The Artificial University ODD May 28, 2020

Summary

We provide an Overview, Design concepts and Details (ODD) document of out viral spread model for an artificial university.

# Purpose

The purpose of the model is to visualize and think throw how an outbreak would spread in a university by using a detailed and deliberate contact network based on the specified and predictable contacts that occur on a university schedule.. The purpose is also to test intervention and test strategies considered by universities. The model also considers the differences between student, faculty, and staff population.

# Entities, State Variables, and Scales

This model includes two kinds of entities: people, and ConnectionOfAgents entities.

## People

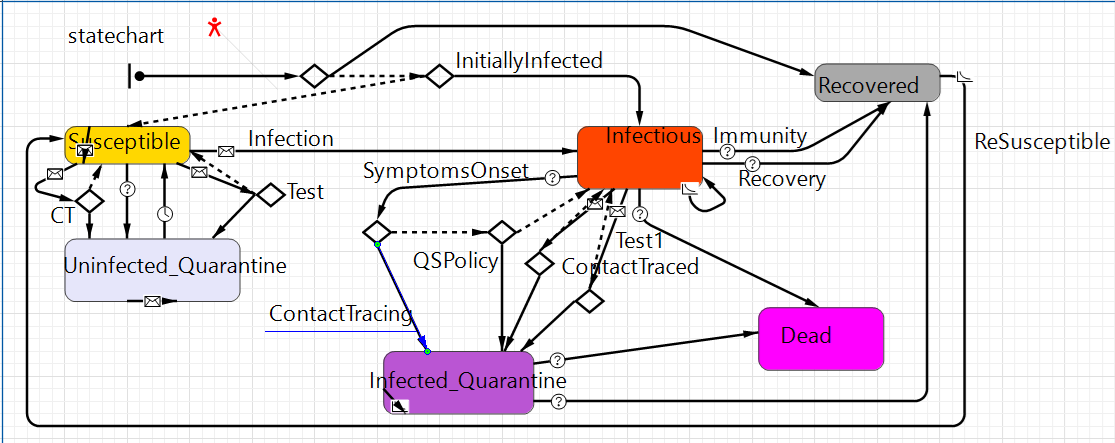
### Variables

People are initialized with several different variables, which are pulled randomly from an input distribution.

* Age
* ContactRate
* QS\_Compliance: compliance level associated with quarantining when infectious
* CT\_Compliance: compliance level associated with reporting symptoms to a contact tracing body
* Testing\_Compliance: compliance level associated with taking a viral or antibody test
* CT\_Traced\_Compliance: compliance level associated with isolating when contact traced
* Propensity to social distance
* Times per week going to gym
* Times per week going to campus events
* hasPreexistingCondition

### Susceptible-Infectious-Recovered State chart

All people follow a statechart for their infection and isolation status, pictured below.



People are initialized to the state of Susceptible, Infectious, or Recovered. A typical simulation run will start with a small percentage of the population initialized to Infectious, and this is the source of the infection spread.

#### Susceptible

Agents will generally begin in Susceptible. From susceptible, an agent may enter the Uninfected\_Quarantine state via being contact traced or having a falsely positive viral test. A susceptible agent may receive a “contact\_traced” message to signal that they have been contact traced. When receiving this message, the susceptible agent will transition to Uninfected\_Quarantine with a likelihood that is equal to their CT\_Traced\_Compliance value. A susceptible agent may receive a “Test” message, signaling they are being tested. An agent may refuse to test based on their Test\_Compliance, in which case they stay Susceptible. Since the agent is a true negative for the viral test, they will only transition to Uninfected\_Quarantine with a probability of 1 – Specificity for the viral test. Note that for true positive tests, we do not consider the delay in test results, since it is not reasonable that a person in the Susceptible state would have had a different viral test result in the time of the test delay, as agents spend considerable time in the Recovered state before becoming resusceptible.

A susceptible agent may also receive an “Infection” message to transition to the Infectious state.

#### Uninfected\_Quarantine

This state captures the real-world situation when a person isolates under the presumption of potentially carrying the virus, due to contact tracing or a false positive, when in actuality they do not. An agent will stay in this state for time defined by a global parameter CTQuarantineTime.

#### Infectious

The infectious state represents that an agent can spread the disease. At entry to the infection state, some values are calculated to determine the agent’s disease trajectory. First, an agent determines if it will recover immediately from immunity. The agent will recover immediately from immunity with a probability that is its immunity level (which is initially zero). If yes, the agent will immediately transition to recovered. Otherwise, the agent next needs to determine if it will die by checking it’s probability of death for its age group, and drawing from a uniform distribution between 0 and 1 to see if the agent will die. Regardless of whether or not the agent will die, the agent also needs to determine its illness duration and its time to show symptoms – note that an agent may show symptoms immediately (if its time to show symptoms is drawn to be zero) or it may not show symptoms at all (if the time to show symptoms is greater than the illness duration).

