

# Designing an ABM

Multiple approaches

## Groups (4)

▼ Group 1	3 / 4 students	⋮
⋮ Francisco Alves Camello Ne... ⋮	⋮ Ahmed Hanif ⋮	⋮ Otobong Nse ⋮
▼ Group 2	<b>Full</b> 4 / 4 students	⋮
⋮ Pitchaporn Likitpanjamanon ⋮	⋮ Surya Naganathan ⋮	⋮ Jay Pandya ⋮
⋮ Helina Shrestha ⋮		
▼ Group 3	<b>Full</b> 4 / 4 students	⋮
⋮ Lana Annisa Rizqilana ⋮	⋮ Irene Irene Caroline Sihom... ⋮	⋮ Kiki Winda Kiki Winda Ver... ⋮
⋮ William William Arthurius ⋮		
▼ <u>Group 4</u>	3 / 4 students	⋮
⋮ Monalaine Bermoy ⋮	⋮ Rezaul Bhuiyan ⋮	⋮ Thierry Tuyizere ⋮

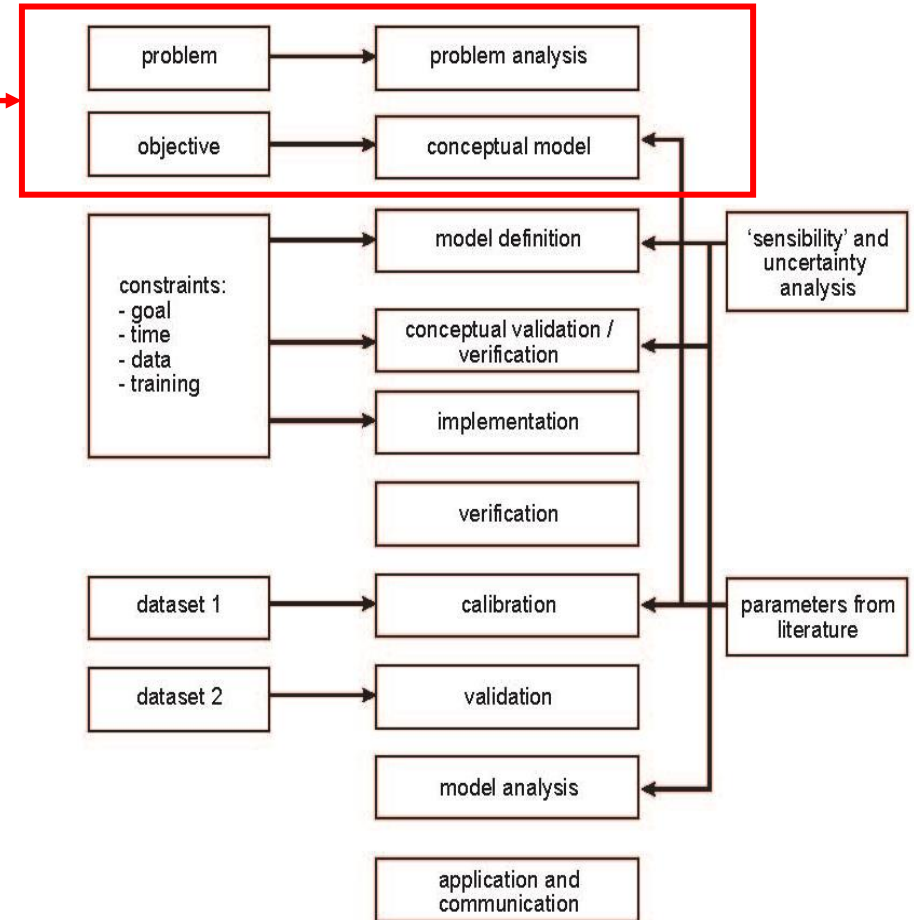
## Individual projects:

