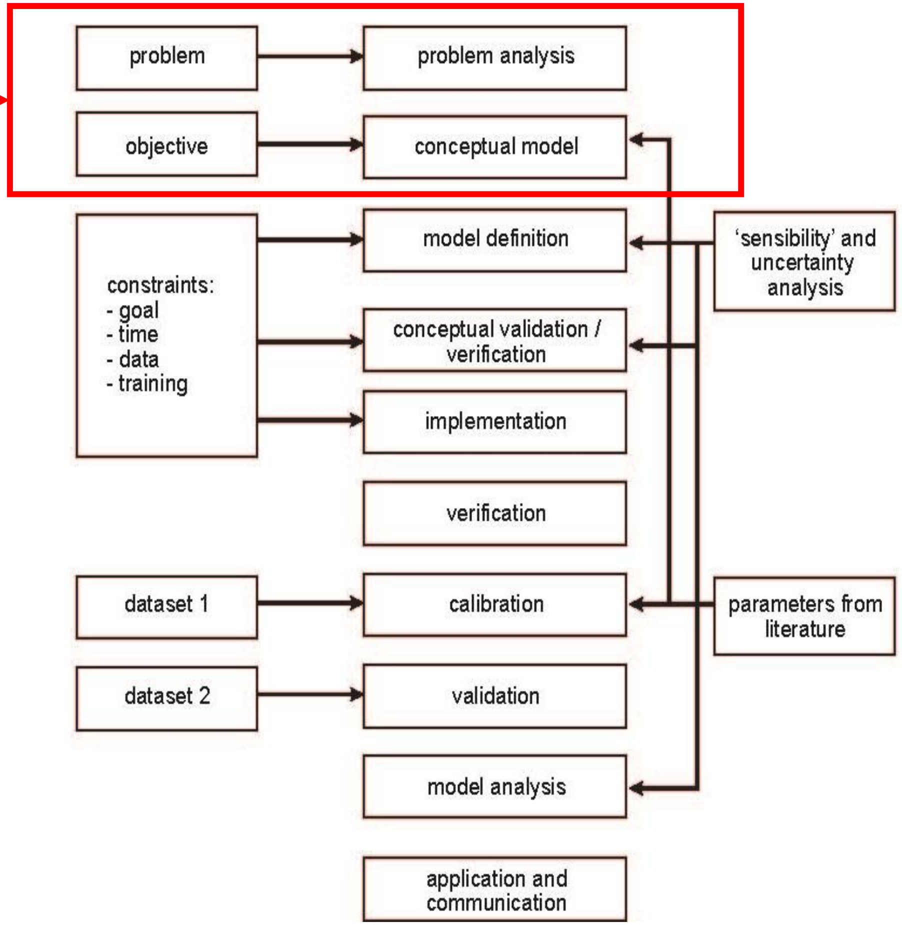
- Monalaine Bermoy
- Rezaul Bhuiyan
- Thierry Tuyizere

Individual projects:

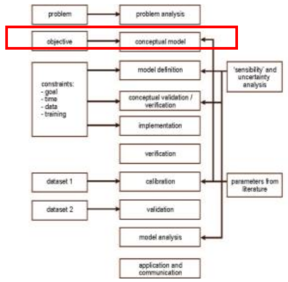
- Amira Al Balushi
- Jazib Imran

TOPICS OF TODAY

- Model Design
- Model Implementation
- Calibration/Validation
- Running Experiments



