Building Agent-Based Models

(Based on Rand & Wilensky, 2015)

* Model design
  + Phenomena-based vs. exploratory modeling?
    1. Phenomena-based = begin with known phenomena/characteristic pattern, create model capable of capture that pattern
    2. Exploratory = create agents, rules, and environments that embody characteristics of interest and explore patterns that emerge
  + Intellective vs. emulative modeling?
    1. Intellective = translating generalizable/narrative theory into computational model
    2. Emulative = developing model to fit a precise context, domain, or situation
  + Specify question?
    1. Start with a specific/focused question(s), build model to address question
       1. e.g., How does conflict emerge and get resolved in teams?
    2. Start with a broad/abstract question(s), clarify question as model is built
       1. e.g., General interest in team conflict, not sure what yet
  + Top-down vs. bottom-up design
    1. Top-down = work out agents, rules, environment and pseudocode before any programming takes place
    2. Bottom-up = start with general idea of what things are of interest and how they might interact, but begin to sketch out specific details as model is programmed
  + Tips & tricks/design principles
    1. Start simple
       1. Add complexity as needed to address question of interest
    2. Examine model and “trim fat” as you go
       1. Something that seemed like a good idea/way of doing things at first might change as model begins to be built out; don’t be afraid to eliminate redundancies
    3. Consider end-goal, verification/analytic strategy, and data output early and often
       1. Make sure you are building in and recording data that allows you to evaluate question of interest and explore potentially unanticipated patterns
* Consider model characteristics
  + What/who are the agents?
    1. What are the agent attributes (i.e., defining characteristics of agents)?
    2. What are the agent states (i.e., changing/emergent properties of agents)?
  + What is the environment and how is it defined?
    1. What are the characteristics of the environment?
    2. What are the environmental structures that organize agent interaction?
  + What are the processes/rules that govern agent action?
    1. What are the processes/rules that describe how agents interact with other agents?
    2. What are the processes/rules that describe how agents change or react over time?
  + Tips & tricks/design principles
    1. Don’t need to know/have all these considerations worked out in advance…but having a solid direction to start with is often very helpful
    2. Write your pseudocode in the program as comments, then work through programming each step of the pseudocode
       1. Ok to skip and/or revisit previous steps as model architecture is sketched out
    3. Organize your model code in a way that flows logically and is easily understood
       1. Will depend on specific software you program model in…however, many (simple) models I program in R follow a common structure. Presented in order from top to bottom of code:
          1. Model parameters – fixed and/or manipulable variables
          2. Model functions – custom-built functions that help with model initialization or that will be called by agents to carry out a particular behavior
          3. Initialization – create objects that will be used to hold/record data about agents, environment, output, etc. during simulation
          4. Model – the procedure and execution of model steps (i.e., pseudocode) that represent model behavior
          5. Data output – create, combine, and/or manipulate data output generated by simulation into data files useful for specific analyses
* Model evaluation and simulations
  + Identify parameters that will be fixed and those that will be manipulated
    1. Typically decided on basis of question of interest
  + Identify ranges/candidate values for manipulable parameters
    1. Can rely on empirical data for ranges that make sense if possible and model is programmed to accept those types of data
    2. More frequently, parameter values themselves are not of interest but rather their relative magnitude
       1. i.e., select values for variables that are relatively high, medium, and low
    3. Can also perform a “parameter sweep” to evaluate all possible combinations of parameters
       1. Can use Monte Carlo approach that places probability distributions over parameter values or sample randomly
  + Identify number of iterations to run (i.e., observation per “cell”)
    1. Models that contain stochasticity should be run (many) multiple times to facilitate drawing general insights about parameters
  + Tips & tricks/design principles
    1. Keep in mind the type of inferences you want to draw
       1. General main or interaction effects of parameters require varying parameters across range of values
       2. Exploring “dynamics” and pursuing explanations of process not often summarized neatly in summary/descriptive output data
          1. Consider ways to record data that allow you to plot how particular variables within a single run of the model change over time