- ⋮ Amira Al Balushi +
- ⋮ Jazib Imran +

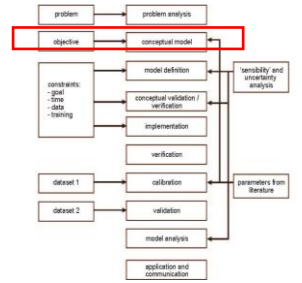


## TOPICS OF TODAY

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



# MUSHROOM SEARCH IN FOREST

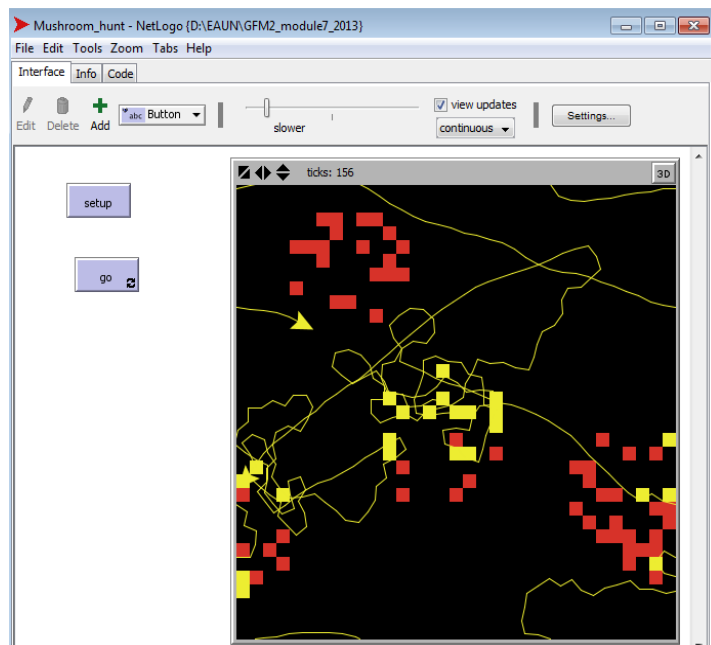


A video of a conceptual modeling session is uploaded in Canvas



## 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

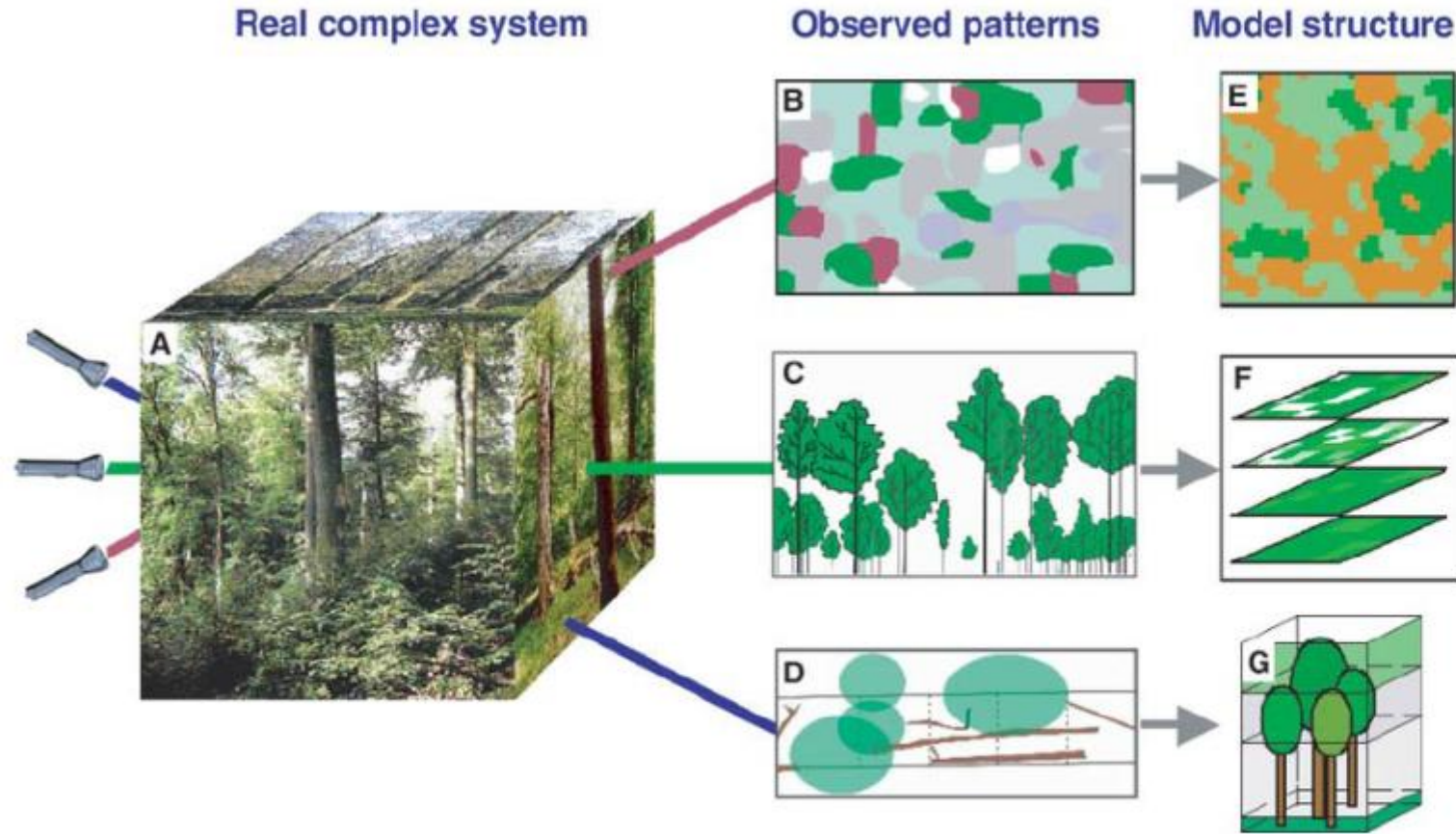


