Getting started with Agent Based modeling in AnyLogic

International System
Dynamics Conference 2006
Nijmegen

Workshop program

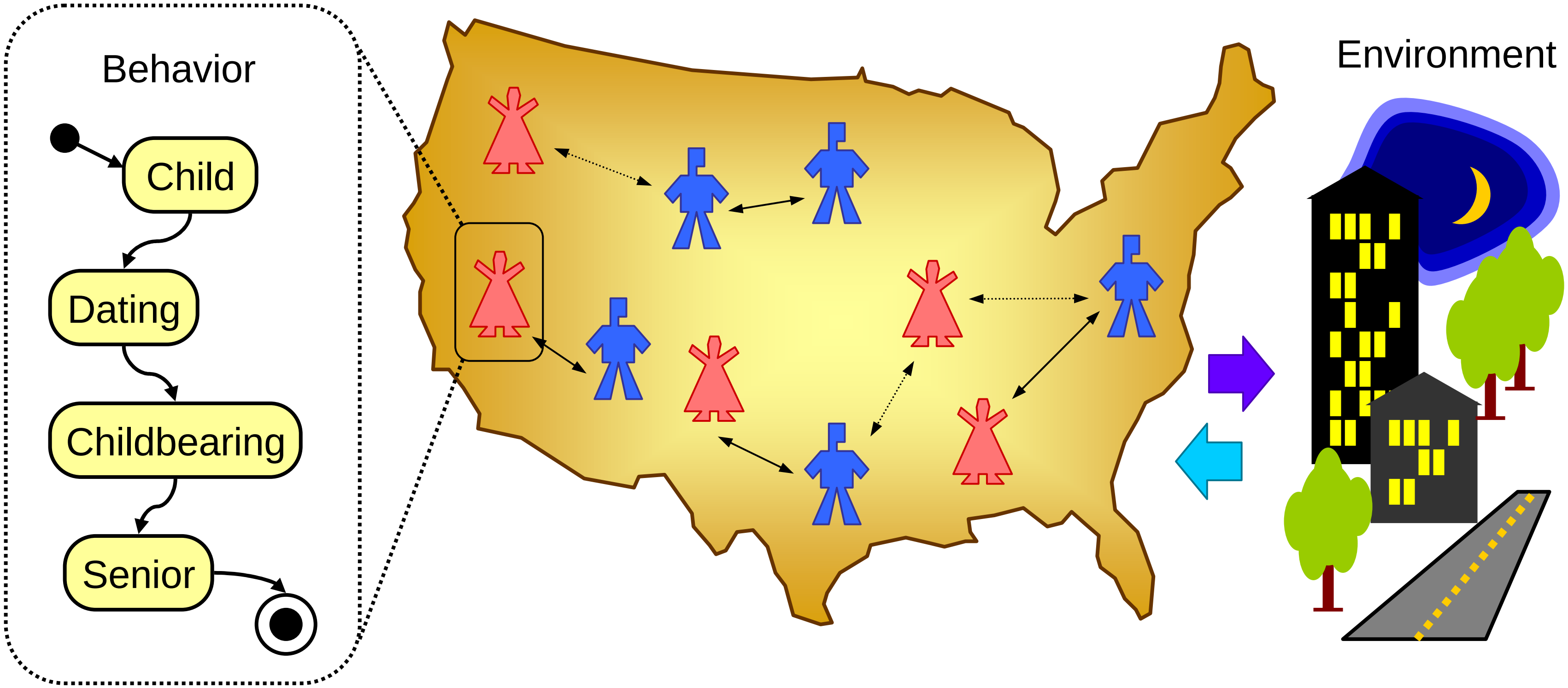
- Introduction: SD and Agent Based modeling
- Problem definition, SD model explained
- Build a simple AB model, discuss results
- Add physical space and social network to AB model, simulate and compare results
- Add a SD model of vaccination to the AB model and run the combined AB + SD model

Your CD:

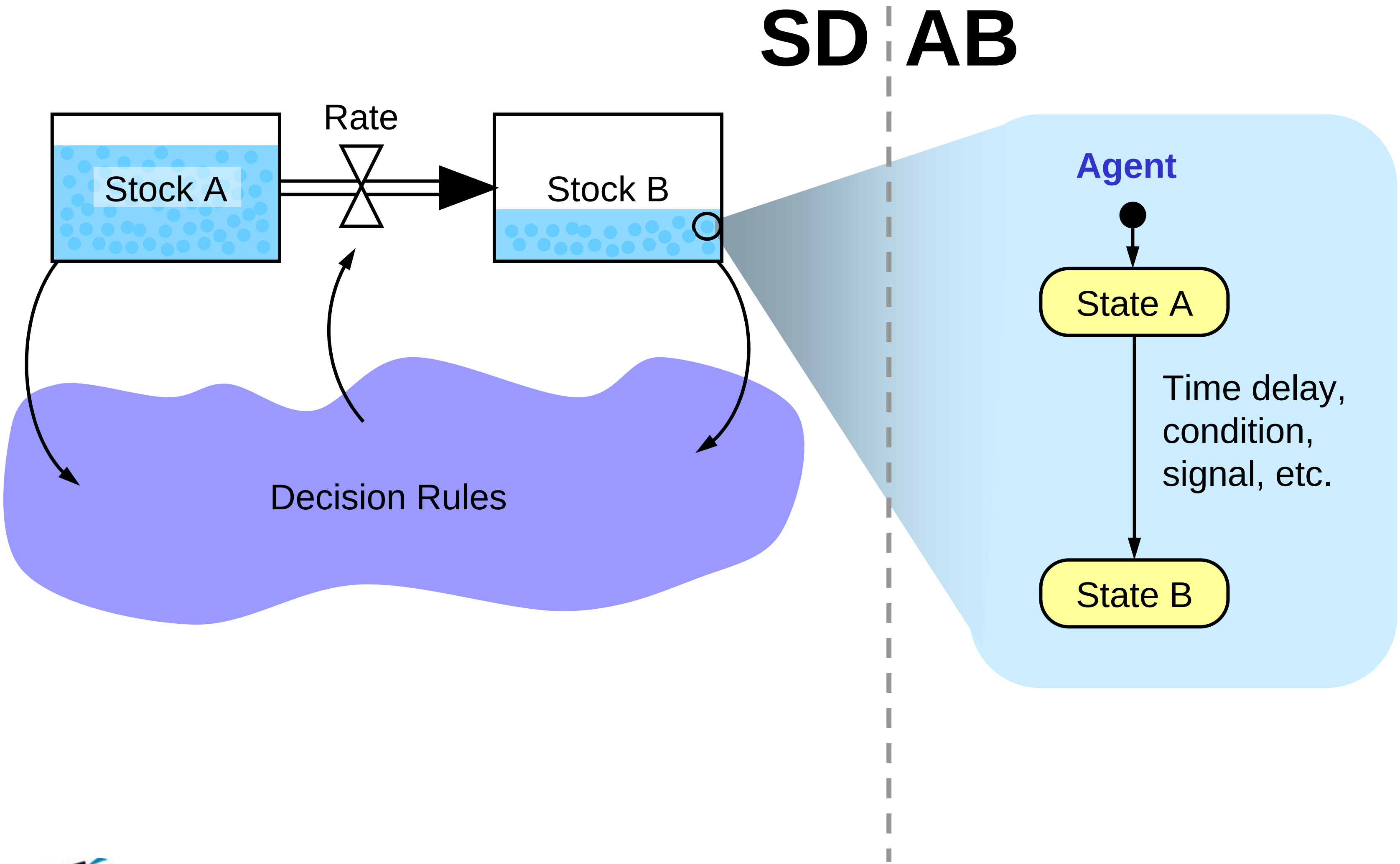
- AnyLogic 5 (anylogic-5.5.exe)
- AnyLogic 5 evaluation key (in file readme.txt)
- Today's workshop exercise model (phase by phase)
- This presentation

Agent Based Modeling Approach

- Agents. Individual behavior rules. Decentralized. Communication with each other and environment



Correspondence Between SD and AB



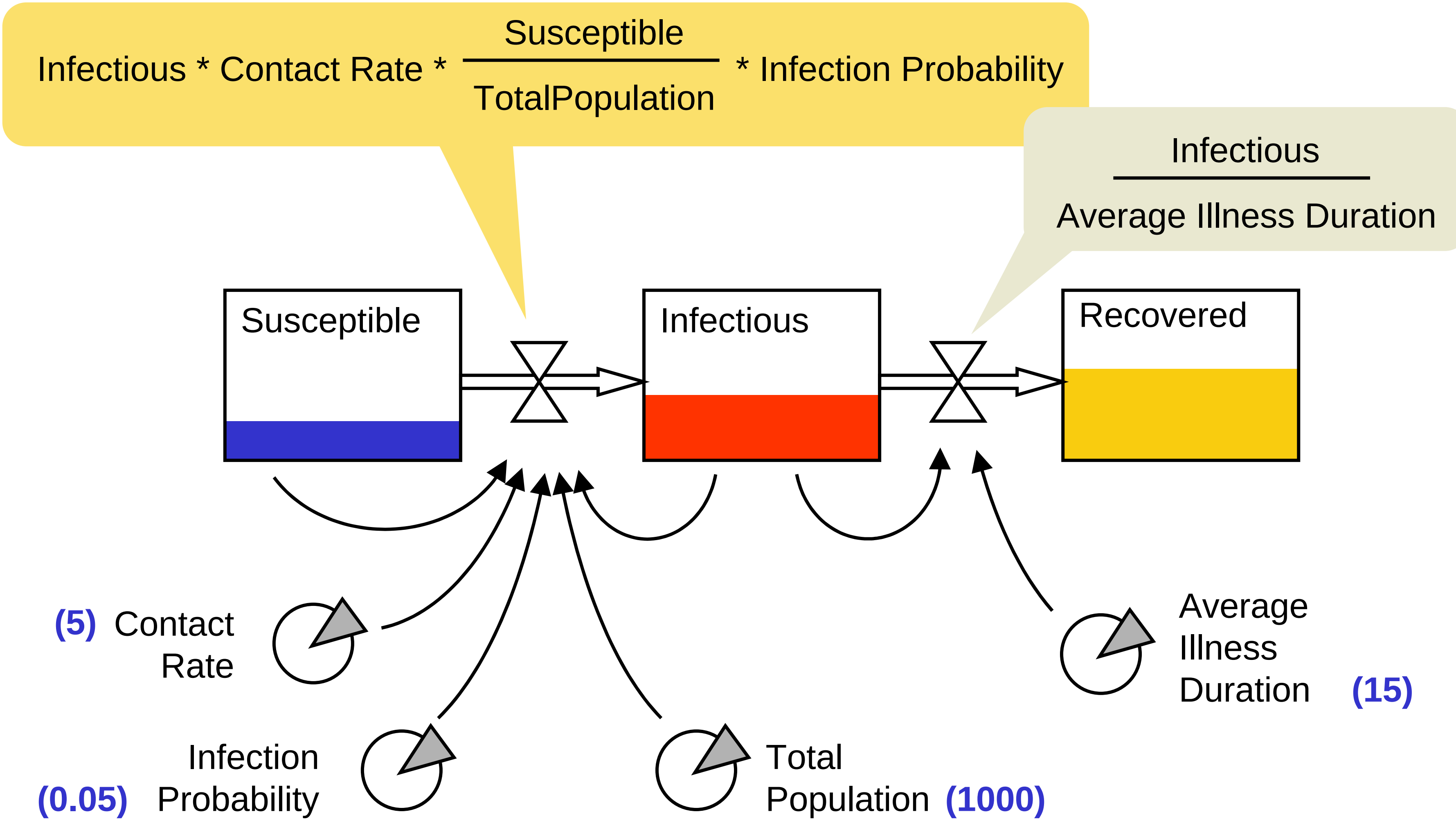
What can AB give you?

