Agent-Based Modelling for Social Simulation

EASSS 2018 | Maastricht

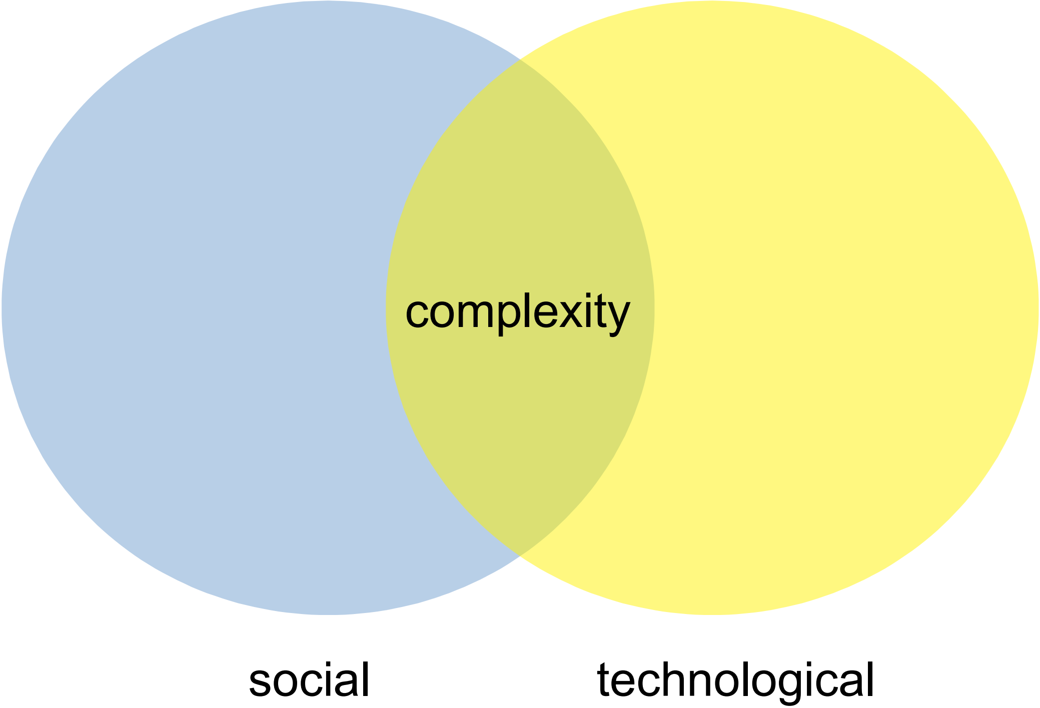
Neil Yorke-Smith
n.yorke-smith@tudelft.nl

Thanks to I. Nikolic

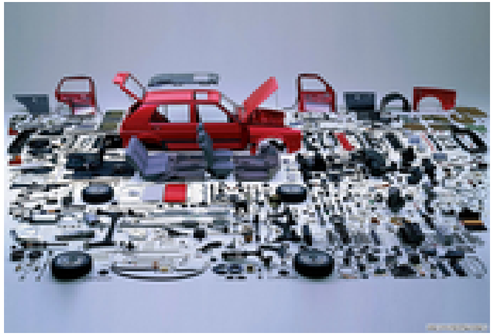
Outline

1. Socio-technical systems
2. Generative modelling
3. ABSS tools (hands on)
4. Modelling process (hands on)
5. Resources

Socio-technical systems



Complex systems



Complex systems

self-similarity

information

order

adaptiveness

path-dependency

diversity

robustness

emergence

non-linearity

evolution

observer-dependency

networks

randomness

chaos

instability

Social science research questions

- Why is this happening?
- How does it affect these stakeholders?
- What are the values behind these actors' interactions?
- What are the links between these factors?
- What policy should the government take?

TUDelft

Hopkins and King (2010)

“Computer scientists may be interested in finding the needle in the haystack – but social scientists are more commonly interested in characterizing the haystack.”

Schools of agent thinking

- Artificial Intelligence
  - Agents as autonomous identities solving problems
- Multi-Agent Systems
  - Distributed control of systems
- Agent-Based Modelling (and Simulation)
  - Simulating (real world) phenomena

Agent-Based Modelling and Sim.

- Bottom-up perspective
- Model social reality with agents and their interactions
- **Key:** How could the decentralised local interactions of heterogeneous autonomous agents generate the observed regularity?

ABSS applications

- Energy market deregulation (Macal & North 2005)
- Epidemic spread (Zhang et al 2016)
- National-scale employment (Axtell 2016)
- University admissions (Reardon et al 2016)
- Eurovision Song Contest collusion (Gatherer 2006)

Case study:
Deflzijl industrial network

Summary

- Research questions in social science concern causality and explanation
- Agent-based modelling: bottom-up perspective
- Used in anthropology, business, ecology, economics, political science, sociology

TU Delft

Outline

1. Socio-technical systems
2. **Generative modelling**
3. ABSS tools (hands on)
4. Modelling process (hands on)
5. Resources

Models simplify reality

All models are wrong,
some are useful!

- Every model is a simplification of reality…
- ...is it a useful one?

Model a problem, not a system!

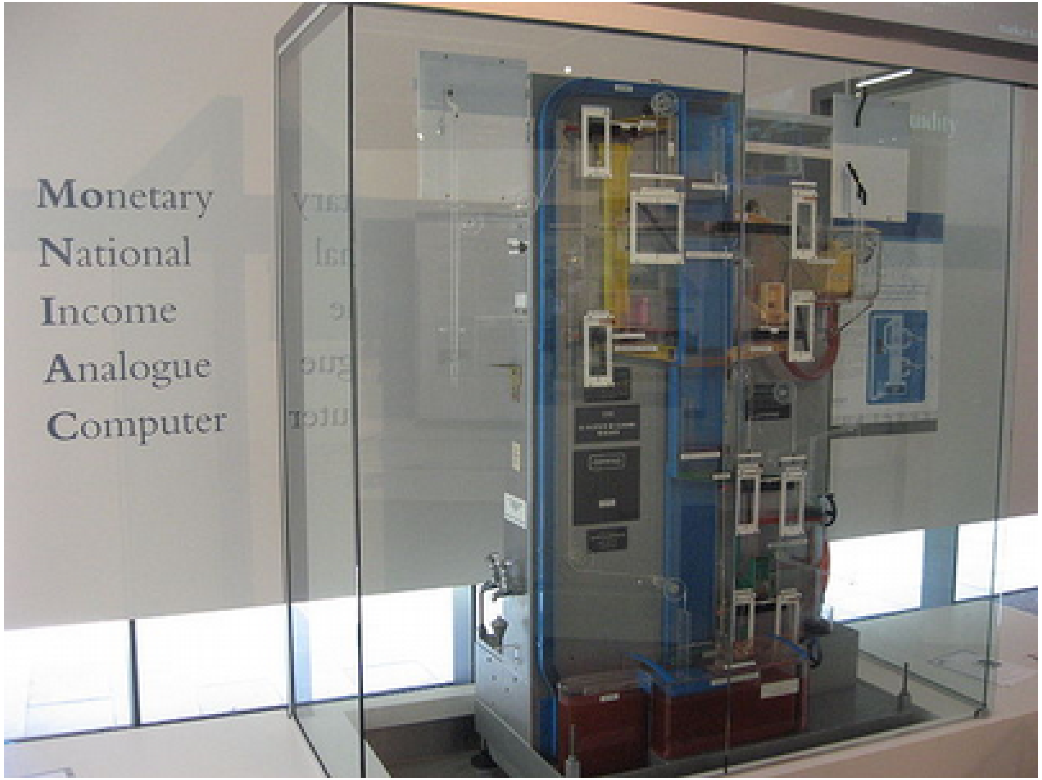
Models simplify reality

All models are wrong,
some are useful!

- Usefulness of a model is measured by the speed it is replaced

Insight is the goal, not numbers!