# Methods to design an ABM

- 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.



# Pattern Oriented modelling



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

- 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

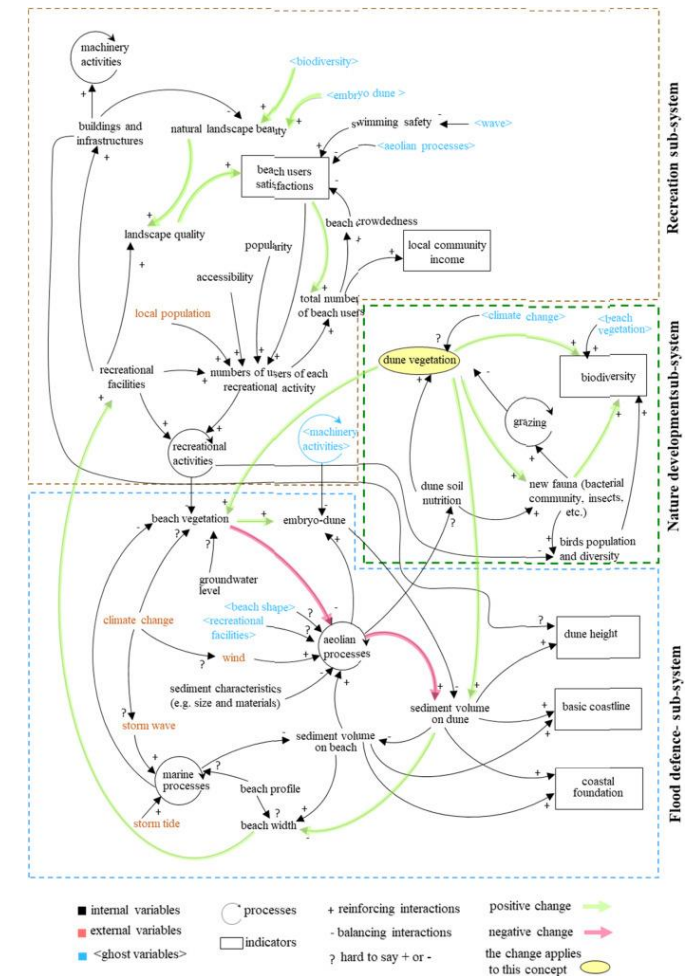
- Identify alternative submodels that implement alternative hypothesis
- Implement the submodels
- Contract the alternatives
- Repeat until a submodel has been found that reproduces all patterns

