# GIS-E4020: Advanced Spatial Analytics
# Lecture 5: Agent-Based Simulation with Spatial Data


- Jussi Nikander
  ~~Natural Resources Institute Finland~~
  Aalto University, Department of Built Environment




Aalto University
School of Engineering

# Contents of this Lecture

- Basics of Modeling and Simulation
- Agent-Based Models
- Agents and environments
- Design and Implementation of Agent-Based Models
- Verification, Calibration, and Validation

# Literature for this topic

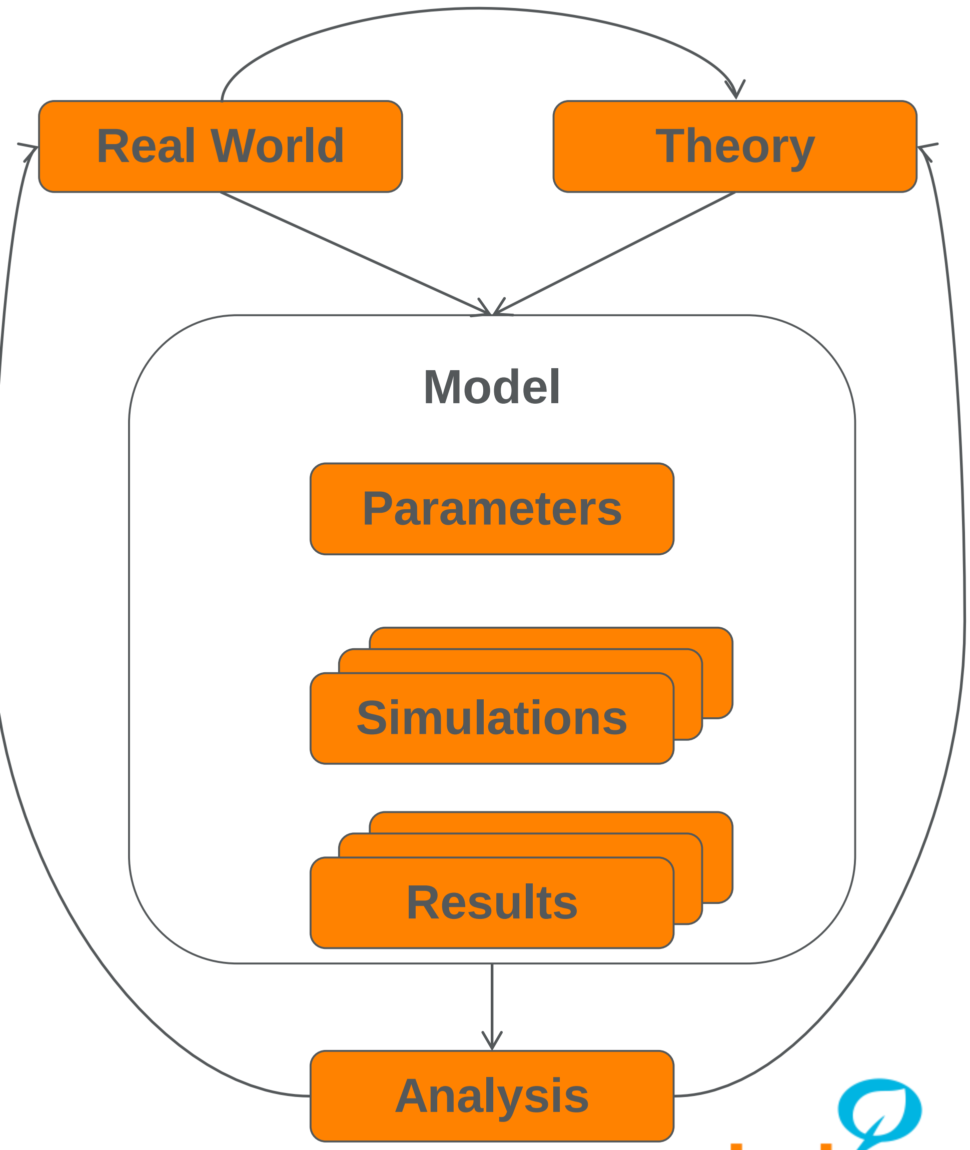

- Heppenstall, A.J., Crooks, A.T., See, L.M. and Batty, M. eds., 2011. *Agent-based models of geographical systems*. Springer Science & Business Media.
- Macal, C.M. and North, M.J., 2010. Tutorial on agent-based modelling and simulation. *Journal of simulation*, *4*(3), pp.151-162.
- Crooks, A.T., 2010. Constructing and implementing an agent-based model of residential segregation through vector GIS. *International Journal of Geographical Information Science*, *24*(5), pp.661-675.
- De Ligt, V., 2010. *Practical and conceptual issues in the use of agent-based modelling for disaster management* (Doctoral dissertation, University of Nottingham).
- http://gama-platform.org/