Top-down modelling



© 2018

Limitations of top-down modelling

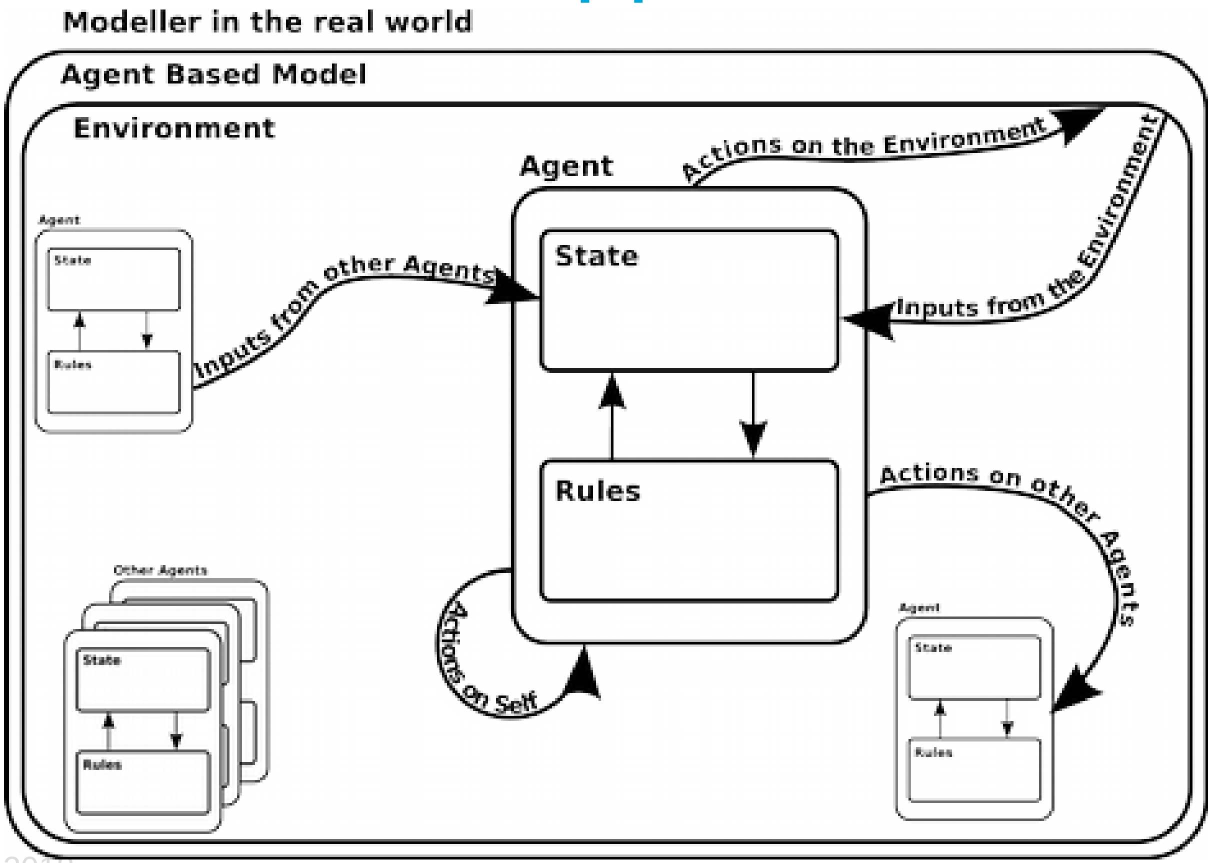
- Understand system in entirety
- Understand exactly how components interact with each other
- Good for complicated systems, e.g. cars

- Fails for complex systems

Generative principle

- Build understanding from the bottom up
  - “If you did not grow it, you did not explain it!” (Epstein 1999)
- **Principle:** phenomena can be described in terms of interconnected networks of (relatively) simple units
  - Deterministic, finite rules and parameters of natural phenomena interact with each other to generate complex behaviour

Generative approach

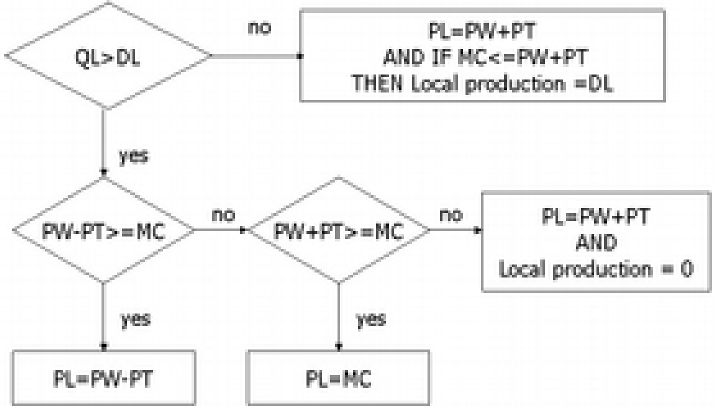


ABM entities

- Agent = thing that does things to other things
- Agent state and behaviour, model state and behaviour
- Rules = agents' internal models
- Behaviour = set of observable actions
- Environment = everything relevant that's not an agent
- Discrete time

Types of rules

- Decision and transformation rules: inputs, states → action, behaviour
- Can be static or dynamic
- Rules, MCDM, inference engines, ML, GA



Limitations of bottom-up modelling

- Data requirements (of individuals)
- Implementation not straightforward
- Different from equation-based models

- Excessive for simple and complicated systems

Case study: Epidemic modelling

Stoch Environ Res Risk Assess (2016) 30:2079–2093
DOI 10.1007/s00477-015-1199-x

ORIGINAL PAPER

Agent-based modelling of cholera diffusion

Ellen-Wien Augustijn[1] · Tom Doldersum[1,3] · Juliana Useya[4] · Denie Augustijn[2]

Published online: 5 January 2016

Abstract This paper introduces a spatially explicit agent-based simulation model for micro-scale cholera diffusion. The model simulates both an environmental reservoir of naturally occurring V. cholerae bacteria and hyperinfectious V. cholerae. Objective of the research is to test if runoff from open refuse dumpsites plays a role in cholera diffusion. A number of experiments were conducted with the model for a case study in Kumasi, Ghana, based on an epidemic in 2005. Experiments confirm the importance of the hyperinfectious transmission route, however, they also reveal the importance of a representative spatial distribution of the income classes. Although the contribution of runoff from dumpsites can never be conclusively proven, the experiments show that modelling the epidemic via this mechanism is possible and improves the model results. Relevance of this research is that it shows the possibilities of agent-based modelling combined with pattern reproduction for cholera diffusion studies. The proposed model is simple in its setup but can be extended by adding additional elements such as human movement and change of behaviour of individuals based on disease awareness. Eventually, agent-based models will open opportunities to explore policy related research questions related to interventions to influence the diffusion process.

Keywords Agent-based modelling · Cholera · Dumpsite runoff · Ghana

1 Introduction

Cholera is a disease spread by Vibrio cholerae, causing diarrhea and severe dehydration in about one out of 20 patients. Cholera can be endemic, leading to seasonal outbreaks, or epidemic. According to the World Health Organization (WHO), cholera incidence has increased globally since 2005 with in 2012 48 % of cholera cases occurring in Africa (WHO 2014).