- Capture much more sophisticated behaviors
- Capture individual properties, history, contacts
  - AB is free from SD "PERFECT MIX" assumption
- AB may be much more natural way of modeling!
  - In many cases the modeler is much more confident and comfortable with describing the system behavior at individual, low abstraction level than with trying to identify the system-level dependencies and flows
- Much better visualization of model behavior
  - You can develop appealing colorful dynamic animations that speak much better than static stock & flow structures and plots

SIR epidemic model: the assumptions

- An infectious disease outbreak is modeled
- We distinguish three different states of a person: **Susceptible**, **Infectious**, **Recovered**
- Recovered are immune to the disease
- A susceptible person may get infected if contacted by an infectious person
  - (with a certain probability)
- Contacts occur between any people
  - (at a certain rate)

SIR model: SD version



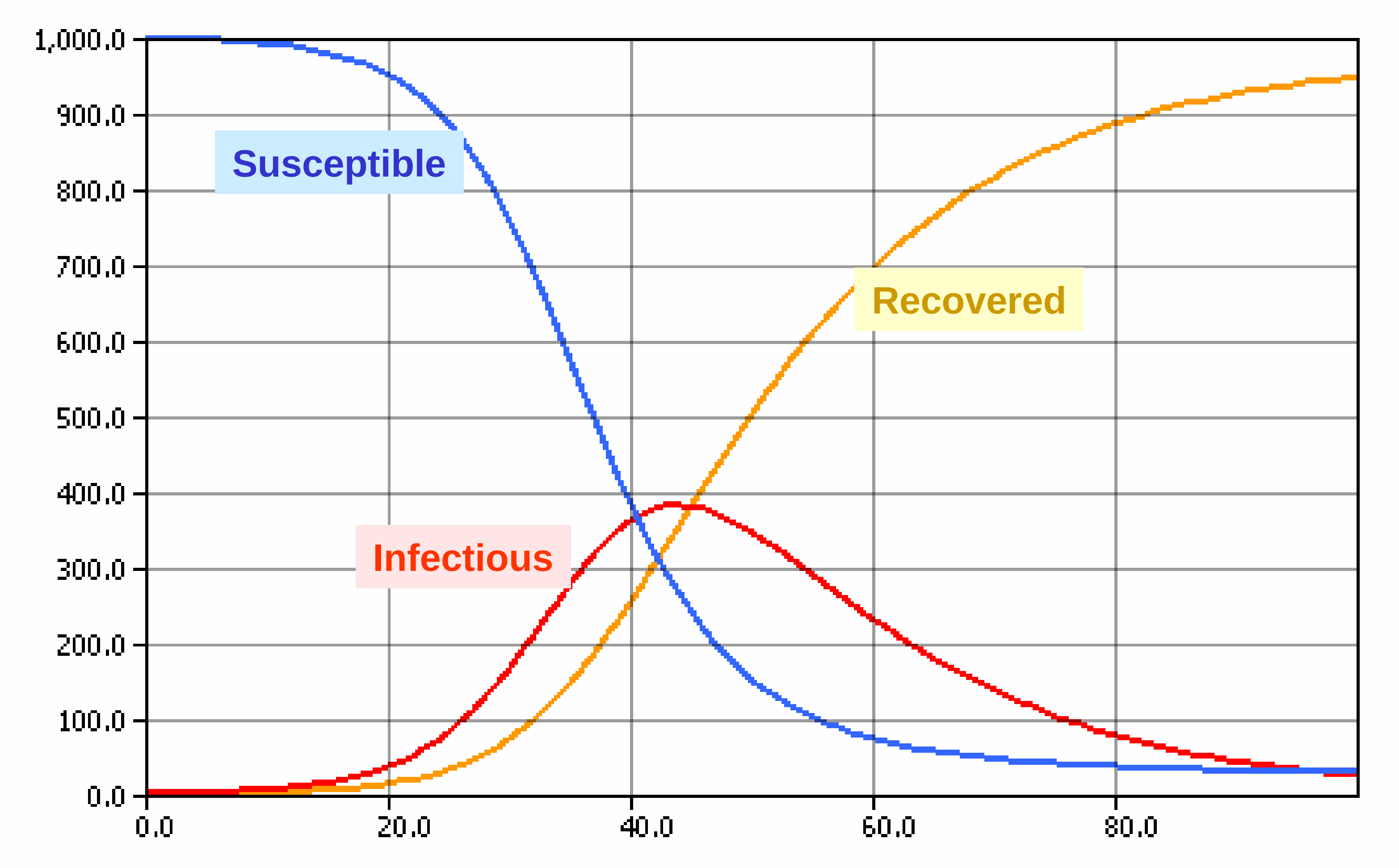
The diffusion equation

- How can we interpret this:

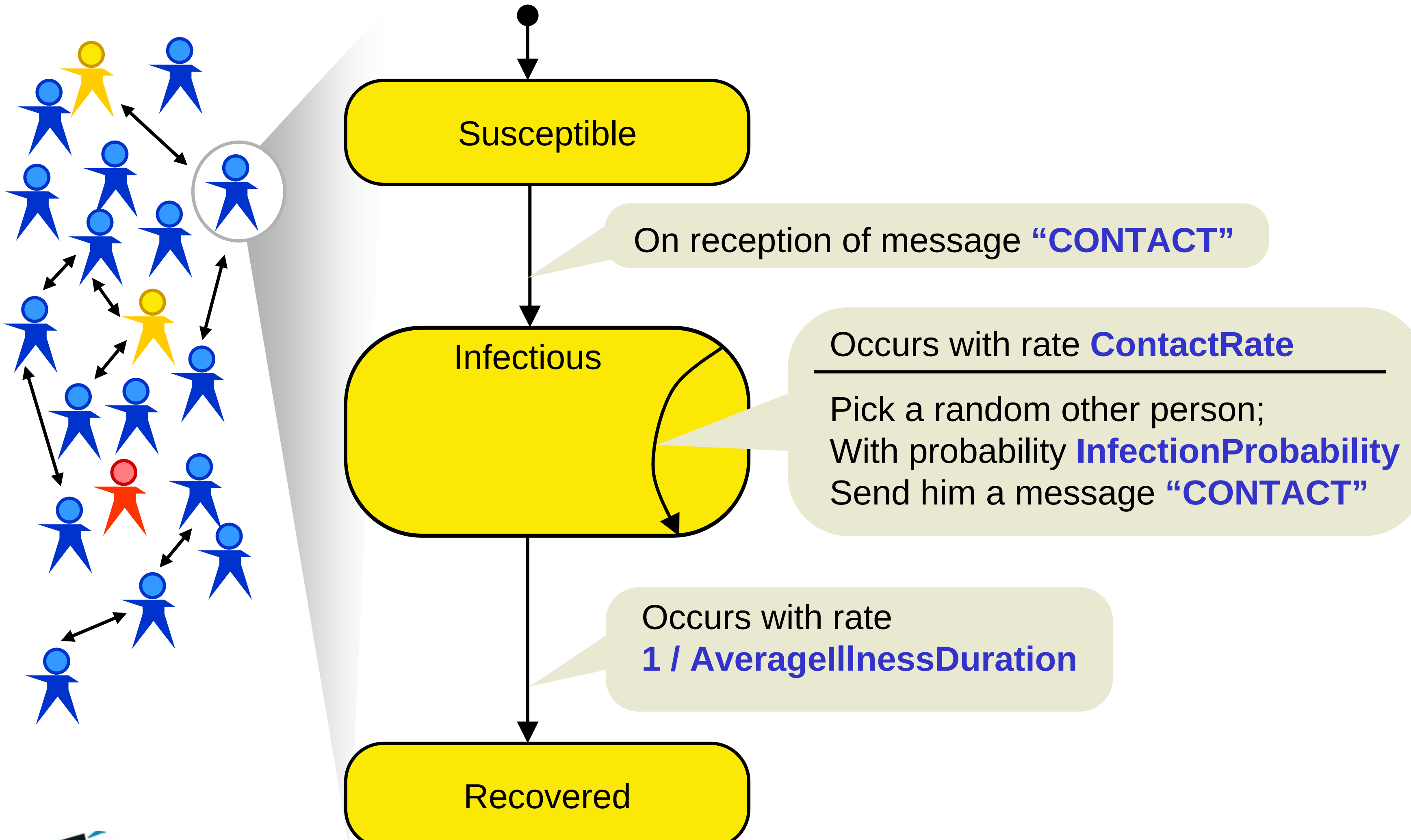
$$\text{Infectious} * \text{Contact Rate} * \frac{\text{Susceptible}}{\text{TotalPopulation}} * \text{Infection Probability} \quad ?$$

- Every **Infectious** person…
- Every day contacts **Contact Rate** other people…
- Of which **Susceptible / Total Population** are susceptible (on average)…
- And if contacted, the susceptible person gets infected with **Infection Probability**