# Basics of Modeling and Simulation

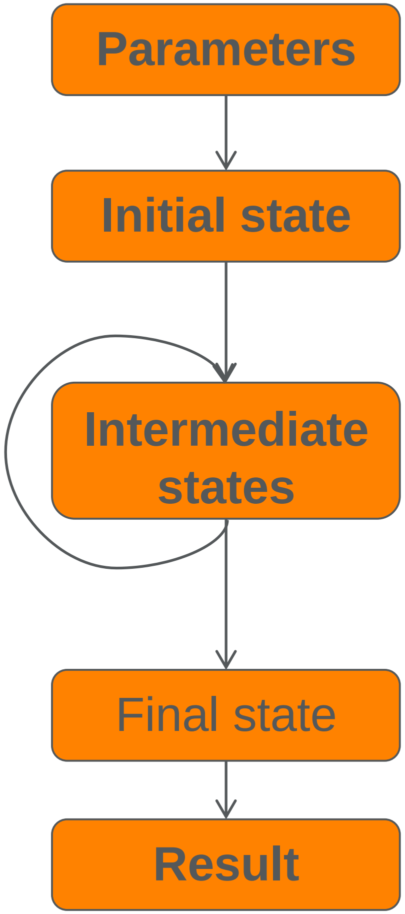
# Modeling and Simulation

- In research, theories are created to explain how an aspect of the real world works
- These theories can then be represented using models
- Simulation models can be used to simulate the theory
- Results of the simulation can be analyzed in order to find out how well the model reflects reality and the theory. If required, the theory, or the model, may be refined



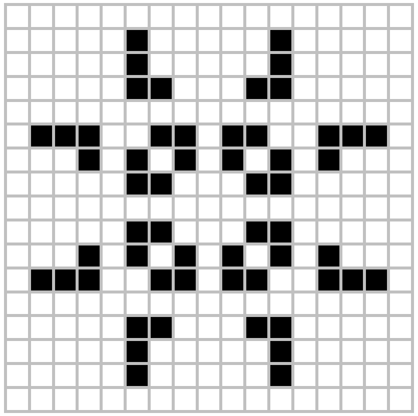
# Simulation

- A simulation often consists of:
- A set of parameters that are used to create a specific initial state
- a set of states that represent intermediate situations during the simulation
- A set of transition rules that dictate how elements of the simulation behave in a given state
  - Transitions often stand for time steps
- A condition that states when the simulation ends



# Simple Simulation Example: Game of Life

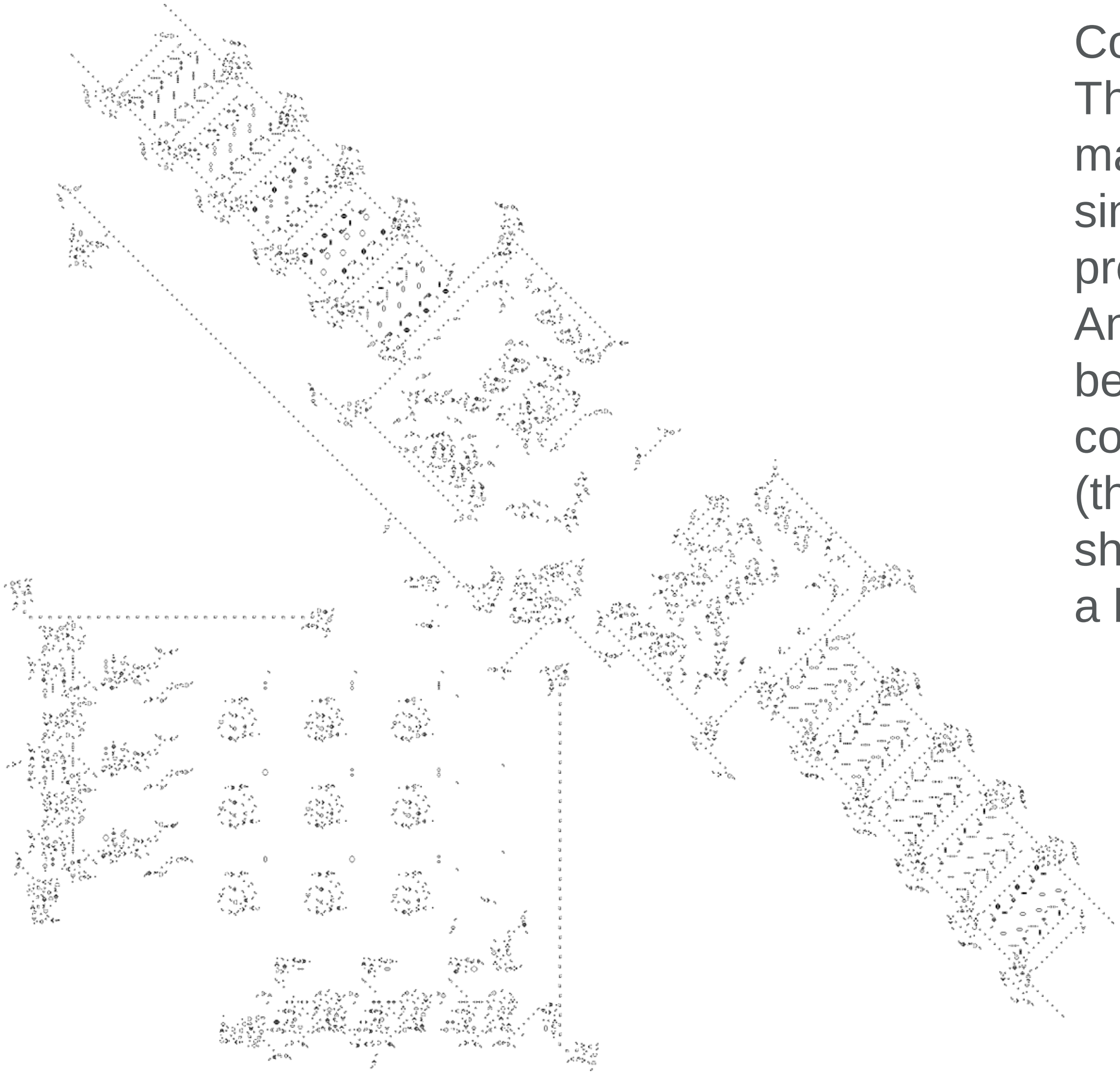
- Extremely simple rules
- Extremely expressive: Life is Turing-complete
  - You can build a simulation of a general computer using Life
- Despite this, it is mostly a mathematical toy
  - Most simulations using Life are extremely inefficient