Cholera infection can be caused by ingestion of food or water contaminated by V. cholerae and has two distinct life-cycles, one in the environment and another in humans (Harris et al. 2012). The pathogen occurs naturally in coastal waters, preferring brackish water and can live in association with zooplankton and shellfish (Harris et al. 2012; Sedas 2007). The intake and passage of V. cholerae through the human body results in conversion of the pathogen to a hyperinfectious state. When shed via faecal excretion of infected individuals, hyperinfectious bacteria can be re-introduced into the environment and pose a severe risk to other individuals as the infectious dose is 10–100 times lower compared to natural, non-human shed low-infectious organisms (Harris et al. 2012). When

✉ Ellen-Wien Augustijn
p.w.m.augustijn@utwente.nl

Tom Doldersum
t.doldersum@gmail.com

Juliana Useya
useya21997@alumni.itc.nl

Denie Augustijn
d.c.m.augustijn@utwente.nl

[1] Faculty Geo-Information Science and Earth Observation, University of Twente, Enschede, The Netherlands

[2] Faculty of Engineering and Technology, University of Twente, Enschede, The Netherlands

[3] Present Address: Rijkswaterstaat, Ministry of Infrastructure and the Environment, Lelystad, The Netherlands

[4] Department of Geoinformatics and Surveying, University of Zimbabwe, Harare, Zimbabwe

Summary

- Generative models are wrong,
  but can be very useful
- Agents – states – rules – actions (behaviours) – environment – time

Outline

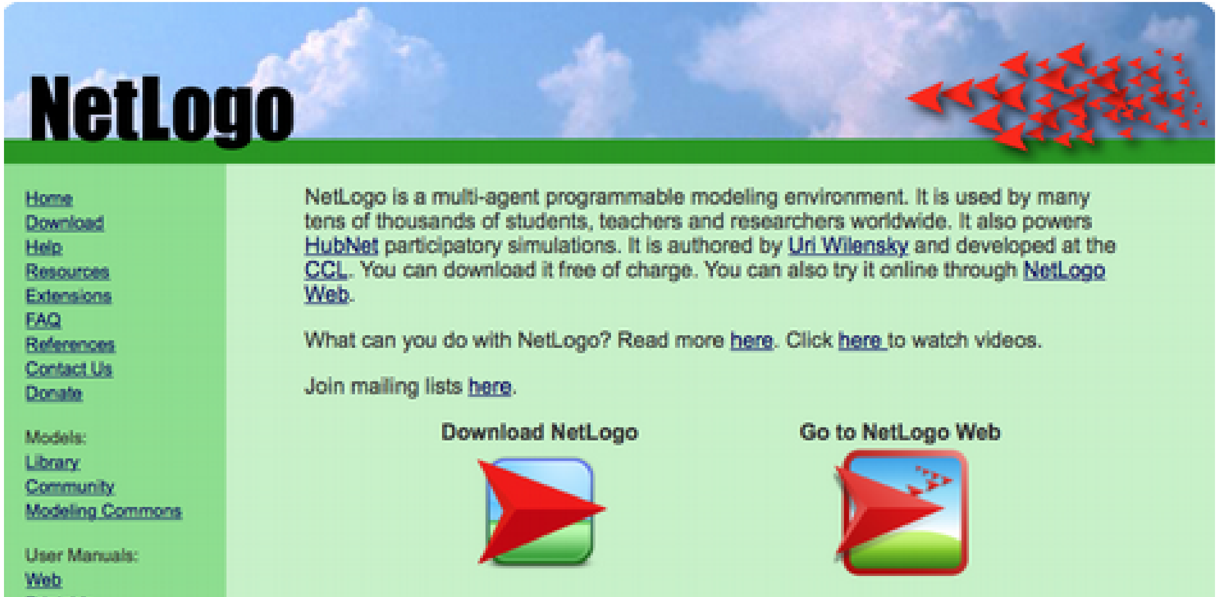
1. Socio-technical systems
2. Generative modelling
3. **ABSS tools (hands on)**
4. Modelling process (hands on)
5. Resources

Four types of ABSS tools

- General agent platform
- Dedicated ABM platform
- Dedicated ABM library
- General programming language

NetLogo hands-on

- Today we’ll try modelling with NetLogo
- https://ccl.northwestern.edu/netlogo/



NetLogo

Home
Download
Help
Resources
Extensions
FAQ
References
Contact Us
Donate

Models:
Library
Community
Modeling Commons

User Manuals:
Web

NetLogo is a multi-agent programmable modeling environment. It is used by many tens of thousands of students, teachers and researchers worldwide. It also powers HubNet participatory simulations. It is authored by Uri Wilensky and developed at the CCL. You can download it free of charge. You can also try it online through NetLogo Web.

What can you do with NetLogo? Read more here. Click here to watch videos.

Join mailing lists here.

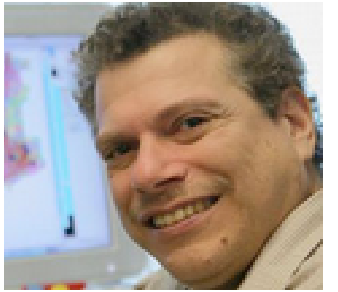
Download NetLogo

Go to NetLogo Web

“Low threshold, no ceiling”

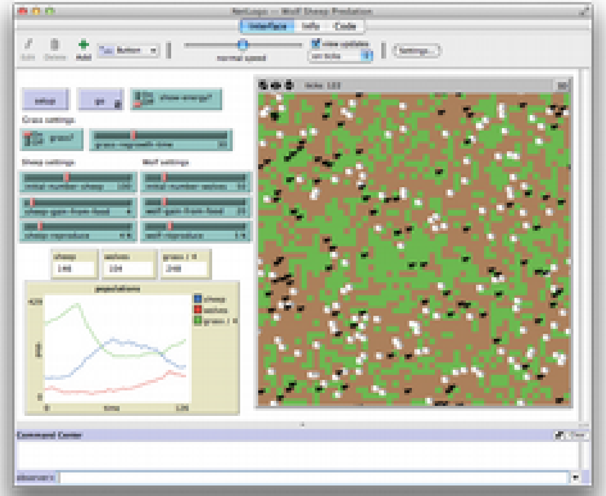
- Commonly-used ABSS platform
- Scripting language + UI
- Logo + Lisp → StarLogo → NetLogo
- Open source Scala/Java
- NetLogo Web

Wilensky (1999)

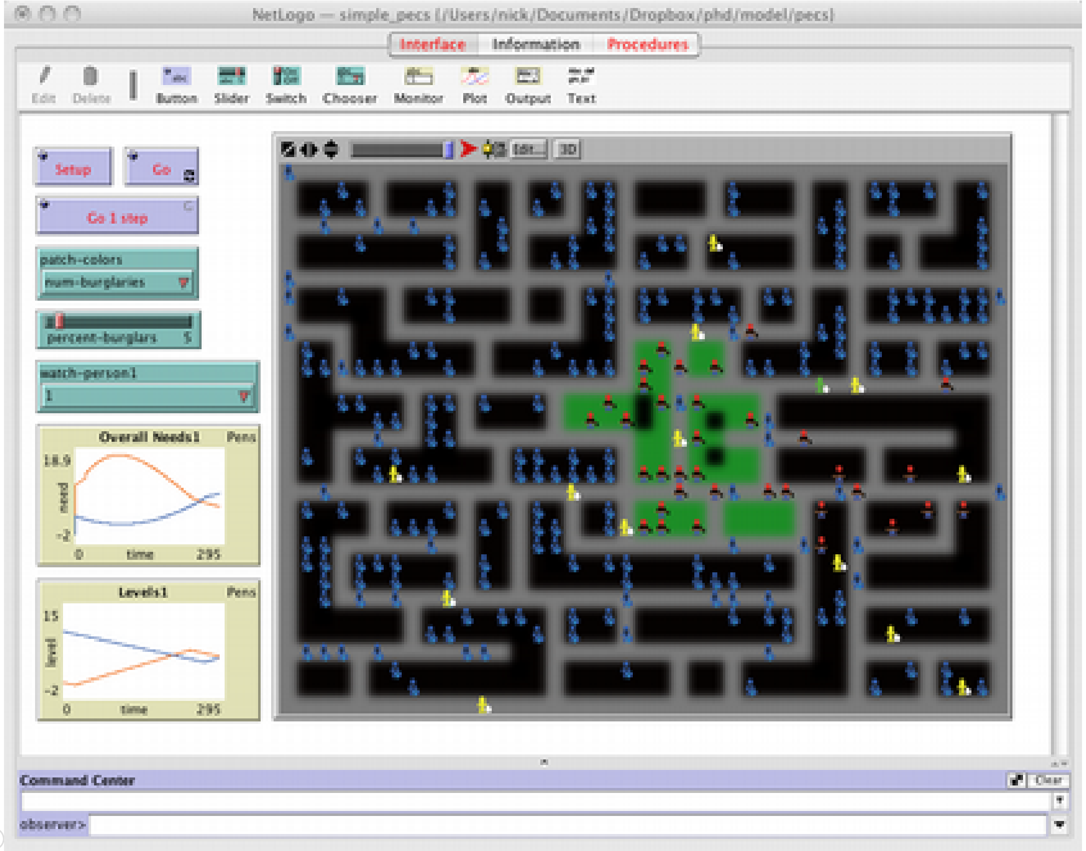


NetLogo is not perfect

- Some language limitations
- Not object orientated
- Scaling
- Not Python
  (but PyNetLogo)



