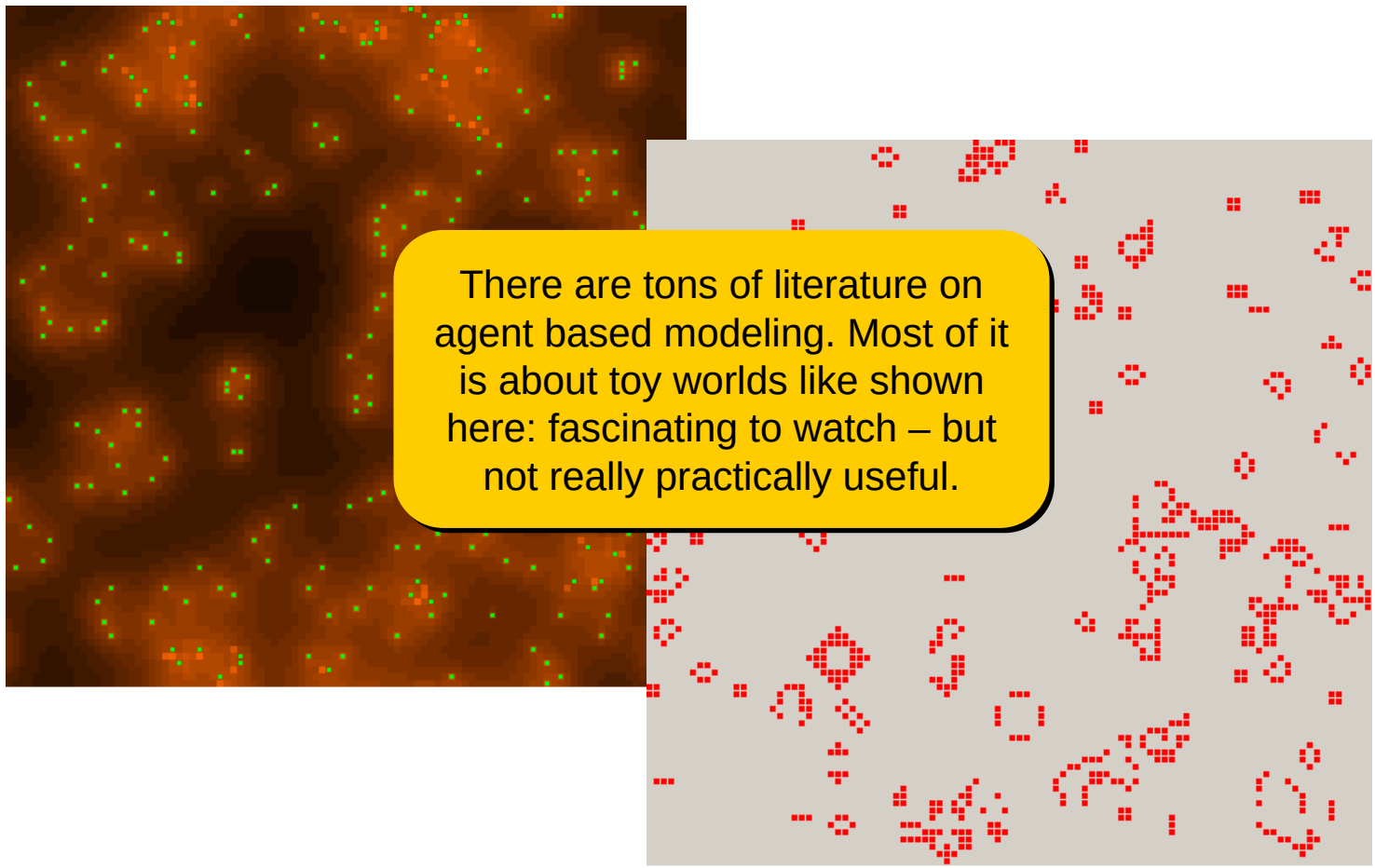
System Dynamics and Applied Agent Based Modeling

by Andrei Borshchev

Workshop
“Agent Based Modeling:
Why Bother?”

International System
Dynamics Conference
Boston, July 2005

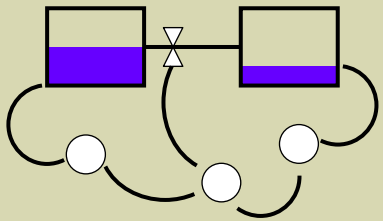
Warning!



Modeling from different perspectives

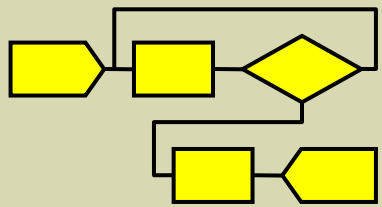
System Dynamics Perspective

Key aggregate variables,
Global feedbacks

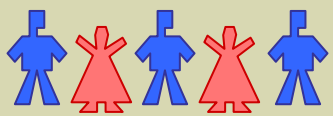


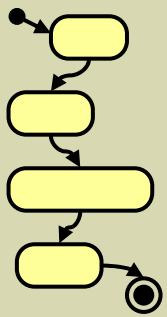
Discrete Event Perspective

Processes: sequence of
operations, resources



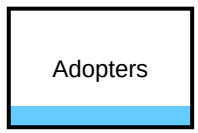
Individual parameters
and state variables,
Personal decisions





Agent Based Perspective

From SD to AB: model of diffusion



SD | **AB**

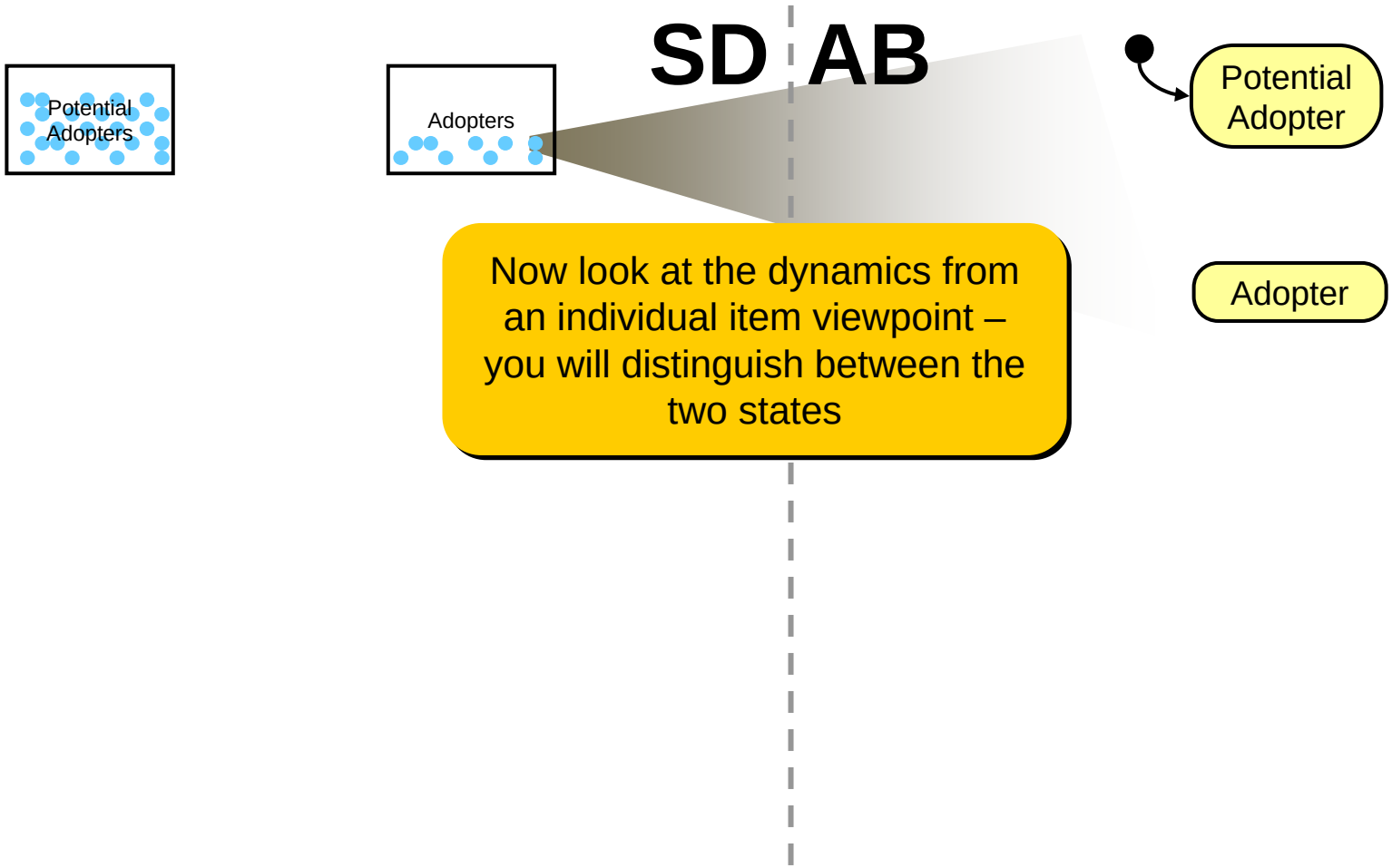
Let us consider how an AB model of product (or innovation) diffusion can be built on the basis of an SD model

First step: disaggregate

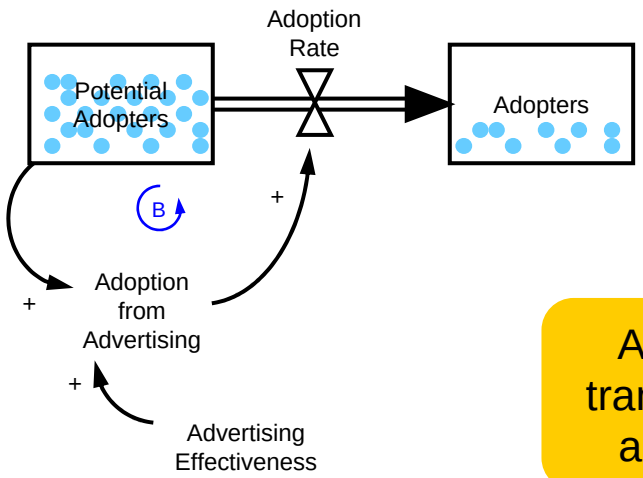
Imagine stocks are not tanks with liquid but boxes with discrete items