# Simple Simulation Example: game of life

- The simulated world consists of a arbitrarily large two-dimensional orthogonal grid of square cells
  - Each cell has one of two states: dead or alive
  - Each cell has exactly eight neighbors (Moore neighborhood)
- At each step of the simulation:
  - A dead cell with exactly three live neighbors becomes alive
  - A live cell with exactly two or three live neighbors stays alive
  - A live cell with less than two or more than three live neighbors dies

# Turing Machine on Game of Life



Contains 36549 cells
This is not an universal machine, but one that simulates a specific program.
An universal machine has been constructed and contains ~250 000 cells (this page is too small to show it as anything but a blur)

# Abstracting Reality into Models

- A model is always a simplified representation of reality
- First computer models aimed at reproducing the behavior of a given real world phenomenon while staying as simple as possible
  - The goal was to find a minimalist explanation for the phenomenon
- Systems including human activity are intrisically complex
  - Aggregate ”average” behavior is not very good at explaining human systems
  - People are heterogenous individuals
- Simple models are good at explaining existing situations, but often bad at predicting future behavior
- Thus, there is a need for complex, heterogenous models

# Abstracting Reality into Models

- All models require abstract, formal representation of the system being modeled
- There are numerous ways to formalize this abstraction, depending on the model
  - In many types of model there is no need to **explicitly** formalize these abstractions
  - In the case of computational models the abstractions **need** to be formal and well-defined



NATURAL RESOURCES INSTITUTE FINLAND

# Formal Representation

- A model can consist, for example, of an *environment* and a *population* that acts in the environment
  - What is the environment and what is the population are rather involved questions and the answers are strongly dependent on the model
- In many cases we are modeling events that happen in the real world and are tied to specific locations
  - Thus many models are **spatial** in character: the environment tries to model the locations of the real world
- A dynamic, spatial model covers some space at certain time window. Typically the model is discretized into time steps

# Formal Representation: Environment and Population

- The space covered by the model is typically part of the environment
  - The space is indexed using, for example, coordinates for all locations (x,y)
  - Each location may have attributes that change over time
  - Locations may be a discretized field (raster) or distinct objects
- The environment is thus formalized into set E, where each location can be depicted as $E_{xyt}$
- On each time step, each location $E_{xyt}$ in the environment has a certain population
  - This view aggregates the population by location, whether this happens in the actual model depends
- Thus population P consists of populations at each location $P_{xyt}$

# Simulation, Feedback, and Model Dynamics

- A single step of a simulation thus consists of the state of the Environment and the Population at that step
- Simulation step $\boldsymbol{S_t} = \langle \boldsymbol{E_t}, \boldsymbol{P_t} \rangle = \left\langle \sum_{xy} \boldsymbol{E_{xyt}}, \sum \boldsymbol{P_{xyt}} \right\rangle$
  – A simulation step consists of the state of the simulation at that particular time step
- The whole simulation $\boldsymbol{S} = \sum_t \boldsymbol{S_t}$
  – A whole simulation consists of all the time steps
- A dynamic model consists of a number of time steps, where the state of the environment and the population changes
  – The state of the simulation on time step t+1 depends
    - on the state of the simulation on time step t
    - on the actions the active elements of the simulation between the two time steps
- Thus, $\boldsymbol{S_{t+1}} = \boldsymbol{f}(\boldsymbol{S_t})$ that can also be written as $\langle \boldsymbol{E_{t+1}}, \boldsymbol{P_{t+1}} \rangle = \boldsymbol{f}(\langle \boldsymbol{E_t}, \boldsymbol{P_t} \rangle)$

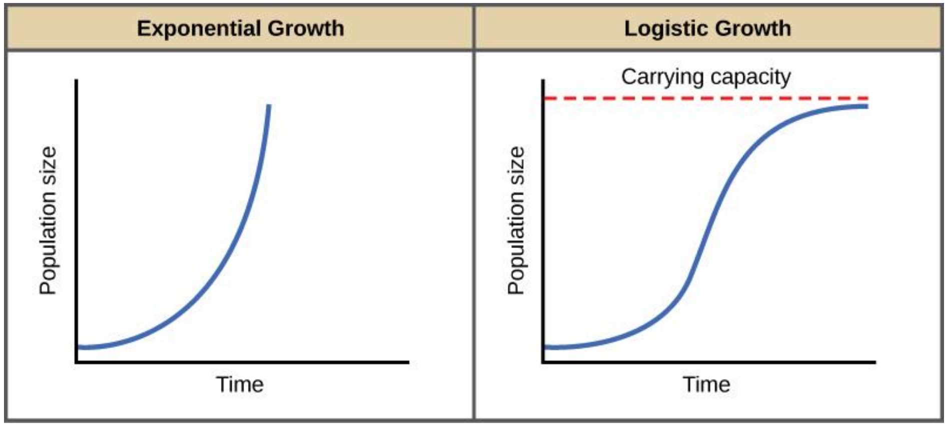
# Feedback, Dynamics, and Change Example