NetLogo UI has 3 tabs



NetLogo — simple_pecs (/Users/nick/Documents/Dropbox/phd/model/pecs)

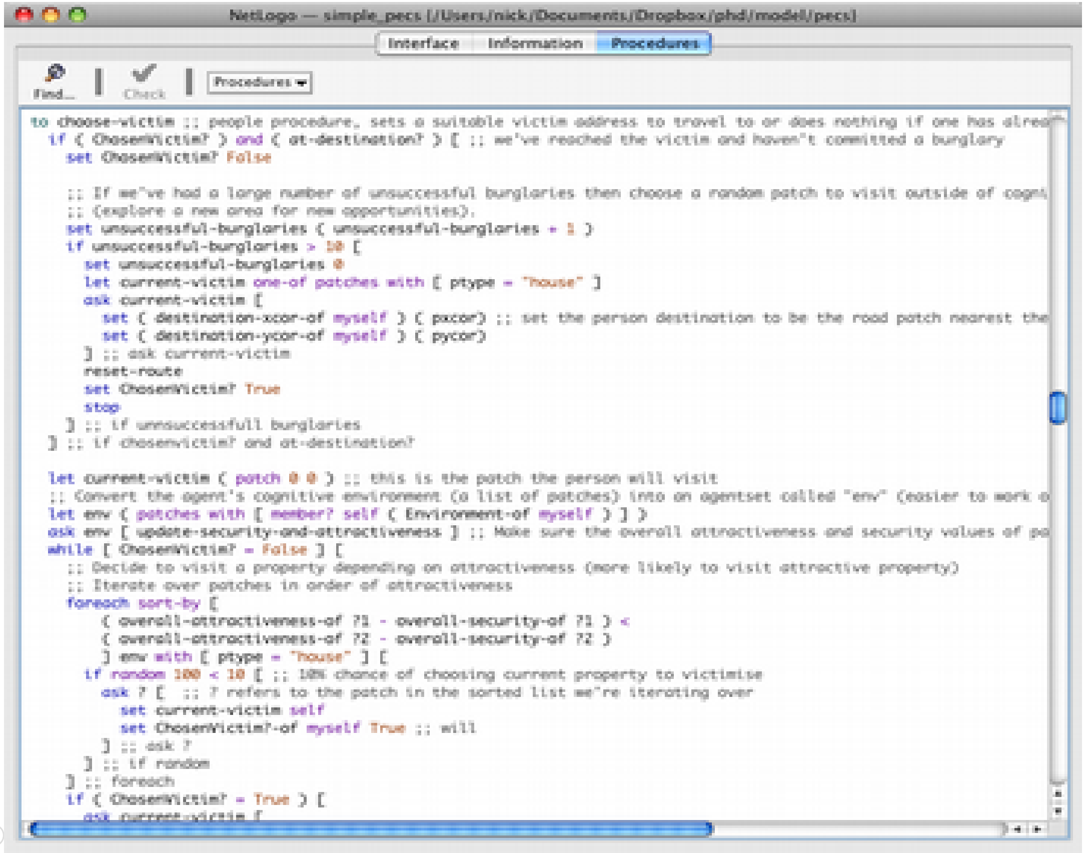
patch-colors: num-burglaries

percent-burglars 5

watch-person1: 1

observer>

NetLogo UI has 3 tabs

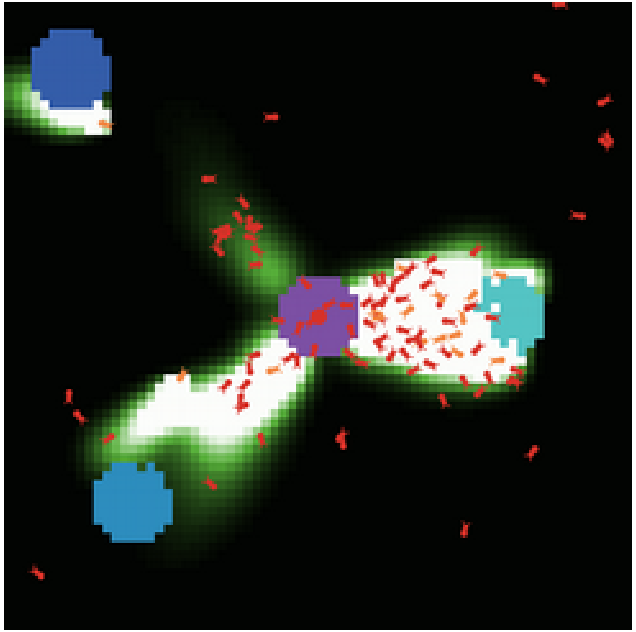


NetLogo — simple_pecs [/Users/nick/Documents/Dropbox/phd/model/pecs]

```
to choose-victim ;; people procedure, sets a suitable victim address to travel to or does nothing if one has alrea
  if ( ChosenVictim? ) and ( at-destination? ) [ ;; we've reached the victim and haven't committed a burglary
    set ChosenVictim? False

    ;; If we've had a large number of unsuccessful burglaries then choose a random patch to visit outside of cogni
    ;; (explore a new area for new opportunities).
    set unsuccessful-burglaries ( unsuccessful-burglaries + 1 )
    if unsuccessful-burglaries > 10 [
      set unsuccessful-burglaries 0
      let current-victim one-of patches with [ ptype = "house" ]
      ask current-victim [
        set ( destination-xcor-of myself ) ( pxcor) ;; set the person destination to be the road patch nearest the
        set ( destination-ycor-of myself ) ( pycor)
      ] ;; ask current-victim
      reset-route
      set ChosenVictim? True
      stop
    ] ;; if unsuccessfull burglaries
  ] ;; if chosenvictim? and at-destination?

  let current-victim ( patch 0 0 ) ;; this is the patch the person will visit
  ;; Convert the agent's cognitive environment (a list of patches) into an agentset called "env" (easier to work o
  let env ( patches with [ member? self ( Environment-of myself ) ] )
  ask env [ update-security-and-attractiveness ] ;; Make sure the overall attractiveness and security values of pa
  while [ ChosenVictim? = False ] [
    ;; Decide to visit a property depending on attractiveness (more likely to visit attractive property)
    ;; Iterate over patches in order of attractiveness
    foreach sort-by [
        ( overall-attractiveness-of ?1 - overall-security-of ?1 ) <
        ( overall-attractiveness-of ?2 - overall-security-of ?2 )
        ] env with [ ptype = "house" ] [
      if random 100 < 10 [ ;; 10% chance of choosing current property to victimise
        ask ? [  ;; ? refers to the patch in the sorted list we're iterating over
          set current-victim self
          set ChosenVictim?-of myself True ;; will
        ] ;; ask ?
      ] ;; if random
    ] ;; foreach
    if ( ChosenVictim? = True ) [
      ask current-victim [
```

