Nancy Darling & Richard Salter

Oberlin College Oberlin, OH US

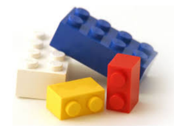
This work was supported by the National Science Foundation
(NSF Grant Number 0939153)

Oberlin Modeling Initiative (OMnI)

- NSF funded program to introduce **dynamic systems thinking** across the Oberlin curriculum
  - *How do you think about complex problems where key elements interact and feed back into each other?*

- Challenge
  - Build a tool to support dynamic systems thinking for a broad, multi-disciplinary user group
  - Teach ***thinking*** not ***programming***
  - Support collaborative, multi-disciplinary problem solving

Nova concept



- Flexible modeling platform suitable for student exercises but powerful enough for serious research applications
- Capable of a full range of dynamic models
  - Stock and flow
  - Spatial
  - Agent based
- Multi-platform
- Modular

No cost license!

Nova: A Java-based platform that operates at multiple levels

Agent based and spatial modeling on a stock & flow core

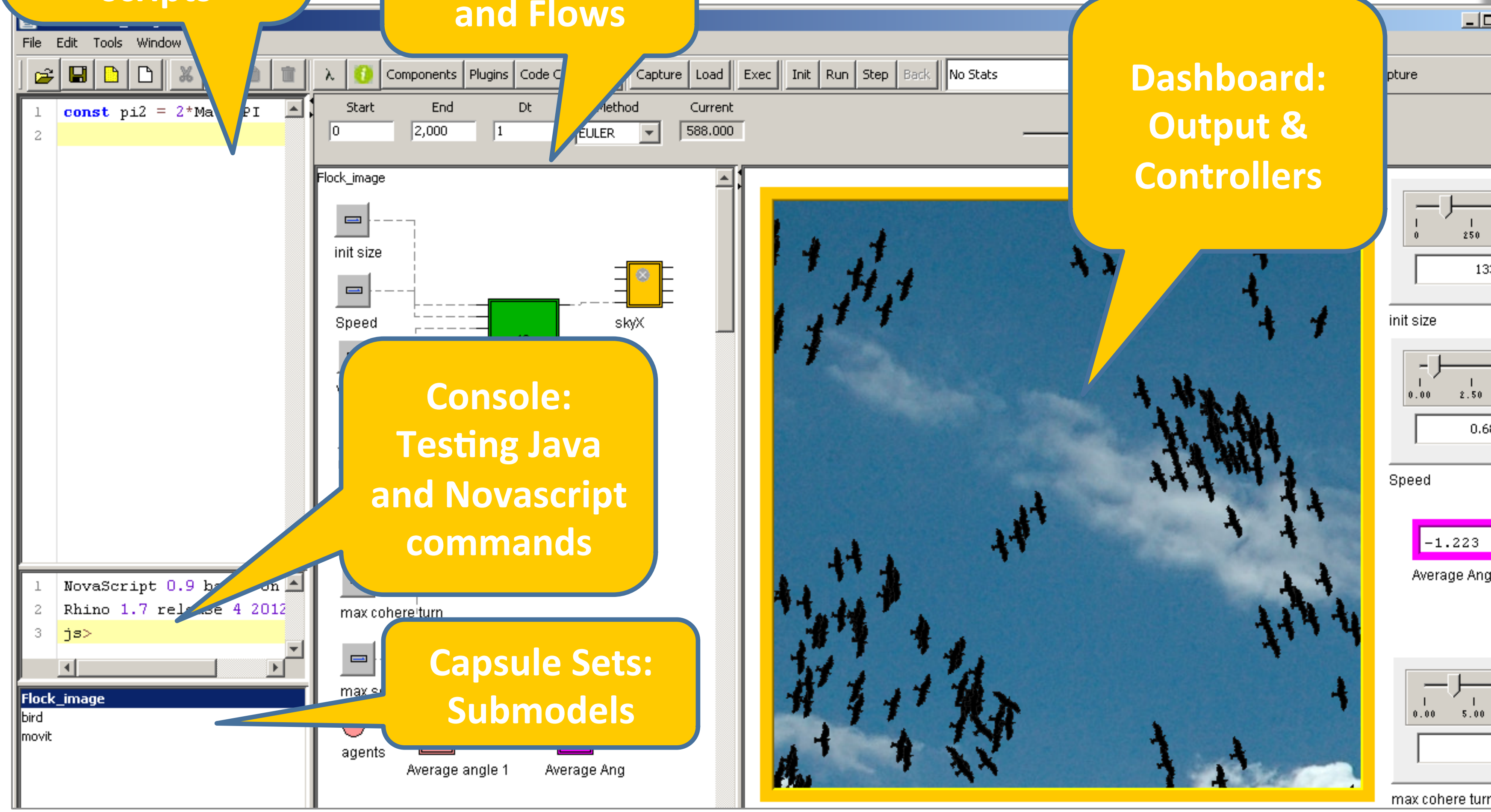
Netlogo or Agentsheets

OM I
Oberlin Modeling Initiative

ew: Nova can operate entirely

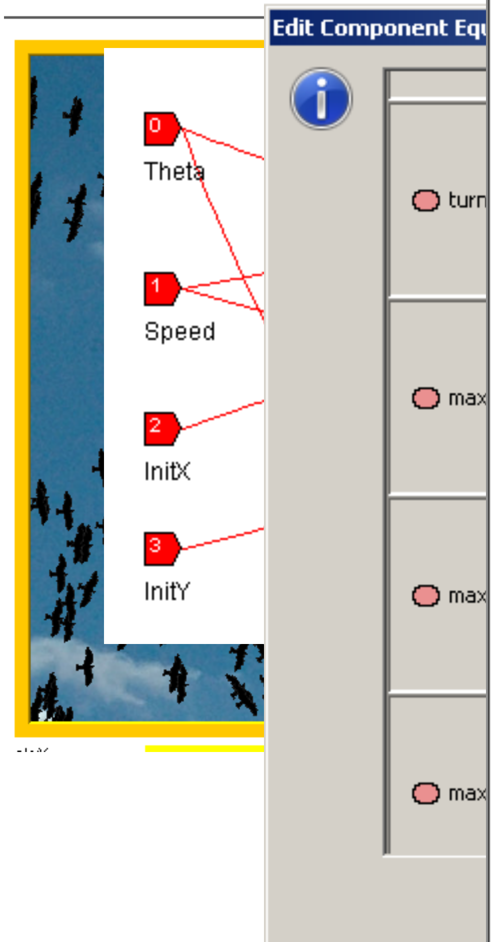
he

Stella, Vensim, Madonna