##### Infectious: Symptoms Onset

When becoming infectious, it is calculated when the onset of symptoms will occur. When this calculated time passes, agents will report their symptoms to a contact tracing body with a likelihood equal to their CT\_Compliance.

##### Infectious: Viral test

An infectious agent may receive a “Test” message to signal a viral test. An agent may refuse testing based on their Test\_Compliance. In order to model the delay in test results, we will have to use the agent’s state from the current model time minus the testing delay. In this way, the “Test” message does not really represent a current test happening, but rather the arrival of test results for a test that was administer previously. If the agent was infectious at the current model time minus the testing delay (when the test was “administered”), they will move to Infected\_Quarantine with a likelihood equal to the viral test sensitivity (a global parameter). If the agent was Susceptible at the current model time minus the testing delay, they will move to Infected\_Quarantine with a likelihood equal to 1 – Specificity; in this scenario, the agent actually had a false positive test, but became infected in the time of the test results delay. If the agent tests positive (meaning they will transition to Infected\_Quarantine), they will also do contact tracing.

##### Infectious: Contact Traced

An infectious agent may receive a “Contact\_Traced” message to signal they have been contact traced. They will transition to the Infected\_Quarantine state with a likelihood that is equal to their CT\_Traced\_Compliance.

##### Infectious: Death

If determined to die, the agent will transition to Dead after their illness duration.

#### Infected\_Quarantine

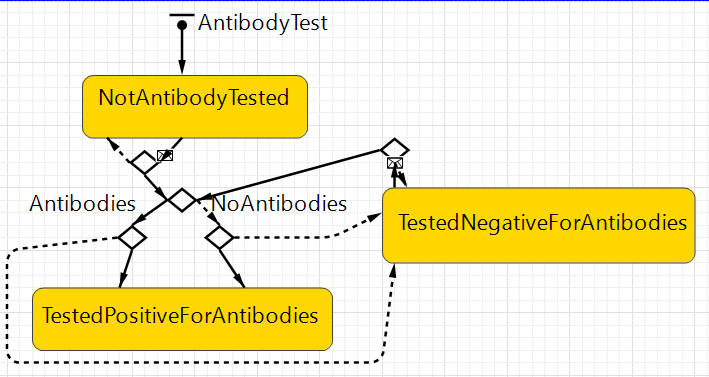
This state represents when someone is quarantining and they are infectious. Once they enter this state, an agent will stay in this state for the remainder of their illness duration. If determined to die when they transitioned to Infectious, the agent will then move to the Dead state. Otherwise, the agent will move to the Recovered state.

#### Recovered

When entering Recovered, the agent will record its entry time for use in antibody testing. The agent will also gain a immunity level. The agent will stay in the Recovered state until eventually becoming susceptible again, as specified by global parameters.

### Antibody testing state chart

For the purposes of our model, an agent has antibodies as soon as they enter the Recovered status in the SIR statechart. In a real world scenario, some time may need to pass before a person has antibodies. This delay can be added to our antibody testing delay to account for it. For instance, if the delay to get antibody test results is two days, and it takes thirty days of recovery for a person to have antibodies that can be registered on a test, one can set the global parameter for antibody testing to thirty two days.



All agents start in the state of NotAntibodyTested. The agent can receive an “AntibodyTest” message to signal antibody testing. The agent may refuse to test with likelihood 1 – Test\_Compliance, and remain in the NotAntibodyTested state. Otherwise, the agent will check if they had entered the Recovered state at time current – antibody test delay in order to determine the true outcome of the antibody test. From the true outcome, the agent will either test positive or negative based on the antibody test sensitivity and specificity: if the truth is that the agent has antibodies, they will test positive with likelihood of antibody test sensitivity; if the truth is that the agent does not have antibodies, they will test positive with likelihood of 1 – Specificity. If an agent tests positive, they will remain in the TestedPostiiveForAntibodies state for the remainder of the simulation. If an agent tests negative, they will go to the TestedNegativeForAntibodies state, and they may be tested again later in the simulation.

## People types

People are subtyped to Students, Faculty, and Staff. A person’s subtype will affect their initialization, such as having different age distributions or connections.

## ConnectionOfAgents

A connection of agents represents a group of agents who may potentially contact each other. By default, a connection has a frequency (number of days between occurrences) and an infectivity (likelihood of contact to result in infection).