- A simple, equation-based population growth
  - Exponential model, unlimited growth: $P_{t+1} = P_t + \alpha P_t$
  - Exponential model with carrying capacity $\bar{P}$

    $$\begin{cases} P_{t+1} = P_t + \alpha P_t, P_t < \bar{P} \\ P_{t+1} = P_t, P_t \geq \bar{P} \end{cases}$$

  - Logistic model with carrying capacity

    $\bar{P}$: $P_{t+1} = P_t + \beta P_t \left(1 - \frac{P_t}{\bar{P}}\right)$








Luke
NATURAL RESOURCES INSTITUTE FINLAND

# External Factors in Change

- Often system is affected by external factors
  - Random chance, large upheavals, etc.
  - For example, logistic population growth can be rewritten as
    $$P_{t+1} = P_t + \beta P_t \left(1 - \frac{P_t}{\bar{P}}\right) + X_t$$
    where the last term is an external (positive or negative or even random) factor
- Sufficiently large external factors may affect the equilibrium of the system

# Group work, task 1

- As part of this lecture, you will do a group work
  - The group work is done iteratively, in stages
    - Between stages I'll talk (and you'll hopefully ask questions when needed)
  - The outcome for the group work is a poster
- At the end of the lecture, we'll report on the group work using the gallery walk method: creating new groups that contain member(s) from each group
  - Then you'll go through each poster, and the group representative(s) shortly describe(s) what you've done

- Divide into groups of equal size
- Each group should move together, so they'll have easier time working
- Each group can come and fetch materials (poster, markers, post-its)

# Group exercise: task 2

- Decide on a simple model you'll design here
- What's the goal of your simulation?
  - What will you simulate?
  - What's the population?
  - What is the environment? (please use a spatial environment)

- Possible topics could include rabbits and foxes (or generally predators and prey), fire or rescue services, animal migration, human movements (daily movement, moving houses, etc.), insect colonies, etc.
- When you've decided on what you'll model, consult me so we don't get people doing **exactly** the same thing
- Let's spend around 5 minutes on this

# Agent-Based Models

# What Are Agent-Based Models?

- Agent-based model consists of
  - A collection of **agents**, an **environment**, and **rules** that govern the **interaction** between the agents and the environment
  - The *population* in the model consists of agents
- In an agent-based model agents are the primary proactive elements
  - Meaning: agents are the parts of the model that actively DO things
  - The environment reacts to agent activity and is otherwise passive

# What are Agents?

- Agents are autonomous entities acting inside a model
  - Agents have a **behavior** that controls its actions, typically expressed as a set of rules
  - Agents **interact** with other elements of the model, including other agents. The interaction affects their future behavior
  - Interaction requires a set of **relationships** between agents that govern which agents a specific agent is capable of interacting with
- Agents typically act inside a given *environment*, which affects the agent's behavior
- It should be noted that there is no one single definition of what agents and agent-based models are

# What Is the Environment?

- The agents typically act in an environment that they can affect
- The environment consists of those aspects of the model that are not implemented as agents
  - What exactly is environment depends a lot
- In many cases environmental aspects are elements of the simulation that the agents can affect
  - These elements are typically otherwise passive

# Time, Space, and Agent-Based Models

- Population in an Agent-Based Model consists of agents
- The agents act in the environment, which may react to agent activity
  - In each time step, the state of the agent depends on the agent's previous state, the states of the agents it interacts with, and the state of the local environment
  - In each time step, the state of a location depends on the state of the same location on the previous time step as well as all agents that affect the location
- For an agent $\boldsymbol{A_i}$ on time step $\boldsymbol{t+1}$: $\boldsymbol{A_{i,t+1} = f(A_{i,t}, E_{i,t}, \sum_{k \neq i} A_{k,t})}$
  - $\boldsymbol{E_{i,t}}$ is the environment at the agent's location
  - $\boldsymbol{A_{k,t}}$ is another agent this agent interacts with
- For a location $\boldsymbol{E}$ in the environment: $\boldsymbol{E_{i,t+1} = f(E_{i,t}, Z_{i,t}, \sum A_{i,t})}$
  - $\boldsymbol{Z_{i,t}}$ is the neighborhood of the location
  - $\boldsymbol{A_{i,t}}$ is an agent interacting with the location

# What Are Agents: An Example

- Prey agents represent members of a prey species (e.g. rabbits)
  - They try to avoid predators, find vegetation to eat, mate in certain conditions, etc.
- Predator agents represent members of a predator species (e.g. foxes)
  - The try to find prey to eat, mate in certain conditions, etc.
- The agents live in an environment that contains food for prey (vegetation) and possibly other characteristics
  - E.g. good locations for nests/lairs