SD | **AB**

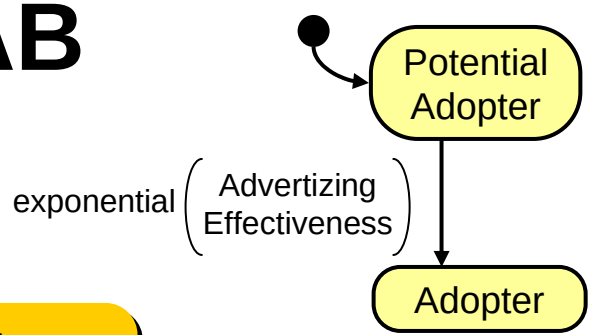
Think in terms of States…



…and transitions

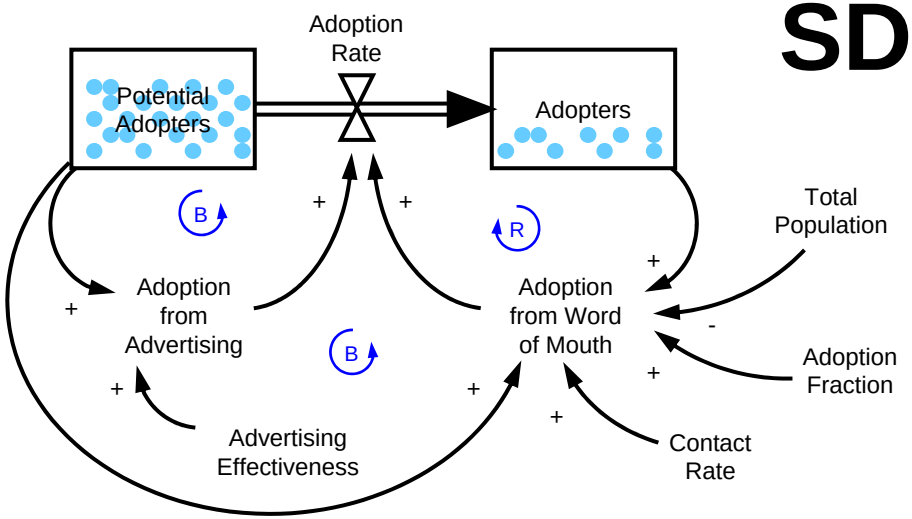


SD | **AB**

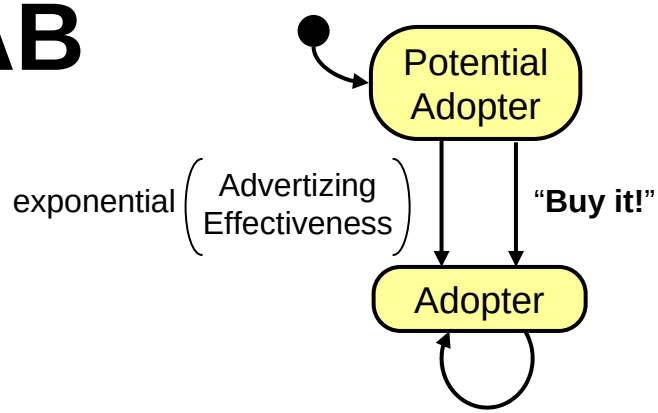


Adoption from Ad is a transition happening with a (stochastic) timeout

Use direct interaction between agents



SD **AB**

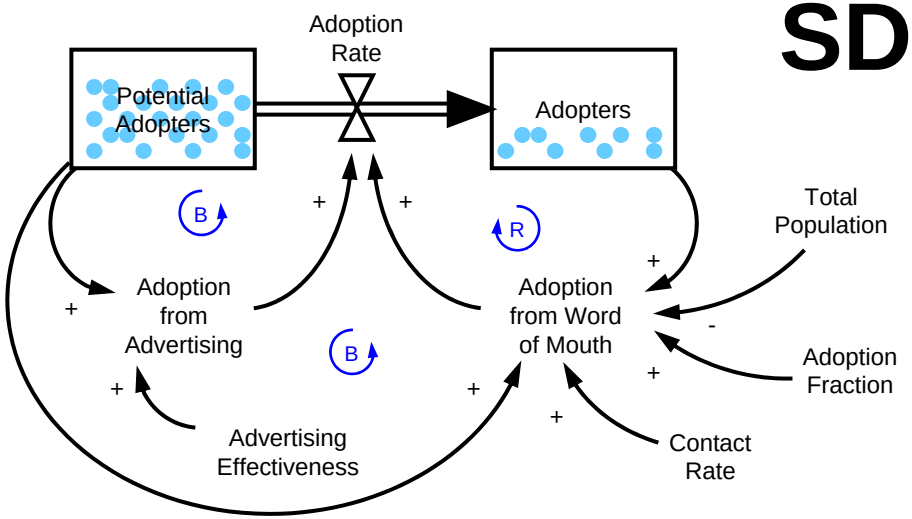


exponential(Contact Rate * Adoption Fraction)

<random agent>."**Buy it!**"

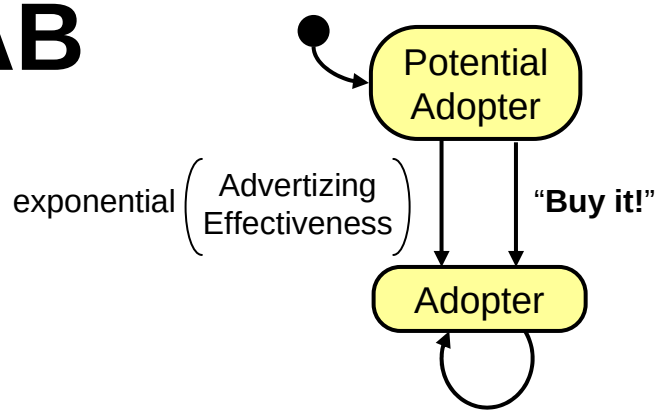
Adoption from WOM is
a) agents telling other agents "Buy it!"
b) Other agents reacting to this by taking a transition

Simulation results: few agents



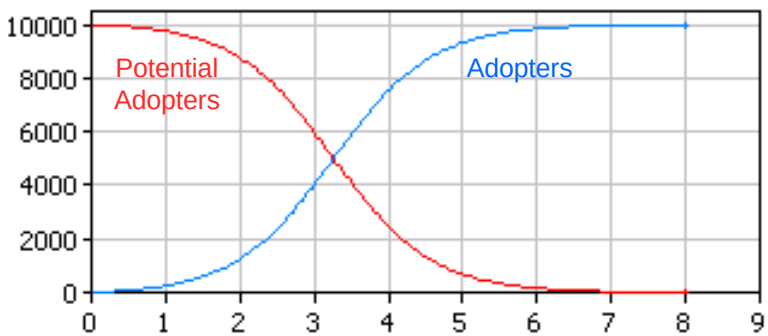
SD

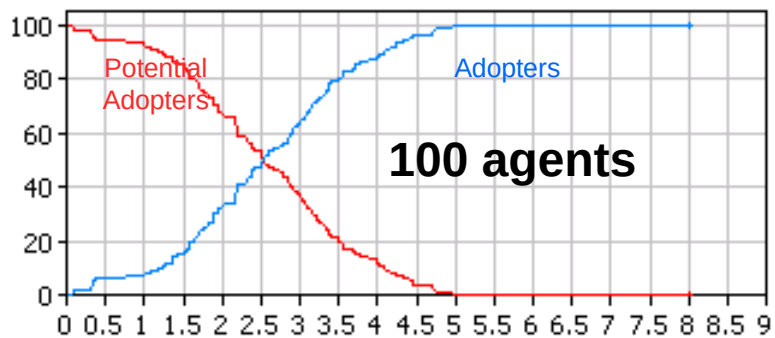
AB



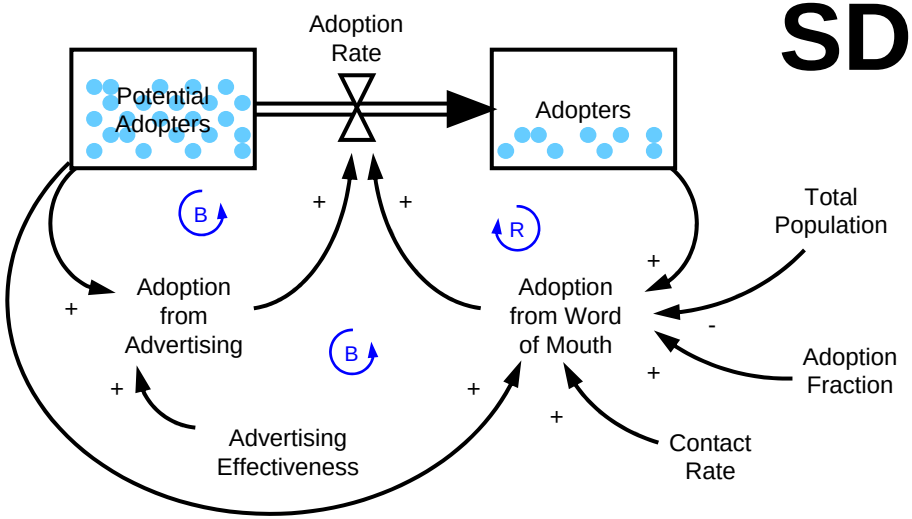
exponential(Contact Rate * Adoption Fraction)

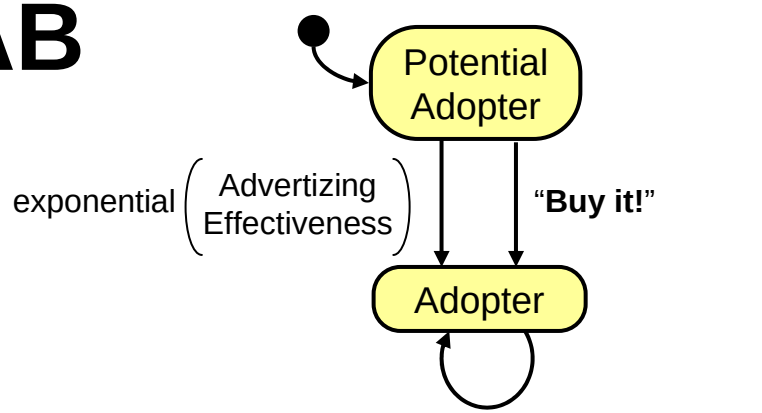
<random agent>."**Buy it!**"

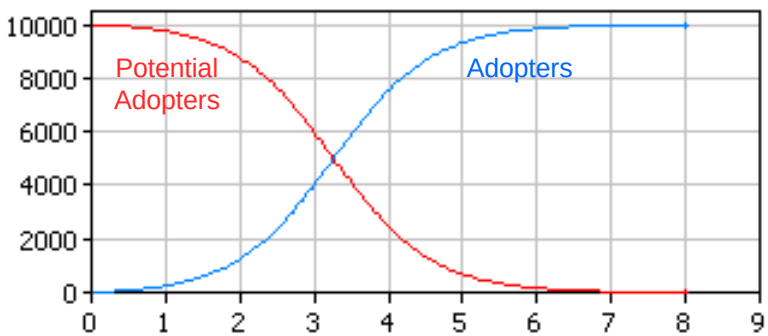


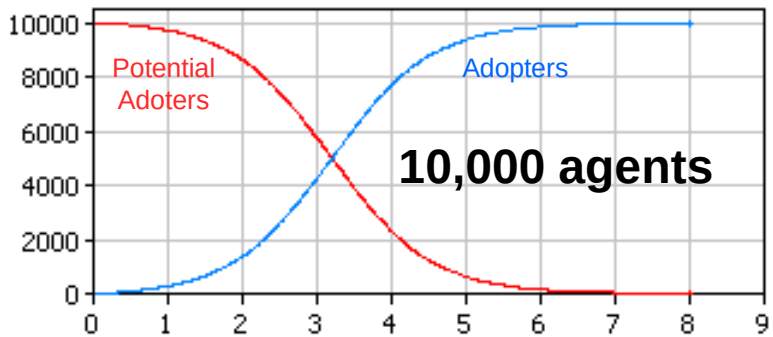


Simulation results: more agents





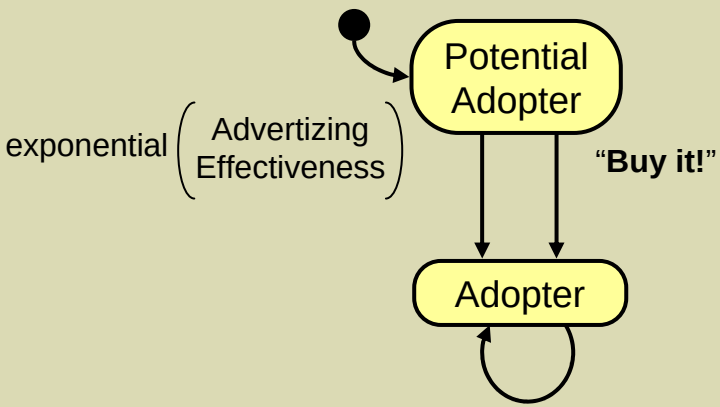




Capturing more with AB model

What if WOM effect of a person depends on how recent is a purchase (adoption)?