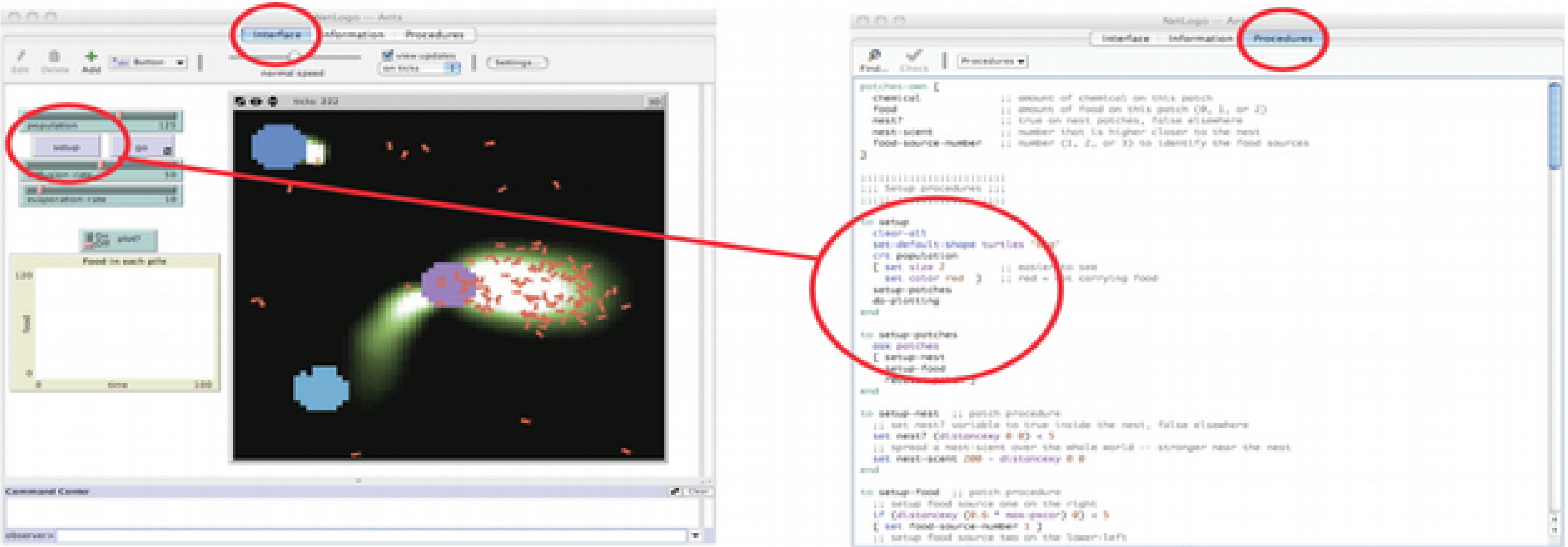
Let’s try model *Ants*

- File → Models Library → Biology → Ants



© 2018

Interface ↔ Code



```
patches-own [
  chemical             ;; amount of chemical on this patch
  food                 ;; amount of food on this patch (0, 1, or 2)
  nest?                ;; true on nest patches, false elsewhere
  nest-scent           ;; number that is higher closer to the nest
  food-source-number   ;; number (1, 2, or 3) to identify the food sources
]

;;;;;;;;;;;;;;;;;;;;;;;;
;;; Setup procedures ;;;
;;;;;;;;;;;;;;;;;;;;;;;;

to setup
  clear-all
  set-default-shape turtles "bug"
  crt population
  [ set size 2         ;; easier to see
    set color red  ]   ;; red = not carrying food
  setup-patches
  do-plotting
end

to setup-patches
  ask patches
  [ setup-nest
    setup-food
    recolor-patch ]
end

to setup-nest  ;; patch procedure
  ;; set nest? variable to true inside the nest, false elsewhere
  set nest? (distancexy 0 0) < 5
  ;; spread a nest-scent over the whole world -- stronger near the nest
  set nest-scent 200 - distancexy 0 0
end

to setup-food  ;; patch procedure
  ;; setup food source one on the right
  if (distancexy (0.6 * max-pxcor) 0) < 5
  [ set food-source-number 1 ]
  ;; setup food source two on the lower-left
```

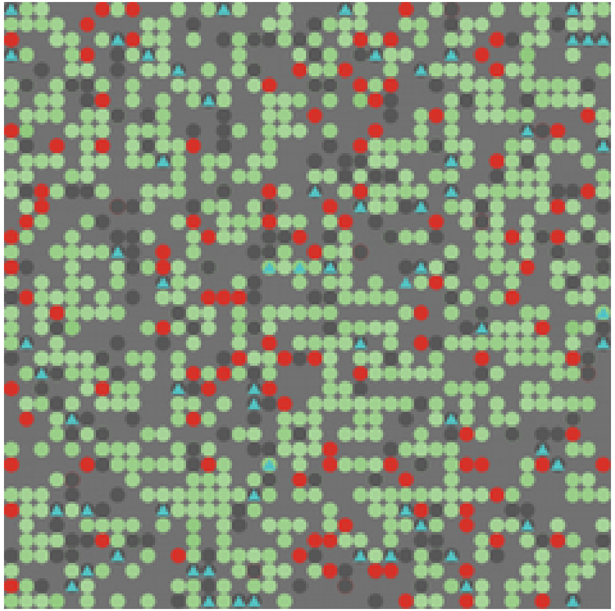
Agents, environment, observer

- Turtle = agents
- Patches = locations (grid cells)
- Observer = control agent

```
; Set the colour of the houses surrounding person fred:
ask person fred [
   ask neighbors4 with [ptype = "house"] [set pcolor blue]
]
```

Let’s try model *Rebellion*

- File → Models Library → Social Science → Rebellion



Epstein (2002)

Things to try

- Parameters
- Watch one agent
- 2D/3D visualization

- Write a reporter procedure that reports `true` when there is a rebellion, `false` during quiescent periods

More NetLogo

- GIS data
- System dynamics
- Distributed models
- Batch experiments
- Python, R interfaces

Summary

- Dedicated ABM vs. Java/Python library
- NetLogo: turtles, patches, observer

Outline

1. Socio-technical systems
2. Generative modelling
3. ABSS tools (hands on)
4. **Modelling process (hands on)**
5. Resources

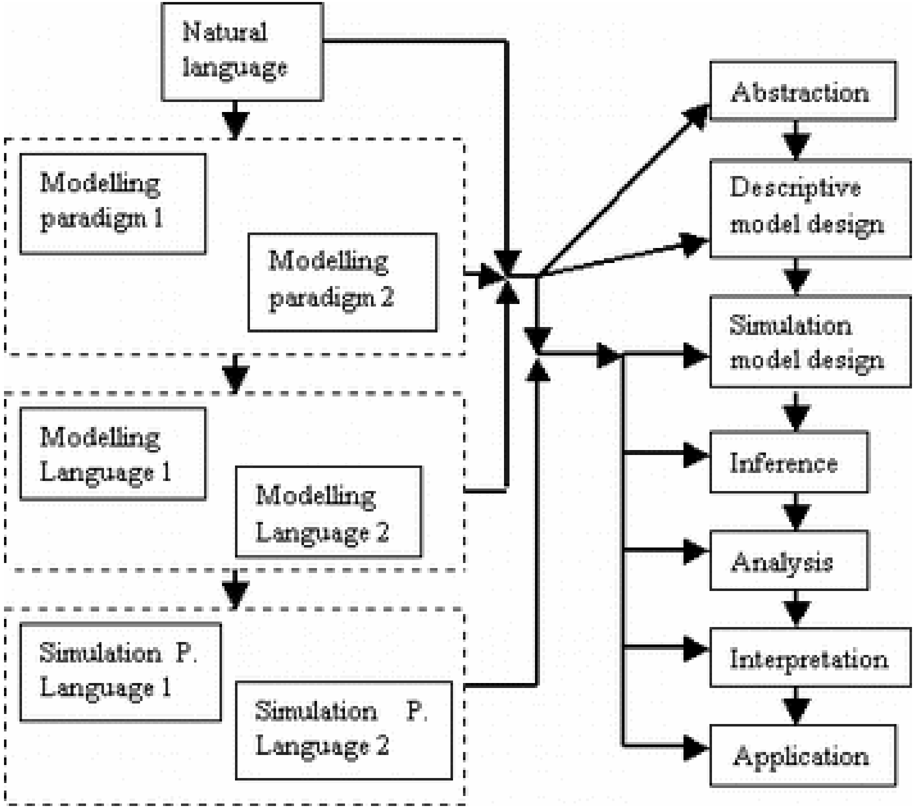
Suggested methodology

1. Purpose of simulation
2. Entities / actors
3. Data
4. High-level design
5. Detailed design
6. Implementation & calibration
7. Verification & validation

Purposes of simulation

- Understand
- Explore
- Predict
- Control
- Design
- Validate
- Perspective

Levels of abstraction



Teran (2004)