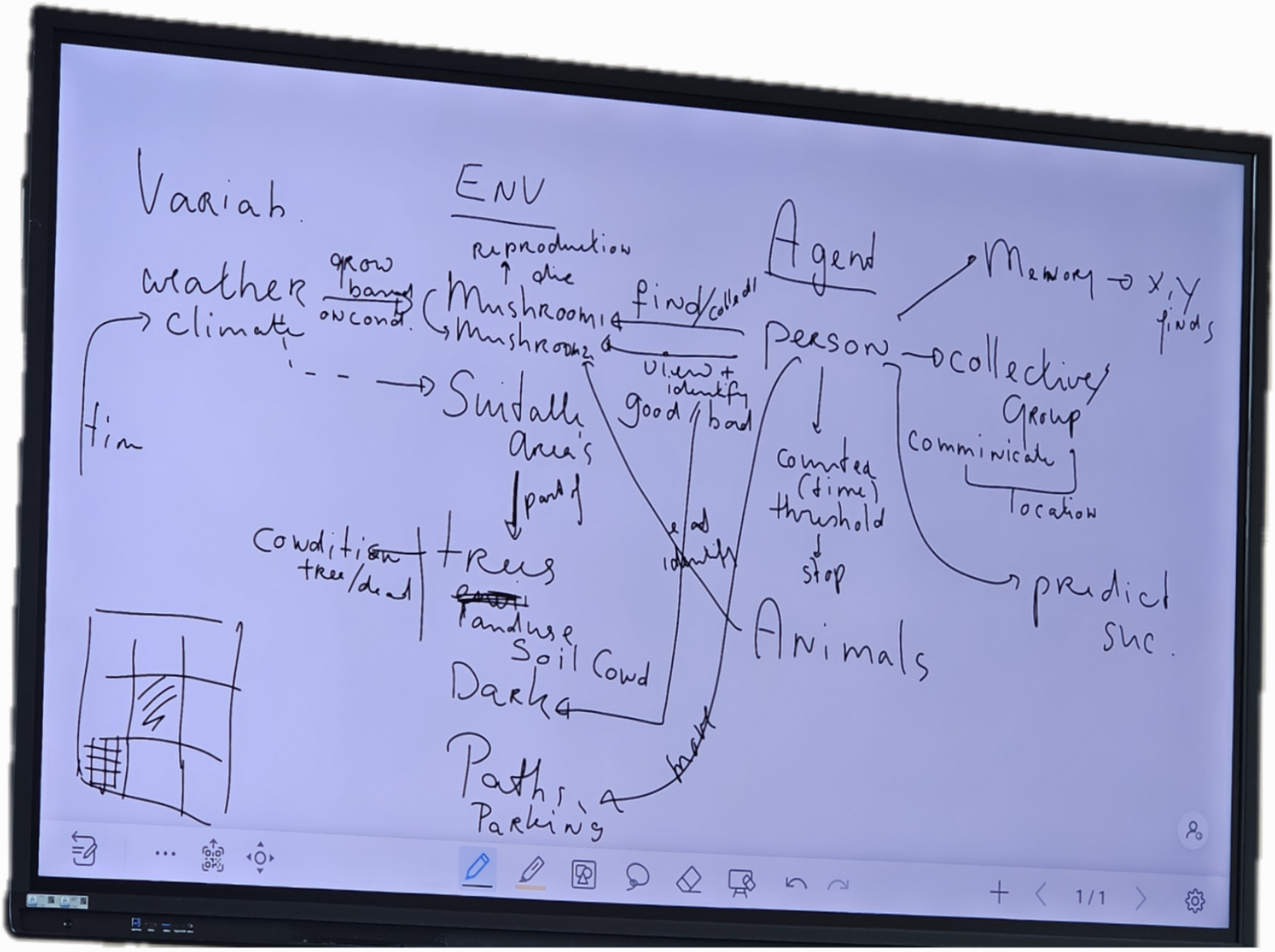
MUSHROOM SEARCH IN FOREST



A video of a conceptual modeling session is uploaded in Canvas

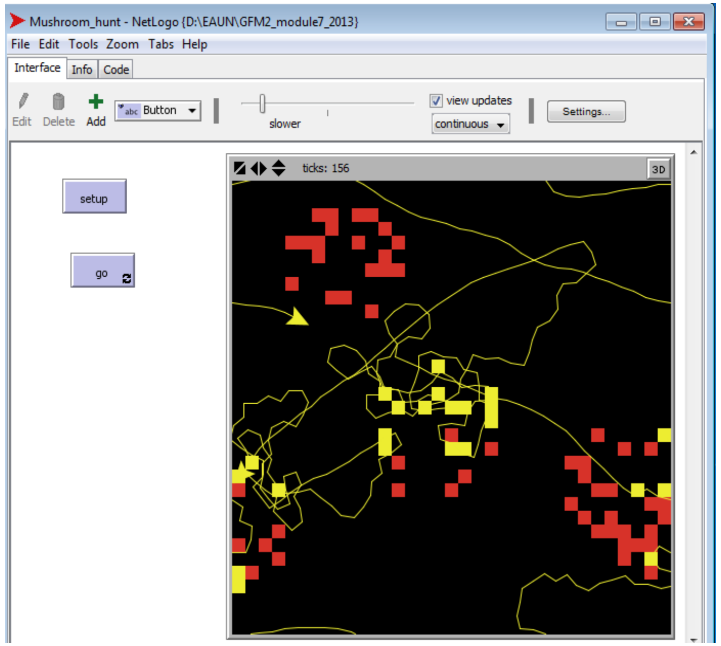
ABM elements for mushroom search

- Agent : hunter / people
- environment : mushroom
- time : tick



MUSHROOM SEARCH IN FOREST

- We need a model (of human behavior)
- How do you find mushrooms?



- Mushrooms grow in clusters
- The agent will scan the neighborhood
- When a mushroom is identified, the agent will switch to smaller-scale movement and more detailed search

→ agent has memory
(they change finding behavior when they find mushroom vs. not finding mushroom)

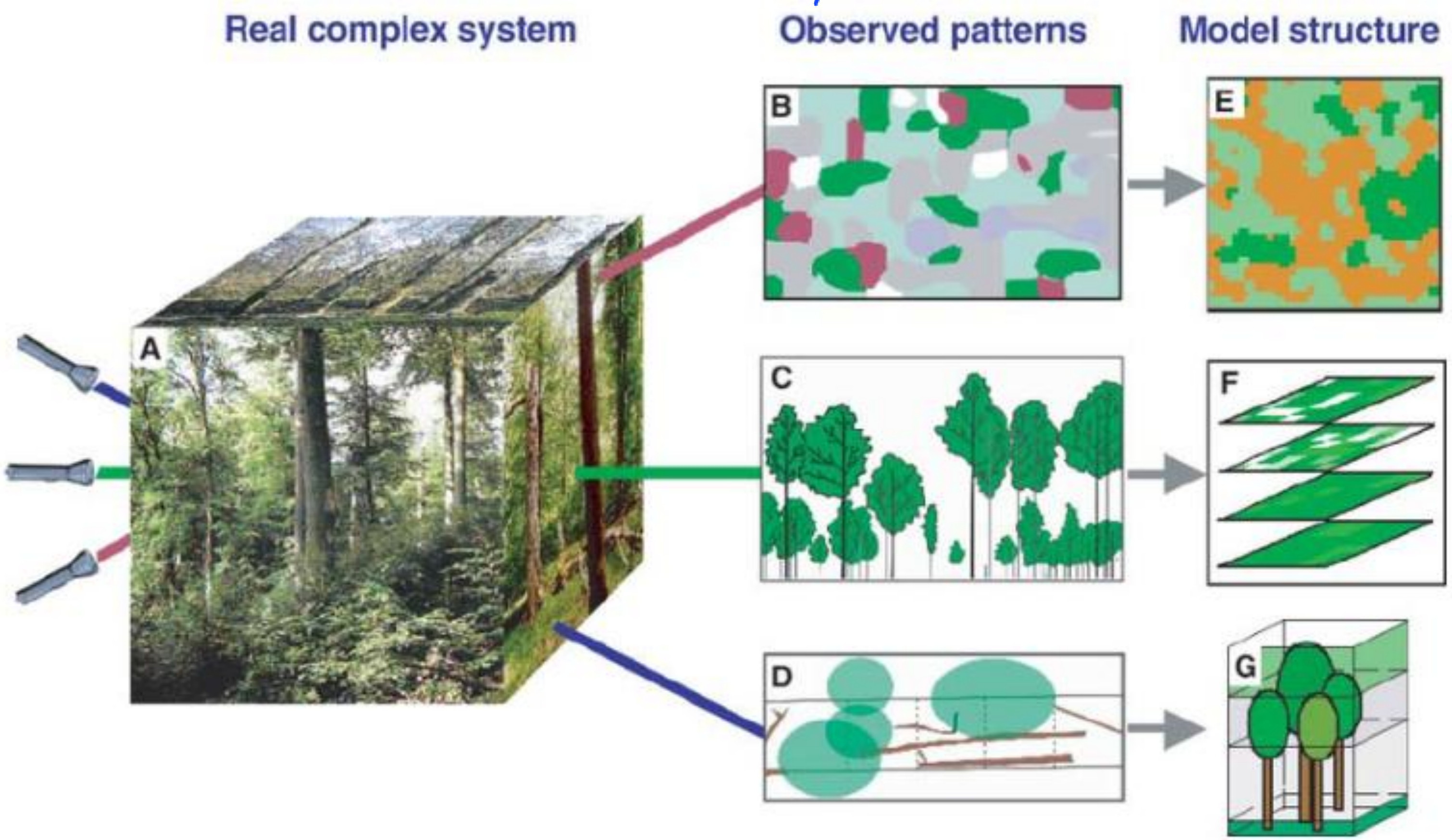
Methods to design an ABM

- Pattern-Oriented Modelling (POM)
- Participatory Modelling: FCM
- Start simple and add complexity
- Build toward a Question: do not add content that does not help to answer this question.