Person

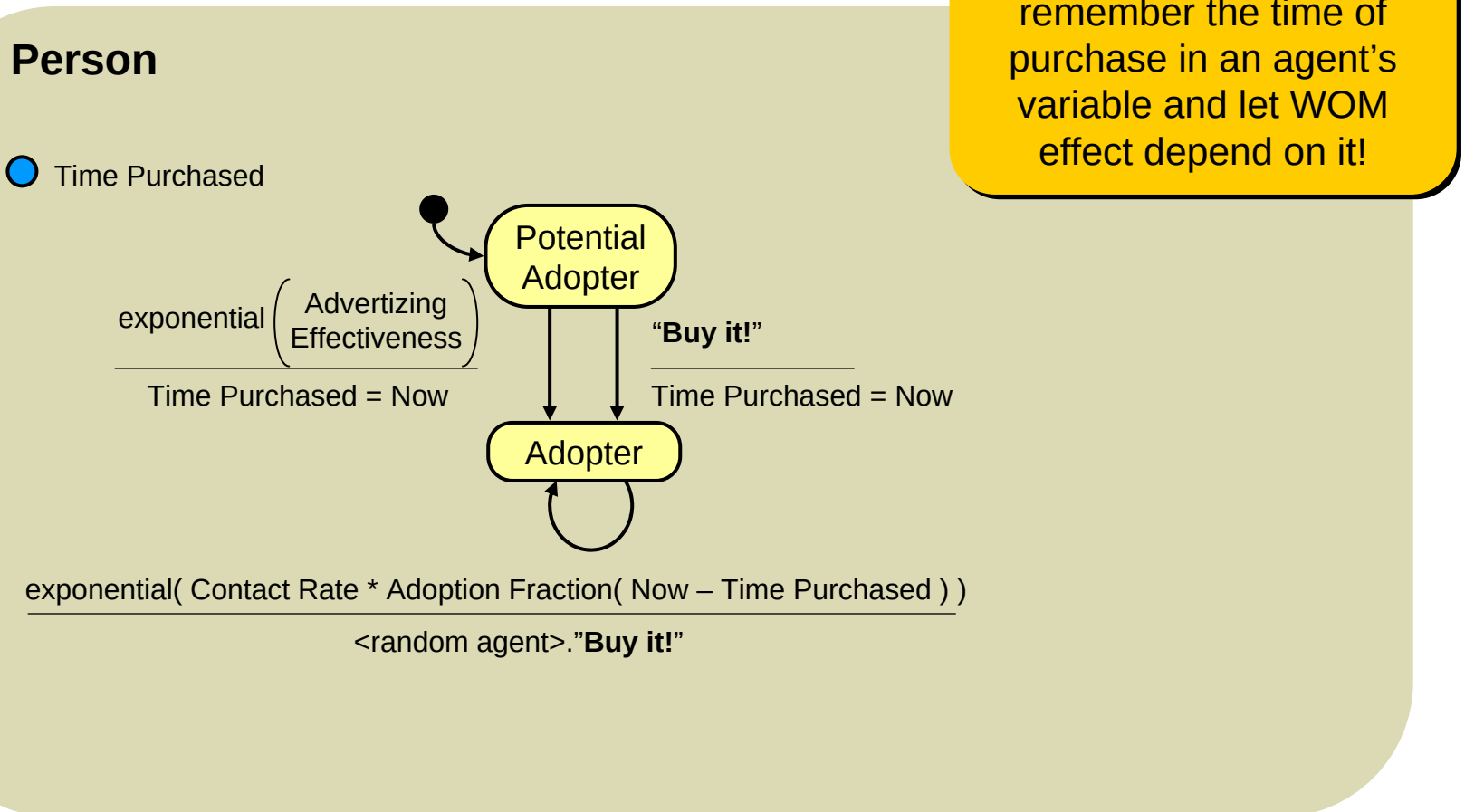


exponential(Contact Rate * Adoption Fraction)

<random agent>."**Buy it!**"

We can store individual information

No problem! We can remember the time of purchase in an agent's variable and let WOM effect depend on it!



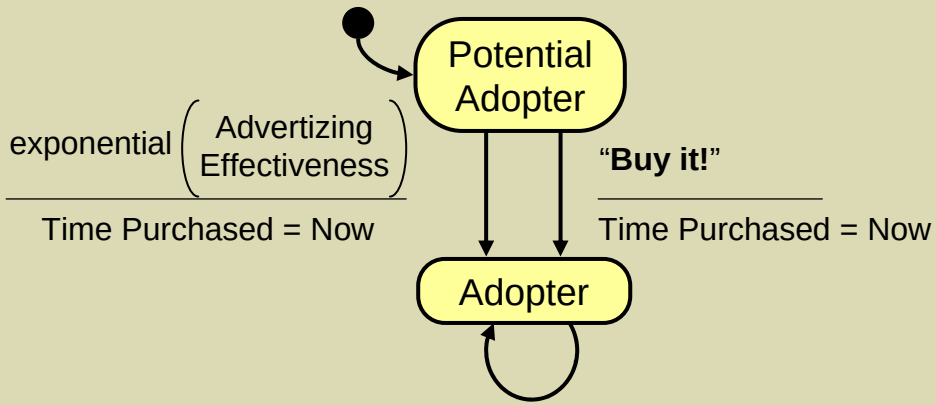
...

Same for Age – just remember the birth date

Person

Birth date

Time Purchased

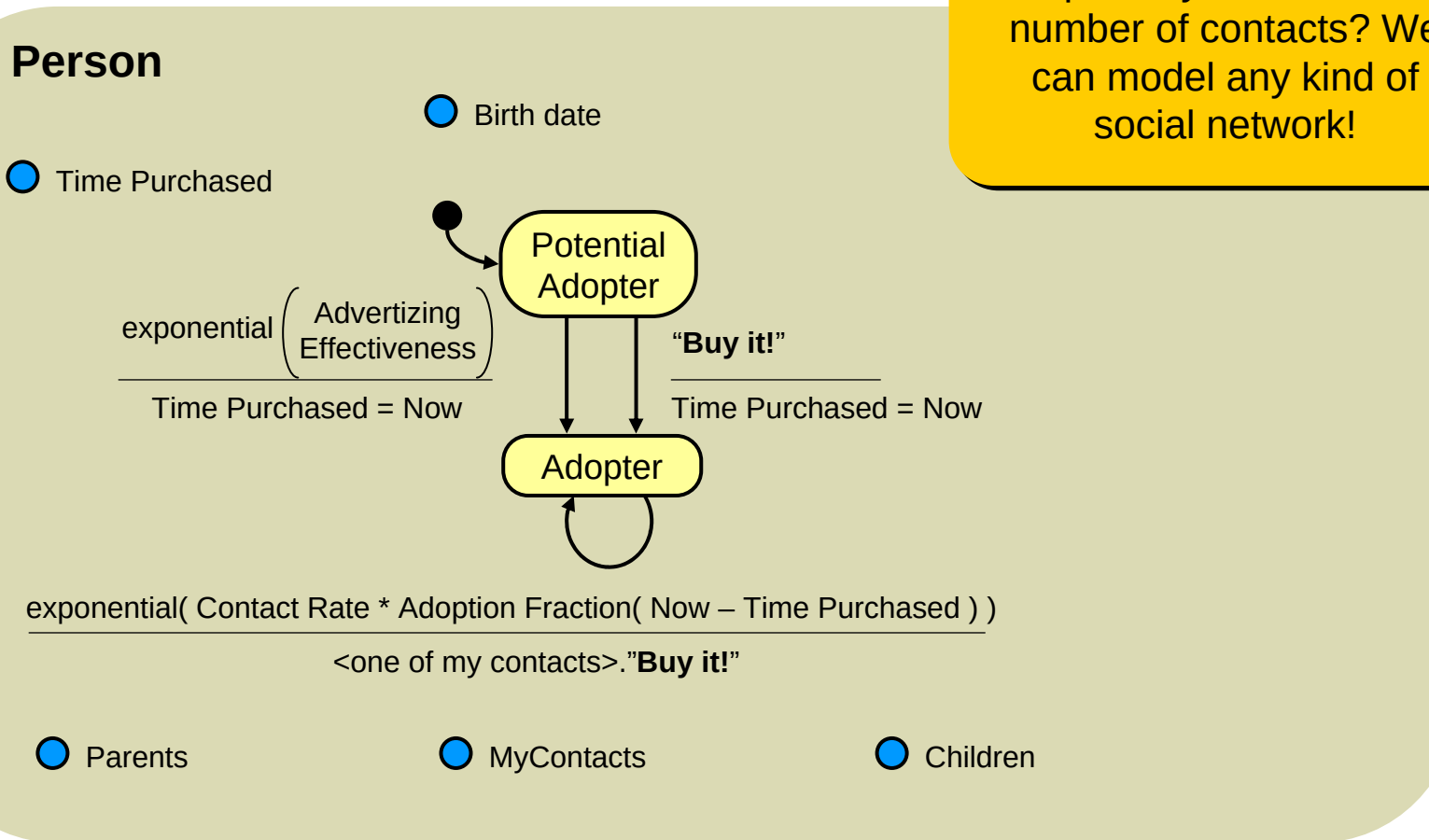


exponential(Contact Rate * Adoption Fraction(Now – Time Purchased))

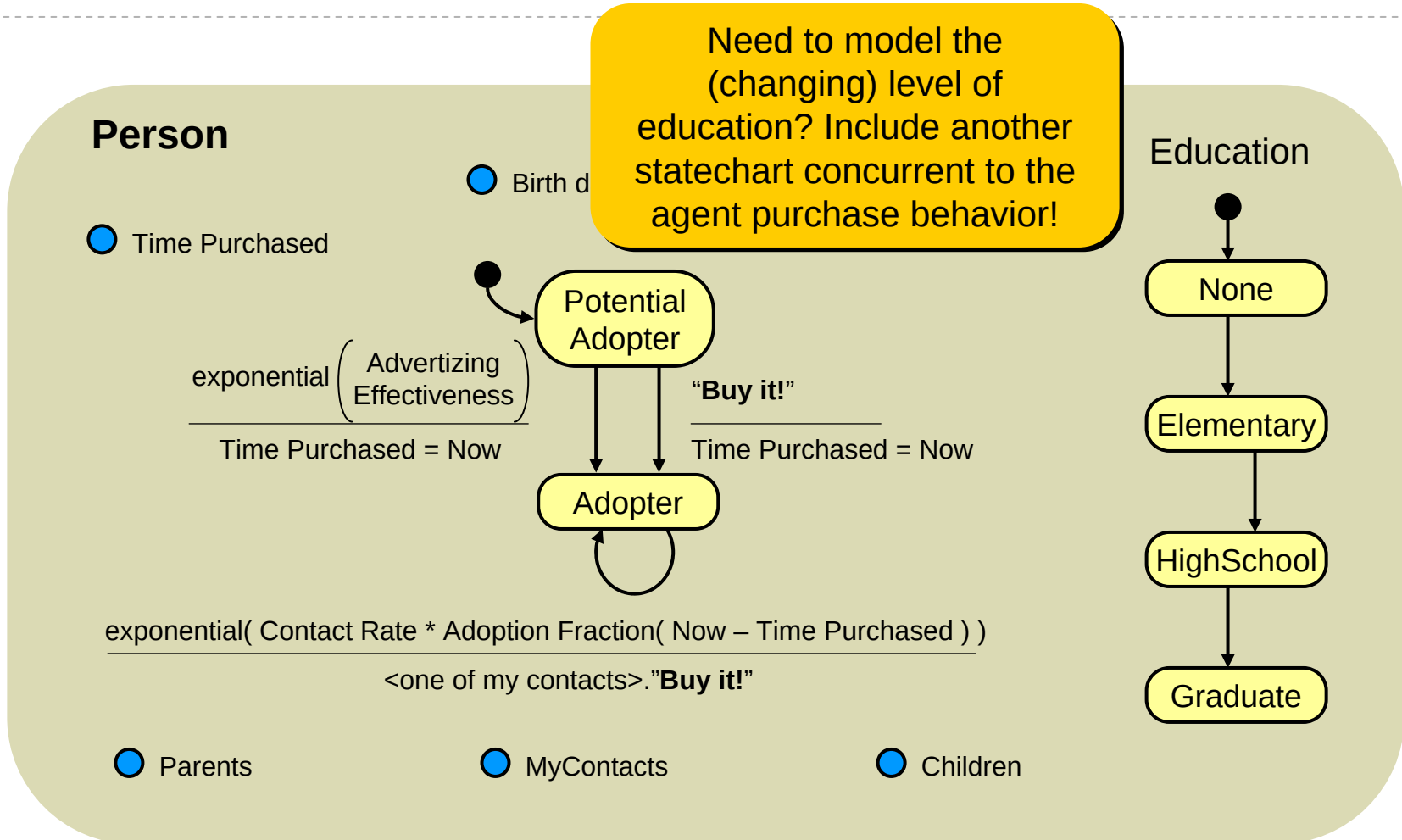
<random agent>."**Buy it!**"

We can maintain social networks

People only have a limited number of contacts? We can model any kind of social network!