# Group Exercise: task 3

- In groups: **design a simple model for your simulation**
  - The model should contain agents, and everything else you think is needed to give them a proper environment
  - The agents need some sort of behavior
  - Agents may interact with each other and the environment
- Please consider at least the following:
  - The environment
  - Agent behavior, interactions, and relationships
  - Time step length, environment size and resolution (if applicable)
- You have 15 minutes. Go grab a cup of coffee if you want. You don't need to finish the whole thing now (we'll continue and improve on the model later on)

# Emergent Behavior

- The results of an agent-based scenario depend on actions and interactions of the agents
- More complex behavior arises through agent interaction
  - This is called emergent behavior
- If there are problems in the model, the emergent behavior can be odd

Fire trucks are, in my opinion, the most annoying type of SimCity service vehicle because they can create huge traffic problems and their 'pack mentality' makes them ineffective at their primary job - fighting fires.
When a fire breaks out in your city ALL fire trucks are called to respond, so long as they're not on their tea break. ALL fire trucks will head towards that fire and will continue to head towards it until either the fire has been put out, or it has burned out. This is a serious problem because if a second fire breaks out in the meantime it has to wait until the first fire has gone before ANY of your fire trucks will go to it. There's no division of labor (i.e. all trucks go to fire no.1, all vehicles except 2 go to fire no.2, all except 2 of those go to fire no.3 etc).





Luke NATURAL RESOURCES INSTITUTE FINLAND

# Reasons for the Unexpected Behavior in this Case

- Simplistic logic
  - All fire trucks respond to each fire
- Bad simplifications from reality
  - Other traffic cannot give way to emergency vehicles
  - Vehicles are unable to overtake each other
  - Vehicles are unable to reassess their route except at specific locations
  - All of these have probably been done in order to make the development simpler and the game run faster

Just like all SimCity vehicles, fire trucks use the same inadequate pathfinding decision making, not just for deciding which fires to go to (I think we've firmly established that they don't). They also spend a lot of time stuck in traffic whilst your buildings burn because for some utterly retarded reason ROAD TRAFFIC DOES NOT GIVE WAY TO EMERGENCY VEHICLES (head-meets-desk, why Maxis, why?).

# Agents and Environments in More Detail

# Characteristics of Agents

- Agents are **self-contained**
  - they can be easily distinguished and identified
- Agents are **autonomous**
  - An agent can function **independently** of its environment, and has a set of **behaviors** it follows
- Agents have a **state**
- Agents are **social**
  - They interact with other agents
- Agents may be **goal-oriented**
  - An agent may have some objective it tries to achieve
- Agents may be **heterogenous**
  - There may be several different types of agents in one simulation

# Characteristics of Agents Continued

- Agents typically have only **limited information**
  - They only know their surroundings instead of the whole environment
  - Thus, in at least some occasions, agents act according to bounded rationality
- An agent may be **connected** to a number of other agents
- An agent **sees** a part of the world around it
  - The agent has a neighborhood
- Agents may be **adaptive**
  - They may be able to **change the rules that dictate their behavior** through learning
- Agents may be **capable of movement**
  - Some researchers actually make this a requirement

# Agent Types

- It is possible to divide agents into different types according to their behavior
  - **Reactive** agents observe their surroundings and react according to their behavior rules
  - **Goal-oriented** agents attempt to accomplish something
  - **Adaptative** agents are able to change their behavior rules
- Categorizations vary

# What Is the Environment?

- The agents typically act in an **environment** that they can affect
- The environment consists of those aspects of the model that are not implemented as agents
  - What exactly is the environment varies **considerably**
- In many cases environmental aspects are elements of the simulation that the agents can affect
  - These elements are typically otherwise passive
  - The difference between environment and reactive agents can be vague
- The environment can also define a **neighborhood** for each location and agent

# Examples of Different Types of Neighborhoods

- Regular grid (von Neumann and Moore -neighborhoods)
- Euclidean space (with specified distance)
- Network model (explicit relations between locations / agents)
- Geographic space (agents work in a model of the real world; spatial attributes may affect agent behavior – for example the neighborhood of a specific location can be defined by the viewshed)

# Elements of the Environment

- What the environment is like depends entirely on the application area
  - City, forest, desert, empty space, road network, geographic data set with numerous attributes, network of social relationships, etc
- The environment is typically passive and static for the most part, but can include dynamic elements
  - Weather, growing plants, etc.
  - What is a part of the environment and what are agents is primarily an implementation decision
  - If it moves or can directly affect things around it, it probably should be an agent

# Advantages of Agent-Based Models

- Good for heterogenous individual-based simulation
  - Lots of real world entities, entities have different characteristics
- Can easily be used to explore phenomena
- Captures emergent phenomena
- Ability to simulate complex, adaptive, interactive, non-linear behavior
  - Perhaps better than aggregate models
- Easily enables stochastic modeling
  - Randomness can be applied to specific elements of the model
- Easily maps to the object-oriented programming paradigm

# Limitations of Agent-Based Models

- Consumes a lot of resources
  - Many agents, complex agents, several runs, etc.
- Not suitable for working with aggregates
- Agent behavior may not be a good indicator for real-world behavior
  - Recall the fire trucks in Sim City
  - Simplifications done to create the model may lose important details of the real-world phenomenon