# Methods to design an ABM

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

# Cognitive mapping

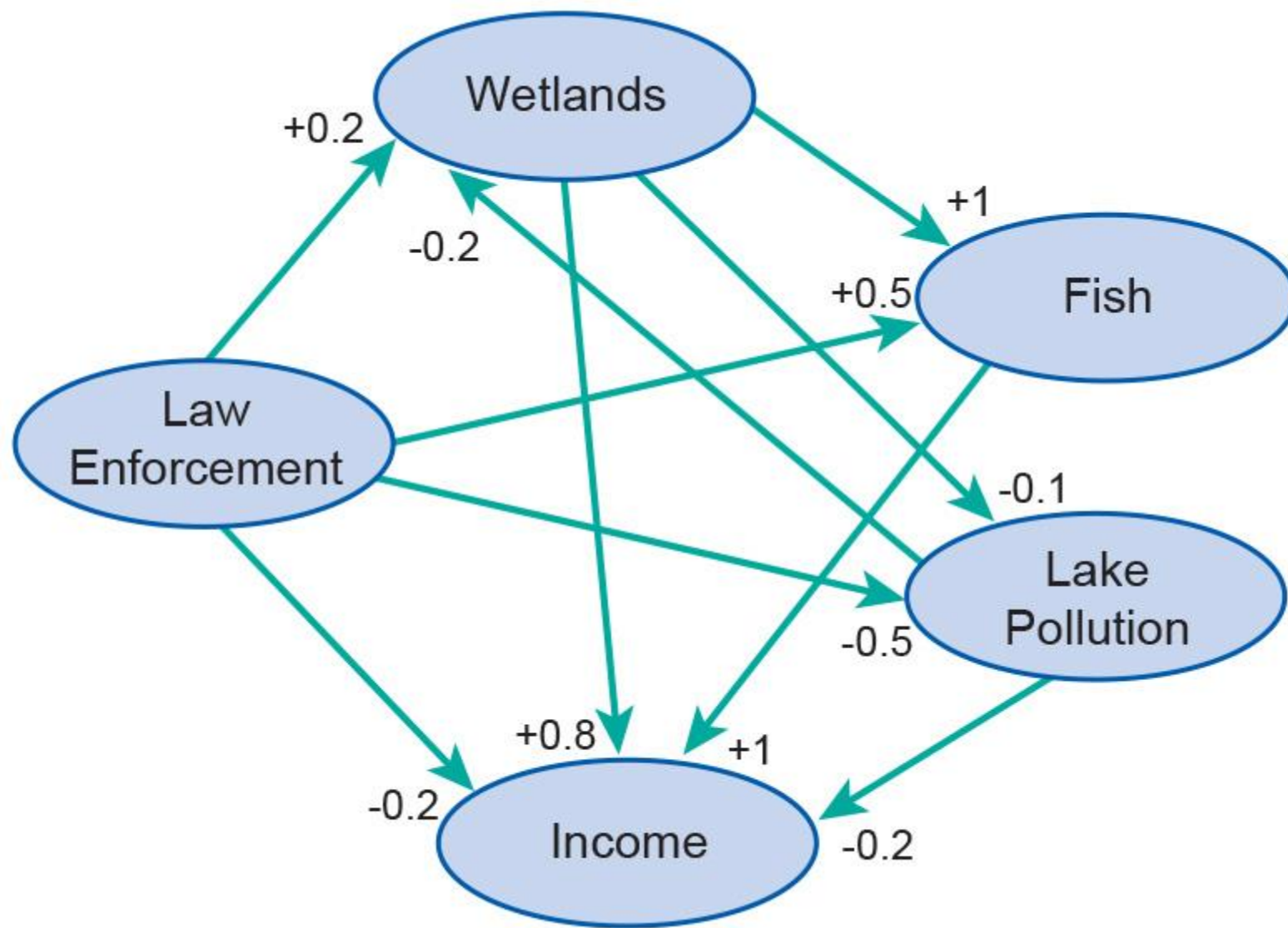
- **Cognitive mapping** is a mapping method used to create a **visual representation of a person's (or a group's) mental model** for a process or concept.
- The **relationships** in cognitive maps are often binary, indicating whether there is a **positive or negative association** between two concepts.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

[illegible]