SIR model: SD version output

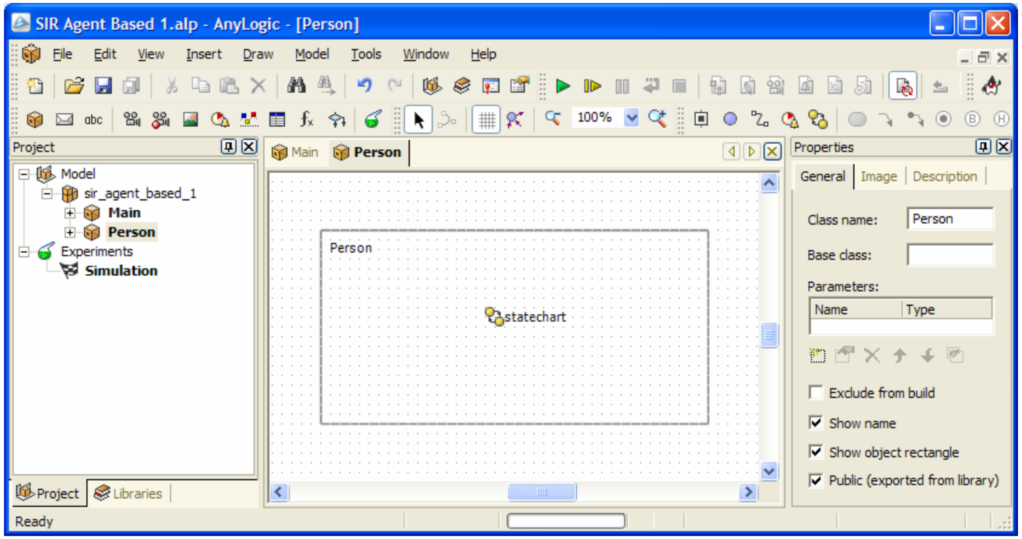


Simple AB Model. The concept



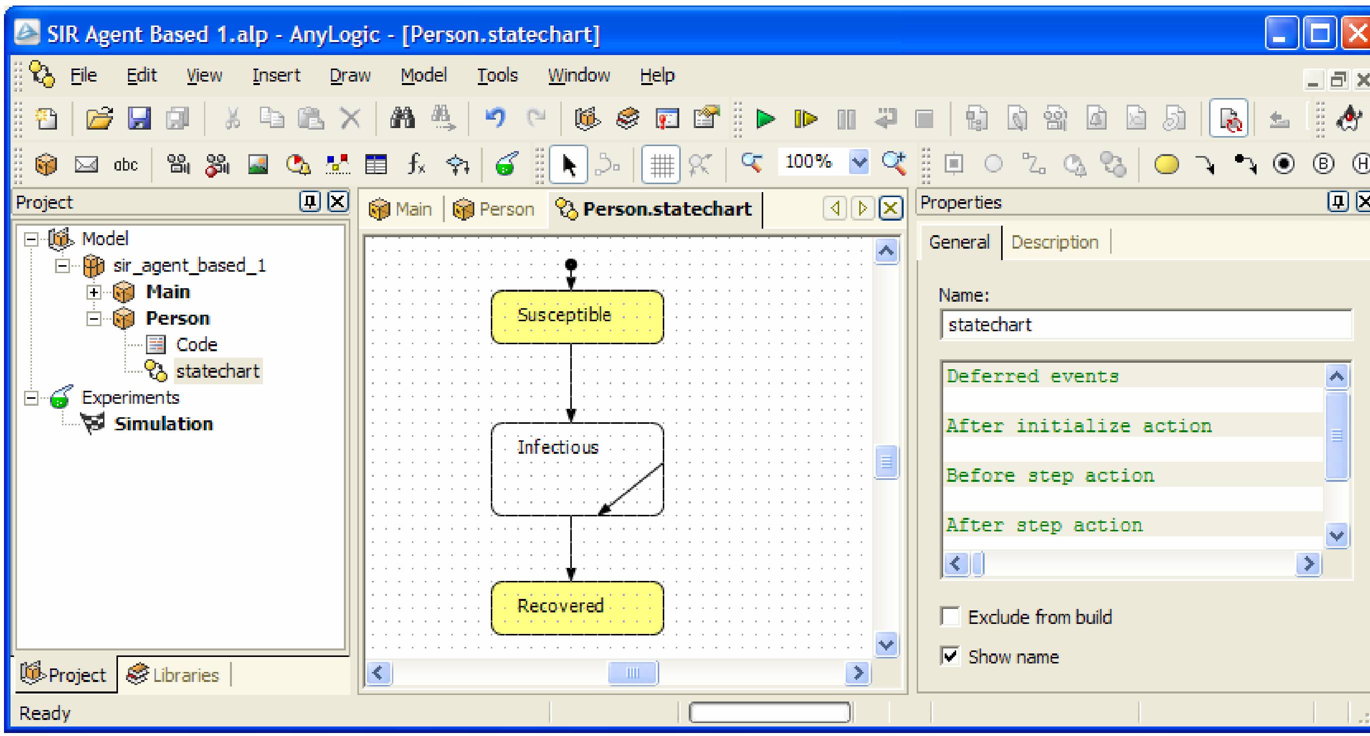
AB Model. Phase 1. Step 1

- Open AnyLogic
- Create a new project “SIR Agent Based”
- Create a new Active Object class Person
- In Person create a statechart



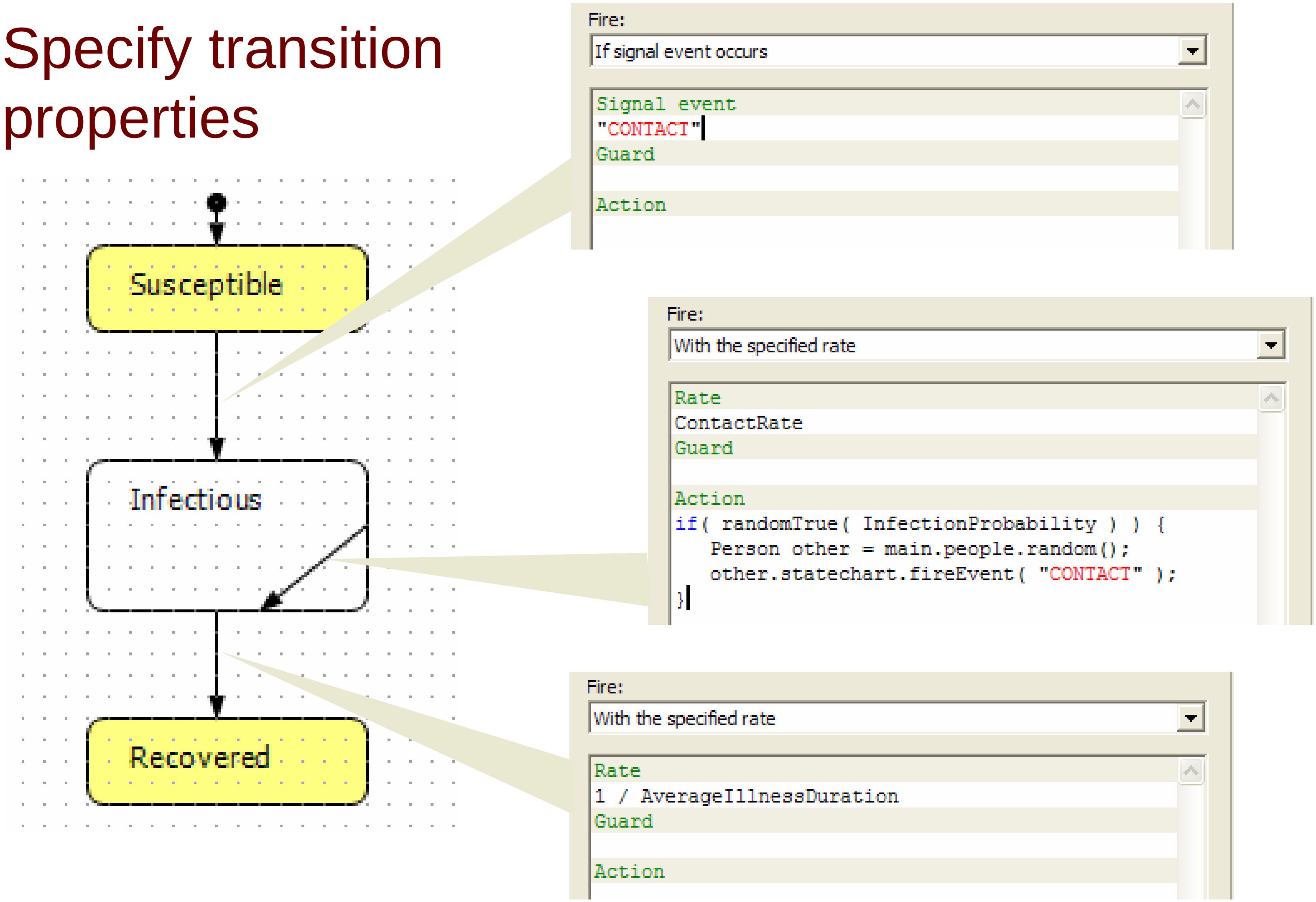
AB Model. Phase 1. Step 2

- Open the statechart editor
- Draw a statechart as we designed



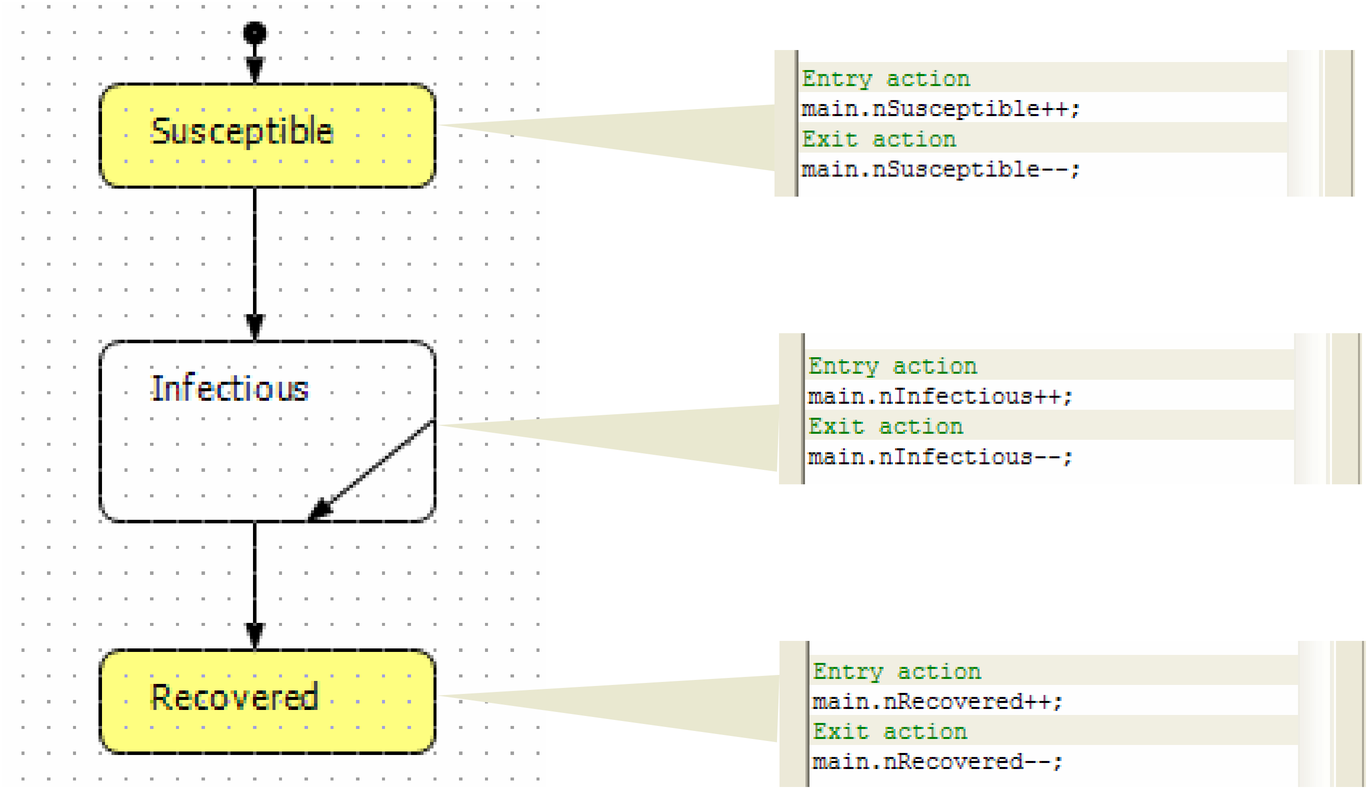
AB Model. Phase 1. Step 3

- Specify transition properties



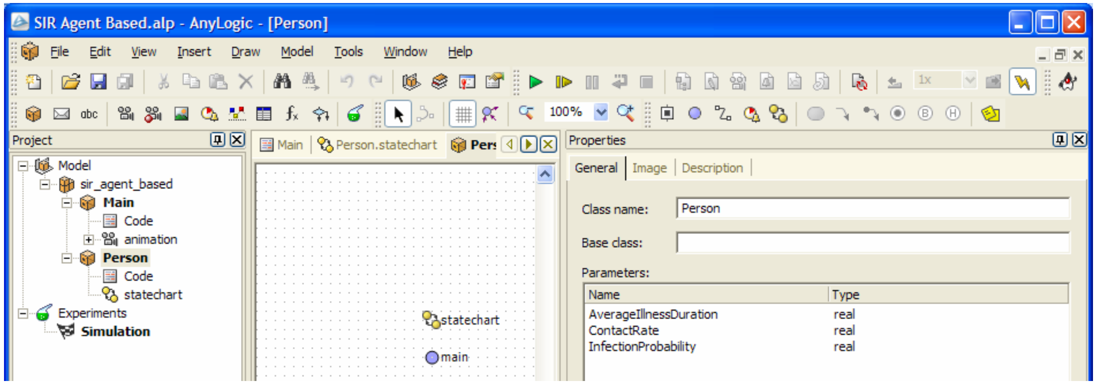
AB Model. Phase 1. Step 4

- Specify state properties
  - This is just for stats



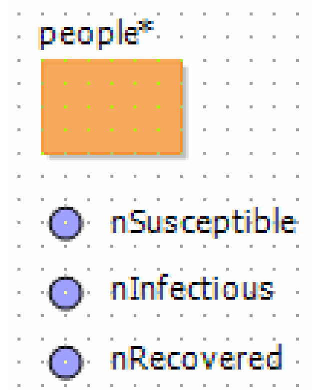
AB Model. Phase 1. Step 5

- Return to the structure diagram of Person
- Define a variable
  - main of type Main, initial value (Main)getOwner()
- Define three parameters
  - AverageIllnessDuration of type double, default value 15
  - ContactRate of type double, default value 5
  - InfectionProbability of type double, default value 0.05