Can be pattern in temporal / spatial ?

Pattern Oriented modelling

for the whole system, sub components



Pattern-Oriented Modeling of Agent-Based Complex Systems: Lessons from Ecology. Source: Grimm, 2005.

Pattern Oriented Modelling (POM)

- Complex Systems consist of hierarchies – POM promotes modelling systems across scales and at multiple levels.
- POM can be used as a strategy for:
  - Multi-criteria design
  - Selection and calibration/validation of models
- POM starts with identifying a set of patterns at multiple scales
- Patterns are used to:
  - Determine what scales, entities, variables and processes the model needs
  - Test and select sub models
  - Find parameter values during calibration

Three elements of POM

eg. flocking of bird (try run flocking model in NetLogo)

- Patterns for model structure
  - Define a structure of your model verbally, using the model purpose as a filter: only use what is necessary to achieve the purpose
  - Identify the patterns that characterize the system. Sources are empirical observations, literature, other models.
  - Define criteria for deciding whether you reproduced the patterns
  - Revise the model structure (iterate over steps 1-3)

Hypothesis in POM

- Identify alternative submodels that implement alternative hypothesis
- Implement the submodels
- Contract the alternatives
- Repeat until a submodel has been found that reproduces all patterns

can combine two hypothesis in one model

Methods to design an ABM

- Pattern-Oriented Modelling (POM)
- Participatory Modelling: FCM
- Start simple and add complexity
- Build toward a Question: do not add content that does not help to answer this question.

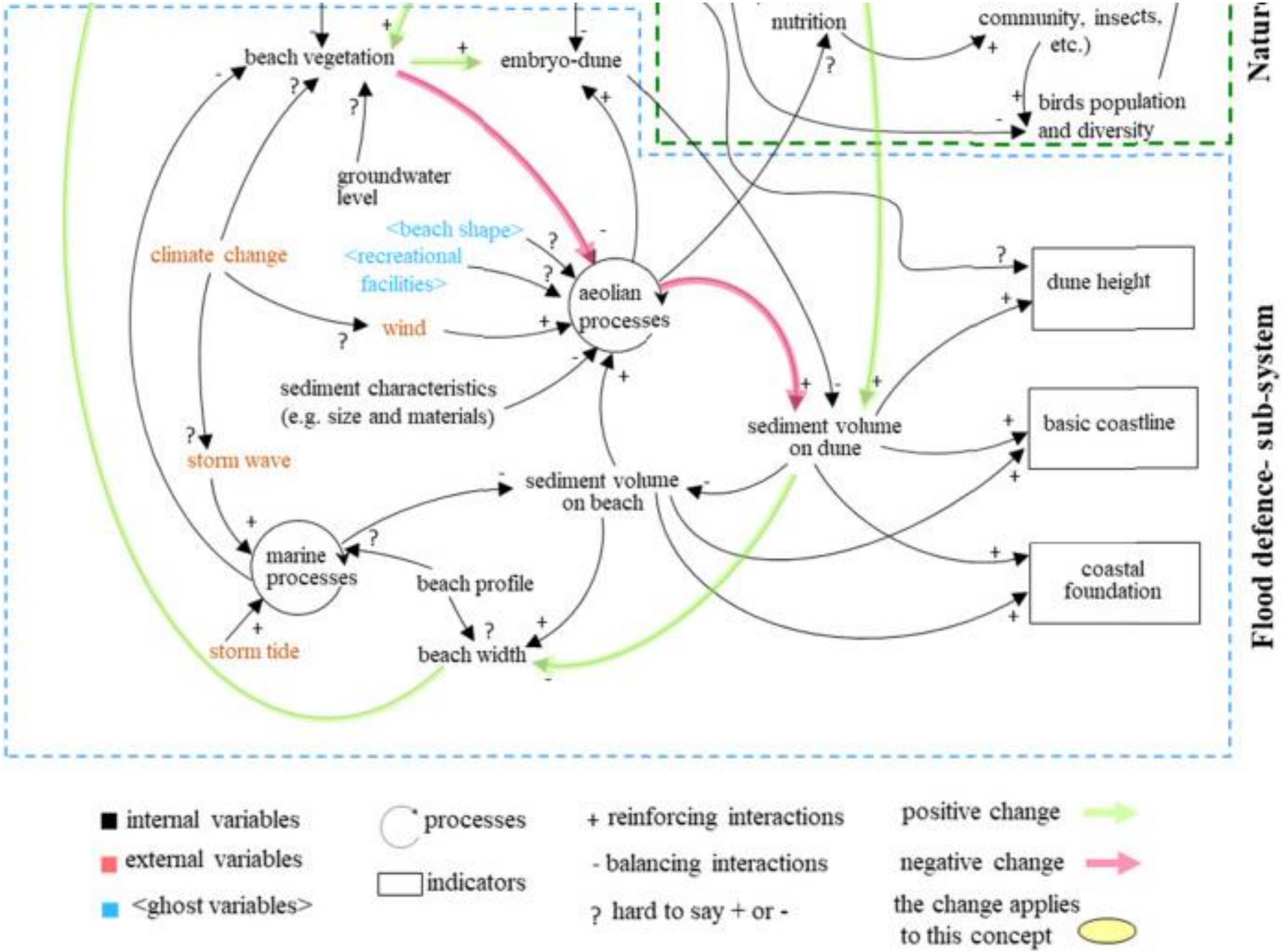
sth. positively / negatively impact sth.