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

# 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.

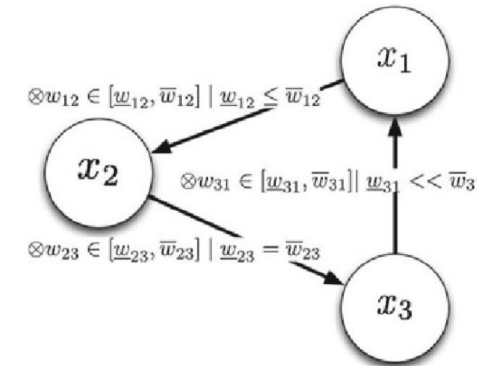


Core book



# 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 RBF-CM relationship between them. Fuzzy rules and defuzzification process to compute the new state  $c_3$

This figure was uploaded by [Elpiniki Papageorgiou](#)  
Content may be subject to copyright.

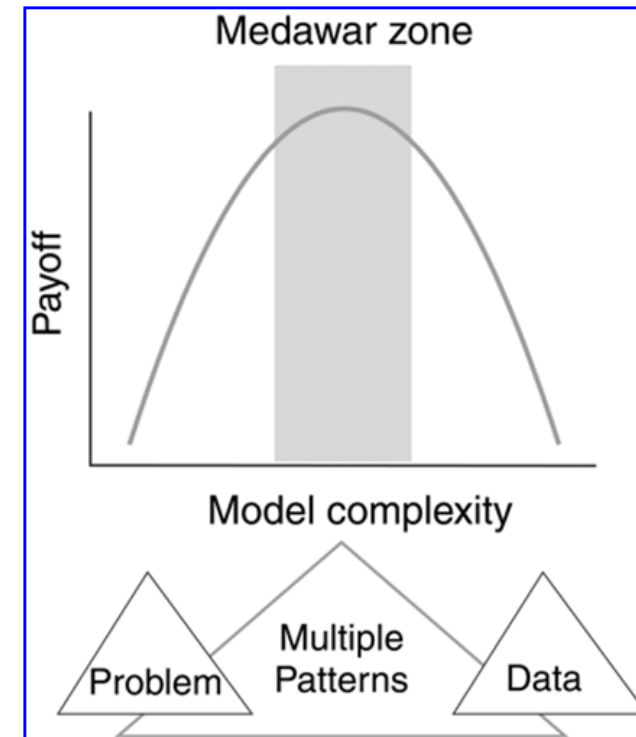
# Methods to design an ABM

- 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.

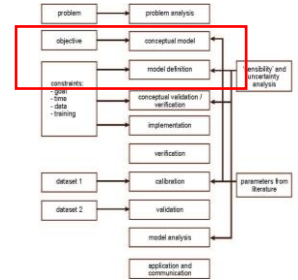
## CONCEPTUAL MODELING (COMPLEXITY)

- 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)