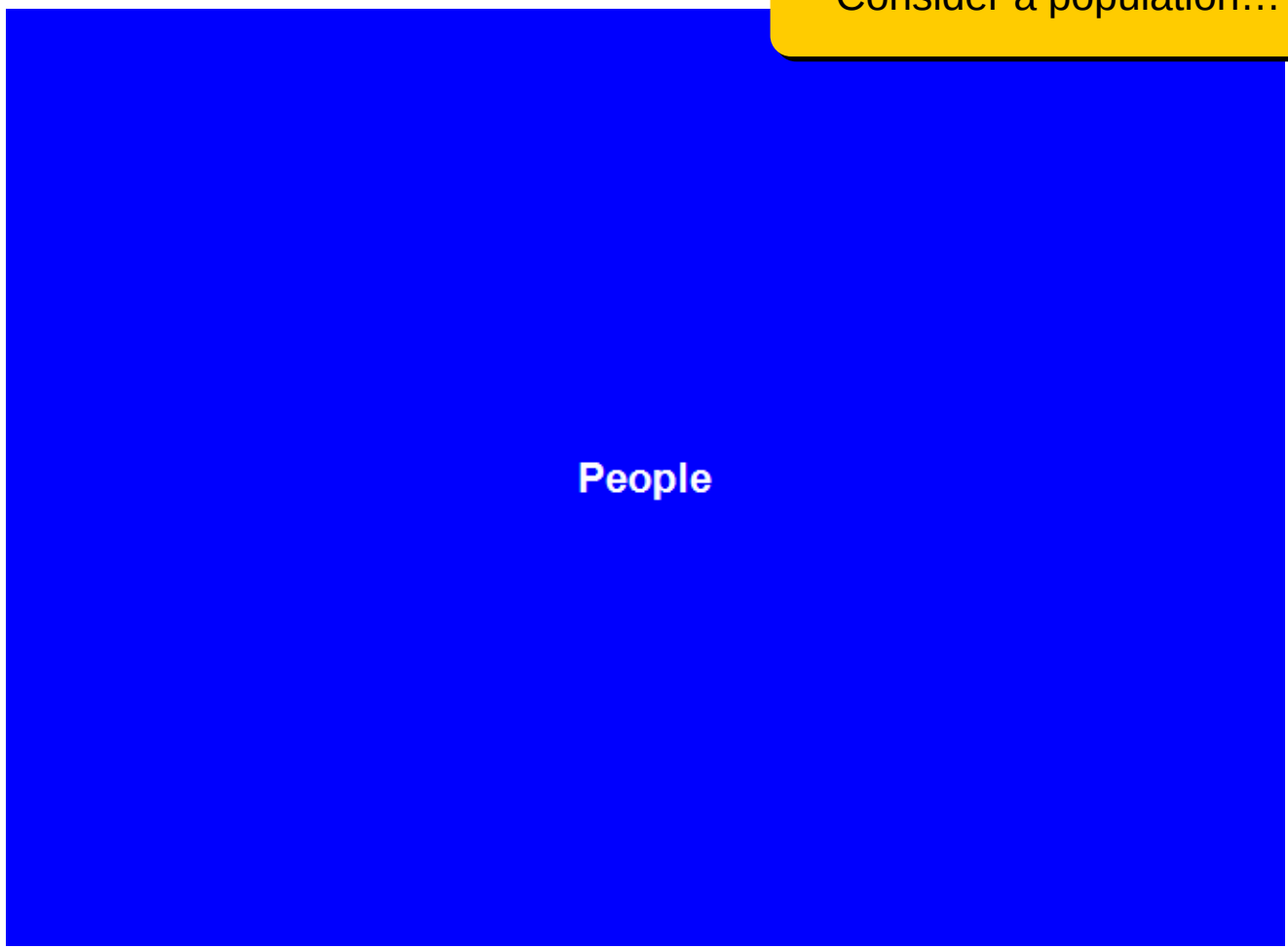


We can model co-behaviors!