AB Model. Phase 1. Step 6

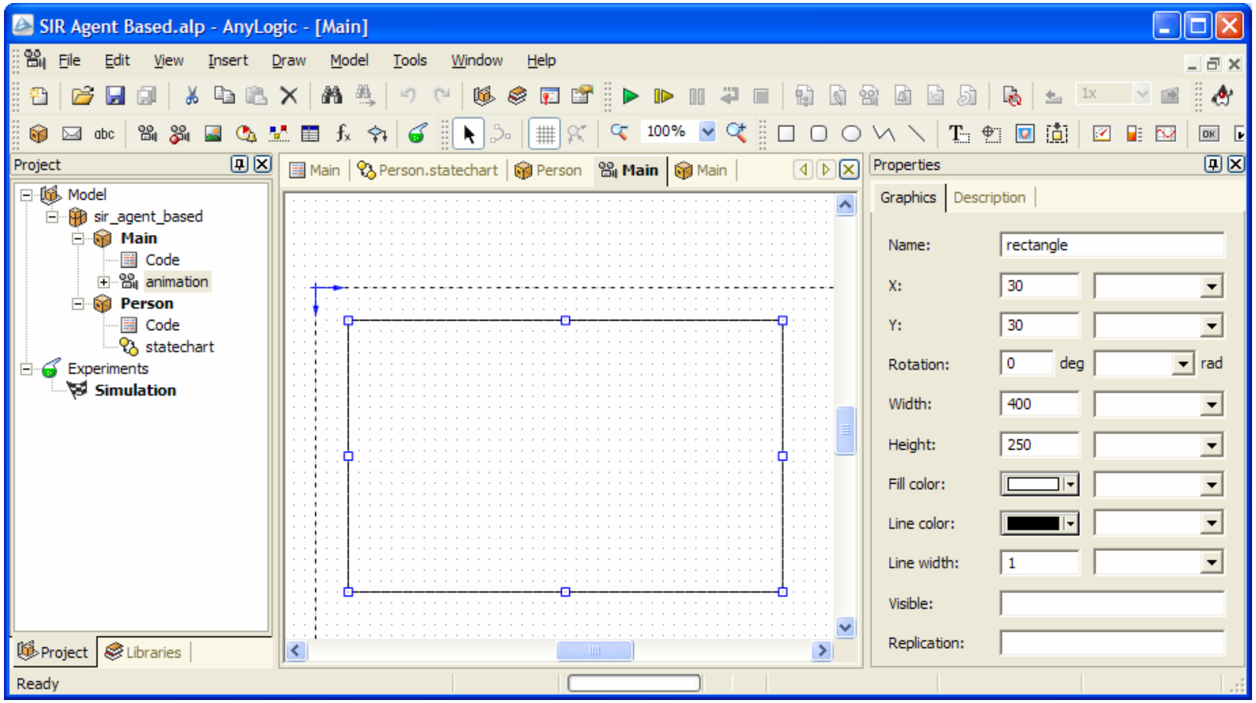
- Open the structure diagram of Main
- Drop there class Person from the tree
- Rename the dropped person to people
- Enter 1000 as the Number of objects in its Replication tab
- Add variables: nSusceptible, nInfectious and nRecovered of type integer
- Open the Code of Main and enter in Startup Code:




```
Startup code
people.item(0).statechart.fireEvent( "CONTACT" );
```

AB Model. Phase 1. Step 7

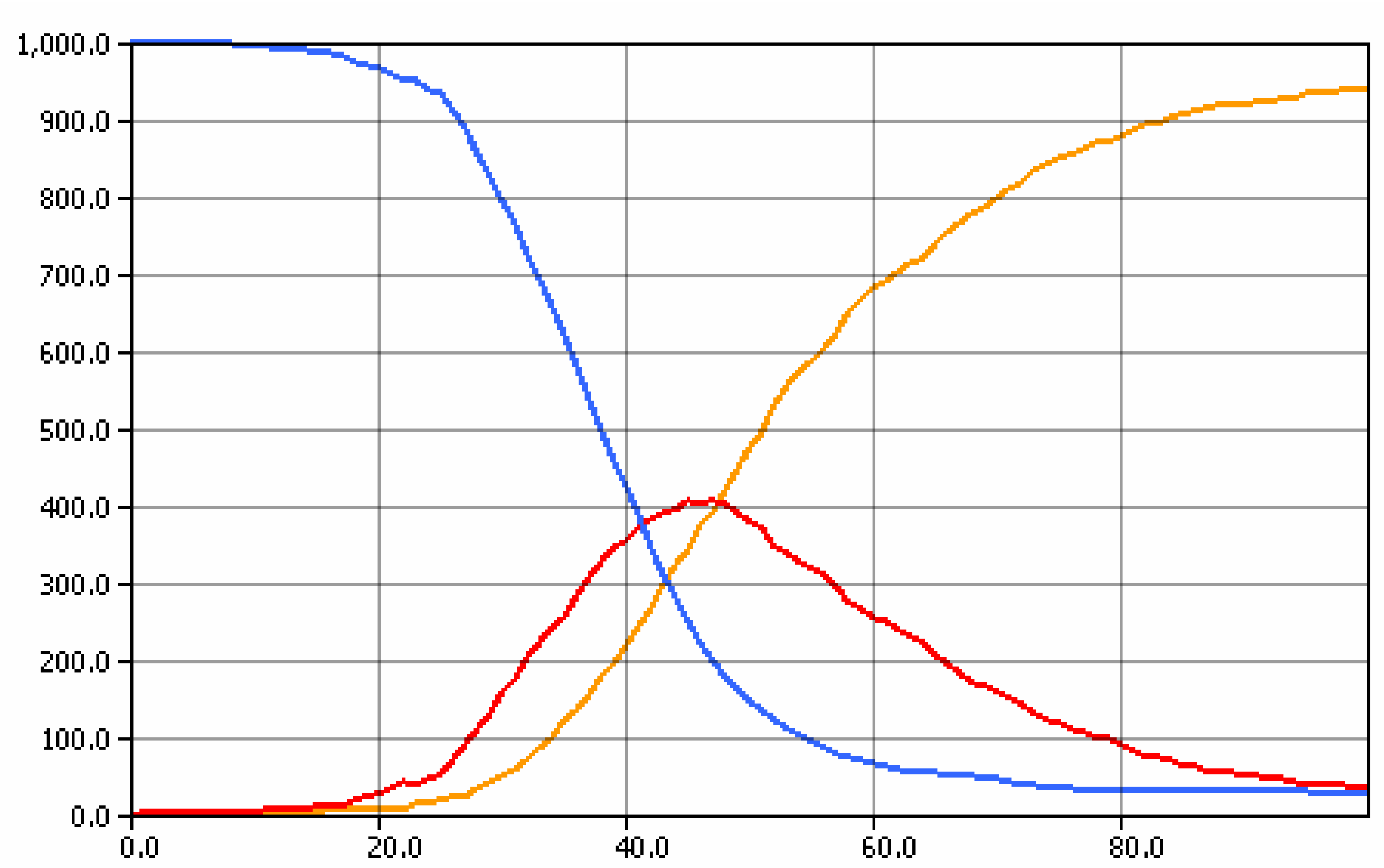
- Add Animation to Main
- Draw a rectangle of size approx 400 x 250 at the position approx (30,30)



AB Model. Phase 1. Step 8

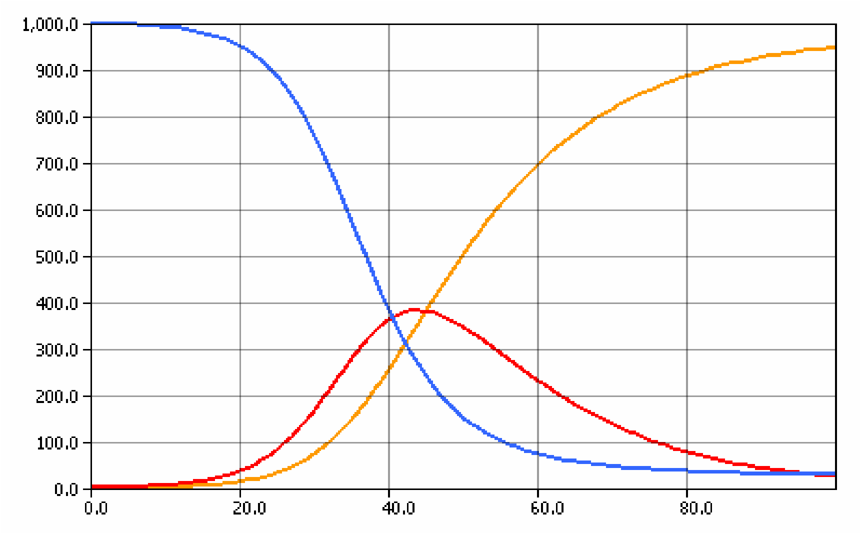
- Switch from animation to structure of Main
- Open the Business Graphics Library tab
- Drop ChartTime object to the structure of Main
- Set these parameters of chartTime:
  - Placeholder: animation.rectangle (use drop-down list)
  - UpdateMode: AUTO - UPDATE Data EVERY TimeStep
  - ScaleType: AUTO SAME FOR ALL DATA
  - Data0: nSusceptible Legend0: “# Susceptible”
  - Data1: nInfectious Legend1: “# Infectious”
  - Data2: nRecovered Legend2: “# Recovered”
- Check Experiment | Simulation | StopAtTime 100

Run the model. You should see this:

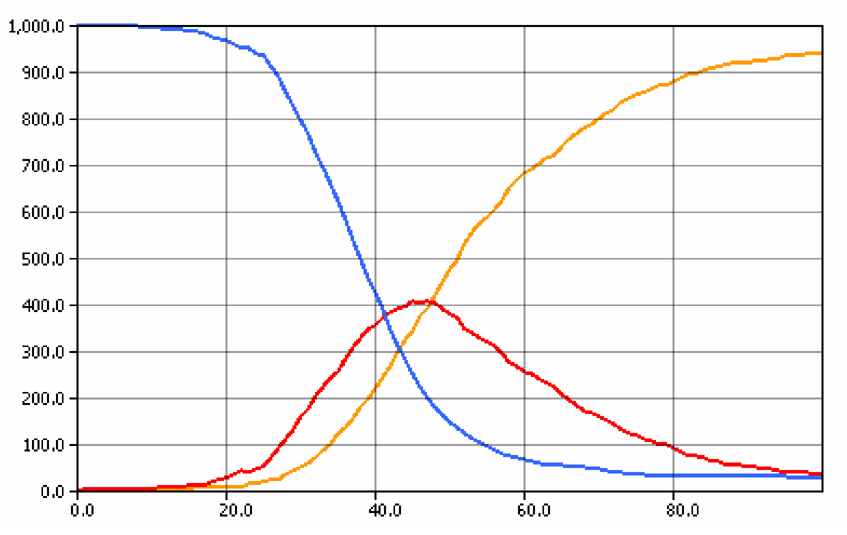


Compare the results

- System Dynamics



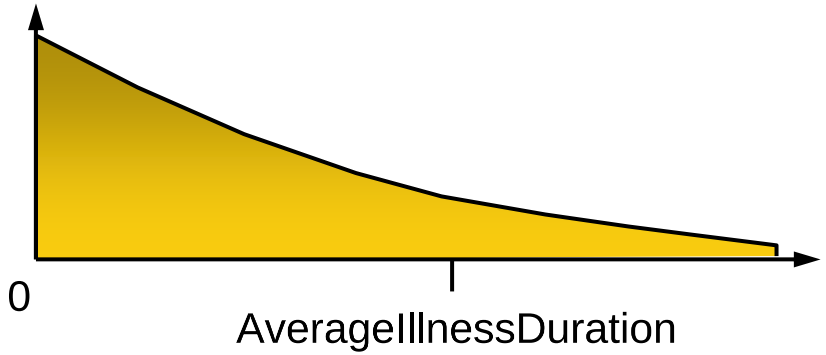
- Agent Based



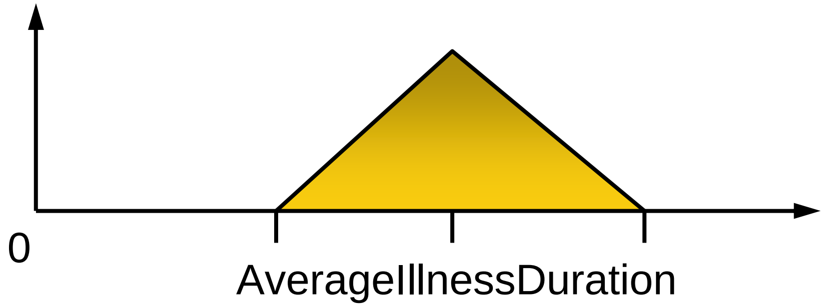
- Very similar
- Discrete stochastic nature of AB shows well
- So why do we need to build AB model???