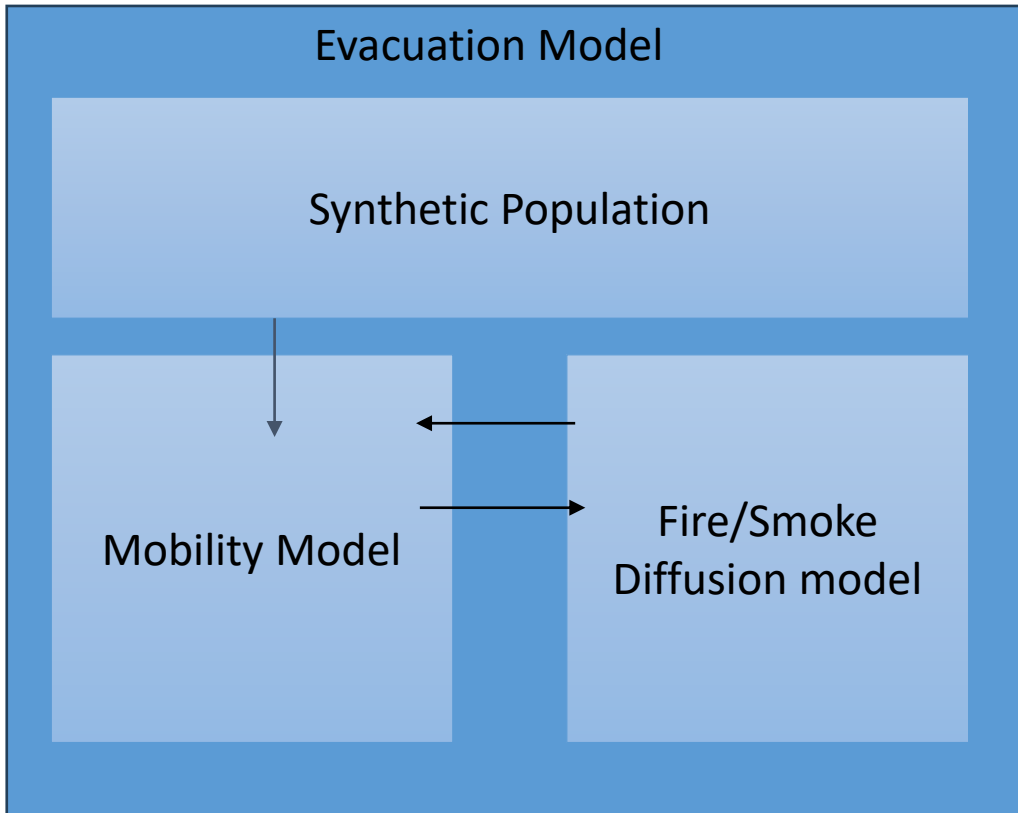
Start with the simplest version of the model,  
iteratively increase the complexity to reach  
“Medawar Zone”



Ref: Volker Grimm et al 2005

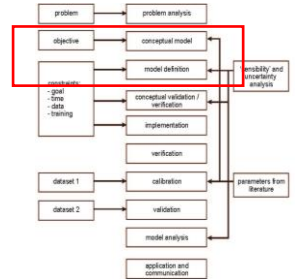


# MODEL COMPONENTS



- Complex systems consist of hierarchies.
- A Complex system consists of sub-systems that we can translate into sub models.
- Sub models can be developed independent of each other.

submodels can run independently



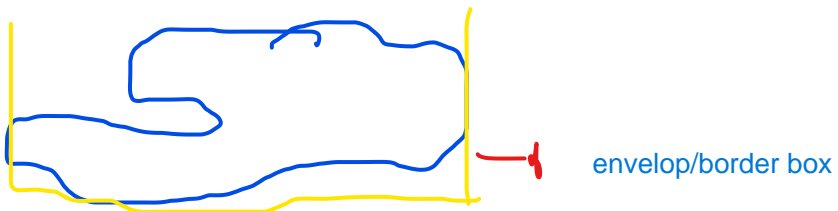
Code examples

## COVID-19 MODEL

`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
```





## 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
  clear-all
  create-turtles 10
end
```

;; comments are written after semicolon(s)  
;; clear everything  
;; make 10 new turtles  
; (one semicolon is enough, but I like two)

Example of a procedure

```
to go
  ask turtles [
    forward 1
    right random 360
  ]
end
```

;; all turtles move forward one step  
;; and turn a random amount

Primitives are colored in blue

```
to go
  ask turtles [
    forward 1
    right random 360
  ]
end
```

;; all turtles move forward one step  
;; and turn a random amount

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?



The image illustrates the NetLogo environment and its code execution flow. It shows the main interface with various controls and a button labeled 'load'. A red arrow points from this button to the 'to load' procedure in the code editor, indicating that the code within this procedure runs when the 'load' button is clicked.

**NetLogo Interface (Top Left):**

- Menu: File Edit Tools Zoom Tabs Help
- Interface Info Code tabs
- Buttons: Edit Delete Add (abc Button)
- Speed: normal speed
- View updates: ☒ view updates
- Continuous: continuous
- Settings...