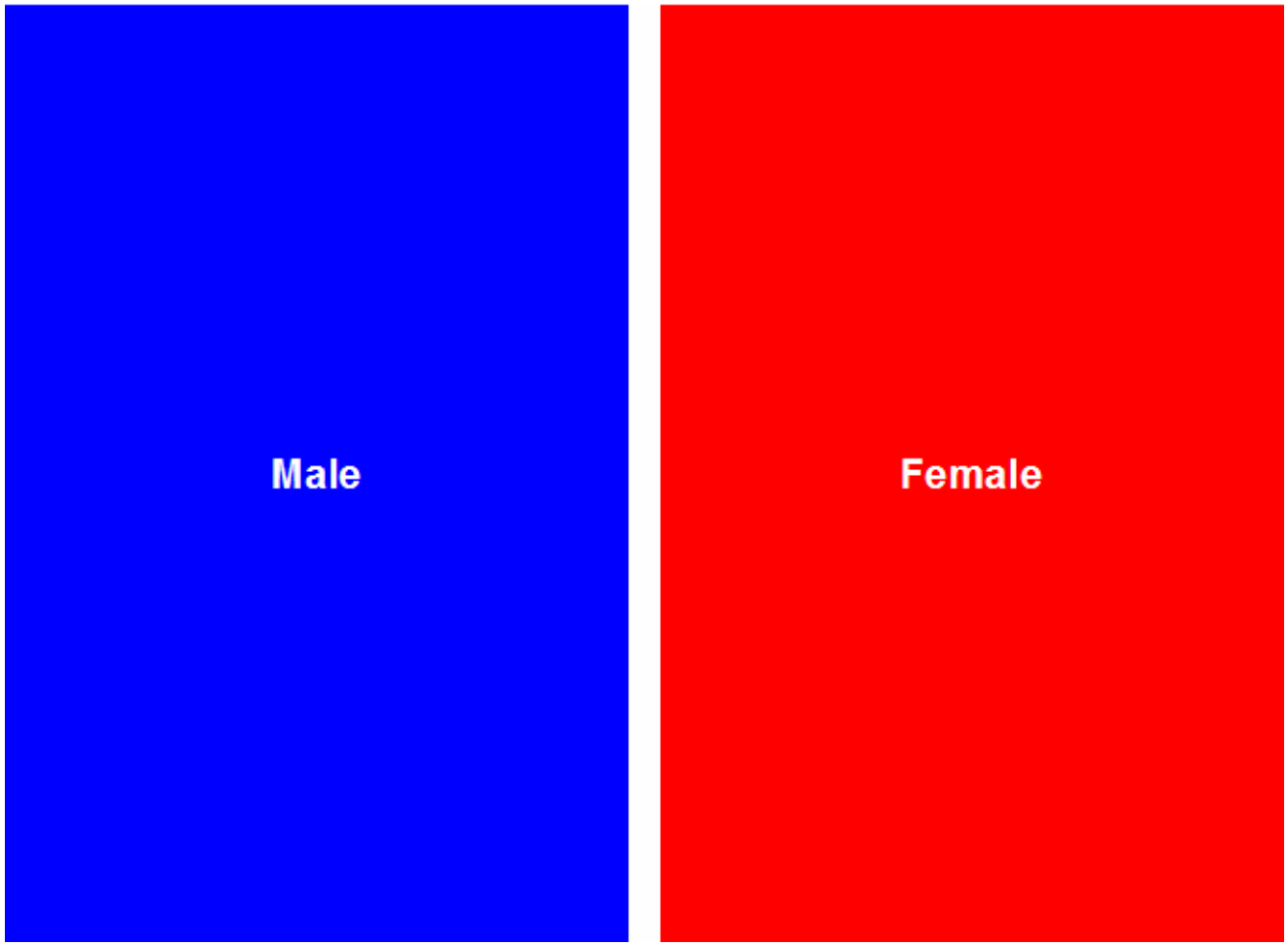


Can you do that in SD? You can try…

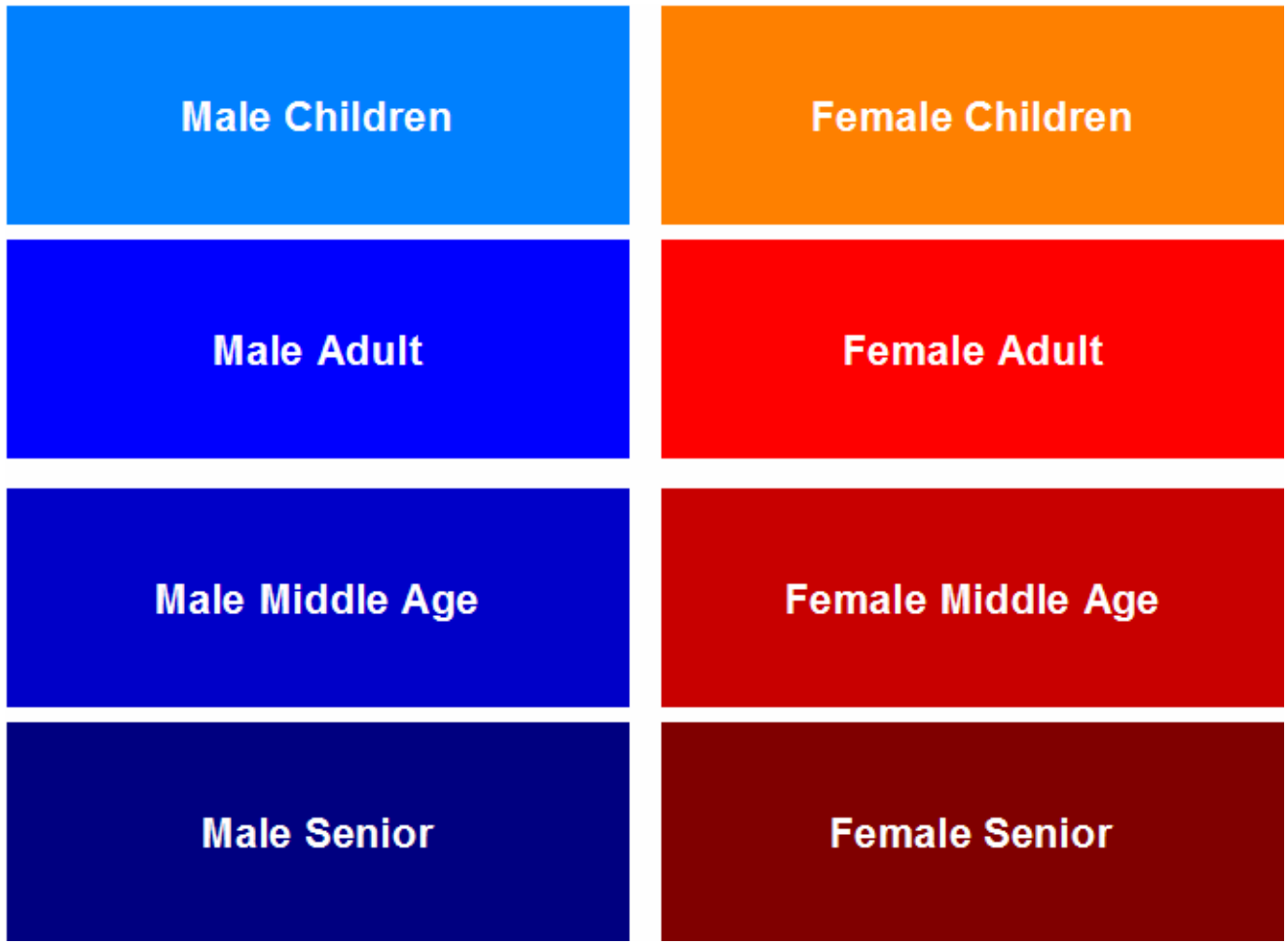
Consider a population…



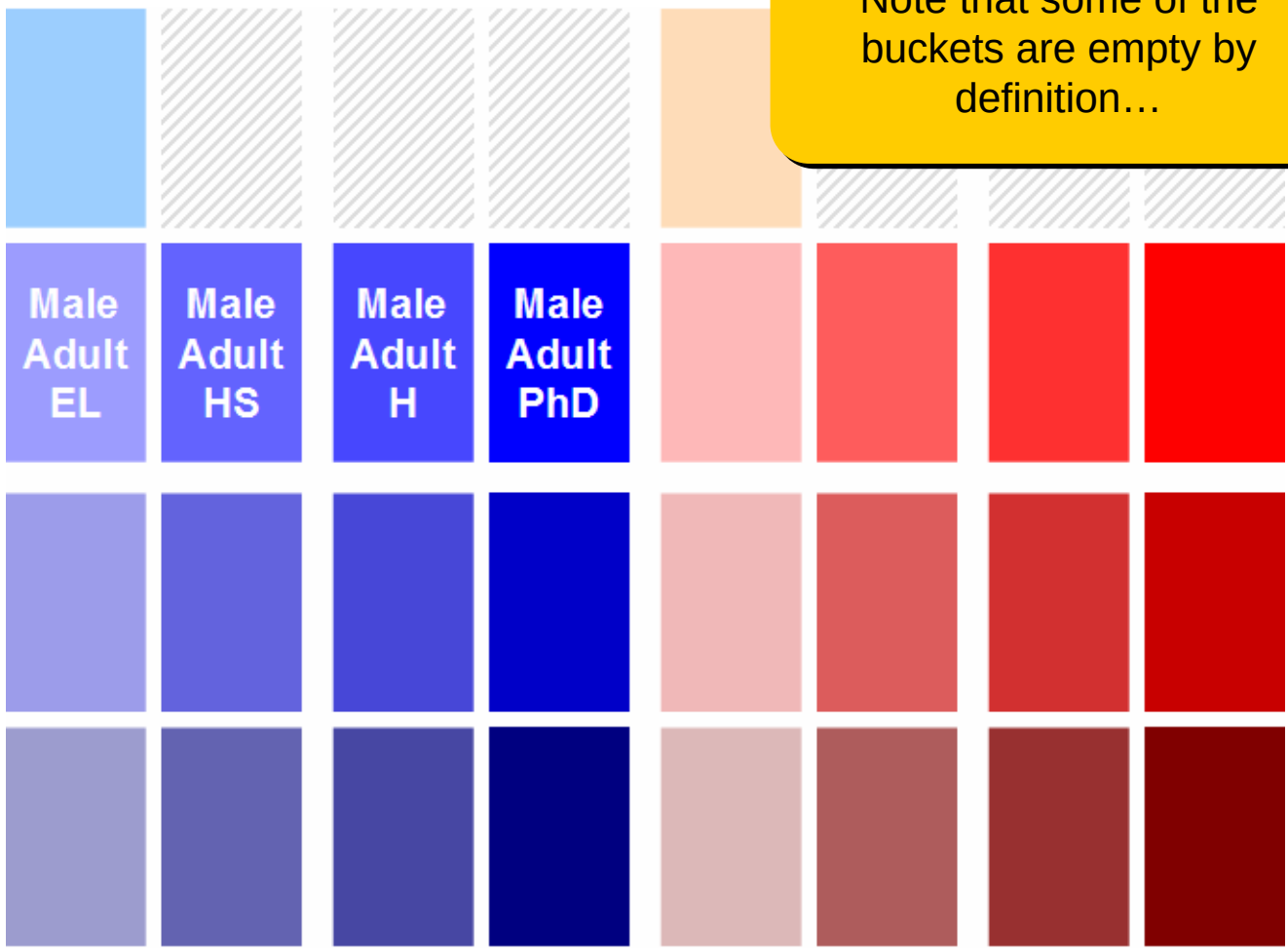
... + Gender



... + Age

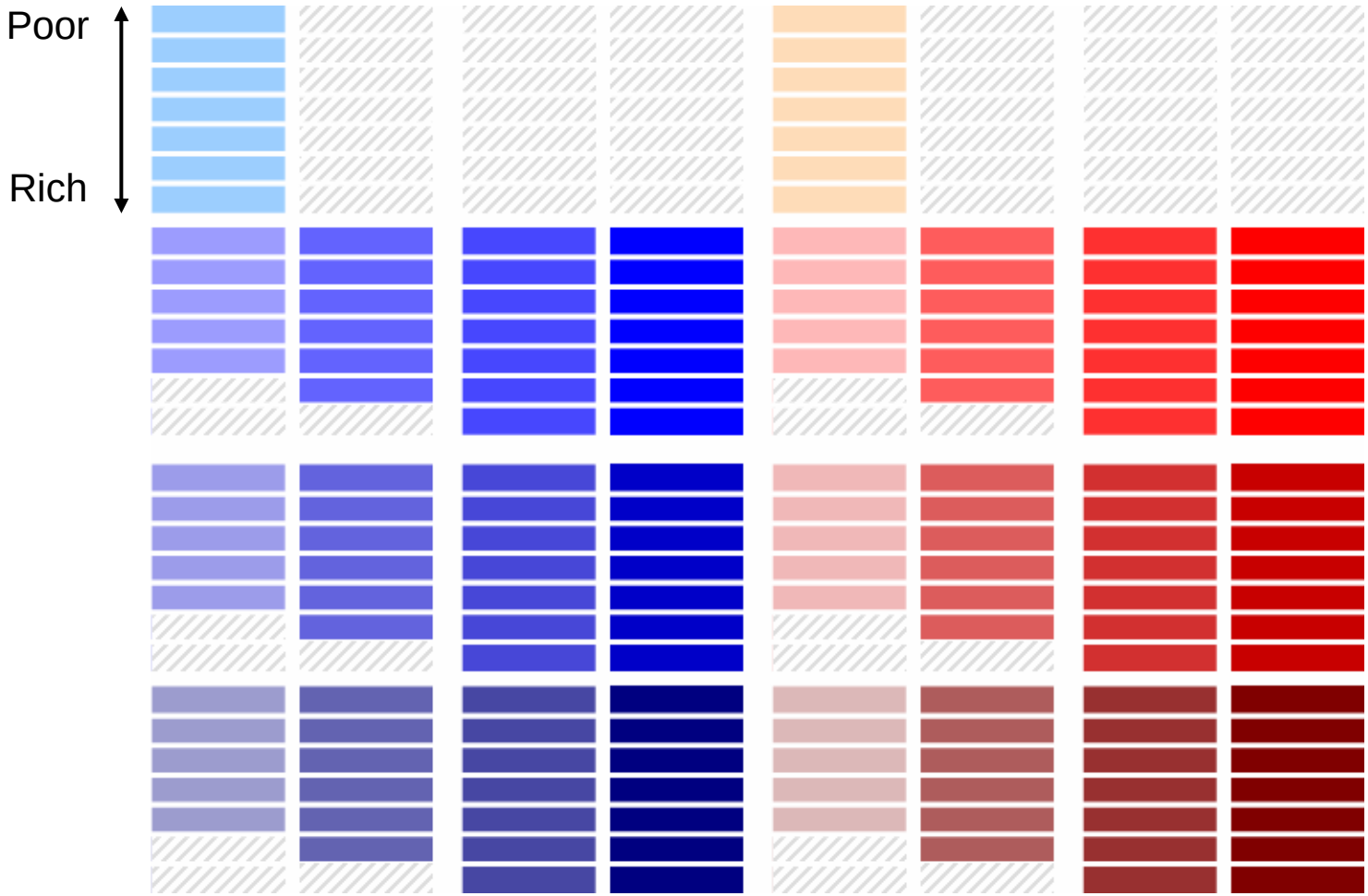


… + Education



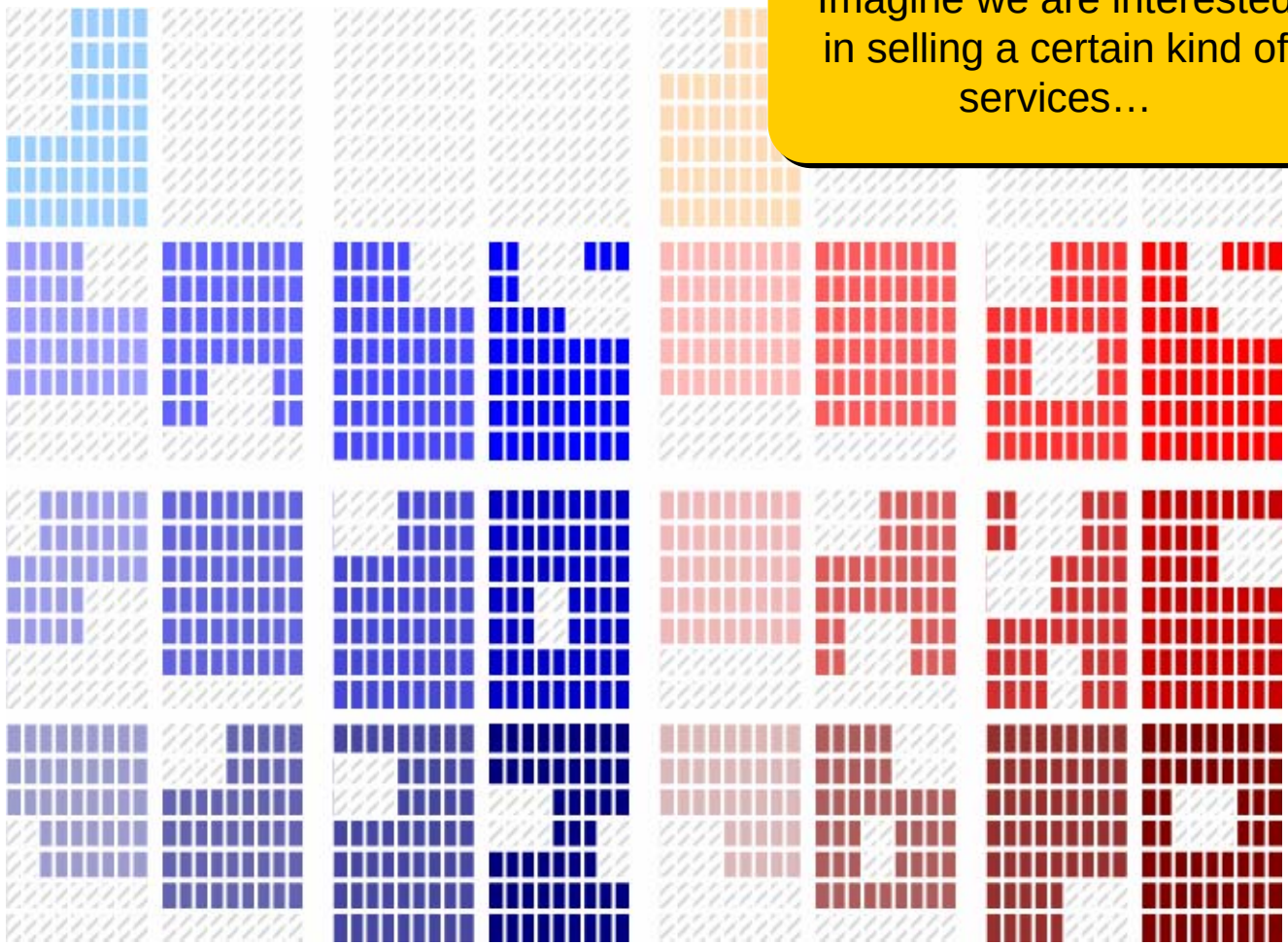
Note that some of the buckets are empty by definition…

… + Wealth

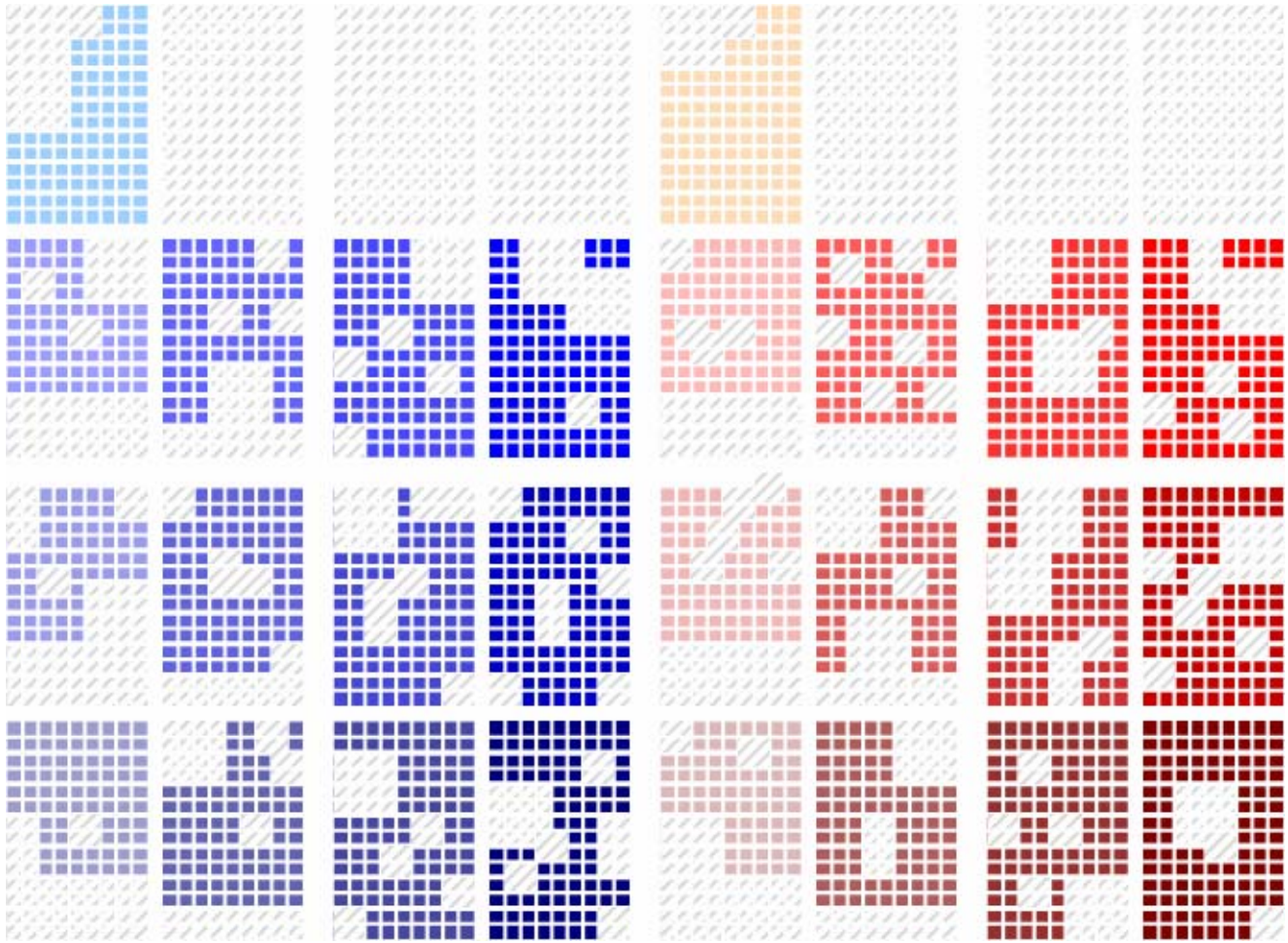


… + Has used our services within N months

Imagine we are interested in selling a certain kind of services…



... + ???