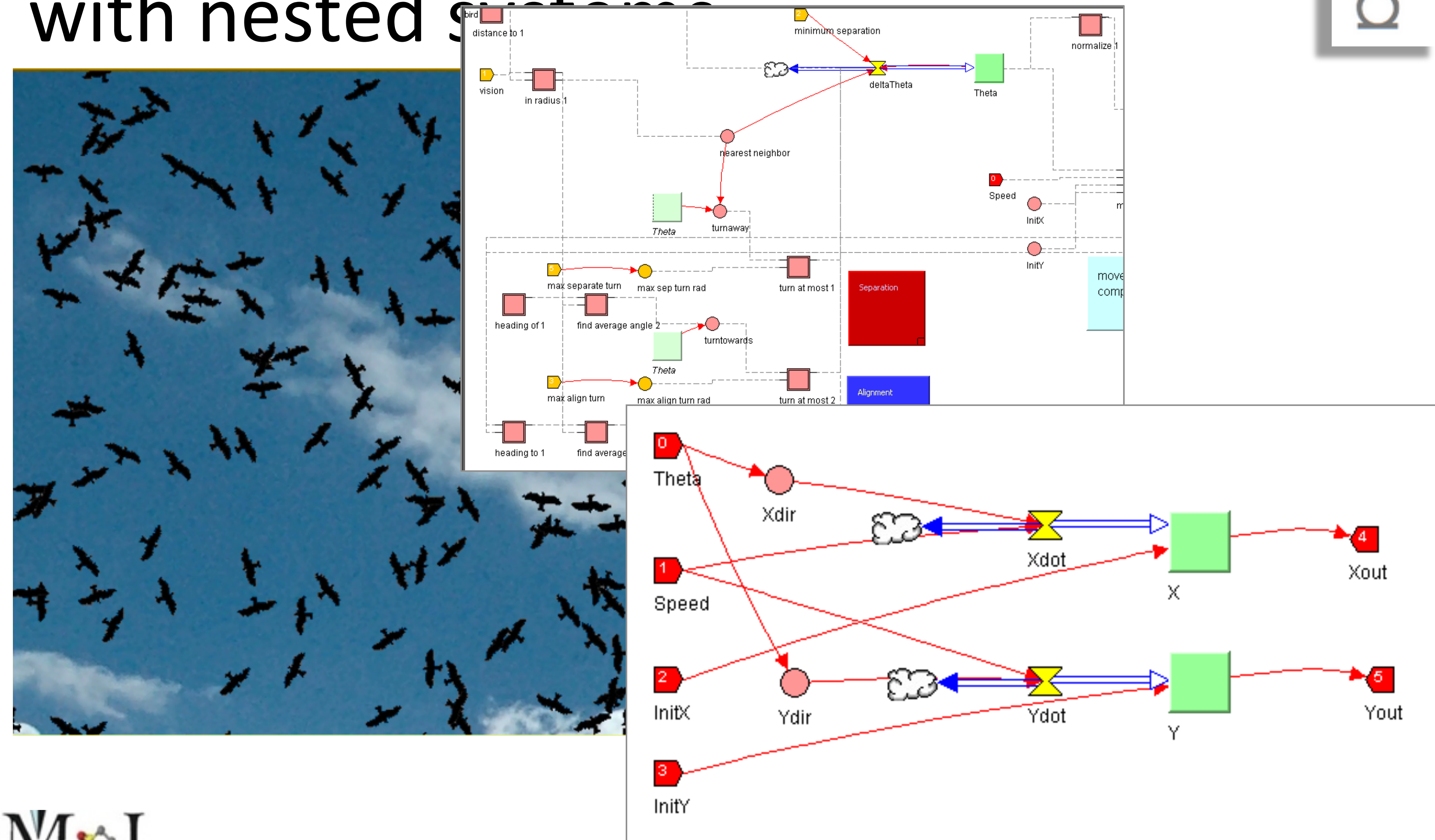
Oberlin Modeling Initiative

Overview: You can work with Nova at multiple le




```
      function() {
        var f;
        f = function(other) {
          var theta = other.Theta;
          var phi = theta % pi2;
          if (phi > Math.PI) phi -= pi2;
          else if (phi <= -Math.PI) phi += pi2;
          return phi;
        }
        return {f:f};
      },
      ['f'], true, false),
    in_radius: Dynamic(
      function(rad,distanceTo) {
        var all,closest;
        var best = Infinity;
        var closest = 0;
        var all = [];
        for (var i = -rad; i < rad; i++)
          for (var j = -rad; j < rad; j++) {
                var coords = CELL_COORDS();
                var y0 = coords.row + i;
                var x0 = coords.col + j;
                if (x0 < 0) x0 = cols + x0;
            if (y0 < 0) y0 = rows + y0;
                if (x0 >= cols) x0 = x0 - cols;
                if (y0 >= rows) y0 = y0 - rows;
            var agentset = AGENTS_AT(y0, x0);
                for (var k = 0; k < agentset.length; k++) {
                var z = agentset[k];
```

Overview: Capsules are what makes **Nova** interesting for people who work with nested s