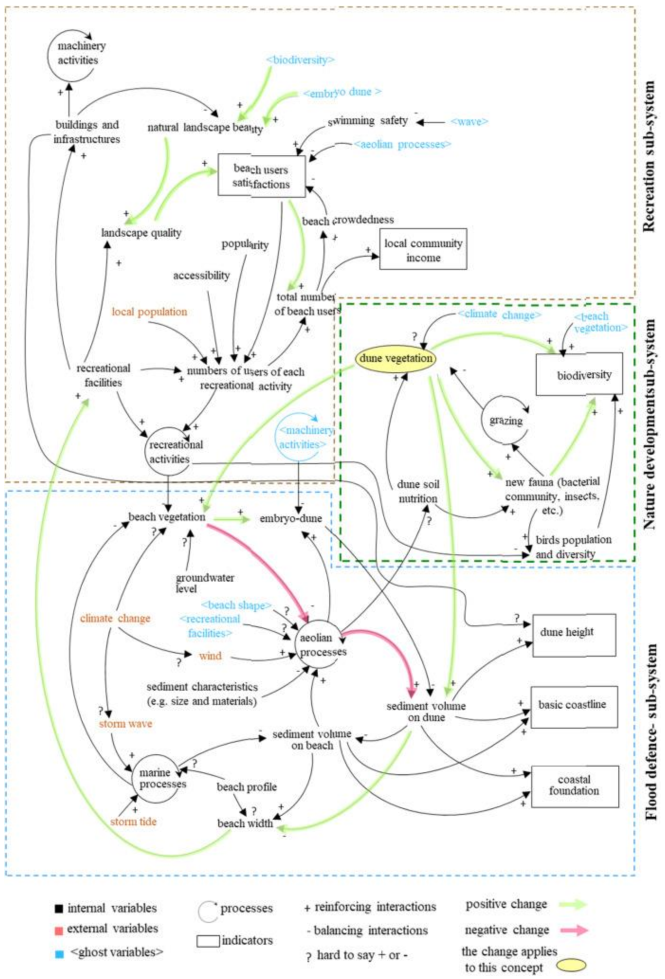
Cognitive mapping

relationship bet. elements

- Cognitive mapping is a mapping method used to create a **visual representation of a person's (or a group's) mental model** for a process or concept.
- The **relationships** in cognitive maps are often binary, indicating whether there is a **positive or negative association** between two concepts. These associations are typically represented using arrows or lines between nodes, with the direction of the arrow indicating the direction of influence or causality.

Cognitive map

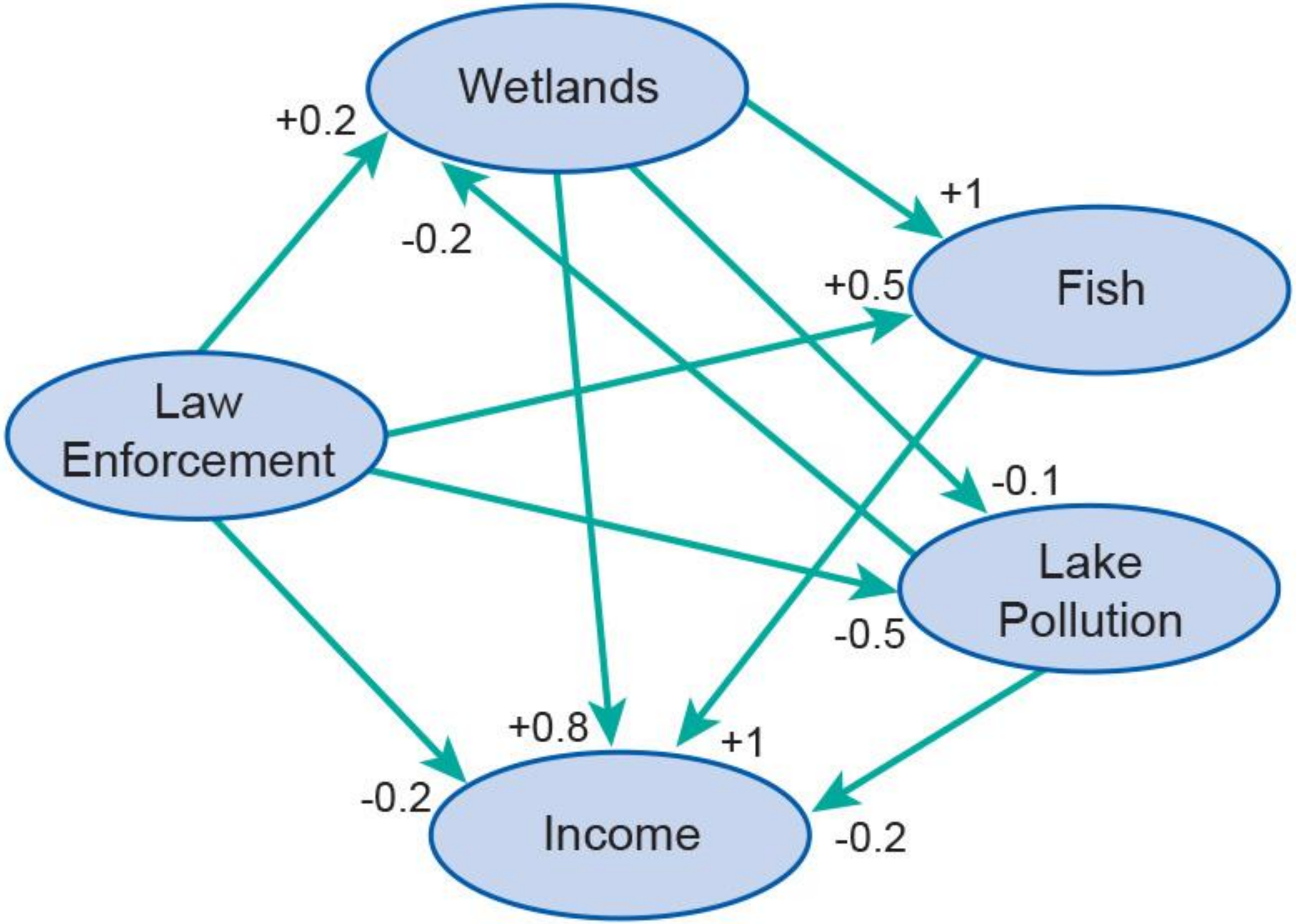




Elham Bakhshianlamouki, Ellen-Wien Augustijn, Marcela Brugnach, Alexey Voinov, Kathelijne Wijnberg, (2023) *A participatory modelling approach to cognitive mapping of the socio-environmental system of sandy anthropogenic shores in the Netherlands*, Ocean & Coastal Management, Volume 243 quantify relationship

Fuzzy Cognitive Mapping

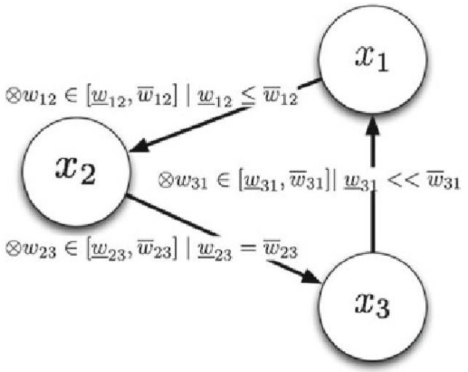
- Fuzzy cognitive mapping extends traditional cognitive mapping by incorporating fuzzy logic to represent the uncertain or imprecise nature of human cognition. In FCM, the relationships between concepts are not limited to binary associations but can instead have varying degrees of strength or uncertainty.
- Each relationship in an FCM is assigned a fuzzy value between 0 and 1, indicating the degree of association between concepts. These fuzzy values capture the ambiguity and subjectivity inherent in human cognition, allowing for more nuanced representations of complex relationships.