Let’s change the illness duration model

- Instead of exponentially distributed duration
  - Which was “inherited” from SD assumptions!

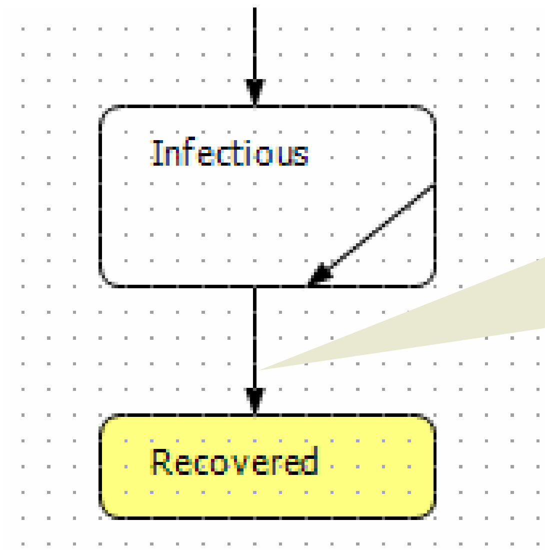


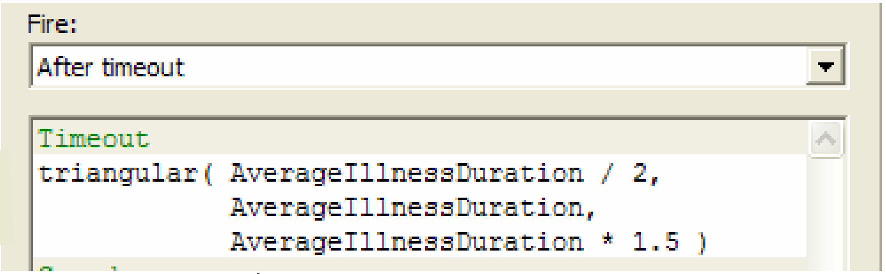
- We will use triangular distribution:



AB Model. Phase 2. Step 1

- Open the statechart of Person
- Modify the trigger of transition between Infectious and Recovered:

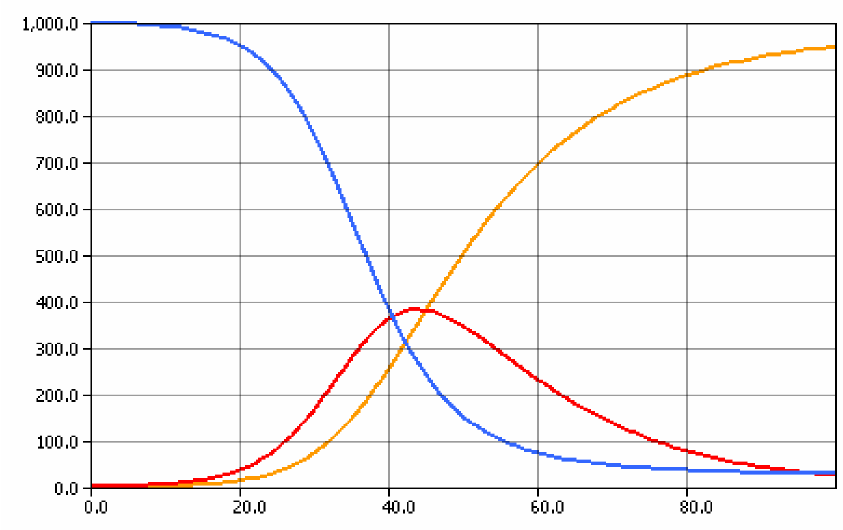


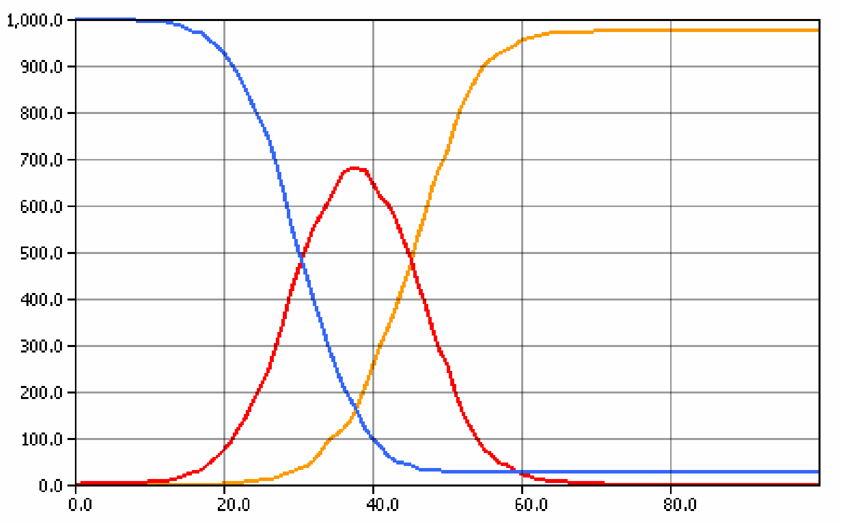


Now the time spent by a person in Infectious state is distributed differently

Run the model again

- Output is different:
- System Dynamics
- Agent Based



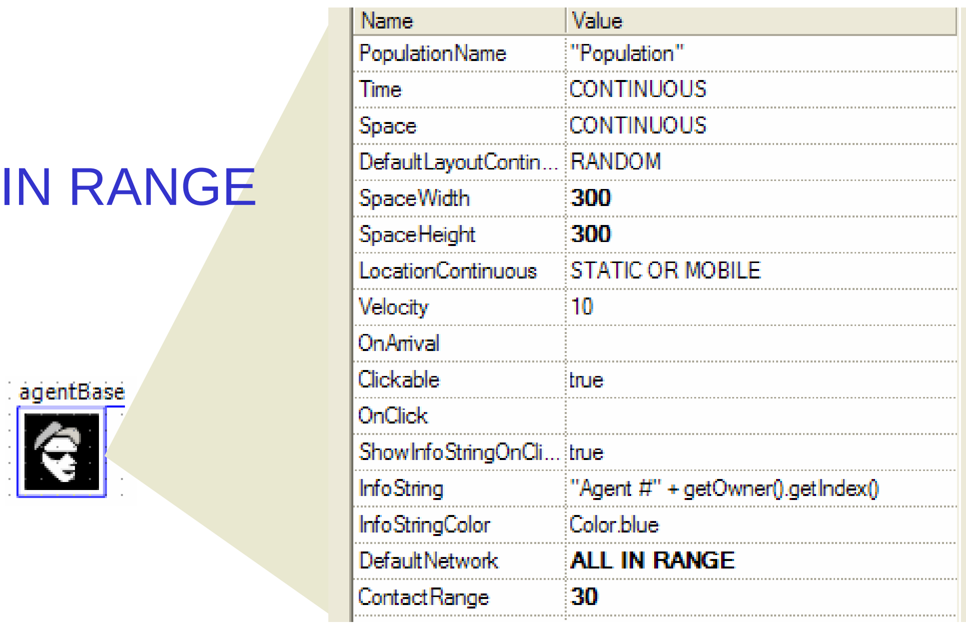


Discussion on the run results

- Can you model this kind of illness duration in SD?
  – Yes, one can try to use high order delays or conveyor to reproduce it…
- But:
  – This may be considered as a certain type of workaround used to reproduce the known results, while AB gives you the direct means of modeling the desired behavior

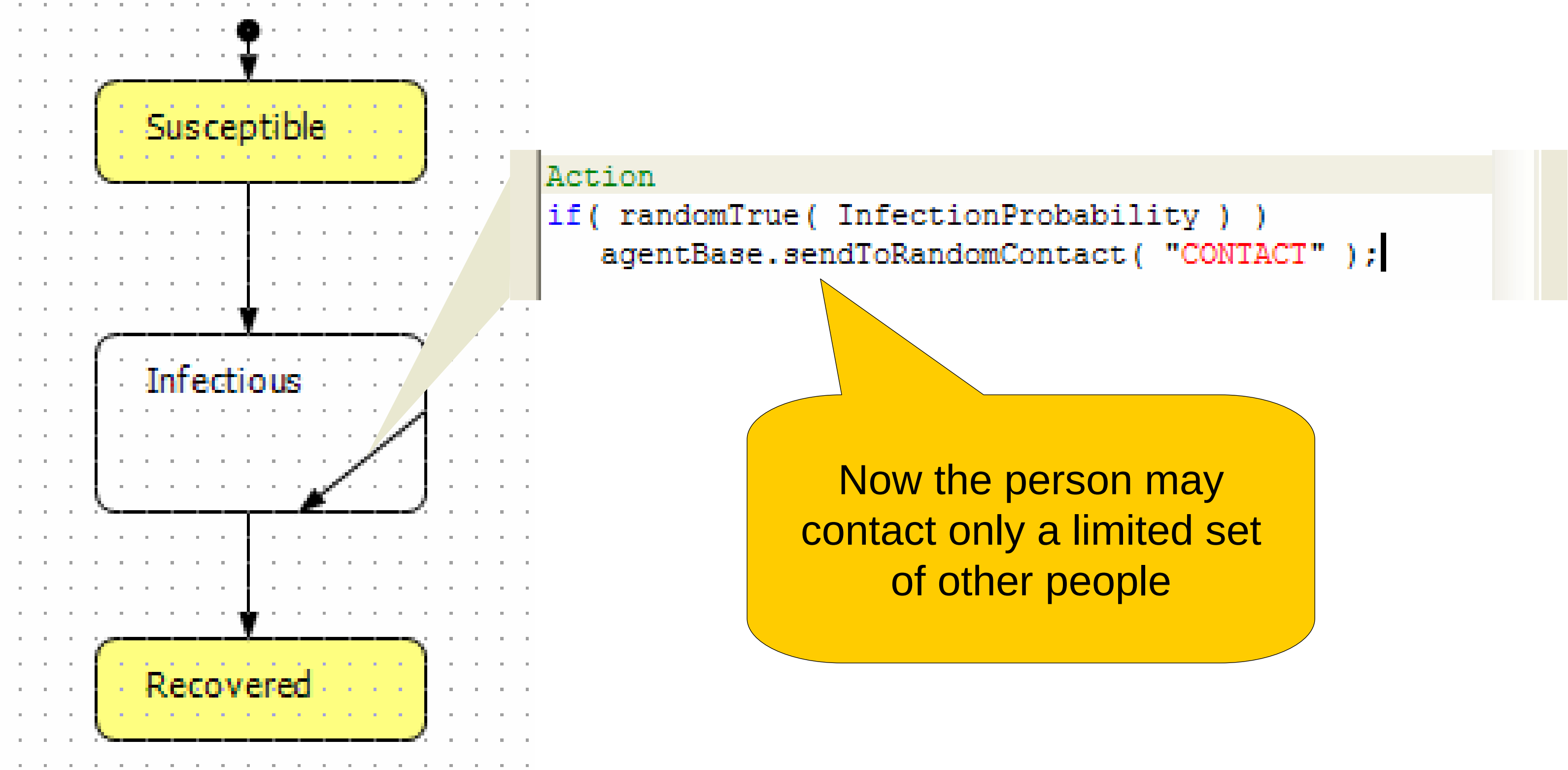
AB Model. Phase 3. Step 1

- Switch to structure of Person
- Open the Agent Based Library tab
- Drop AgentBase object to the structure of Person
- Set these parameters of agentBase:
  - SpaceWidth: 300
  - SpaceHeight: 300
  - DefaultNetwork: ALL IN RANGE
  - ContactRange: 30