Oberlin Modeling Initiative

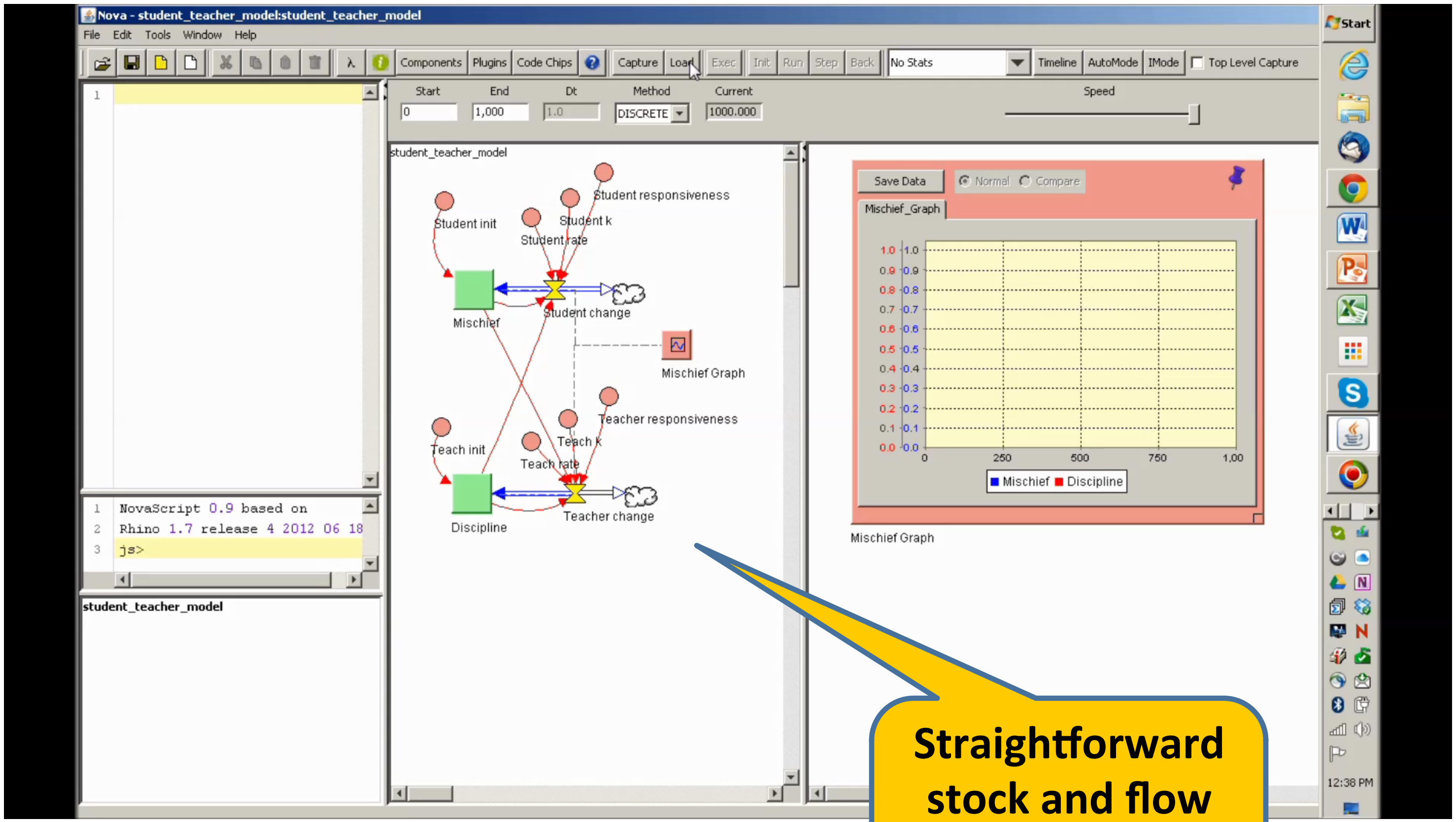
Advantages of Nova as a platform

- Allows people with different expertise to work at different components of the problem
- Allows stock & flow, spatial, and agent models simultaneously, allowing modeling of contagion
- Nested models through capsules and code chips, so you can move from the individual to group level and back down again
- Does not assume homogeneity of the population
- Automated runs across a range of distributions
- Output results graphically, csv, or directly into R
- Allows full integration of R and Java functions

Oberlin Modeling Initiative

Modularity in the Predator-Prey Model:

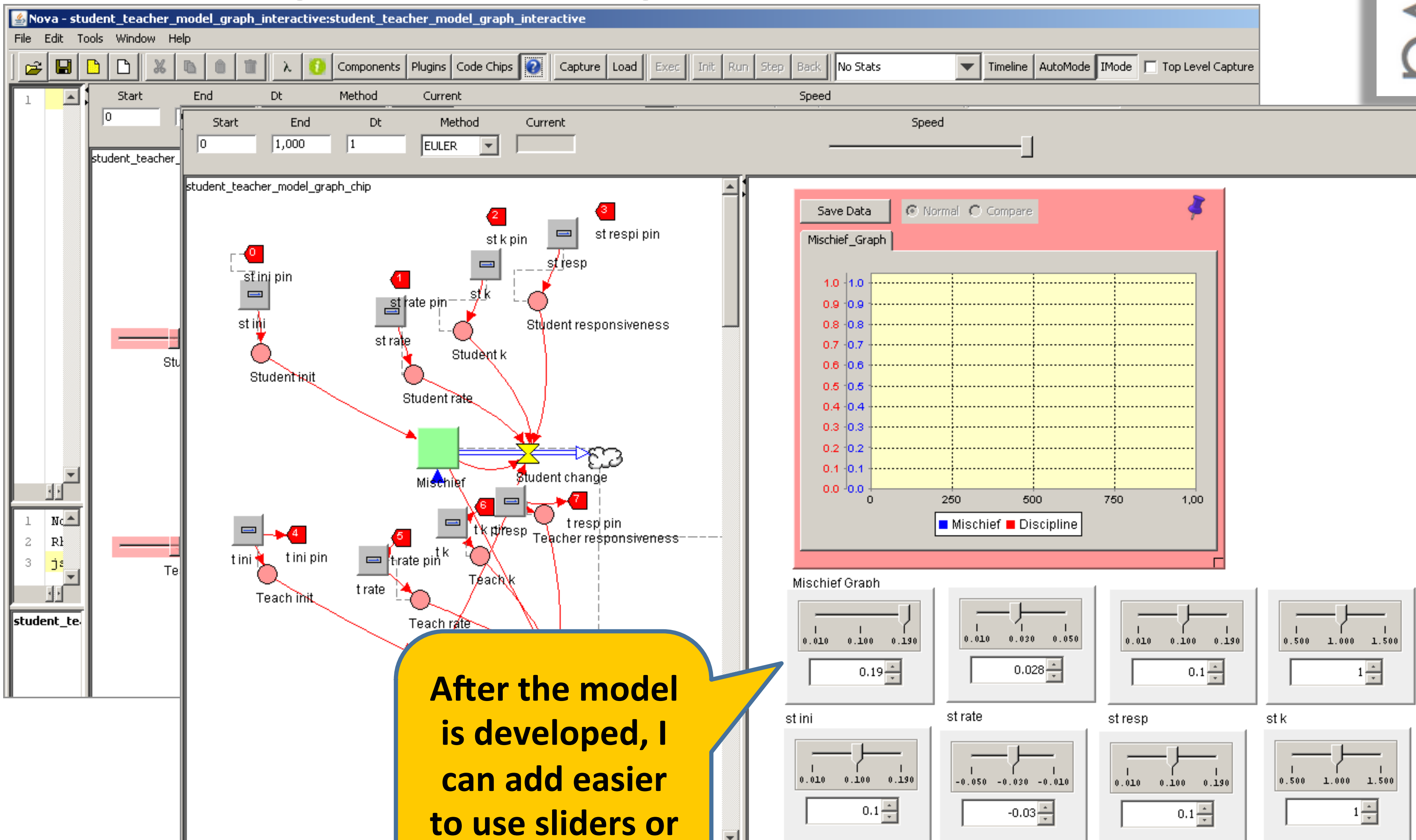
Mischievous Students in a Classroom



Straightforward stock and flow model with output

Oberlin Modeling Initiative

Working and Configured Controllers

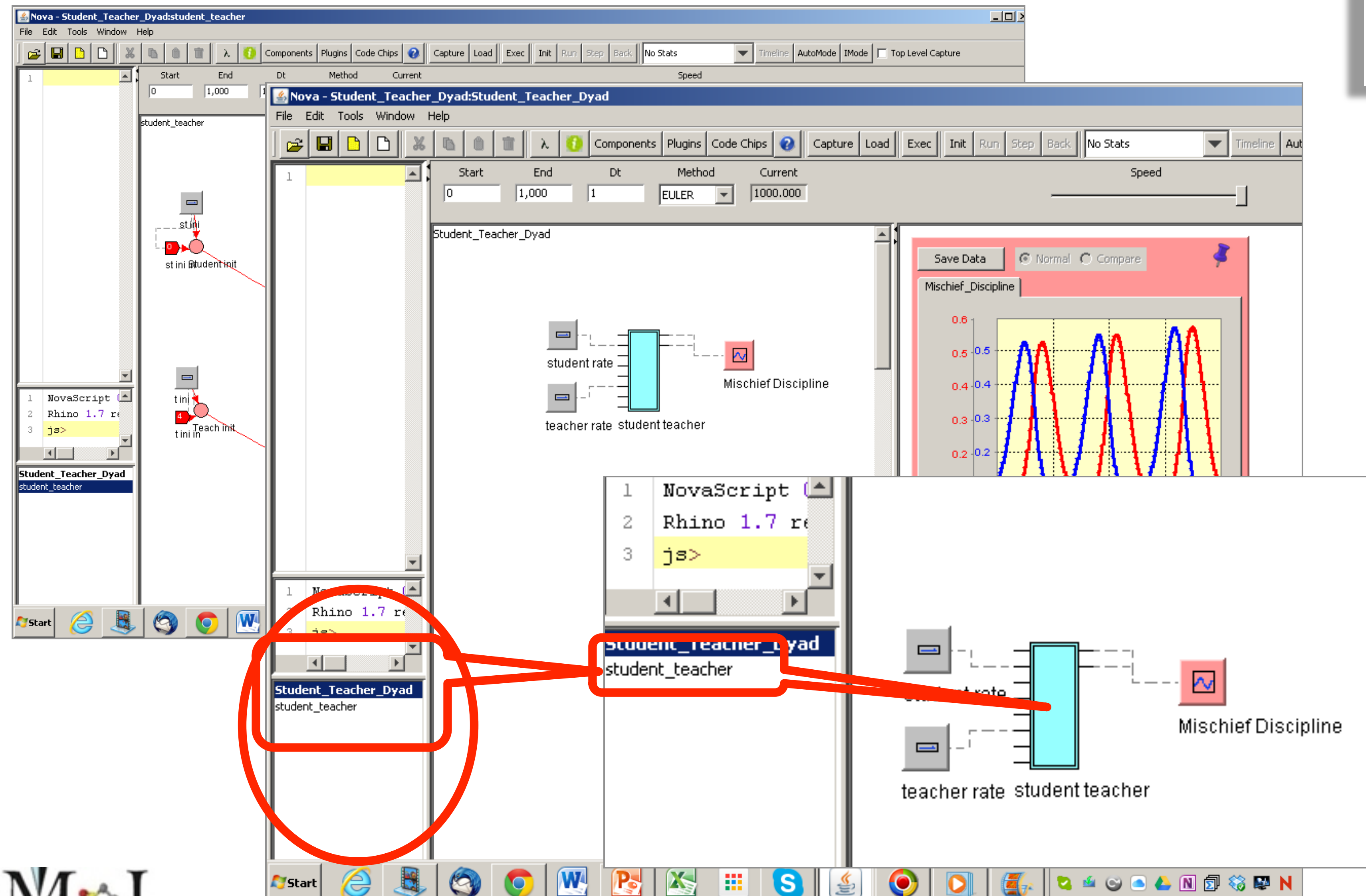


After the model is developed, I can add easier to use sliders or spinners