# Group exercise task 4: Characteristics of Agents and Environments

- Consider your model:
  - Which characteristics the agents in your model include? Are some characteristics that should have been included missing? If there are, improve on your model
  - What is the environment in your model like? Based on this new information, how would you improve it?
  - Are there other things missing?
- Let’s use at least 5-10 minutes, but we can use a longer time if needed

# Design and Implementation of Agent-Based Models

# Designing an Agent-Based Model

- Remember that you are designing an experiment!
  - The purpose of the experiment
  - Research questions
  - Literature review
  - Selection of simulation system
- Serious model design can start only after these steps have been carried out

# The Main Components of Agents and Environments

- The main components and abilities of agents
  - Perception: ability to observe the environment
  - Abilities: a set of actions the agent can do
  - Memory: the Agent's knowledge base
  - Status: the current state of the agent
  - Policy: the behavior of the agent
- The components of the environment
  - The area (if applicable, here we assume it is)
  - A set of features
  - A set of attributes for each location/feature
  - Metric relations
  - Topological relations

# The Components, The Design, and The Tool

- What is the model about?
- The size and the number of runs
  - What will be done with the result data?
- What are the agents?
  - Perception, abilities, memory, status, policy
  - Variables, methods, inputs, outputs, common features, specific variants, etc.
- What is the environment?
  - Area, features, attributes, neighborhoods, topology
- With a serious ABM tool, what you’re doing is very close to designing a computer program using the object oriented paradigm
- The importance of domain knowledge cannot, however, be emphasized enough

# Designing an Agent-Based System

- Design takes time
- Design takes skill
- The earlier a problem is noticed and fixed, the less resources it takes to fix it
  - Meaning that if a problem is noticed during the design phase, the resources required to fix it are much smaller than if it is caught during the implementation

# Implementing an Agent-Based System

- Choice of a tool will always have a large effect on the design and implementation
- Basic ABM implementation can typically be divided into two large parts:
- **Experiment setup**
  - Load relevant input data and parameters
  - Create the environment and the agents
  - Give each element in the simulation the proper initial state
- **Experiment dynamics**
  - Agent behavior
  - Ways for the agent to perceive and interact with other elements of the simulation
  - The termination condition of the simulation

# Verification, Validation, and Calibration of Agent-Based Systems

# Verification, Validation, and Calibration

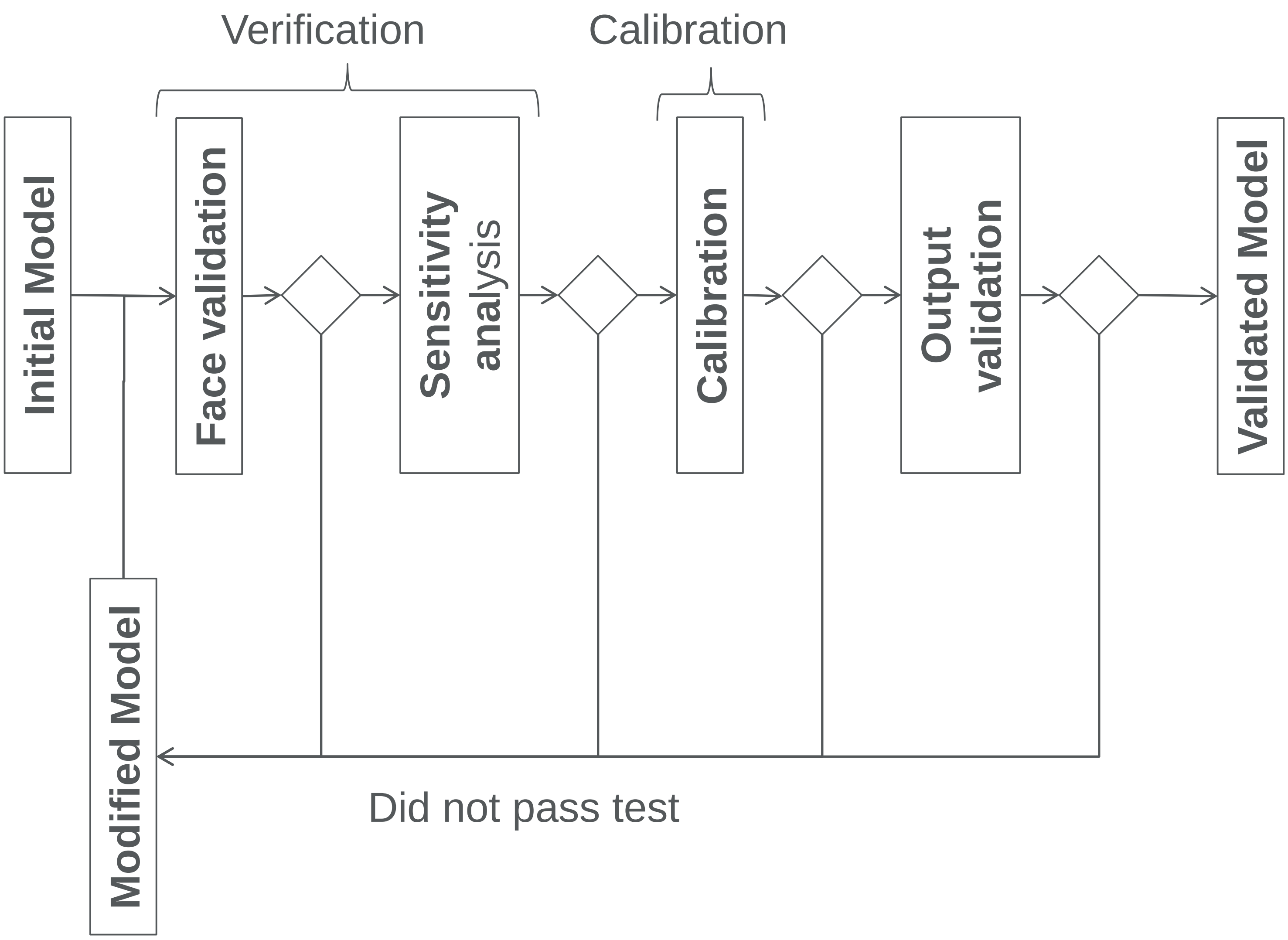
- A simulation model needs to give results that sufficiently accurately reflect the phenomenon being modeled
- Depending on the simulation checking whether the simulation is accurate can vary from easy to extremely difficult
- We can observe how rabbits and foxes interact in the wild
  - We can observe rabbits and foxes, however, we cannot observe all rabbits and foxes, nor can we observe them all the time, so details often need to be inferred
  - Local accuracy of weather simulation is easy enough to ascertain by looking out of the window; global accuracy of weather simulation is significantly more challenging to ascertain
  - Accuracy of a supernova creation mechanism simulation is very difficult to ascertain