AB Model. Phase 3. Step 2

- Open the statechart of Person
- Modify the action of the internal transition inside



AB Model. Phase 3. Step 2

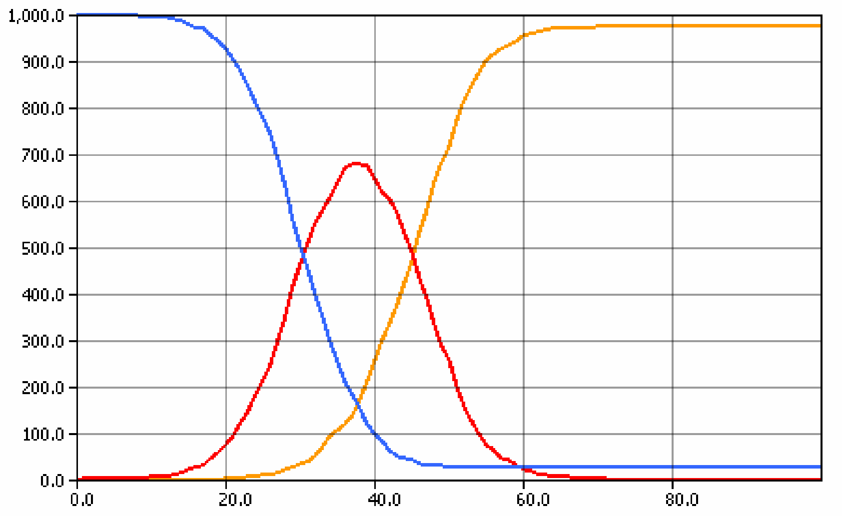
- Switch to structure of Person
- Click on agentBase
- Modify its onReceive property
  - Enter statechart.fireEvent(message);

DefaultNetwork	ALL IN RANGE
ContactRange	30
ShowContacts	ON_CLICK
ContactLineColor	Color.black
ContactLineSemitra...	false
OnReceive	statechart.fireEvent(message);

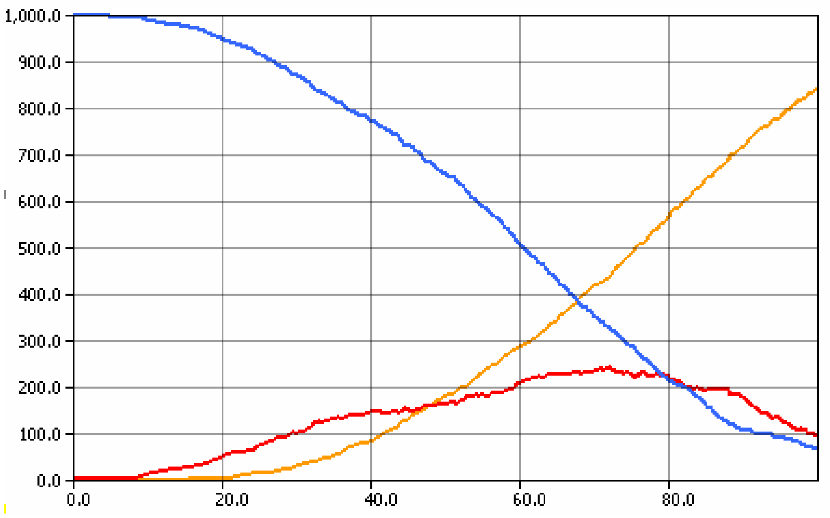
The statechart of this person will be notified when the message is received from the network

Run the model

- Compare outputs:
- AB random contacts

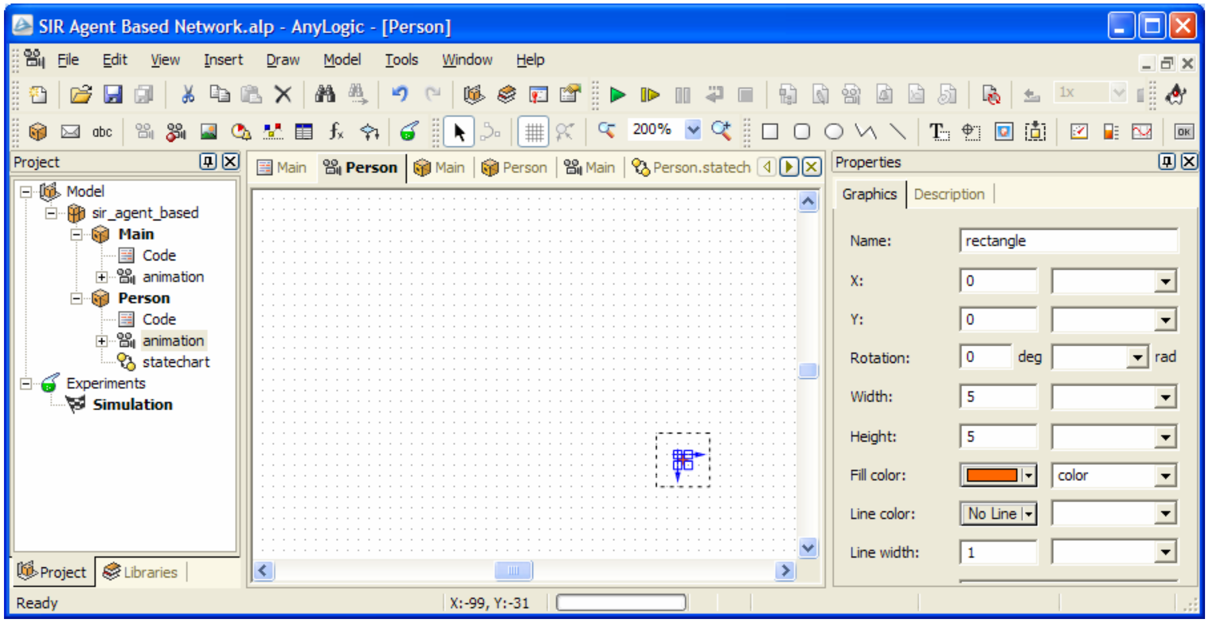


- AB contacts in range 30



AB Model. Phase 4. Step 1

- Add Animation to Person
- Draw a rectangle of size 5 x 5 pixels at the position (0,0), set its Line color to No line
- Type color in its Fill color (dynamic) field

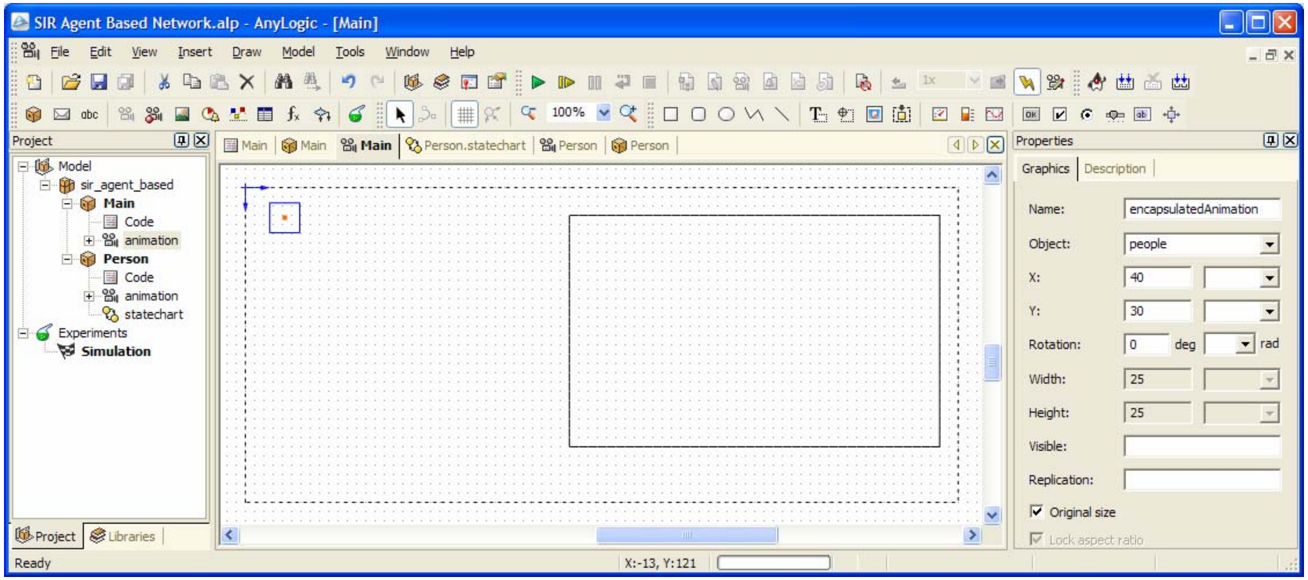


AB Model. Phase 4. Step 2

- Open the structure diagram of Person
- Add a variable color of type Color
- Open the statechart of Person
- Add color changing statements to state actions:
  - Susceptible entry action: add …color = Color.blue;
  - Infectious entry action: add …color = Color.red;
  - Recovered entry action: add …color = Color.yellow;

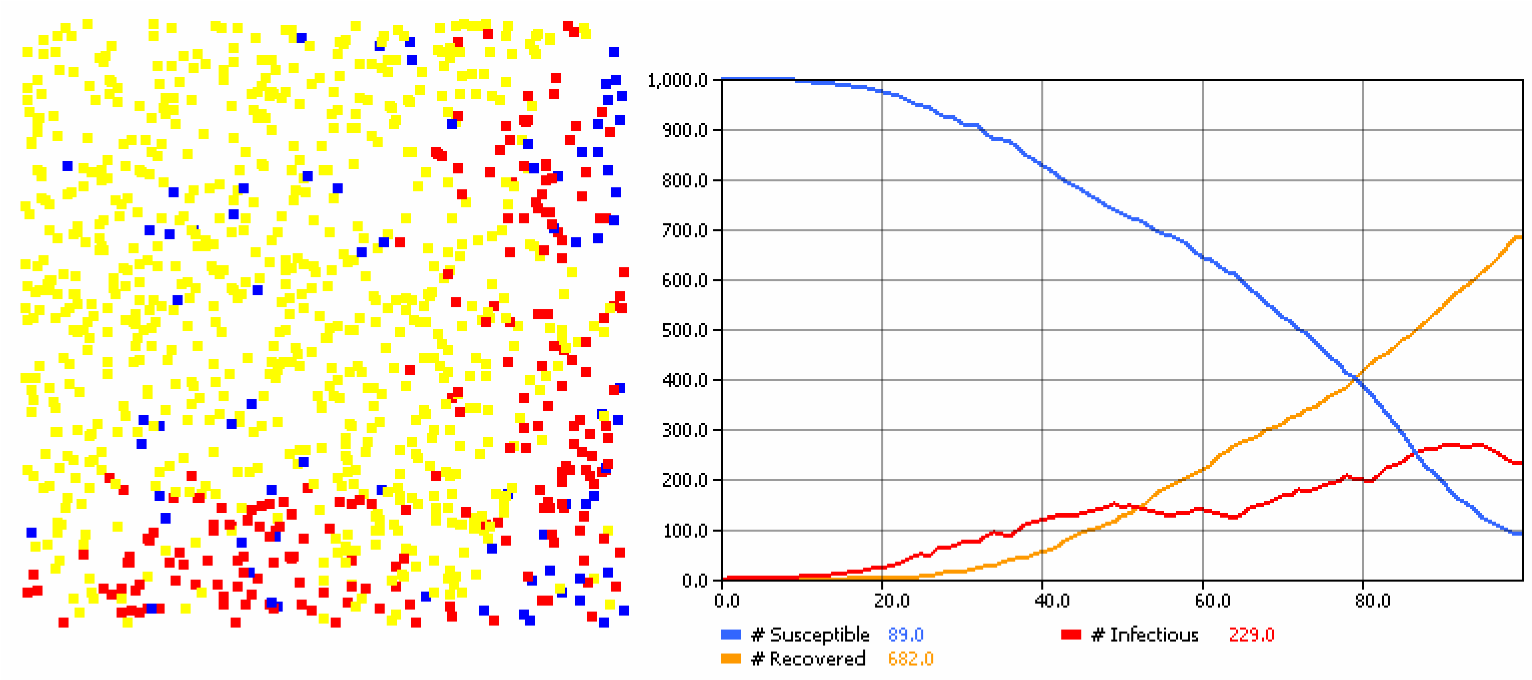
AB Model. Phase 4. Step 3

- Open Animation of Main
- Move the chart placeholder rectangle to (350,30)
- Add Encapsulated animation
- Set its Object property to people