the model

Oberlin Modeling Initiative

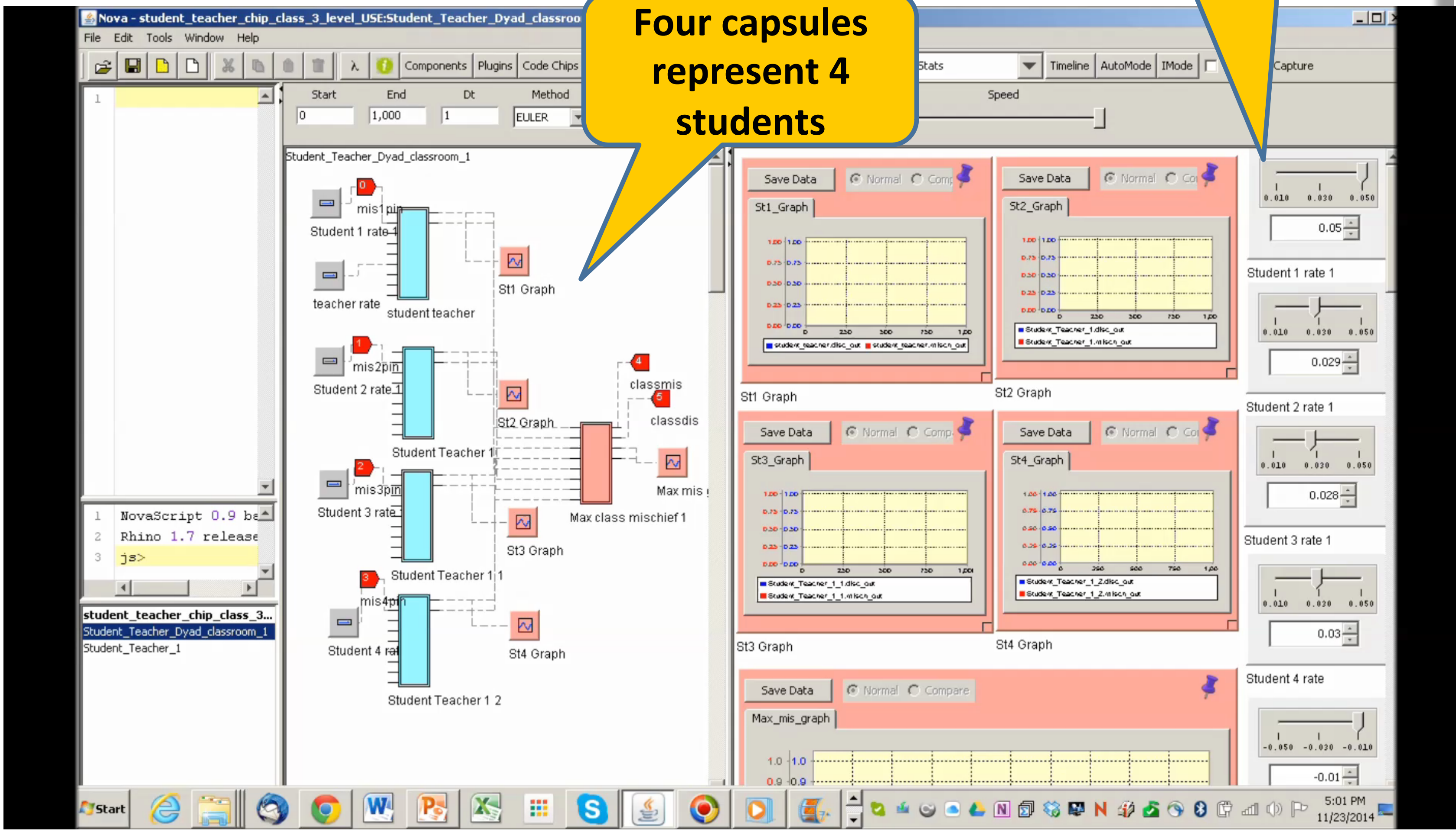
A simpler interface: capsules & chips



What capsules and chips are really useful for is aggregation

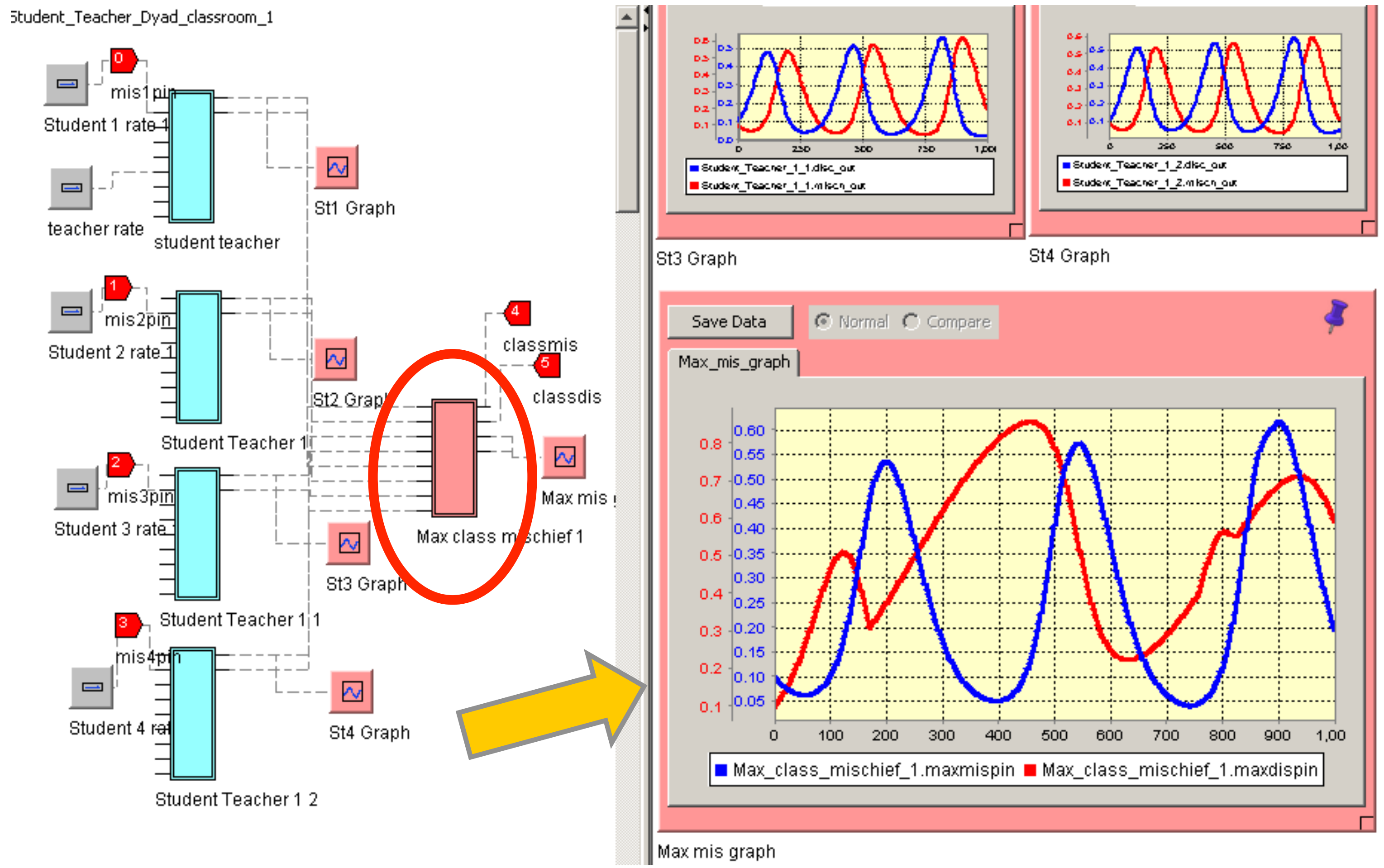
1 mischievous and 3 average students