# Verification, Validation, and Calibration

- **Validation** is the process of making sure that an implemented model matches the real world
- **Verification** is the process of making sure that an implemented model matches its design
- **Calibration** is the process of fine-tuning a model to a particular context

- **Theory** reflects a real world phenomenon
- A **design** reflects a theory and **implementation** reflects the design
- An **implementation** is used for specific task
- **Validation** is concerned with the real world, **verification** with design, and **calibration** with a specific task

# The Validation Process: One Approach



# Face Validation

- Cursory, experience-based checking whether the system seems to work
  - System looks like it works, but has not been shown to work
- Can be divided into smaller elements
  - Animation assessment
  - Immersive assessment
  - Output assessment
- Face validation can also include more sophisticated approaches than ”just looking”
  - E.g. statistical indicators between simulation and existing data can be a part of face validation
- Generally face validation is considered a subjective measure

# Sensitivity Analysis

- Sensitivity analysis in general is a process aimed to ascertain how uncertainty in inputs affects the uncertainty in the output
- For ABM it often means studying how the different input parameters affect the model behavior
- Often valid parameter values are dependent on each other
  - Sustainable number of foxes depends on the number of rabbits, which depends on foxes and vegetation
- With i parameters that can get j values each, there are $i^j$ possible combinations
  - Full exploration of the parameter space is infeasible if the number of parameters (or values) is large
- Typically only some of the (possible) parameters have a large effect on the simulation

# Calibration

- After important parameters have been discovered, the parameter values that best reflect a known or desired situation need to be discovered
  - For example, what values of plant growth, number of rabbits and number of foxes prevent the extintion of any of them and reflect a real world distribution
- If there are several parameters, exploring the whole value space is infeasible

# Parameter Optimization

- The task of finding most suitable parameters is an optimization problem
- Thus optimization methods may be used to discover good parameterizations
- There are numerous different optimization methods available for different situations
  - Linear programming, simulated annealing, genetic algorithms, tabu search, etc.

# Output Validation

- The goal of output validation is to measure how well the simulation represents the real world
- In this part real observed data must be used
  - By comparing simulation results to real situation we can ascertain the validity of the simulation
- This is perhaps the most important part of the validation process
- Same techniques can be used as in fitness measurement
  - Which measure is the best for which situation is a complex issue (that will not be discussed here)

# Group work task 5: Experimental setup and input

- What kind of experimental setup would you model have? What are trying to test with the model?
- What kind of input parameters (initial values) would you need?
- Consider how the initial values of different parameters in the simulation might affect how it works
  - E.g. what happens if there you change the (initial) number of agents?
- Can you think of ways the simulation might go wrong (produce unsuitable results)
- How would you try to find parameters that allow the simulation to succeed?

- Let’s spend at least 10 minute on this

# Methods for and approaches to validation

# Sequential Bifurcation

- Sequential Bifurcation is a way to do **sensetivity analysis**
- One way to find important parameters is to use the idea of divide and conquer
  - First look at the whole set of parameters
  - The divide it into smaller sets and see whether varying the values of parameters in these sets affects the simulation results
  - If varying parameters in a (sub)set does not affect the simulation results, we can conclude that these parameters are not important
  - If varying parameters affects results, the set is divided further
  - After multiple divisions important parameters are found

# Sequential Bifurcation

- Each parameter is simplified to binary
  - 1 corresponds to ”high” simulation output
  - 0 to ”low” simulation output
- Parameter effects to simulation output are simplified to a first-order polynomial
  - $y = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k$
  - Here $\beta_i$ represents the importance of parameter i
  - Task is to find which $\beta_i$ have large effect on output
- First establish that combination of all parameters is important
  - There is a difference between setting all parameters to 0 or to 1 (low < high)
  - Run simulation with all set high ($y_k$) and all set low ($y_0$)and compare the results