Wetlands
+0.2
-0.2
+1
Fish
+0.5
Law Enforcement
-0.1
Lake Pollution
-0.5
+0.8
+1
-0.2
Income
-0.2

Core book

Rule-based FCM

- Rule-based fuzzy cognitive mapping (RFCM) is an extension of fuzzy cognitive mapping (FCM) that integrates rule-based reasoning to enhance the modeling and analysis capabilities of cognitive maps. In RFCM, in addition to representing relationships between concepts using fuzzy logic, explicit rules are defined to govern the behavior of the system based on the fuzzy relationships between concepts.
- **Rule-Based Reasoning**: In RFCM, explicit rules are defined to govern the behavior of the system based on the fuzzy relationships between concepts. These rules specify how the values of certain concepts or variables change based on the values of other concepts or variables in the system. The rules are typically expressed in the form of if-then statements, where specific conditions trigger certain actions or changes in the system.



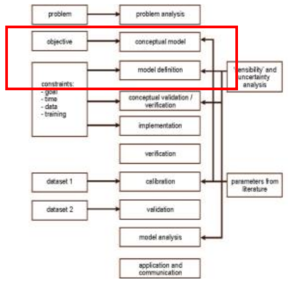
Figure

Caption

Fig. 2 Rule-based fuzzy cognitive maps. It is illustrated with a couple of nodes (c 1 and c 3) and a RBFCM relationship between them. Fuzzy rules and defuzzification process to compute the new state c 3

Methods to design an ABM

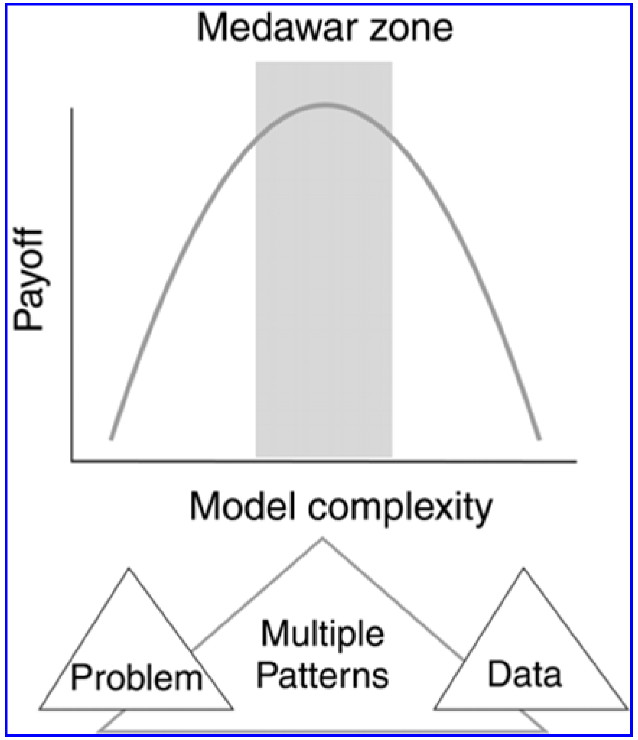
- Pattern-Oriented Modelling (POM)
- Participatory Modelling: FCM
- Start simple and add complexity
- Build toward a Question: do not add content that does not help to answer this question.



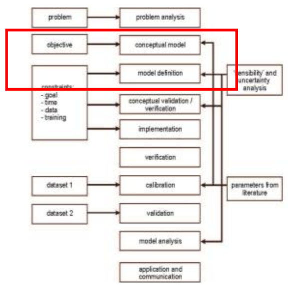
CONCEPTUAL MODELING (COMPLEXITY)

- A model must be built at the right level of description for every phenomenon
- Payoff of bottom-up models versus their complexity. Payoff is determined by its structural realism; i.e. its ability to produce independent predictions that match observations
- Work from simple behavior to complex behavior (making agents move randomly is simpler than moving agents over a road)

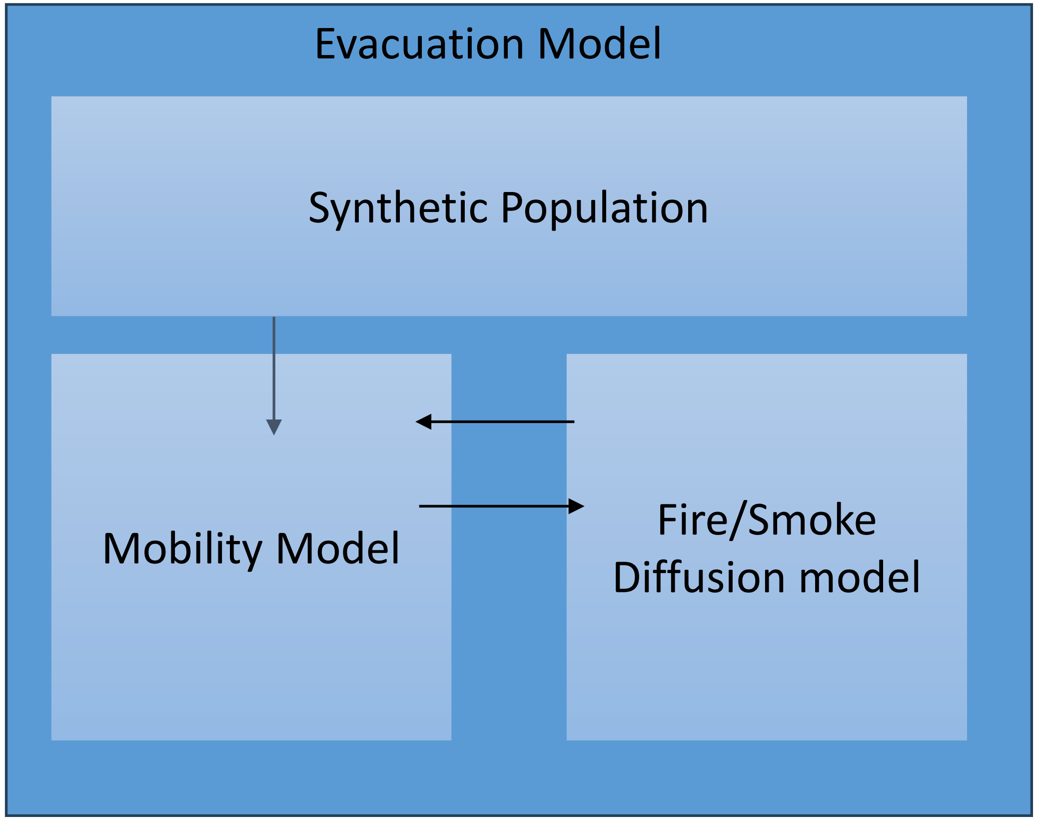
Start with the simplest version of the model, iteratively increase the complexity to reach "Medawar Zone"



Ref: Volker Grimm et al 2005



MODEL COMPONENTS



- Complex systems consist of hierarchies.
- A Complex system consists of sub-systems that we can translate into sub models.

can be run independently

- Sub models can be developed independent of each other.