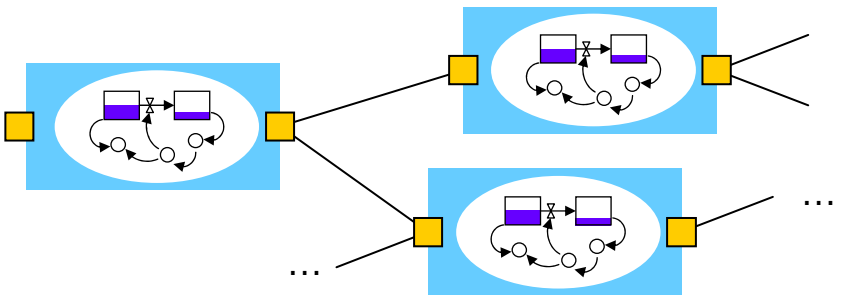


… and finally:

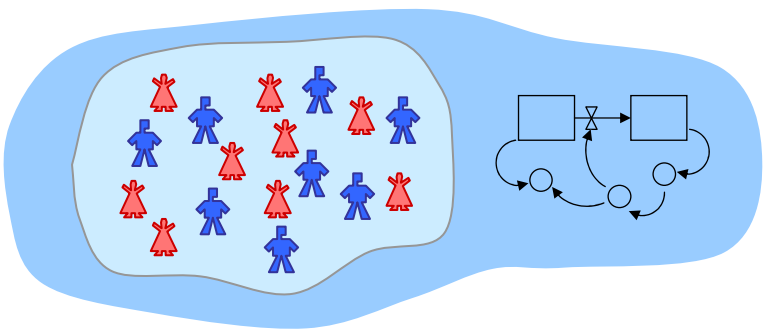
- You may end up having more buckets in the stock than there are people in the region/city/country/world
- In this case agent based model will not only be more compact and adequate, it will be even computationally efficient

Is AB a replacement for SD?

- No! There is a huge class of problems best modeled with SD
- But: there are problems best addressed with AB
- And: in many cases combined, multi-approach modeling is the answer:



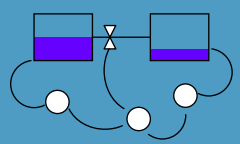
System Dynamics Sub-Models inside discretely communicating Agents
Application example: Supply Chain.



Agents live in an Environment modeled in System Dynamics way
Application example: City Population

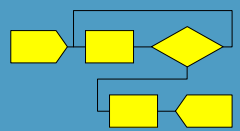
Traditional tools: support one paradigm

SD



VenSim
PowerSim
iThink
ModelMaker

DE

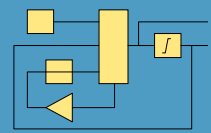


Arena
Extend
SimProcess
AutoMod
PROMODEL
Enterprise Dynamics
FlexSim
eMPlant
…

AB

[Academic software:]
Swarm
RePast
AgentSheets
ASCAPE
SeSam
NetLogo
…

DS



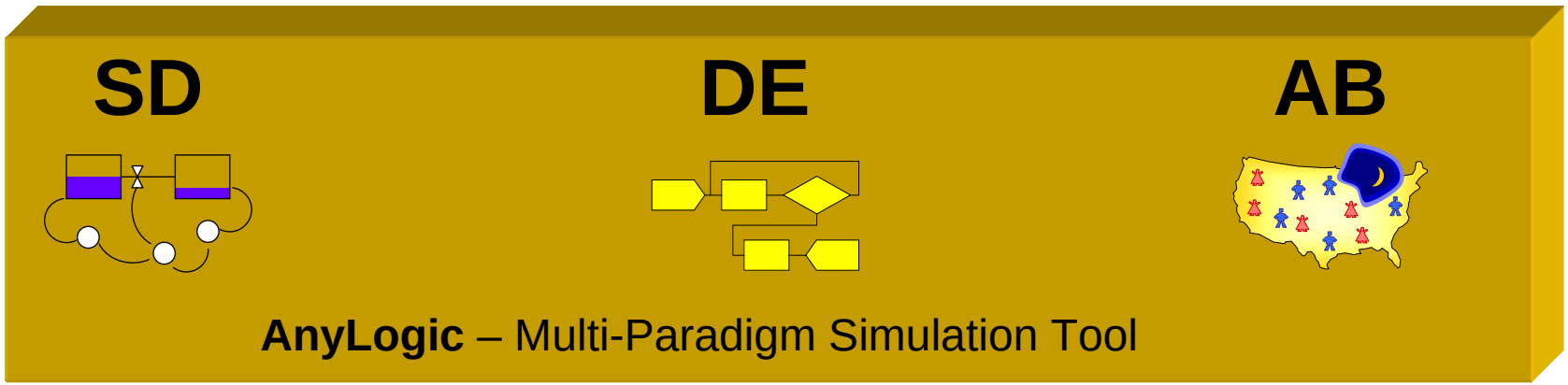
MATLAB
VisSim
LabView
Easy 5
…

The challenge

- The increasing demand for global business optimization have caused leading modelers to look at AB and combined approaches to get deeper insight into complex interdependent processes having very different natures

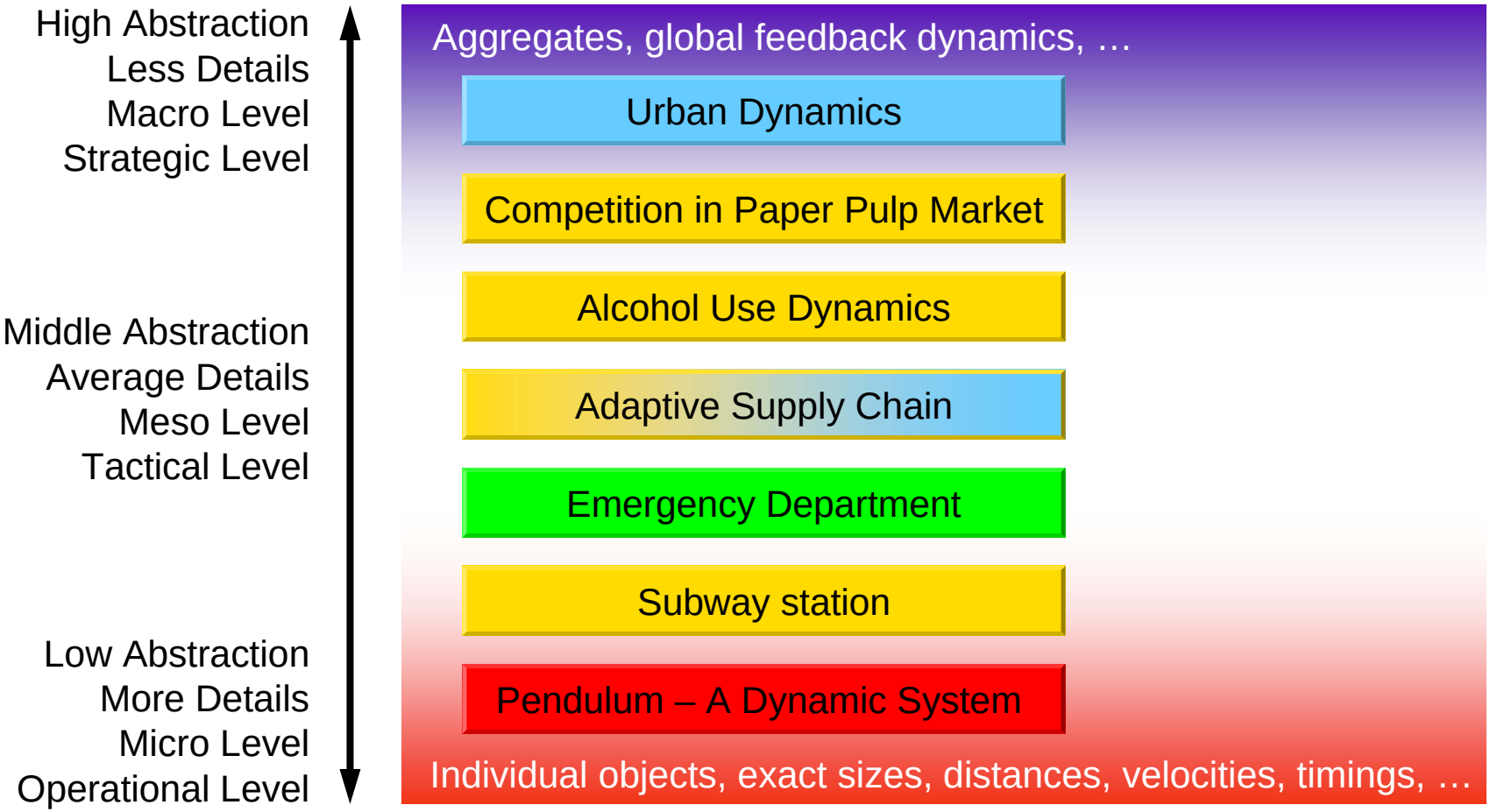
- There is a request for platforms that would allow for integration and efficient cooperation between different modeling paradigms

AnyLogic

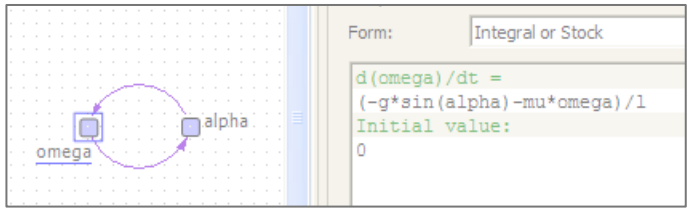


- You can easily vary and adjust the level of abstraction
- You can switch from one approach to another
- You can mix approaches
- All that on one solid object-oriented platform