Four capsules represent 4 students

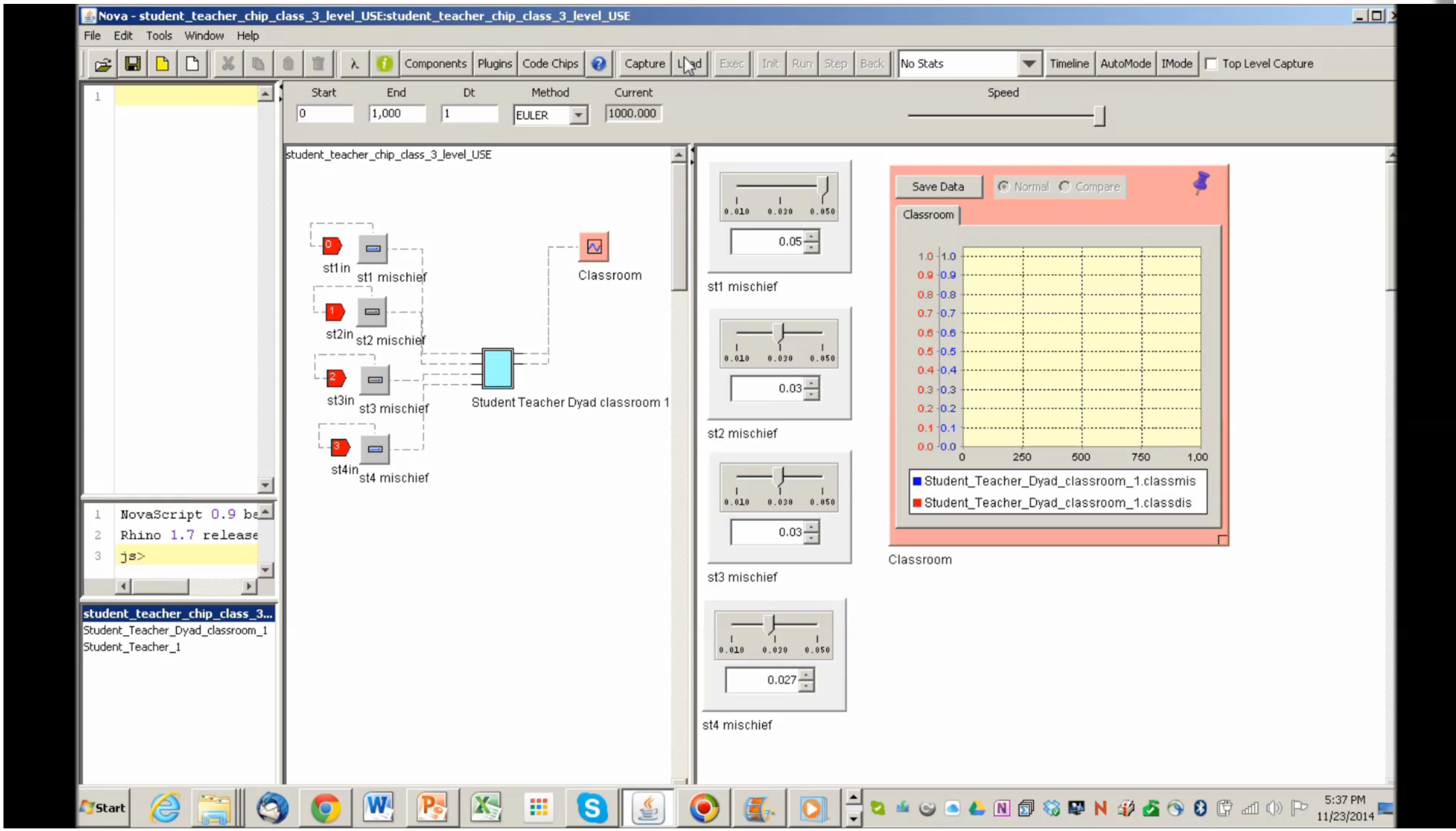


Oberlin Modeling Initiative

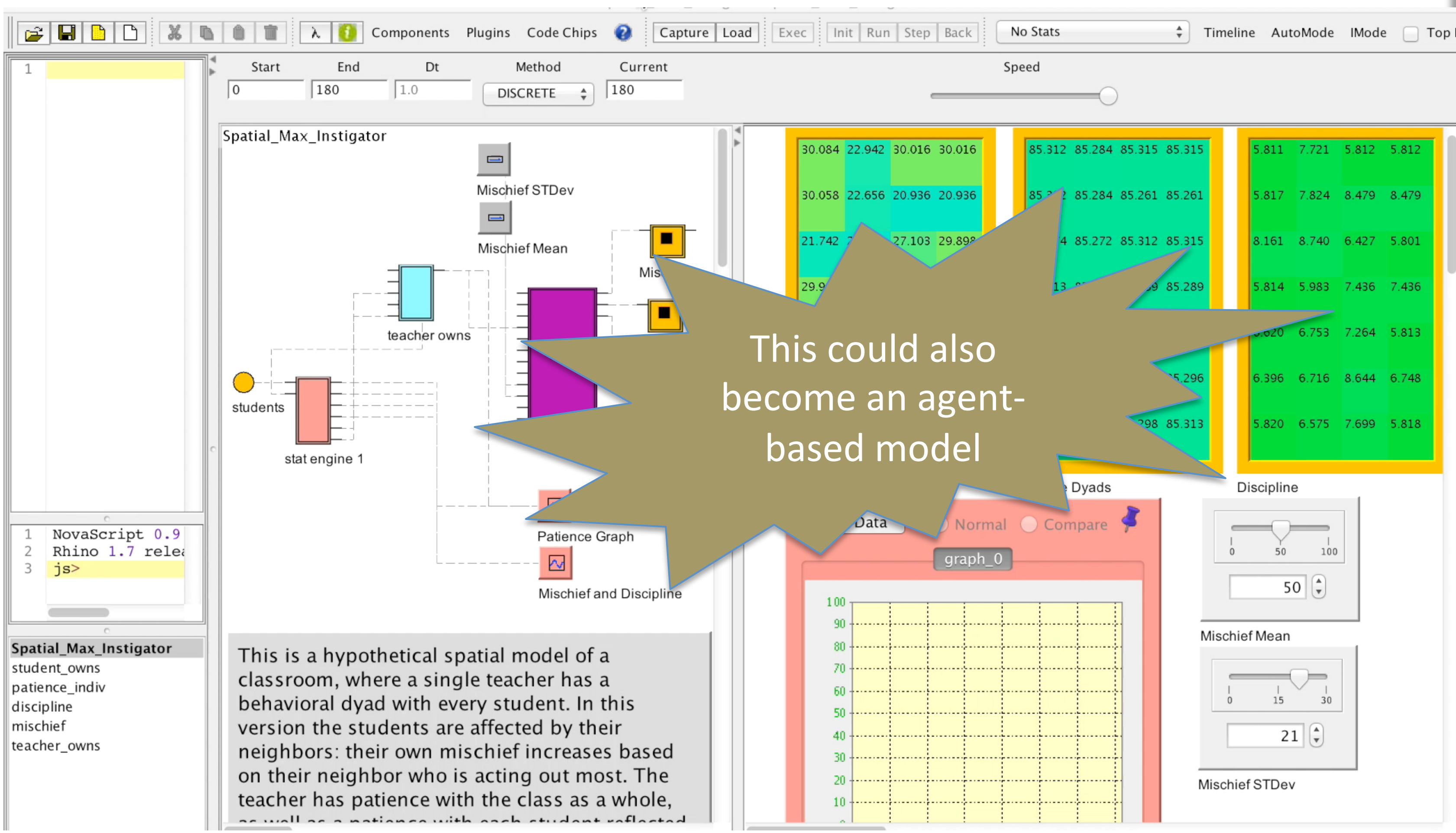
Another chip takes the output from individual dyads and aggregates them at the classroom level. Now I have a **nested model**.