Code examples

COVID-19 MODEL

agent

```
breed [municipalities municipality]   breed [healthunits healthunit]

directed-link-breed [commuterflows commuterflow]
```

```
to read-data
  ;;Load basemap of municipalities
  set basemap gis:load-dataset "Gemeentes2013TrMr.shp" ;; Net
  gis:set-world-envelope gis:envelope-of basemap
  gis:set-drawing-color 5
  gis:fill basemap 0.0
```

(bbox = envelope)

TELLING AGENTS WHAT TO DO

- You can tell agents what to do in two different ways:
  - Via procedures (implemented by the user)
  - Via primitives (built into Netlogo)

```
to setup            ;; comments are written after semicolon(s)
  clear-all         ;; clear everything
  create-turtles 10 ;; make 10 new turtles
end                 ; (one semicolon is enough, but I like two)
```

Example of a procedure

```
to go
  ask turtles [
    forward 1        ;; all turtles move forward one step
    right random 360 ;; and turn a random amount
  ]
end
```

Primitives are colored in blue

```
to go
  ask turtles [
    forward 1        ;; all turtles move forward one step
    right random 360 ;; and turn a random amount
  ]
end
```

Primitive reporters are colored in purple

TWO TYPES OF PROCEDURES

- Command procedures and Reporter procedures
- Reporter procedures use expressions and report a value

```
to-report average-wealth          ;; this reporter returns the
  report mean [wealth] of turtles ;; average wealth in the
end                               ;; population of turtles
```

report is a primitive
[wealth] is the input

GLOBAL VARIABLES

- **Global variables**: If a variable is a global variable, there is only one value for the variable, and every agent can access it.

```
globals [ n-of-strategies ]
```

```
globals [officer_walkspeed leaver_walkspeed follower_walkspeed i
```

LOCAL VARIABLES

- **Local variables**: A local variable is defined and used only in the context of a particular procedure or part of a procedure.

```
to swap-colors [turtle1 turtle2]  ;; turtle1 and turtle2 are inputs
  let temp ([color] of turtle1)   ;; store the color of turtle1 in temp
  ask turtle1 [ set color ([color] of turtle2) ]
        ;; set turtle1’s color to turtle2’s color
  ask turtle2 [ set color temp ]
        ;; now set turtle2’s color to turtle1’s (original) color
end     ;; (which was conveniently stored in local variable “temp”).
```

To create a local variable, use the let command.

If you use it at the top of a procedure, the variable will exist throughout the procedure. If you use it inside a set of square brackets, for example inside an ask, then it will exist only inside those brackets.