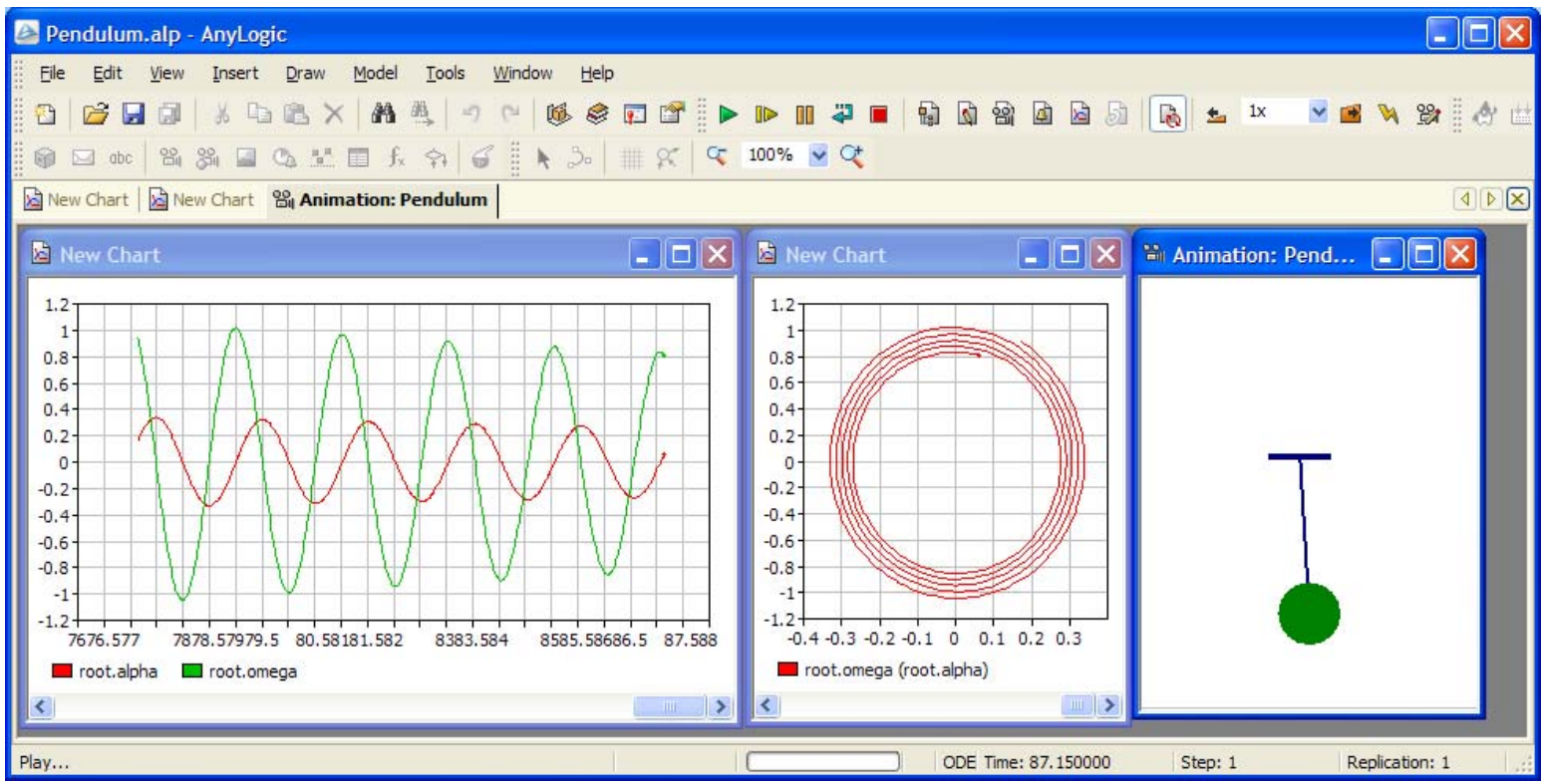
AnyLogic example models



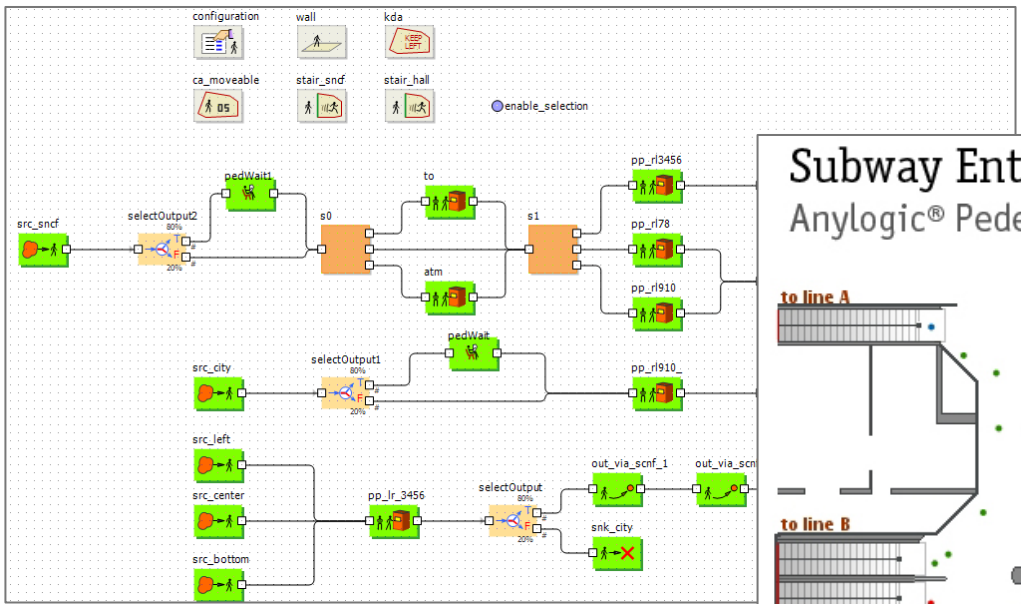
Lowest abstraction level: dynamic system



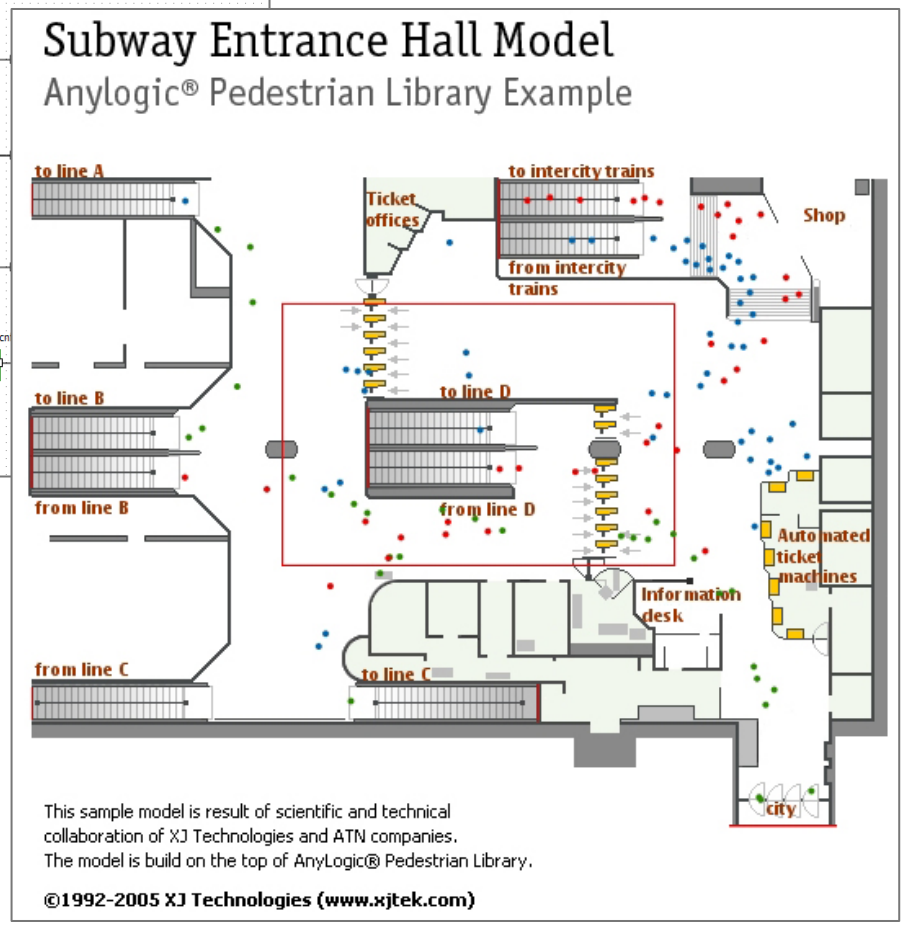
- Design time view: differential equations
- Run time view: charts and animation