**Interface Controls (Middle Left):**

- show-walls: ☐ On ☐ Off
- show-corridors: ☐ On ☐ Off
- show-network: ☐ On ☐ Off
- load (highlighted with a red box)

**Interface Sliders (Bottom Left):**

- num-leavers-staff: 5
- prefer-nearest-exit-staff: 50 %
- pre-evacuation-time-min-staff: 0 seconds
- pre-evacuation-time-mean-staff: 70 seconds
- pre-evacuation-time-max-staff: 250 seconds

**Code Editor (Top Right):**

- Menu: File Edit Tools Zoom Tabs Help
- Interface Info Code tabs
- Find... Check Procedures ☒ Indent automatically

**Code (Bottom Right):**

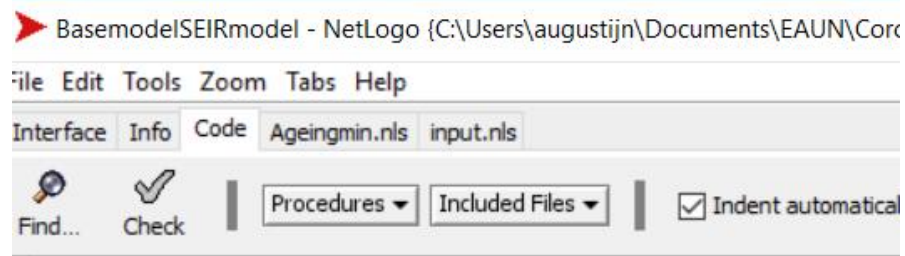
```
;; walk_preference in [0, 1]: 0 when the officers walk on the left of the network, 1 when on the right
;; interaction_delay in [0..10]: nr of seconds that an officer will interact with a leaver
;; round_finished: whether this officer has finished his round, thus scanned all the room
;; team_nr in [0,..]: the team number of the officer: every two officers that run in opposite directions
officers-own [walk_preference interaction_delay round_finished? team_nr]

to load
  ;; (for this model to work with NetLogo's new plotting features,
  ;; __clear-all-and-reset-ticks should be replaced with clear-all at
  ;; the beginning of your setup procedure and reset-ticks at the end
  ;; of the procedure.)
  clear-all-and-reset-ticks
  set officer_walkspeed 1
  set leaver_walkspeed 1
  set follower_walkspeed 1
  set ticks_per_sec 20 ; depends on pixelsize and walkspeed - improve sometime
  set interaction_time 10
  set person_size 10
  set close_to_exit 20000 ; is this centimeters??
  set max_pre_evacuation_delay 300 * ticks_per_sec
  set loaded 1

  file-open "distance_from_main_exit_a.txt"
  foreach sort patches
  [
    ask ?
    r
  ]
end
```

**Button Dialog (Bottom Left):**

- Button: Button
- Agent(s): observer ☐ Forever
- ☐ Disable until ticks start
- Commands: load
- Display name:
- Action key:
- OK Cancel



```
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

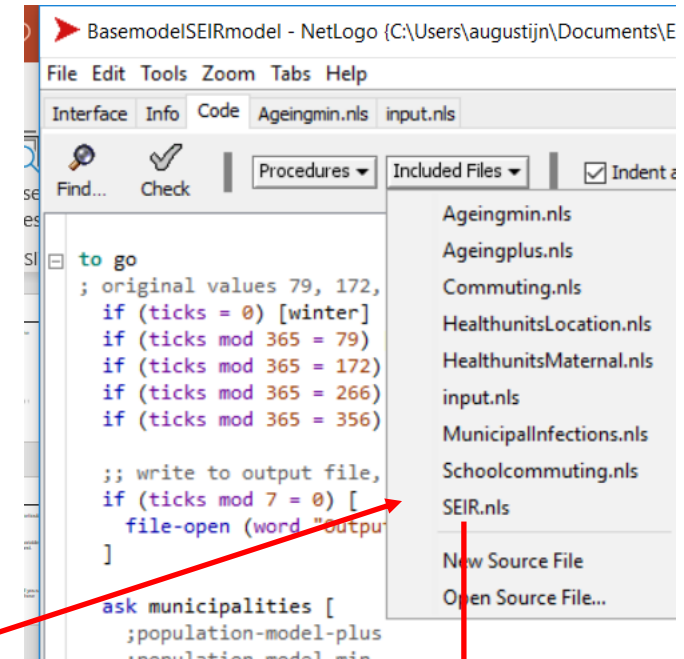
  ;healthunits-maternal-vaccination

  SEIR-model

  ifelse overall-job-commuting? [job-commuting-model] [no]
  ifelse overall-school-commuting? [school-commuting-model] [no]
  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" ; '
  ]
]
```



```
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 S1112 * infected of all classes = 0-24 in list contactrates
;;contact S1212 * infected of all classes = 25-49 in list contactrates
```