And that classroom can be turned into a capsule so it's cleaner

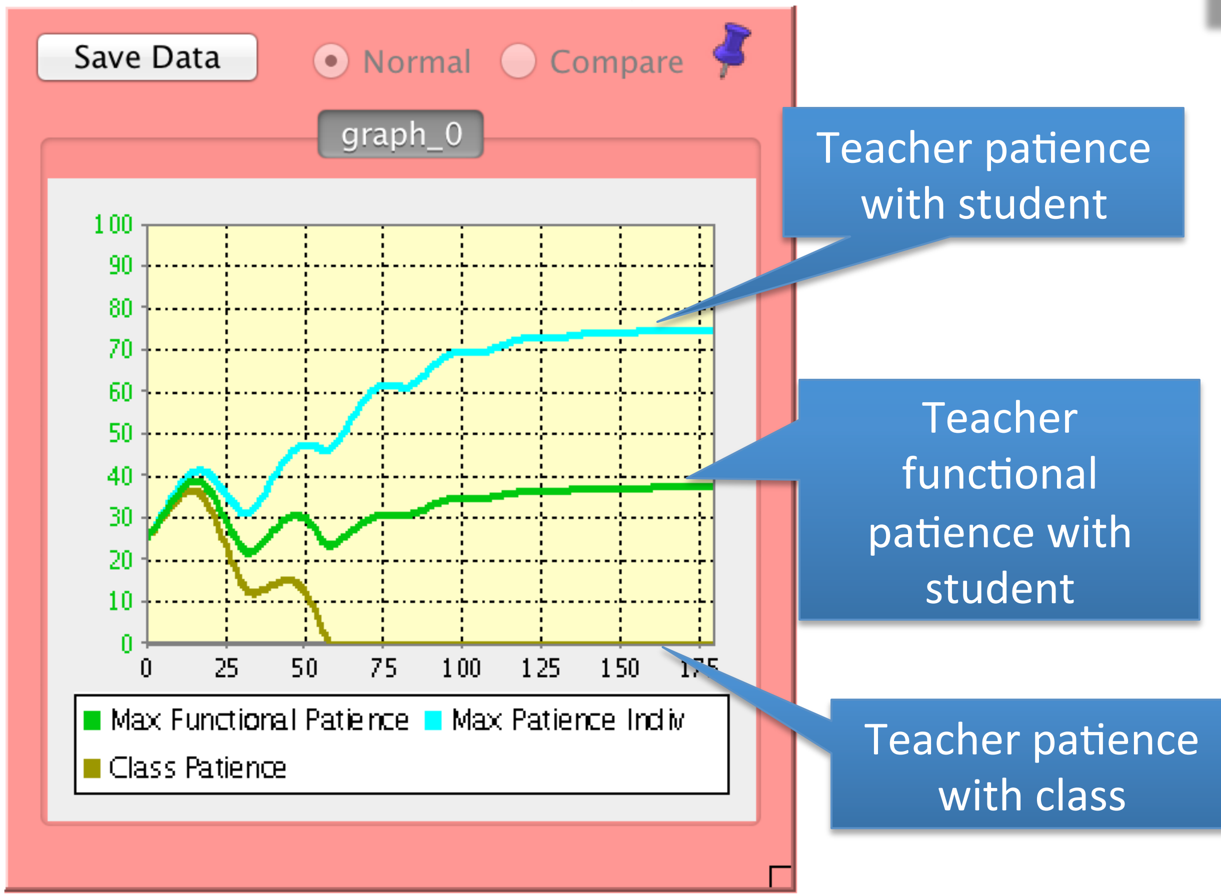


Chips can be configured spatially, so that each student influences the other and the teacher responds to individual and classroom characteristics



Oberlin Modeling Initiative

You can model individual dyads within the classroom



Patience Graph

Why is that cool? More complex models

- You can take the aggregated classroom mischief and create a stock called ‘stress’ that decreases a stock called ‘patience’ that changes the teacher’s dyadic reactivity
- You can create contagion effects so each student’s behavior changes depending on classroom context
- You could create an agent based model with many more children who find others like themselves and create pockets of mischief through contagion

Oberlin Modeling Initiative

Running simulations and outputting data

- Nova has the capacity to automatically run through a range of possible values
- Data can be viewed as:
  - Graphs
  - Tables
- Data can be exported as csv or directly to R

Oberlin Modeling Initiative

Teams Capsules and Modularity: An Example

- Nova Online:
  - Multiple stakeholders working to optimize solutions
- Expertise:
  - How do students respond to teachers?
  - How do teachers respond to individual students?
  - How do teachers respond to classroom dynamics?
  - Peer influence on deviant behavior
  - Classroom dynamics

Teams Capsules and Modularity: An Example

- Working with modeling novices: Attachment
- Process:
  - Sketching ideas
  - 3 different model components, 3 different teams
  - Combining and refining
  - Moving from the individual to the couple