Pedestrian dynamics: Subway station

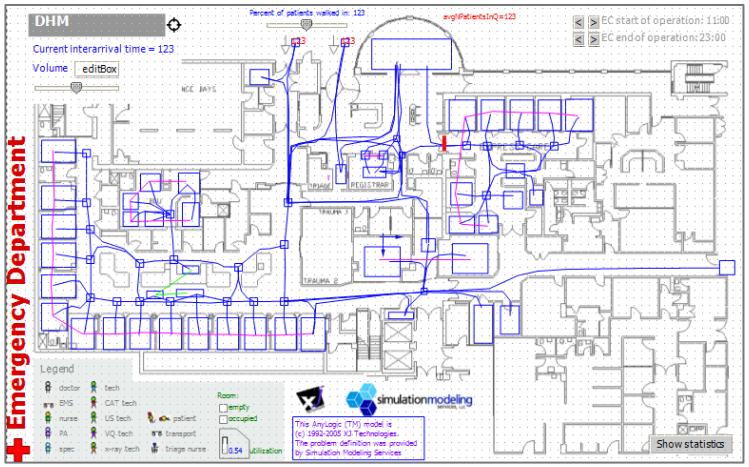


Design time view:
chart defining pedestrian flows



Run time view:
interactive pedestrian simulation

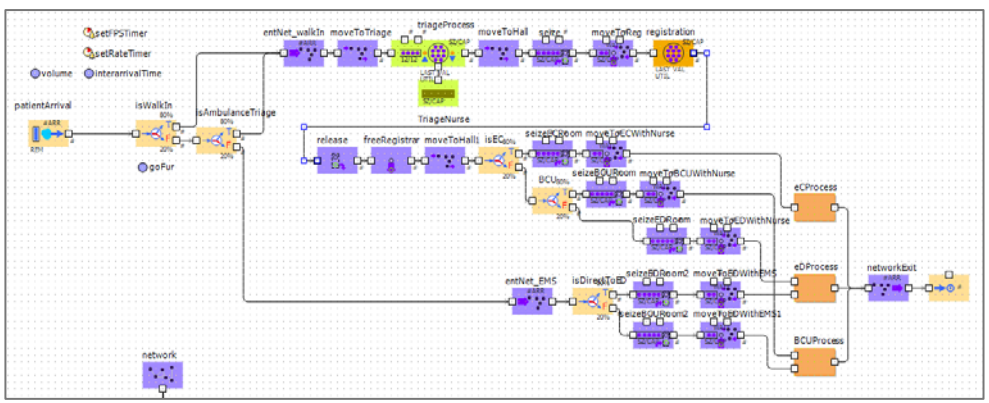
Discrete event: Emergency department

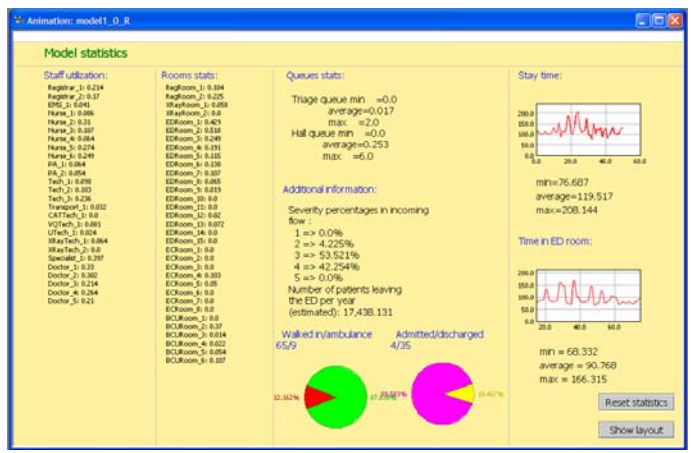


Design time view: layout markup

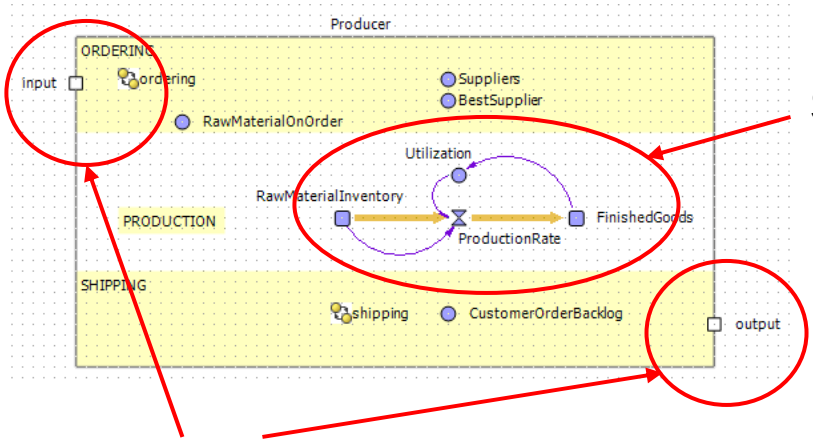
Design time view: process flowchart

Run time view: process animation and statistics





SD+AB: Supply chain

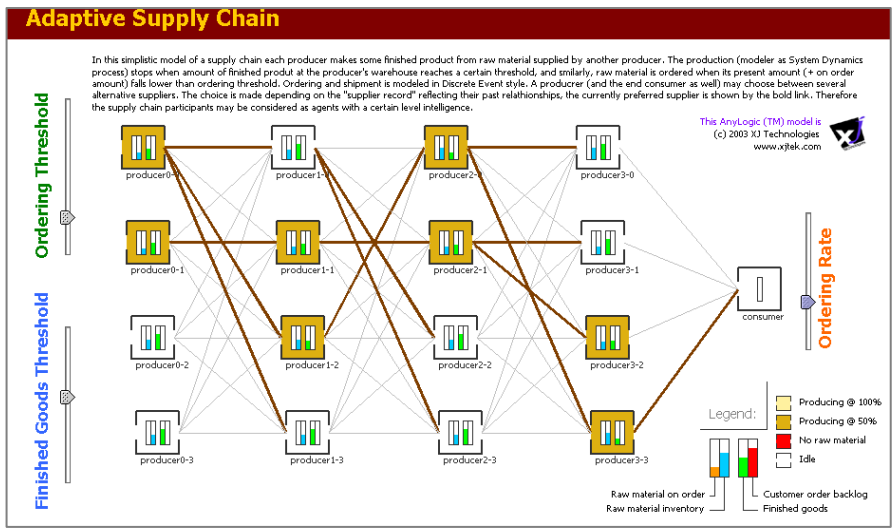


SD model of production process

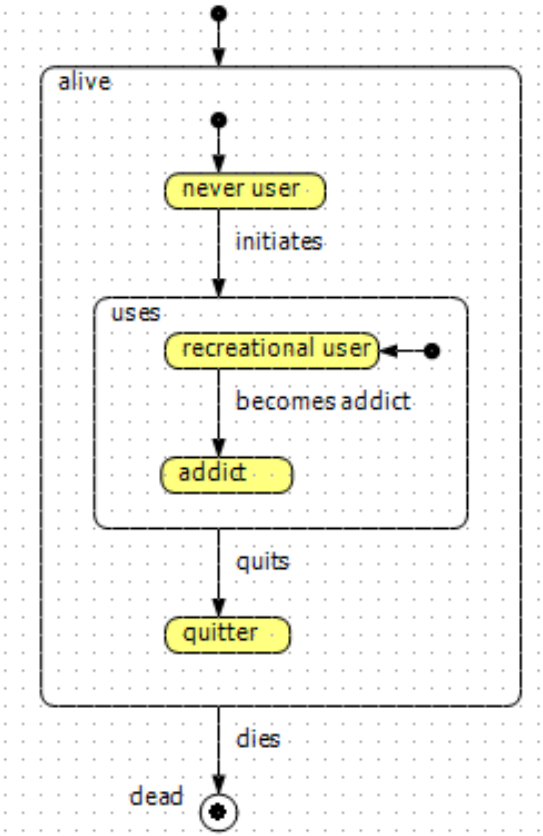
Design time view (Producer)

Discrete model of communication with suppliers and customers

Run time view: ordering pattern

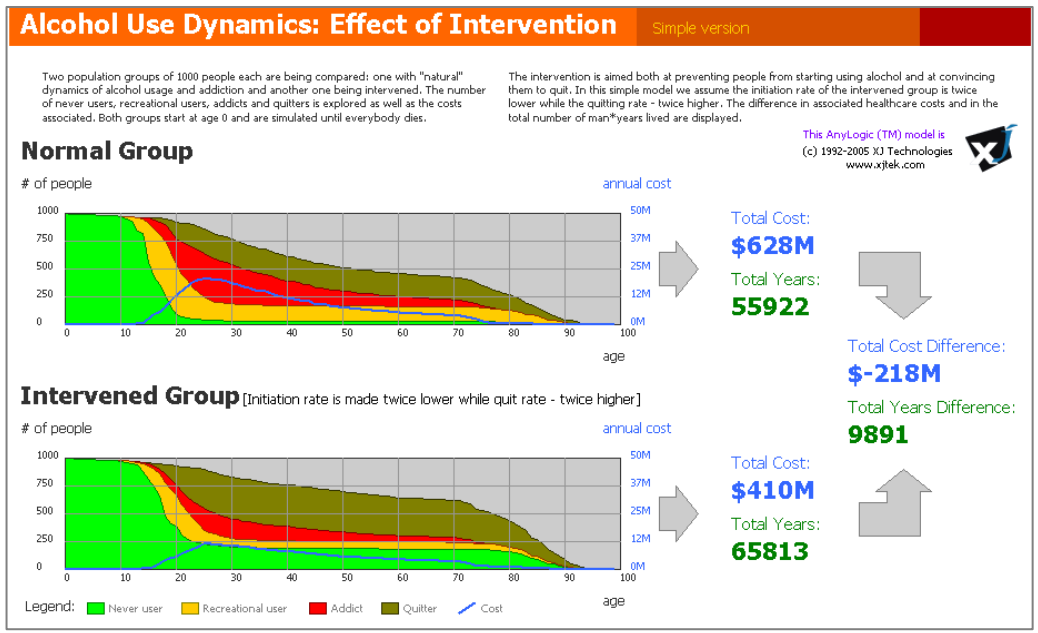


Agent based: Alcohol use dynamics



- Design time view:
  statechart defining individual person behavior

Run time view: control and intervened group dynamics and financial outcome

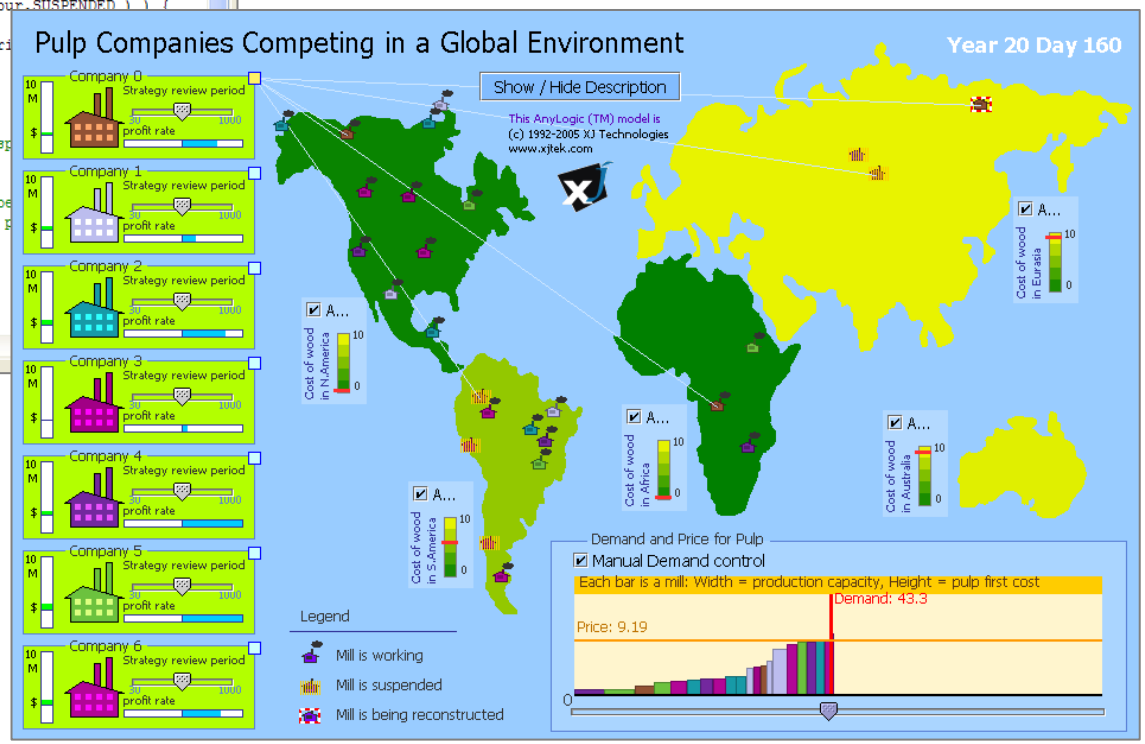


Agent based: Competition in global market

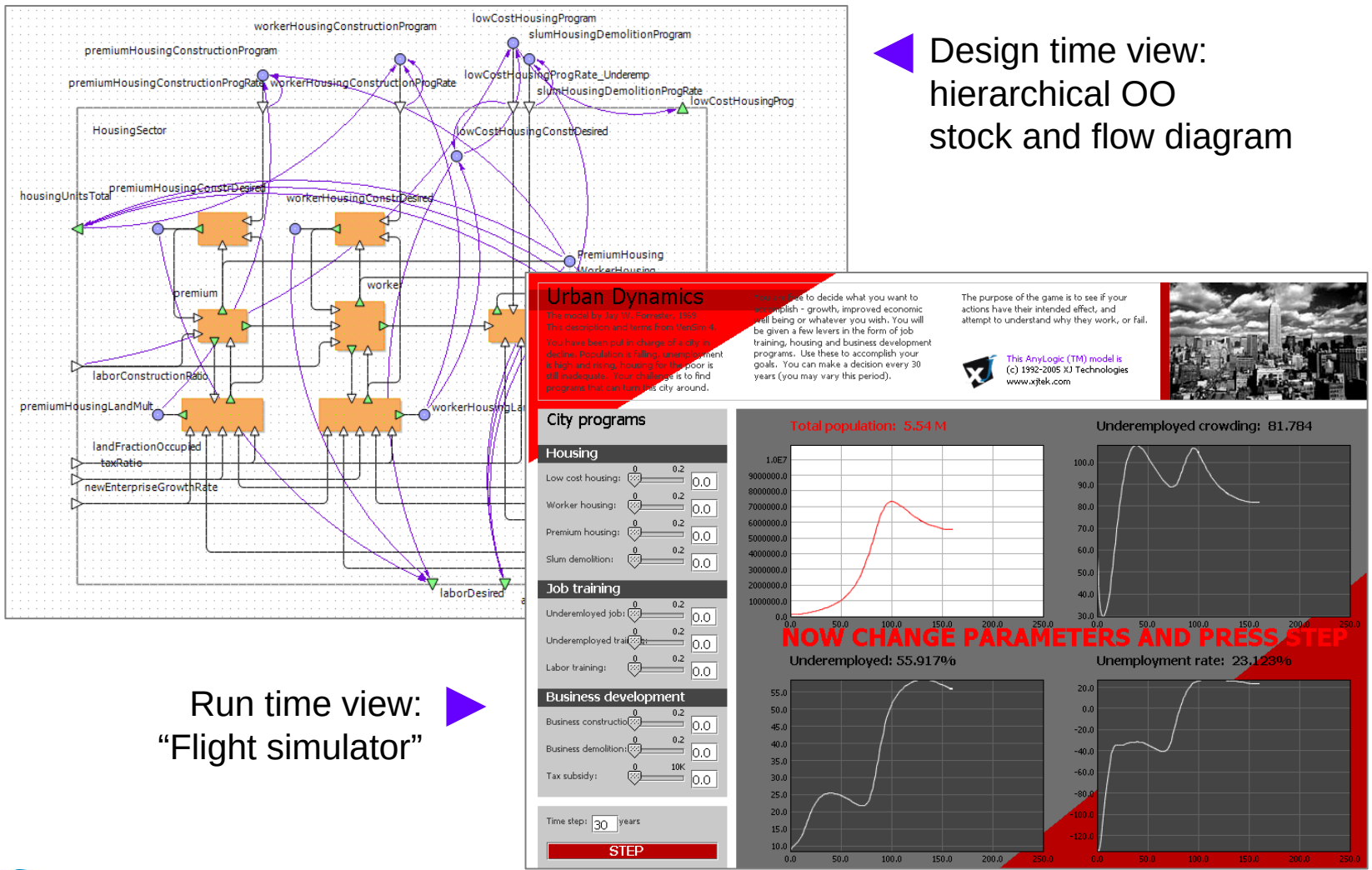
Function body:

```
// Get root object
Model model = (Model)(Engine.getRoot());

// Apply strategy to the existing company mills
for( int i=0; i<mills.size(); i++) {
  Mill mill = mills.item(i);

  if( mill.behaviour.isStateActive( mill.behaviour.WORKING ) ) {

    if( mill.costOfWood + mill.getProcessingCost() > price ) {
      // Suspend unprofitable working mill
      mill.port.receive( "SUSPEND" );
    }

  } else if( mill.behaviour.isStateActive( mill.behaviour.SUSPENDED ) ) {

    if( mill.costOfWood + mill.getProcessingCost() < pri
      // Resume profitable suspended mill
      mill.port.receive( "RESUME" );
    } else {
      if( mill.technologyLevel < 4.5 ) {
        // Reconstruct reconstructable unprofitable susp
        mill.port.receive( "RECONSTRUCT" );
      } else {
        // Close unprofitable suspended mill that has be
        // reconstructed many times but has not become p
        // Unregister the mill at the region
        for( int j=0; j<model.regions.size(); j++ )
          model.regions.item(j).mills.remove( mill );
        // and unregister at the model
        model.mills.remove( mill );
```

- Design time view: company strategy function
- Run time view: geo based competition visualization



System dynamics: Urban dynamics



Design time view:
hierarchical OO
stock and flow diagram

Run time view:
“Flight simulator”

Summary

- Choosing modeling approach and the level of abstraction adequate to the goals of the modeling project is a key to success
- Using a flexible, multi-paradigm platform AnyLogic multiplies your capabilities and saves significant amount of model development efforts

Thank you!

- Questions?