# INCLUDES PROCEDURES (SUB MODELS)

```
File Edit Tools Zoom Tabs Help
Interface Info Code
Find... Check Procedures Included Files Indent automatically
extensions [gis] ;profile
_includes [ "input.nls" "Ageingplus.nls" "SEIR.nls" "Ageingmin.nls" "Commuting.nls" "Schoolcommuting.nls" "HealthunitsMaternal.nls" "HealthunitsLocation.nls" "MunicipalInfections.nls"]
```

gis models include this

```
BasemodelSEIRmodel - NetLogo (C:\Users\augustijn\Documents\EAUN)
File Edit Tools Zoom Tabs Help
Interface Info Code
Find... Check Procedures Included Files Indent automatically
extensions [gis] ;profile
_includes [ "input.nls" "Ageingmin.nls" "Ageingplus.nls" "Commuting.nls" "HealthunitsLocation.nls" "HealthunitsMaternal.nls" "input.nls" "MunicipalInfections.nls" "Schoolcommuting.nls" "SEIR.nls" ]
globals [basemap biblebelt contactS26311 contactS26312 contactS17111 contactS17112 commutercontact commutercontactS26311 commutercontactS26312 commutercontactS17111 commutercontactS17112]
breed [municipalities municipality]
municipalities-own [name G fractioninfected fractioninfectedS26311 fractioninfectedS26312 fractioninfectedS17111 fractioninfectedS17112]
S11112 E11112 I11112 S14112 E14112 I14112
```

BasemodelSEIR changed files

Name	Date modified	Type	Size
Ageingmin.nls	29-3-2016 16:09	NLS File	11 KB
Ageingplus.nls	23-3-2016 13:09	NLS File	22 KB
BasemodelSEIRmodel	3-4-2020 11:31	NetLogo model file	26 KB
BiblebeltTrMr.cpg	26-1-2016 12:48	CPG File	1 KB
BiblebeltTrMr.dbf	26-1-2016 12:48	DBF File	17 KB
BiblebeltTrMr.prj	26-1-2016 12:48	PRJ File	1 KB
BiblebeltTrMr.sbn	26-1-2016 12:48	SBN File	2 KB

```
Ageingmin.nls - NetLogo
File Edit Tools Zoom Tabs Help
Interface Info Code Ageingmin.nls
Find... Check Procedures Close
to population-model-min
;; 0-5 months
if (ticks mod 30 = 0) [
let sum11112 S11112 + E11112 + I11112 + R11112
set S11112 S11112 - (S11112 / sum11112 * 36.2)
set E11112 E11112 - (E11112 / sum11112 * 36.2)
set I11112 I11112 - (I11112 / sum11112 * 36.2)
set R11112 R11112 - (R11112 / sum11112 * 36.2)

;; 5 months - 5 years
let sum12112 S12112 + E12112 + I12112 + R12112
set S12112 S12112 - (S12112 / sum12112 * 36.2)
set E12112 E12112 - (E12112 / sum12112 * 36.2)
set I12112 I12112 - (I12112 / sum12112 * 36.2)
set R12112 R12112 - (R12112 / sum12112 * 36.2)
```



# 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

### summary :

- complex systems are hierarchical, sub-systems are nested in larger systems
- complex systems are non-linear due to interdependency of the elements (state transitions contoh sand, small changes can have large impacts contoh : lempar batu di sungai ngubah aliran air, scaleless behavior (fractals) contoh : same shape repeatit self in different scale, tipping points contoh : critical treshold saat sudah mencapai bisa membuat impact besar dalam system)
- the system is defined by it connection (connectivity)
- due to adaptation, patterns like flocking of birds emerge



## 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.



## 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



## 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



## TEAM-BASED LEARNING

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

- Patterns
- Structures
- Behavior