Luke
NATURAL RESOURCES
INSTITUTE FINLAND

# Sequential bifurcation

- Then divide parameters into two groups:
  $$\mathbf{a} = \{x_1, \dots, x_{k/2}\}, \boldsymbol{b} = \{x_{k+1/2}, \dots, x_k\}$$
- Now set parameters in a to high to get $y_{k/2}$
  – If $y_0 = y_{k/2}$ we can conclude that all these parameters are unimportant
  – Otherwise we need to recursively split the group
- Now set parameters in b to high to get $y_{k'/2}$
  – Do similar analysis to see whether they are important
- Three simulation runs may eliminate half of the parameters
- Maximum number of simulation runs is k*log(k), where k is the number of parameters (all parameters are important)
  – Actual number of runs depends on how many parameters are important

# Fitness Measures

- For **parameter optimization**, the simulation results must be compared to reference data
  - Real data or expected simulation results or something else that represents the desired outcome in a given situation
- The problem is then how to do the comparison, and how to measure the degree of difference
- There are plenty of measures
  - Mean squared error, Root mean squared error, Mean absolute percentage error, Median absolute percentage error, Relative Operating Characteristics, Confusion matrix, Kappa index, etc.

# Root Mean Squared Error

$$\boldsymbol{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}\left(x_{1,i} - x_{2,i}\right)^2}{n}}$$

- $x_{1,i} - x_{2,i}$ is the difference between variable i in simulation and reference data
- n is the number of variables
- RMSE is a weighted average of errors with large errors emphasized

# Relative Operating Characteristic

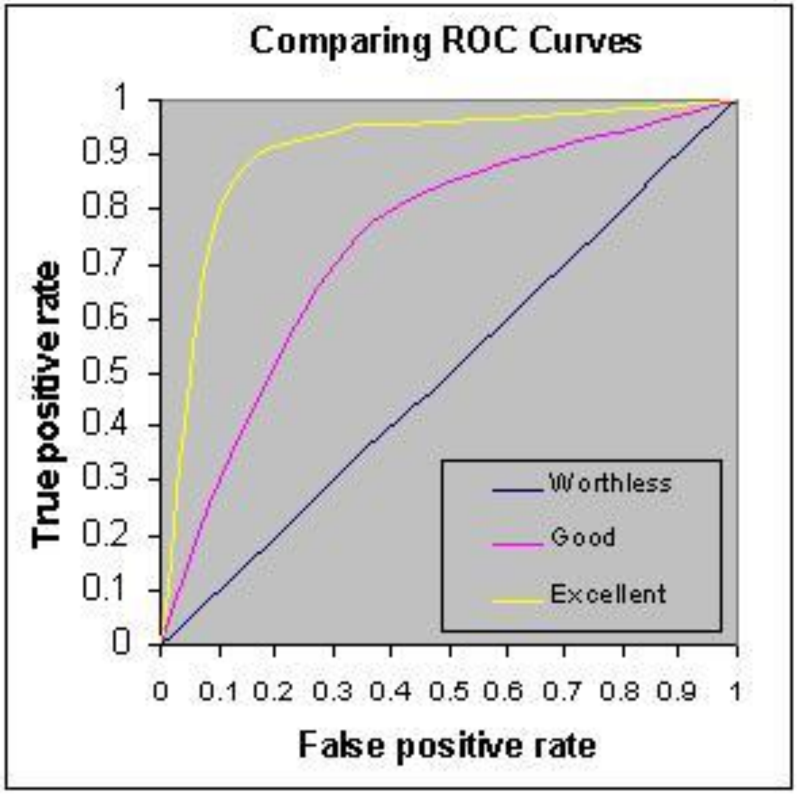
- **Output validation** measures how accurately the simulation results predict real world situation
  - ROC is one method of doing this
- Simulated results on a specific variable (e.g. which rabbits will survive?) are compared to real world results (which rabbits really did survive?)
- Individual results are then compared and categorized:
  - True positive (tp): something happens both in simulation and reality
  - False positive (fp): something happens in simulation, but not in reality
  - True negative (tn): something does not happen both in simulation and reality
  - False negative (fn): something does not happen in simulation, but happens in reality

# Relative Operating Characteristic

- From these values, we can calculate many measures
  - True positive rate (sensitivity) $\mathrm{TPR} = \frac{TP}{P} = \frac{TP}{TP+FN}$
  - False positive rate $FPR = \frac{FP}{N} = \frac{FP}{FP+TN}$
  - True negative rate (specificity) $TNR = \frac{TN}{N} = \frac{TN}{FP+TN}$
  - Etc.

# ROC Diagram

- ROC comparison is typically done by plotting TPR and FPR probabilities
- The larger the area under the curve (the faster TPR rises towards 1) the better the result
- In perfect test the whole area is under curve




Picture from http://gim.unmc.edu/dxtests/roc3.htm

# Group work: reporting

- Now, do you need more time to finalize your poster?
- When the posters are ready, we'll hang them on the classroom walls
- Then split into new groups, which include people from each of the poster groups
- Each group will now then move through each poster; the representative from the group will explain their work
  - You'll have about 10 minutes per poster
- Movement between posters will happen when I give the signal