### Infection daily event for ConnectionOfAgent

Once a day, all ConnectionOfAgent objects are polled. If the connection is active on the current model day, the connection runs through an Infection job. We loop over all agents in the ConnectionOfAgent entity, and, if an agent is in the infectious state, we calculate the likelihood that the agent is present at the ConnectionOfAgent entity on the current model day. The likelihood that an agent is present is usually 1, but this may be less than 1 for some kinds of connections. For instance, for the FitnessCenter ConnectionOfAgent entity, agents have a timesPerWeek value for using the gym, and so an agent’s likelihood of being present is timesPerWeek/7. If the agent is present, we run an infection job for that agent. Note that if the agent was in the Infected\_Quarantine state, we will do the same calculations, except the likelihood that an agent is present is multiplied by 1 - their QS\_Compliance. So, a highly compliant agent will attend few ConnectionOfAgent entities.

### Infection between agents

Once an infectious agent in a ConnectionOfAgent object is determined to be present at that connection, the agent may infect people in that connection. The agent will pick a number of other agents in the connection equal to their ContactRate. For each selected agent, the infectious agent will send an “Infected” message to the selected agent with a likelihood equal to InfectiousAgent.PropensityToSocialDistance \* SelectedAgent.PropensityToSocialDistance \* Connection.Infectivity.

## ConnectionOfAgent types

We will go over an overview of the kinds of connections.

### Course

A course contains a group of students and an instructor, which may be a Faculty or Student. A course has a schedule, which might be MondayWednesdayFriday, TuesdayThursday, or OneDay. The course is active only on days which align with its schedule. A course is also inactive if its instructor is symptomatic. Finally, a course may have splitClasses active, meaning that only a section of the class will be present when active. For MWF classes, the first third of students are presente on Mondays, the second third for Wednesdays, and so on. We similarly split TuTh and OneDay classes into two sections.

### Campus housing connections: Buildings, Floors, Suites, Bathrooms

Students in the world may share buildings, floors, suites, and/or bathrooms with other agents to represent campus housing. Suites and bathrooms are standard ConnectionOfAgent objects, with their own infectivities. Floors contain suites, and when getting the set of people on the floor, a Floor entity will query its contained suites. Similarly, Buildings contain Floors.

### Dining Hall

A dining hall entity represents the cafeteria where on-campus students eat. Dining halls are assigned buildings, and we assume that all students in that building will go to the dining hall.

### Campus Event

A campus event contains a group of people who go to campus events with a size equal to the campus event size. People who go to campus events have a timesPerWeek value to calculate the probability of being at that campus event on a given day.

### Sport Event

Sport events are normal events except people going to sport events have a probability of become infected externally.

### Staff to Student connection

Student facing staff members have a special ConnectionOfAgents entity where the staff member is connected to all students and will interact with a set number of students per day. Infections in a Staff to Student Connection may only involve the student facing staff member (the Staff can infect a student or vice versa). A student cannot infect another student through this kind of connection.

# Process Overview and Scheduling

An agent’s SIR state chart, when infectious, is checked multiple times a day to see if the agent has developed symptoms, recovered, or died.

Infections occur on a daily step. Once daily, all ConnectionOfAgent entities are polled to do infections if the connections are active in the current day.

Testing occurs on a daily step. There are global parameters for the maximum number of viral tests per day. Each day, this number of people are randomly selected for viral testing. By default, each person has a selection likelihood of 1, and selection likelihoods are normalized when drawing a test sample. Selection likelihoods are multiplied to represent varying test strategies, described below.

* Student-facing staff member boost: Student-facing staff members may be selected with a higher likelihood. This is accomplished by multiplying the selection likelihood for student-facing staff members.
* Preexisting conditions: similarly, people are initialized so that some have preexisting conditions, and these people may be prioritized.
* Age: People’s selection likelihood is multiplied by their likelihood of death based on their age.
* Antibody test result: People who have tested positive for antibodies have their selection likelihood divided by some multiple.

The selection likelihood multipliers are global parameters.

External infections occur on a daily step. There are global external infection rate parameters for people living on campus and those living off campus. Each person is randomly sent an Infection message based on their associated likelihood of external infection.

Antibody testing is done on a daily step. A global parameter sets the maximum number of daily antibody tests. Antibody testing population samples are selected randomly. First, agents who have not been tested are selected uniformly at random. If all agents have been tested, then those that tested negative are then selected uniformly at random. If all agents have tested positive, no antibody tests are performed.

# Design Concepts

## Convergence

We can observe a typical infection curve – most agents begin susceptible, some or most become infected, then the infected agents recover or die, then the virus dies when there are no more infected agents. There are also situations where the time to susceptibility is very low, where we might instead see a convergence/cycling of infected to susceptible to infected agents. We can toggle the NPIs to decrease or delay the number of infected people.

## Interaction

The interaction of agents is based on their initialized social network.

## Stochasticity

The university network is initialized with a fixed seed, so the network will be the same across simulation runs. All other aspects of stochasticity, i.e. infection mechanics, use a different pseudo-random number generator, so infection spread can be made to be separately random from the network generation.