Modelling standards

- ODD (Overview, Design concepts, and Design details)
- AGENT UML
- openabm.org

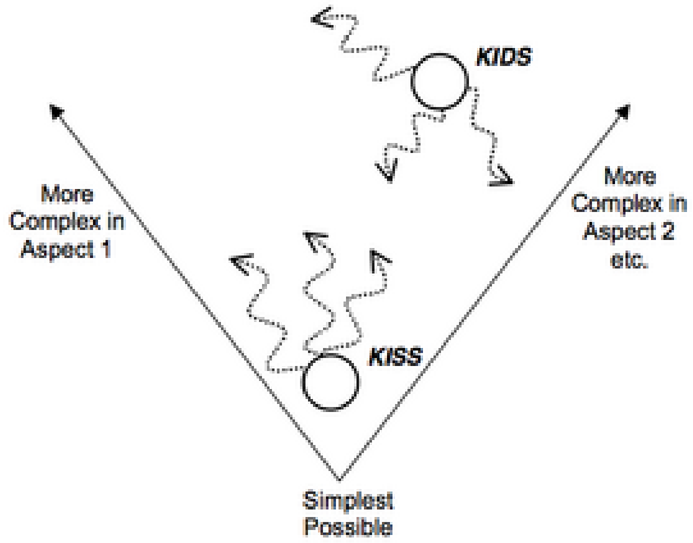
CoMSES | OpenABM
NETWORK

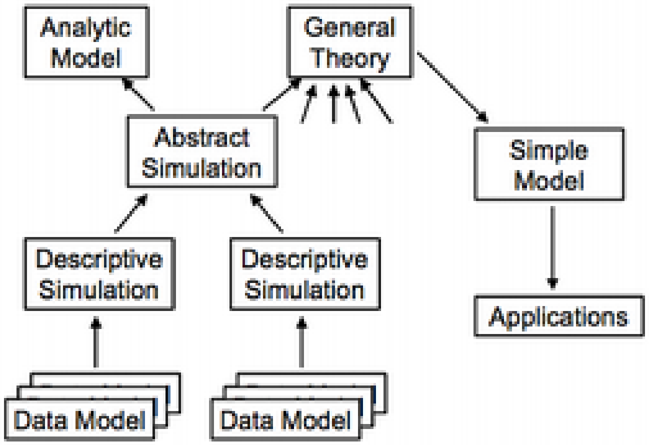
KISS vs KIDS

- **Simple**: abstract as much as possible, only expand the model when this is needed to explain and understand the phenomenon of interest

- **Descriptive**: start with a (complex) descriptive model, only simplify when this turns out to be justified

KISS vs KIDS





Edmonds and Moss (2004)

Verification & validation

- Verification = implementation correctly matches conceptual model
- Validation = conceptual model adequately matches reality

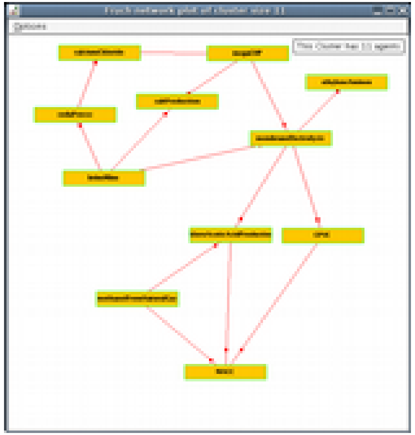
- How would you ensure these properties?

Verification

- Expert checks model, output
- Component verification
- Formal model checking
- Reasonableness under a variety of input parameter settings
- Interactive tracing
- Multiple implementations

“Model adequately matches reality”

- Replicative validity
- Predictive validity
- Structural validity



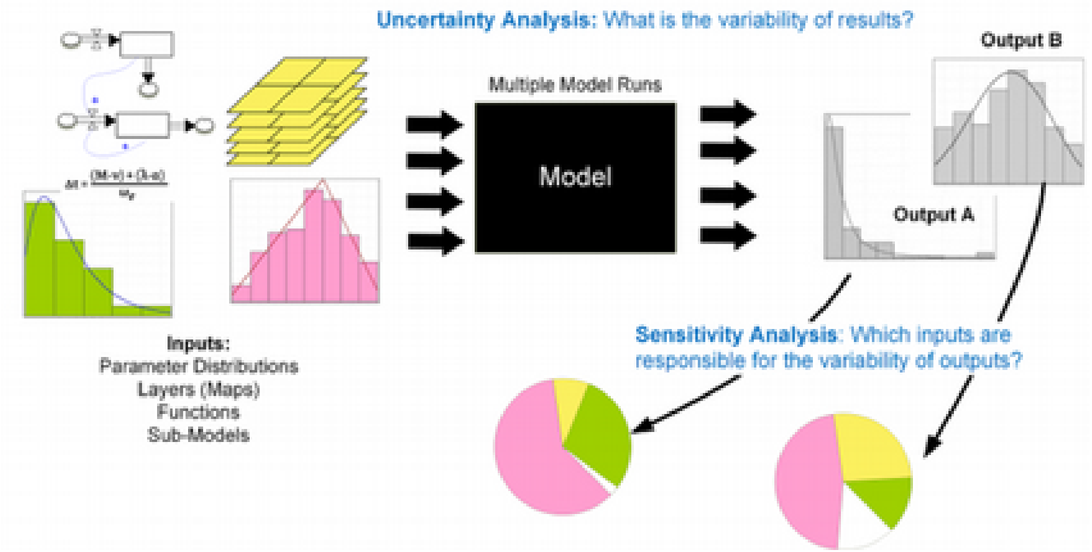
Validation

- Build a model that has high *face validity*
- Validate model assumptions
- Compare the model input-output transformations to corresponding input-output transformations for the real system

Sensitivity analysis

- Which input parameters most affect observed outputs/behaviours?
- How much do outputs depend on precise values of input parameters?