Run the model and see the spatial animation

- The disease now spreads more slowly
  - Because the contacts are only local

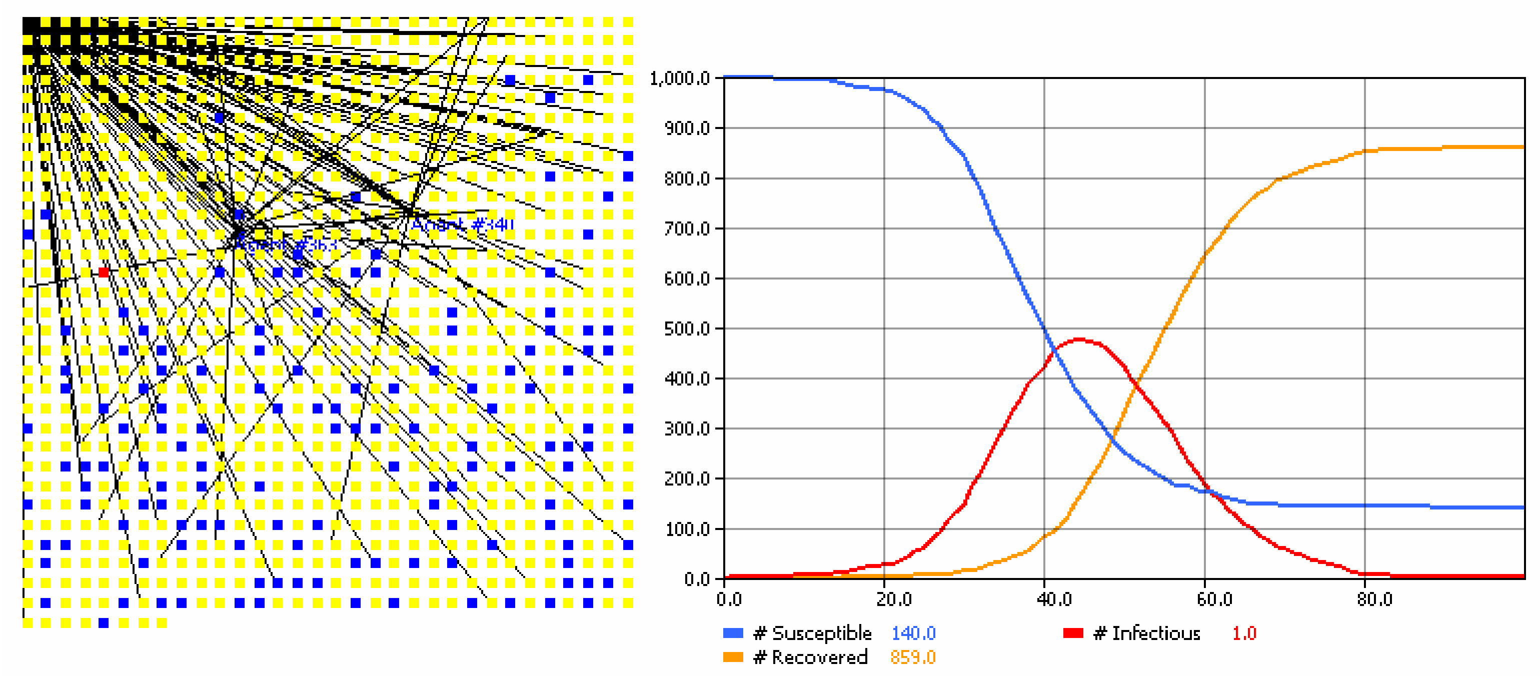


AB Model. Phase 5. Step 1

- Open the structure of Person
- Click on the agentBase
- Specify different layout and network type:
  - Set DefaultLayoutContinuous to ARRANGED
  - Set DefaultNetwork to SCALE FREE

Run the model again

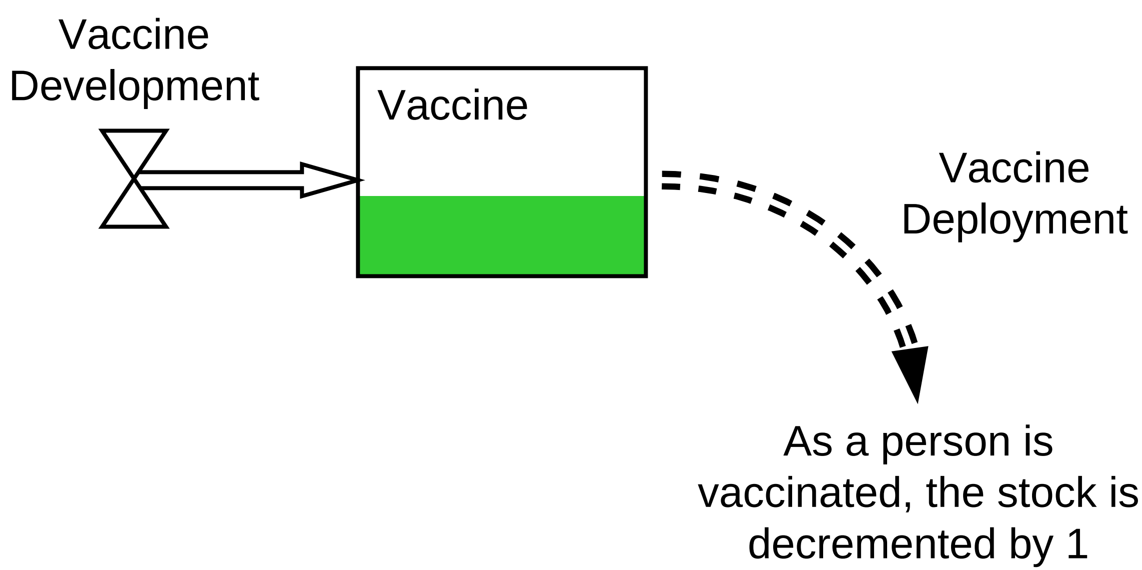
- Click on the top left agent and on couple of others
  - In this network type some agents are hubs with many links



- The disease spread is fast again

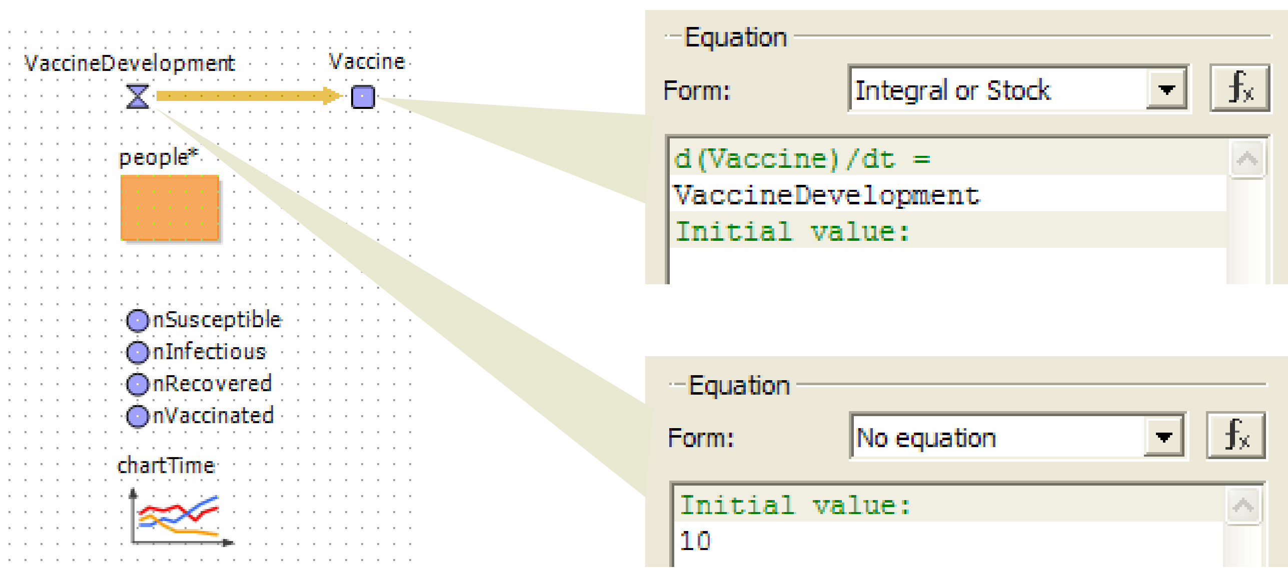
Let us add vaccination model

- We will add vaccination as a System Dynamics model on the global level



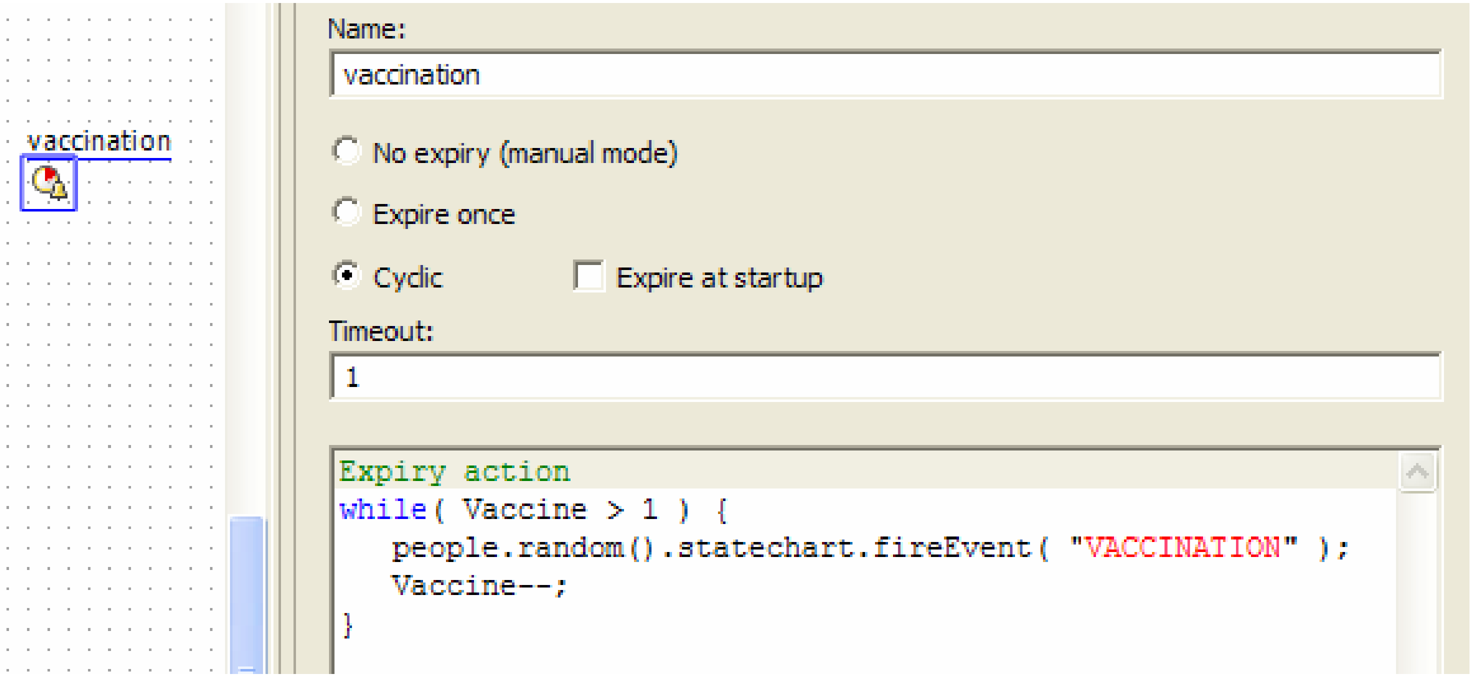
AB+SD Model. Phase 6. Step 1

- Open the diagram of Main
- Add two variables
  - VaccineDevelopment and Vaccine; make Vaccine a stock
- Specify properties of the variables:



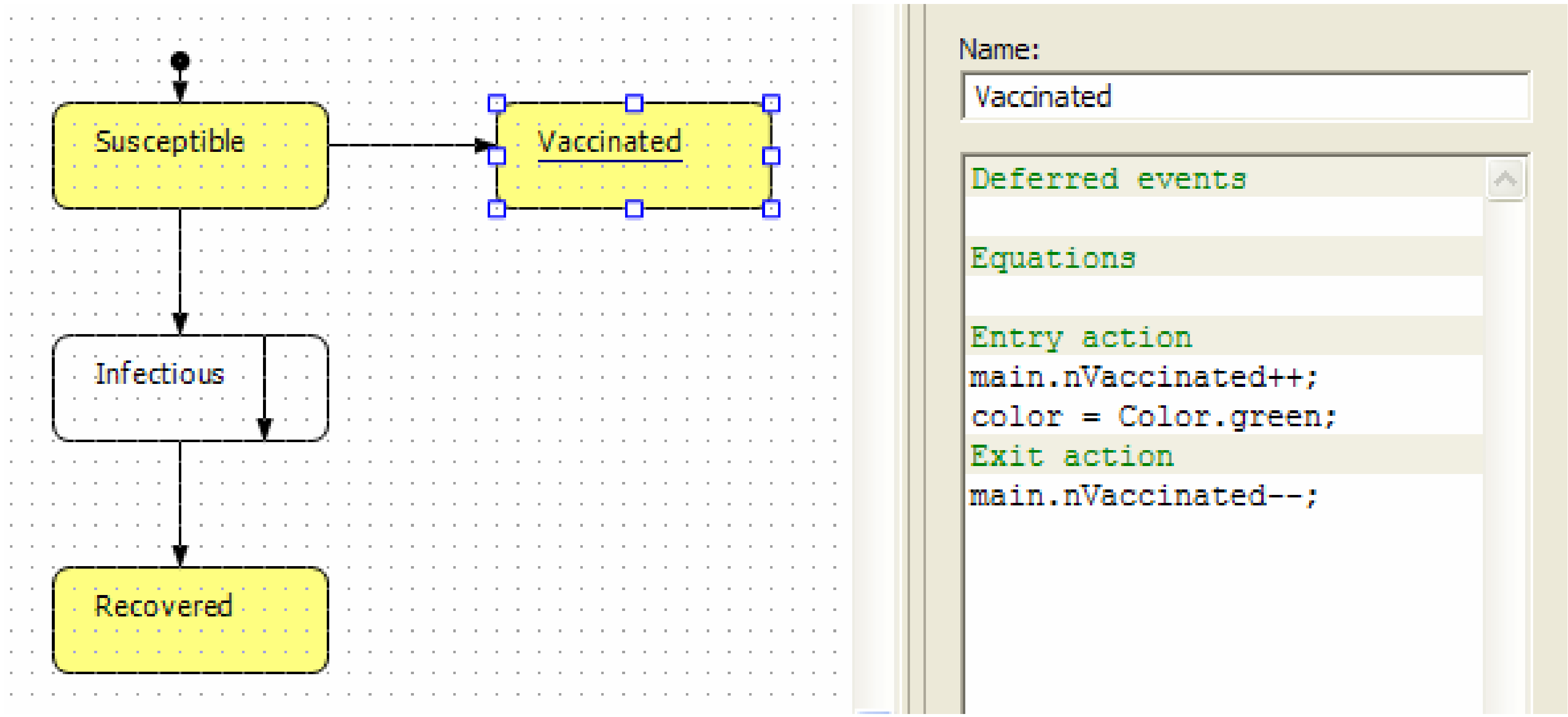
AB+SD Model. Phase 6. Step 2

- Add vaccination timer to the same diagram
- Set the timer to check the vaccine availability and apply it to the agents (random, so far)



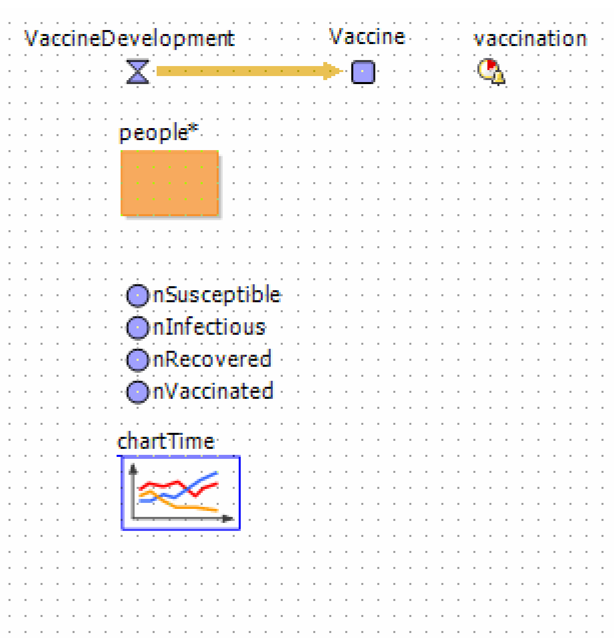
AB+SD Model. Phase 6. Step 3

- Open the statechart of Person
- Add a new state Vaccinated and set its actions



AB+SD Model. Phase 6. Step 4

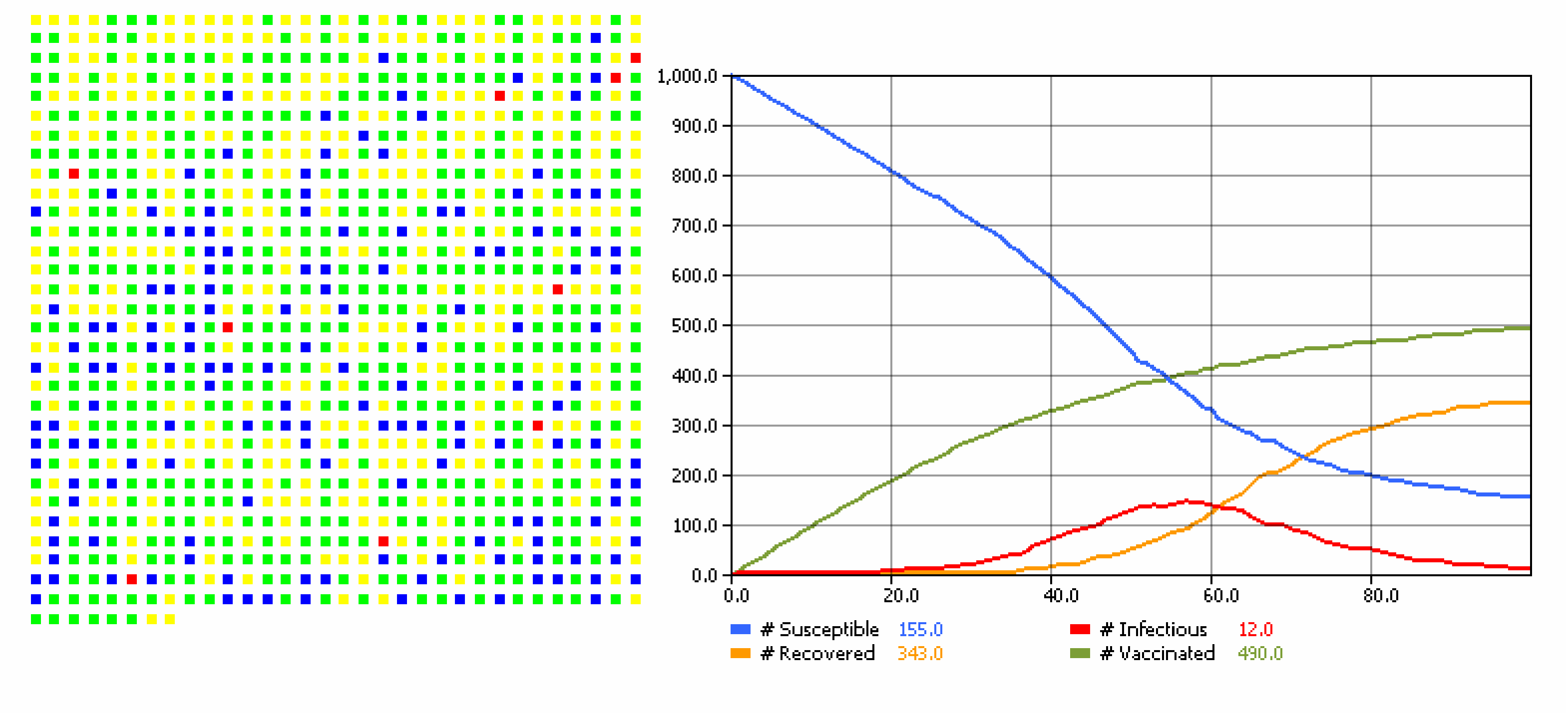
- Switch to the diagram of Main
- Add integer variable nVaccinated
- Add it to the chartTime with green color



ShowLegend	true
LegendTextColor	BGBase.BLACK
LegendPosition	BELOW
Data0	**nSusceptible**
Color0	BGBase.BLUE
LegendText0	**"# Susceptible"**
Data1	**nInfectious**
Color1	BGBase.RED
LegendText1	**"# Infectious"**
Data2	**nRecovered**
Color2	BGBase.YELLOW
LegendText2	**"# Recovered"**
Data3	**nVaccinated**
Color3	**BGBase.GREEN**
LegendText3	**"# Vaccinated"**
Data4	- not specified -

Run the model

- See how vaccination affects the disease dynamics



A note on time steps, etc.

- The AB and SD models coexist in the same time
- There is no notion of time step at AB side – time is “asynchronous”, or “continuous”
  - Therefore agents can generate events at any time moment as needed
  - However, you can make them synchronous by introducing “clock ticks” if you wish
- The numeric solver of course has internal time steps
  - But it works consistently with the discrete events of agents
- Therefore the modeler does not have to care about time steps in AnyLogic

On your own:

- Make the vaccine development rate dependent on the number of infectious people\
  - Let's say the development starts when there are 30 infected
- Change the vaccine deployment so that it is only applied to Susceptible people

Thank you!