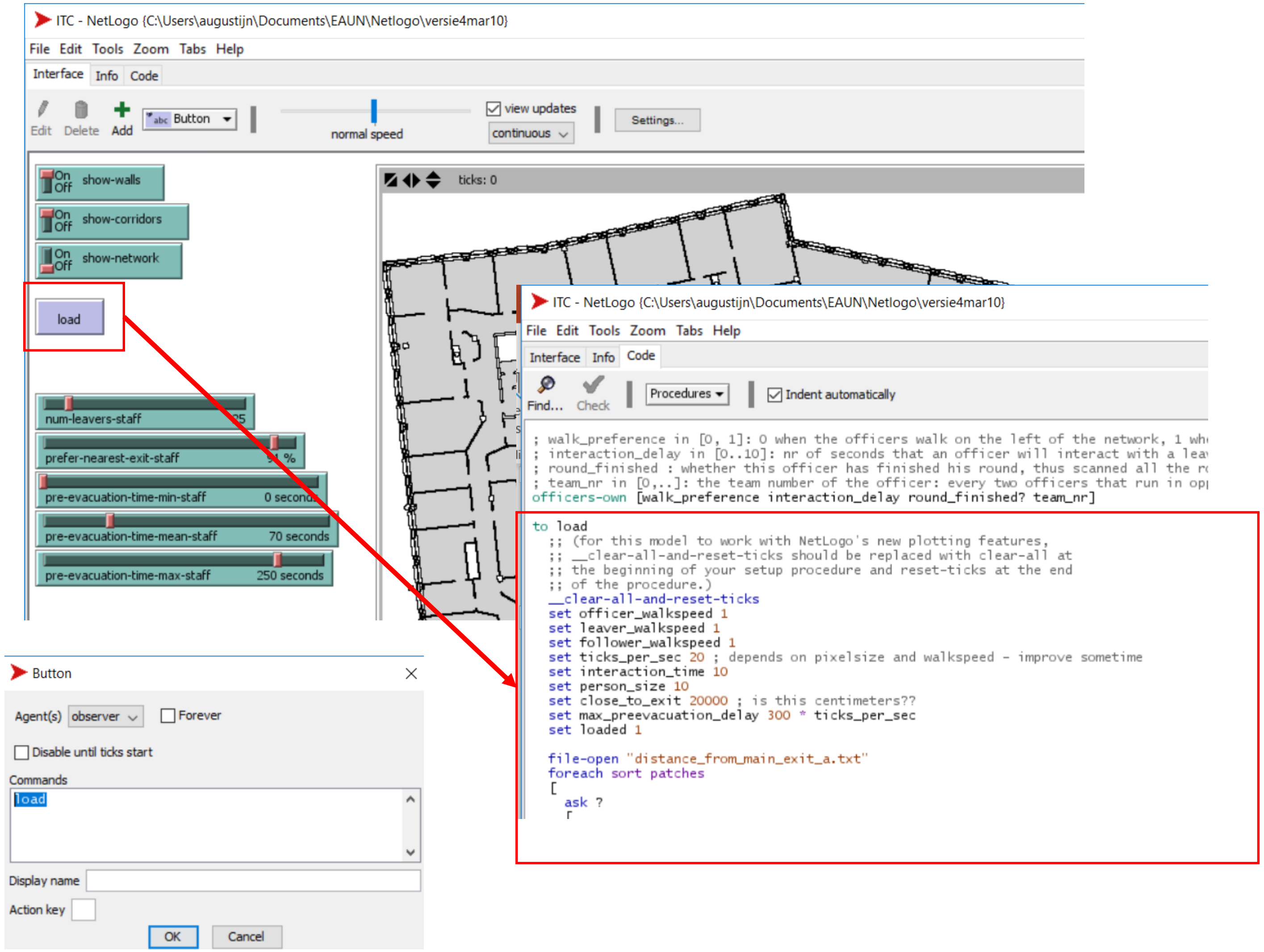
DIFFERENT TYPES OF AGENTS

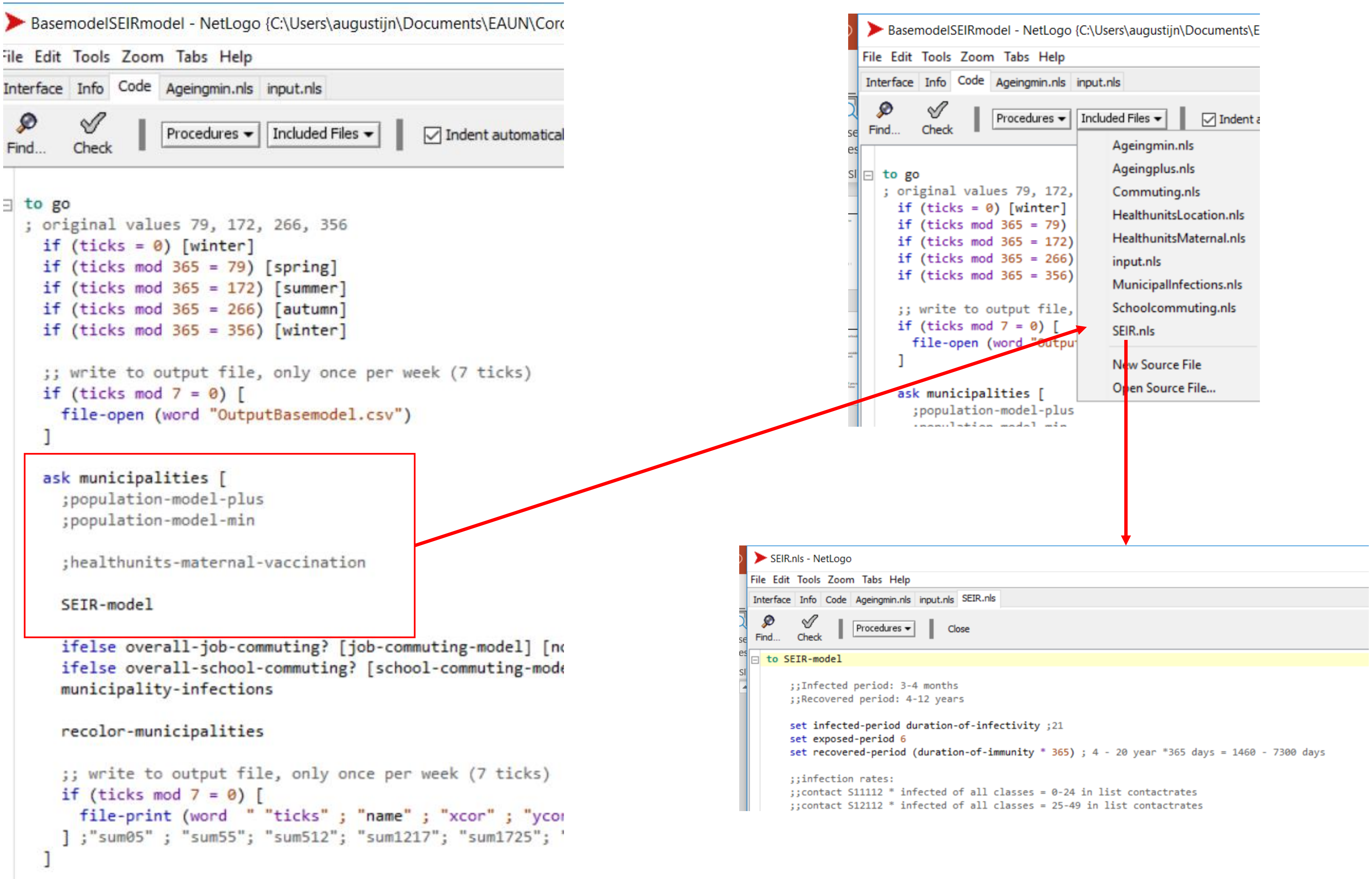
- Different types of turtles are called breeds and are defined with the syntax:

```
breed [plural-name singular-name]
```

```
breed [leavers leaver]
breed [followers follower]
breed [officers officer]
```

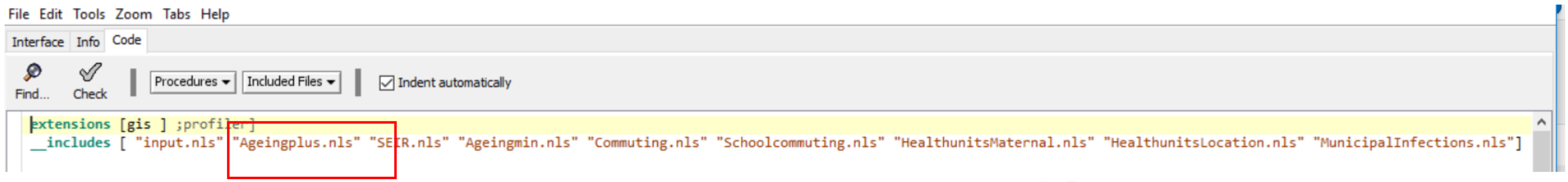
WHEN DOES WHICH PART OF YOUR CODE RUN?