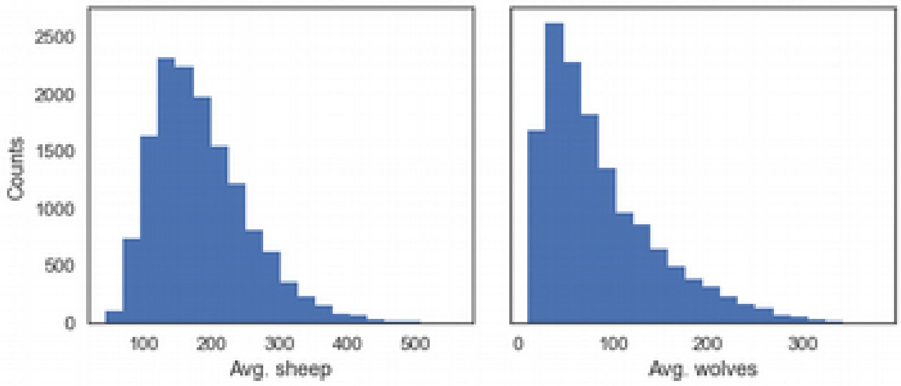
Output of sensitivity analysis

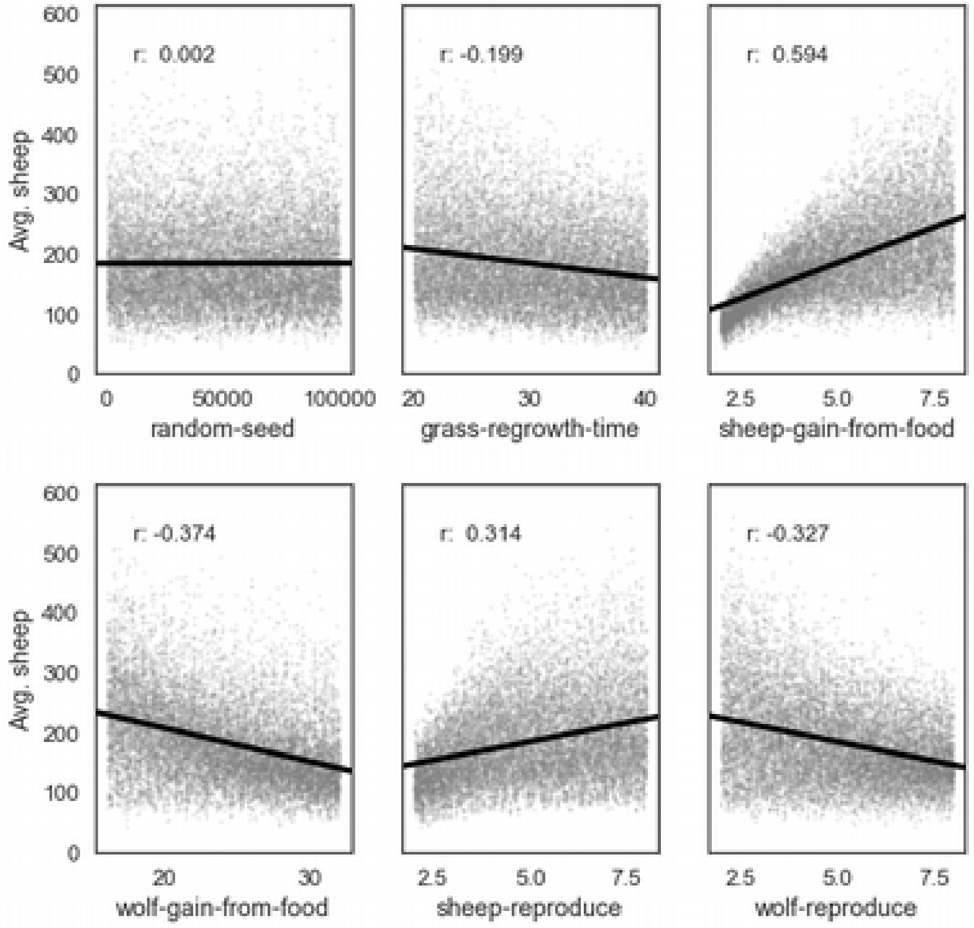


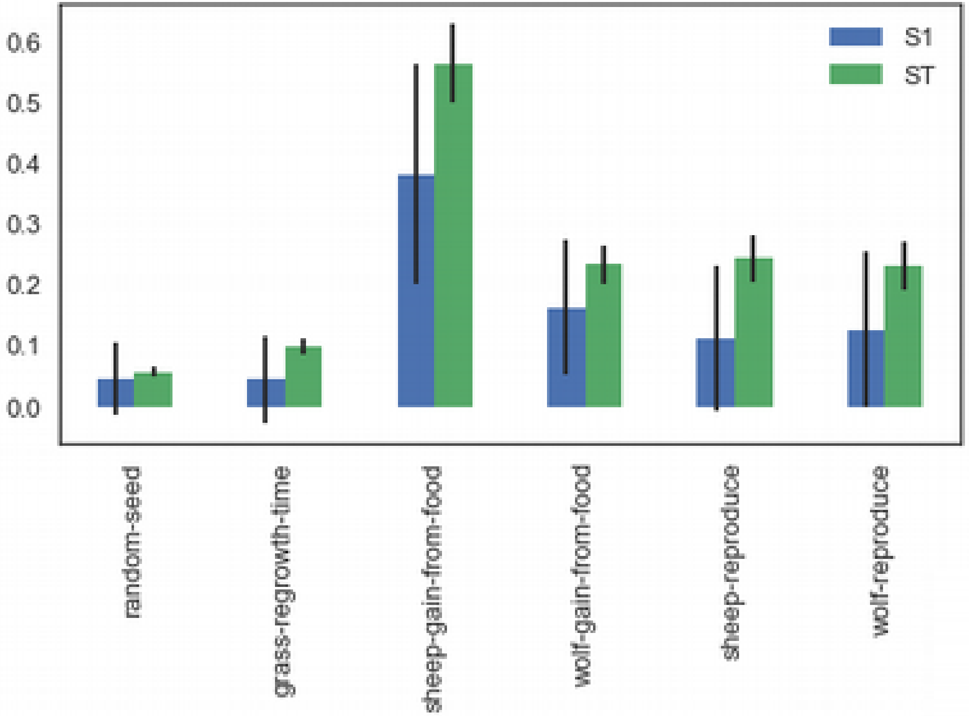
Uncertainty Analysis: What is the variability of results?
Output B
Multiple Model Runs
Model
Output A
Inputs:
Parameter Distributions
Layers (Maps)
Functions
Sub-Models
Sensitivity Analysis: Which inputs are responsible for the variability of outputs?

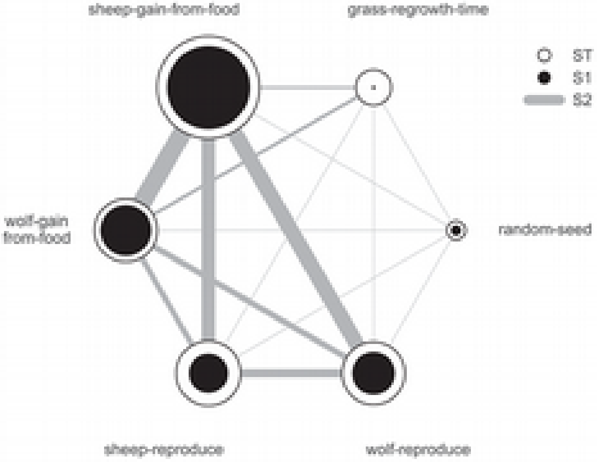
Sensitivity analysis in practice

```
problem = {
  'num_vars': 6,
  'names': ['random-seed', 'grass-regrowth-time', 'sheep-gain-from-food',
            'wolf-gain-from-food', 'sheep-reproduce', 'wolf-reproduce'],
  'bounds': [[1, 100000], [20., 40.], [2., 8.],
             [16., 32.], [2., 8.], [2., 8.]]
}

n = 1000
param_values = saltelli.sample(problem, n, calc_second_order=True)

# Setup, run, analyze
```


Counts
2500
2000
1500
1000
500
0
100
200
300
400
500
Avg. sheep
0
100
200
300
Avg. wolves

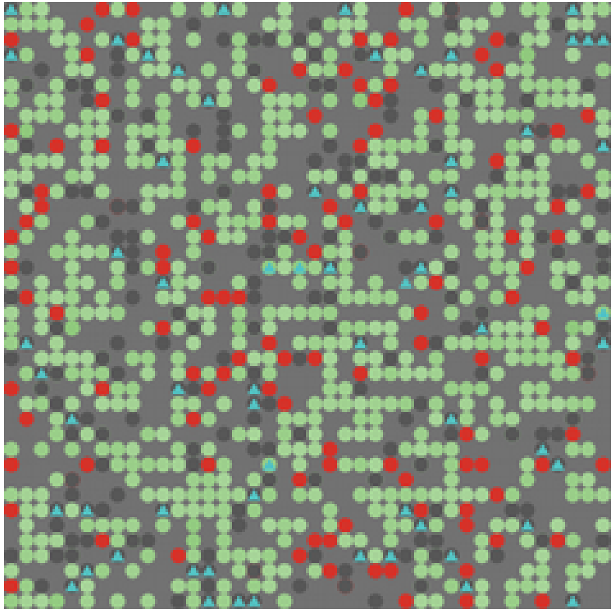

Avg. sheep
600
500
400
300
200
100
0
r: 0.002
0
50000
100000
random-seed
r: -0.199
20
30
40
grass-regrowth-time
r: 0.594
2.5
5.0
7.5
sheep-gain-from-food
Avg. sheep
r: -0.374
20
30
wolf-gain-from-food
r: 0.314
2.5
5.0
7.5
sheep-reproduce
r: -0.327
2.5
5.0
7.5
wolf-reproduce


S1
ST
0.6
0.5
0.4
0.3
0.2
0.1
0.0
random-seed
grass-regrowth-time
sheep-gain-from-food
wolf-gain-from-food
sheep-reproduce
wolf-reproduce


sheep-gain-from-food
grass-regrowth-time
○ ST
● S1
S2
wolf-gain
from-food
random-seed
sheep-reproduce
wolf-reproduce

Rebellion, revisited

- File → Models Library → Social Science → Rebellion



Epstein (2002)

Things to try now

- How sensitive is the model to parameter GOVERNMENT-LEGITIMACY?
- How would you validate this model?
- How would you build a KIDS version?