# Initialization

As mentioned, agents are initialized by drawing their personal values from input distributions. Additionally, a percentage of agents are initialized to be in the Infectious state, a percentage are initialized to Recovered, and the rest are initialized to Susceptible.

The bulk of the initialization is the construction of the university network.

## TAU construction

In our implementation, this occurs in the method Main#initializeSchool. We have two university profiles: a large research university, and a small college. This binary is handled by the global Boolean parameter IRU (IsResearchUniversity). First, the people agents are all created, then the class entities. Students and faculty members are assigned to classes based on global parameters specifying the number of classes each kind of student takes, and the number of classes faculty members teach. Then, student housing entities are created and full time students who live on campus are assigned to suites and bathrooms. FitnessCenter timesPerWeek values are assigned to all people, and the fitness center entity is created. Student groups are created thusly: a student group size is pulled from a distribution, and students are randomly assigned to that group. This process is repeated until the number of students who are in a student group surpasses a global parameter. Campus events are handled similarly, except Staff and Faculty also attend campus events and they have their own attendance distributions. For student facing staff connections, some number of staff are selected to be student-facing, and for each of these, a student-facing staff connection entity is created. Finally, dining halls are created (the number of which is based on a global parameter) and buildings are assigned to dining halls, iteratively selecting buildings until the number of students assigned to that dining hall is greater than or equal to the number of on-campus students divided by the number of dining halls. Then, students are assigned to a sporting event entity.

## Stochasticity

# Input Data

* University parameters
  + Population
    - For each population group
      * Distribution for number of campus events attended per week
      * Distribution for number of times going to the fitness center per week
      * Age distribution, truncated normal
    - Faculty population group
      * Total count
      * Number who teach exactly one class
      * Number who teach exactly two classes
      * Number who teach exactly three classes
      * Number who share an office
    - Staff population group
      * Total count
      * Number who has a student-facing job
      * Number of times per day that student-facing staff interact with a student
      * Number who share an office
    - Students population group
      * Subgroups: {Part-time, Full-time} x {Freshman, Sophomore, Junior, Senior, Graduate}
        + Part time students are assumed not to live on campus
      * For each subgroup:
        + Number of discussion classes per semester
        + Number of non-discussion classes per semester
      * Percentage of students in a student group
  + Classes
    - Number of non-discussion classes
    - Number of discussion classes (which are taught by a graduate student)
    - Students will be assigned to classes such that classes tend to be filled by students of the same class.
  + On-Campus housing
    - Describe a building:
      * Number of floors
      * Number of suites per floor
      * Number of suites per bathroom
      * Number of students per suite
      * Up to two undergraduate classes that are in this building
    - For each building description, we can make N copies of that building. We can handle arbitrary amounts of building descriptions
  + Dining Halls
    - Number of dining halls. On-campus students will be split between dining halls roughly evenly.
  + Student groups
    - Percentage of students in a student group.
    - Student group size distribution
  + Campus Events
    - Distribution of campus event sizes
  + Sport events
    - Number of students who attend sports events
* Intervention parameters
  + Testing
    - Viral test capacity per day
    - Antibody test capacity per day
    - Viral and Antibody test result delays
    - People to test selected randomly on a daily bases. We can vary test strategy by increasing or decreasing likelihood of getting randomly selected for following populations
      * Student-facing staff members
      * More at-risk due to age
      * Having pre-existing condition (if provided a distribution of likelihood of having a preexisting condition)
      * People who have tested positive for antibodies
  + Classes
    - Hybrid class structure – part of class is virtual.
  + Disabling:
    - Student groups
    - Campus events over a certain size
    - Sports events
  + Fitness center
  + Contact tracing history: the traceback time when contact tracing
    - Contact tracing will be triggered by a positive viral test or when someone reports having symptoms
* Compliance parameters (provide a distribution for each agent to pull from at initialization)
  + Reporting symptoms compliance: When developing symptoms, this is the likelihood of an agent to report having symptoms to some contact-tracing body, which will initiate contact tracing
  + Traced compliance: When told by a contact tracing body to have contacted someone who tested positive or has symptoms, this is the likelihood to decide to isolate.
  + Quarantine compliance: When isolating, this is how much the agent actually isolates. I.e. when isolating, they may still perform 10% of their daily contacts.
  + Social distancing compliance
  + Central Tracking: We consider central tracking to mean that people are being strongly tracked, which can enforce Traced compliance and Quarantine compliance.
* Biological parameters
  + Infectivity for each kind of connection entity
  + Min/Max time to symptomatic
  + Min/Max illness duration
  + Min/Max resusceptibility time after recovery
  + Likelihood of death given age