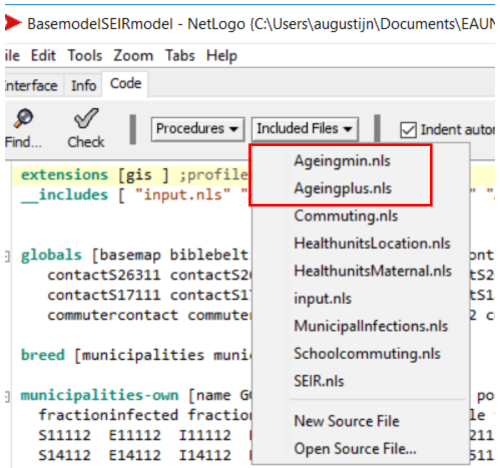


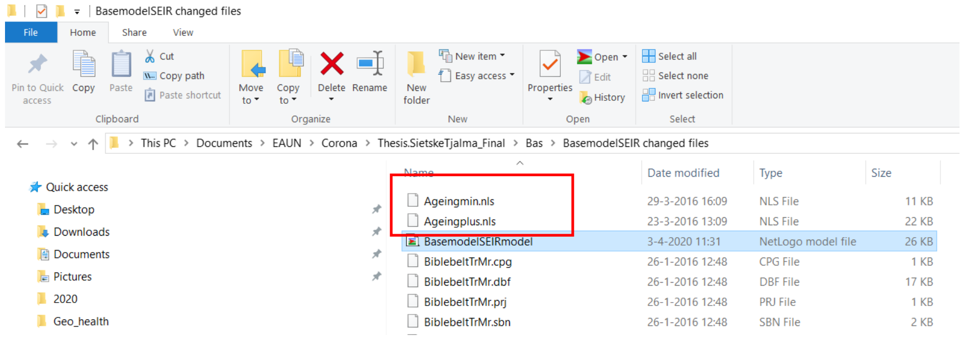


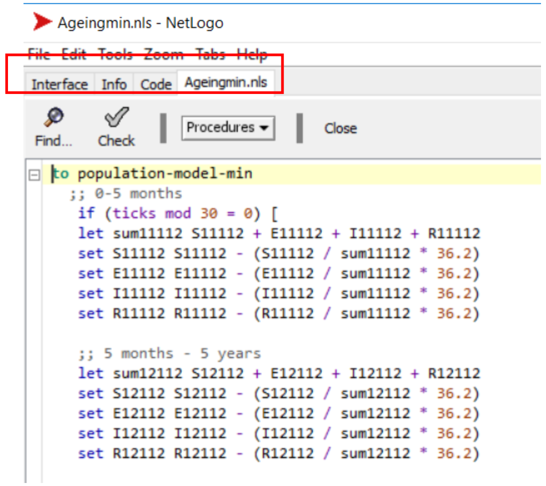
BasemodelSEIRmodel - NetLogo {C:\Users\augustijn\Documents\EAUN\Cor
File Edit Tools Zoom Tabs Help
Interface Info Code Ageingmin.nls input.nls
Find... Check Procedures Included Files Indent automatical
to go
; original values 79, 172, 266, 356
if (ticks = 0) [winter]
if (ticks mod 365 = 79) [spring]
if (ticks mod 365 = 172) [summer]
if (ticks mod 365 = 266) [autumn]
if (ticks mod 365 = 356) [winter]
;; write to output file, only once per week (7 ticks)
if (ticks mod 7 = 0) [
file-open (word "OutputBasemodel.csv")
]
ask municipalities [
;population-model-plus
;population-model-min
;healthunits-maternal-vaccination
SEIR-model
ifelse overall-job-commuting? [job-commuting-model] [no
ifelse overall-school-commuting? [school-commuting-mode
municipality-infections
recolor-municipalities
;; write to output file, only once per week (7 ticks)
if (ticks mod 7 = 0) [
file-print (word " "ticks" ; "name" ; "xcor" ; "ycor
] ;"sum05" ; "sum55"; "sum512"; "sum1217"; "sum1725";
]
BasemodelSEIRmodel - NetLogo {C:\Users\augustijn\Documents\E
Ageingmin.nls
Ageingplus.nls
Commuting.nls
HealthunitsLocation.nls
HealthunitsMaternal.nls
input.nls
MunicipalInfections.nls
Schoolcommuting.nls
SEIR.nls
New Source File
Open Source File...
SEIR.nls - NetLogo
File Edit Tools Zoom Tabs Help
Interface Info Code Ageingmin.nls input.nls SEIR.nls
Find... Check Procedures Close
to SEIR-model
;;Infected period: 3-4 months
;;Recovered period: 4-12 years
set infected-period duration-of-infectivity ;21
set exposed-period 6
set recovered-period (duration-of-immunity * 365) ; 4 - 20 year *365 days = 1460 - 7300 days
;;infection rates:
;;contact S11112 * infected of all classes = 0-24 in list contactrates
;;contact S12112 * infected of all classes = 25-49 in list contactrates

INCLUDES PROCEDURES (SUB MODELS)









Team-Based Learning

WOLF – SHEEP MODEL

- Go to https://www.netlogoweb.org
- Select “Wolf Sheep Predation”
- Note that you can check the “Model Info”
- Run the model to answer the following questions

TEAM BASED LEARNING

1. A complex system consists of elements, and these elements have connectivity. Which of the following answers best describes the situation in the Wolf Sheep Predation model?
   a. This model contains two elements, which in this case are agents and one connection when the wolf eat the sheep.
   b. This model contains three elements, two agents and one environment and the connection between these elements are that the sheep eat the grass, the wolf eat the sheep.
   c. The model contains many elements, many sheep and many wolf, the links between them are that some sheep are eaten by wolf.
   d. This model contains many elements, many sheep and many wolf and many patches (cells) with different levels of grass development. The links between these elements is that grass is eaten by sheep and sheep are eaten by wolf.

Note:- c & d is incorrect because elements can't be considered identical. It should be unique.

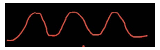
TEAM BASED LEARNING

2. For the wolf-sheep model, which of the following statements about interactions is correct?

a. The model contains positive feedback loops

b. The model contains negative feedback loops

c. The model does not contain any feedback loops

Note: - For option b, reason is growth of wolves leads to less sheep. When model runs by looking at the graph it ties to stabilize the system limited on grass (food supply). 
For option a, the graph will look like this: 

TEAM BASED LEARNING

3. Complex systems show non-linear behaviour. The wolf-sheep model represents a complex system because it has the following examples of non-linearity. Select all correct answers.

a. The model has state transitions
b. In this model, small changes can have large impacts
c. The model is scale-less (fractal)
d. The model has tipping points
e. All of the above

Note: -
a. State Transition is generally irreversible (ex. we loose all wolf/sheep)
b. based on changing the parameters of wolf sheep model (ex: number of wolf/sheep population, reproducibility rate), if we set high number of sheep population --> It will still gone at last as we run model.
c. It should be emerged spontaneously for the model.

TEAM-BASED LEARNING

4. The wolf-sheep example shows emergence. What emerges?

- Patterns
- Structures
- Behavior

Notes: -

- Patterns in time/population:
- Structure: Snowflake
- Behavior: Ex: Agent learn to find new evacuation way.