- Change the model s.t. each agent's `grievance` is influenced by the value of other nearby agents

Summary

- Garbage in, garbage out?
- Methodology is important
- Verification and validation – or stakeholders reject ABM

Outline

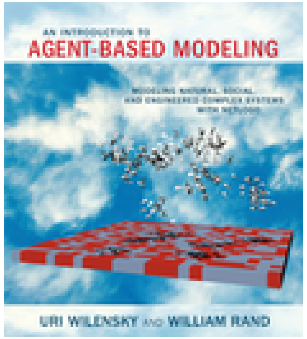
1. Socio-technical systems
2. Generative modelling
3. ABSS tools (hands on)
4. Modelling process (hands on)
5. **Resources**

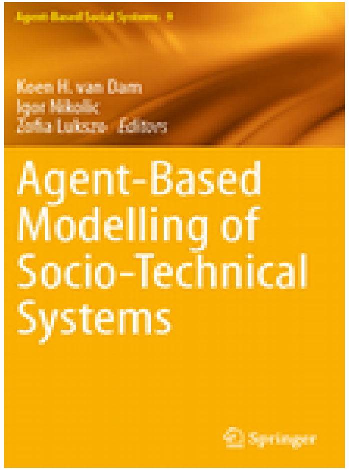
ABM platforms

- AnyLogic: www.anylogic.com
- GAMA: www.gama-platform.org
- MASON: cs.gmu.edu/~eclab/projects/mason/
- Mesa: www.github.com/projectmesa/mesa
- NetLogo: ccl.northwestern.edu/netlogo/
- Repast: repast.github.io/index.html
- SOIL: www.github.com/gsi-upm/soil

Recommended reading

- van Dam et al (eds), *Agent-Based Modelling of STS*, Springer, 2013
- Edmonds & Meyer, *Simulating Social Complexity*, Springer, 2013
- Edmonds & Moss, From KISS to KIDS, *MABS*, Springer, 2004
- Epstein, Agent-based Computational Models and Generative Social Science, *Complexity*, 1999
- Lee et al, Complexities of Agent-Based Modeling Output Analysis, *JASSS*, 2015
- Macal, Everything You Need to Know about ABMS, *J. Simulation*, 2016
- Teran, Understanding MABS and Social Sim., *JASSS*, 2004
- Wallach, Computational Social Science, *CACM*, 2018
- Wilensky & Rand, *Intro to ABM*, MIT Press, 2015
- JASSS journal: http://jasss.soc.surrey.ac.uk/JASSS.html


AN INTRODUCTION TO
AGENT-BASED MODELING
MODELING NATURAL, SOCIAL, AND ENGINEERED COMPLEX SYSTEMS WITH NETLOGO
URI WILENSKY AND WILLIAM RAND


Agent-Based Social Systems 9

Koen H. van Dam
Igor Nikolic
Zofia Lukszo Editors

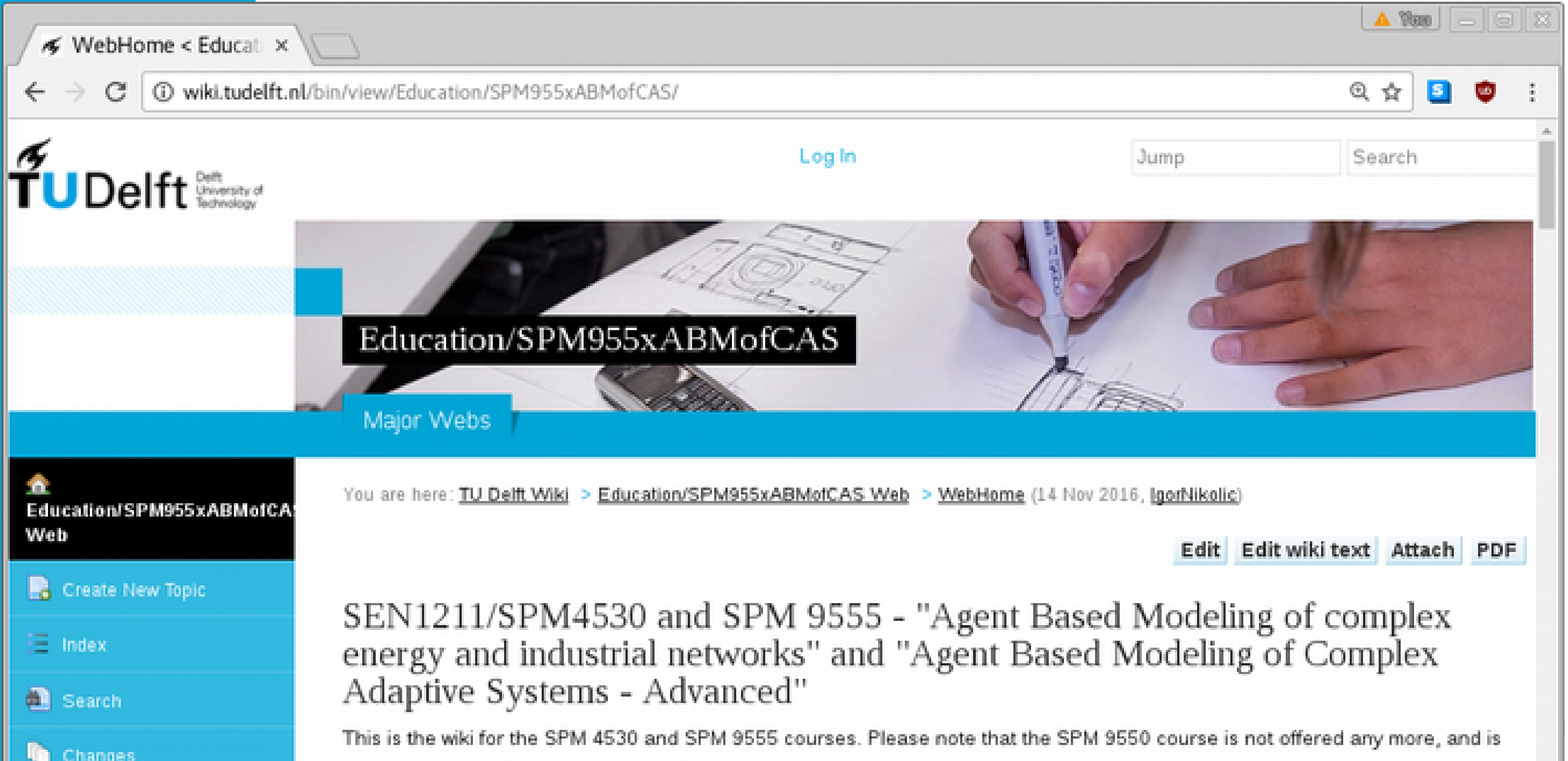
Agent-Based Modelling of Socio-Technical Systems

Springer

EASSS’15 tutorials

- Multiagent Simulation of Complex Systems
- MAS Prototyping Tool: First Steps with Netlogo
- Agents in Complex Networks

TU Delft OpenCourseWare



WebHome < Educat ×
wiki.tudelft.nl/bin/view/Education/SPM955xABMofCAS/
TUDelft Delft University of Technology
Log In
Jump
Search
Major Webs
Education/SPM955xABMofCAS Web
Create New Topic
Index
Search
Changes
Edit
Edit wiki text
Attach
PDF

Education/SPM955xABMofCAS

You are here: TU Delft Wiki > Education/SPM955xABMofCAS Web > WebHome (14 Nov 2016, IgorNikolic)

SEN1211/SPM4530 and SPM 9555 - "Agent Based Modeling of complex energy and industrial networks" and "Agent Based Modeling of Complex Adaptive Systems - Advanced"

This is the wiki for the SPM 4530 and SPM 9555 courses. Please note that the SPM 9550 course is not offered any more, and is

Colophon

- Contact: n.yorke-smith@tudelft.nl

- Thanks to: A. Evans, I. Nikolic, A. Sharpanskykh, G. Wurzer; U. Wilensky and the NetLogo team