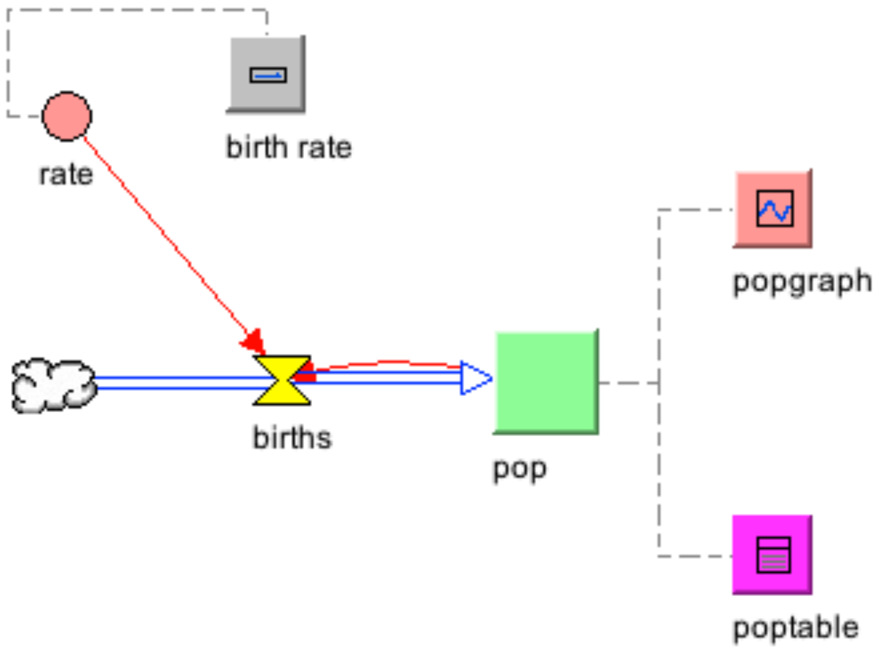
Summary

- Nova is a free, flexible program:
  - Stock and flow
  - Spatial
  - Agent based models
- Strengths:
  - One platform for multiple purposes minimizes learning time
  - Nested models
  - Heterogeneous populations with different distributions
  - Good entry-level tutorials
- Weaknesses:
  - Beta
  - Weak documentation of some features

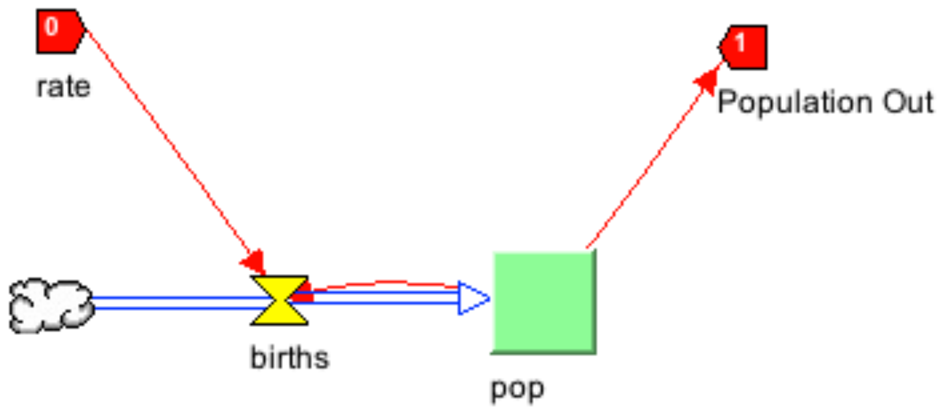
NUTS AND BOLTS

- A single framework for an eclectic set of systems.
- Expressive power derives from
  - modularity
  - abstraction
  - extensibility

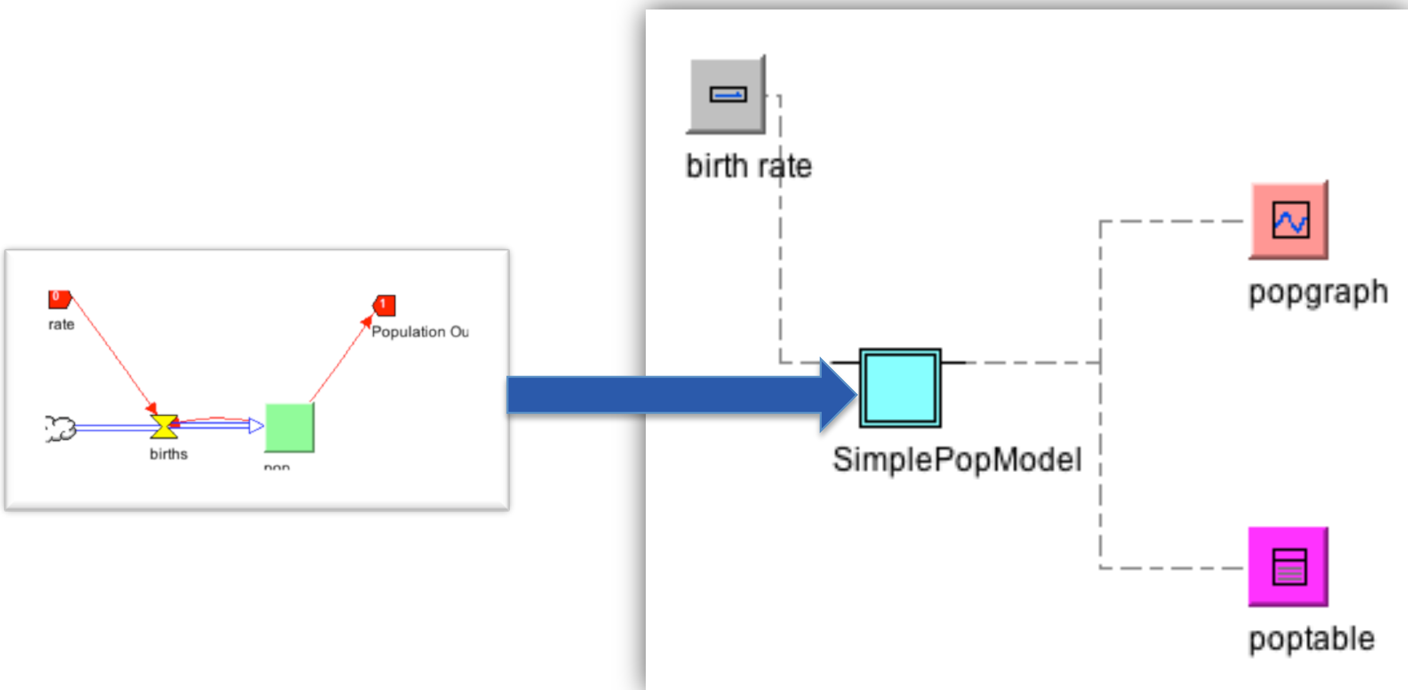
Capsule



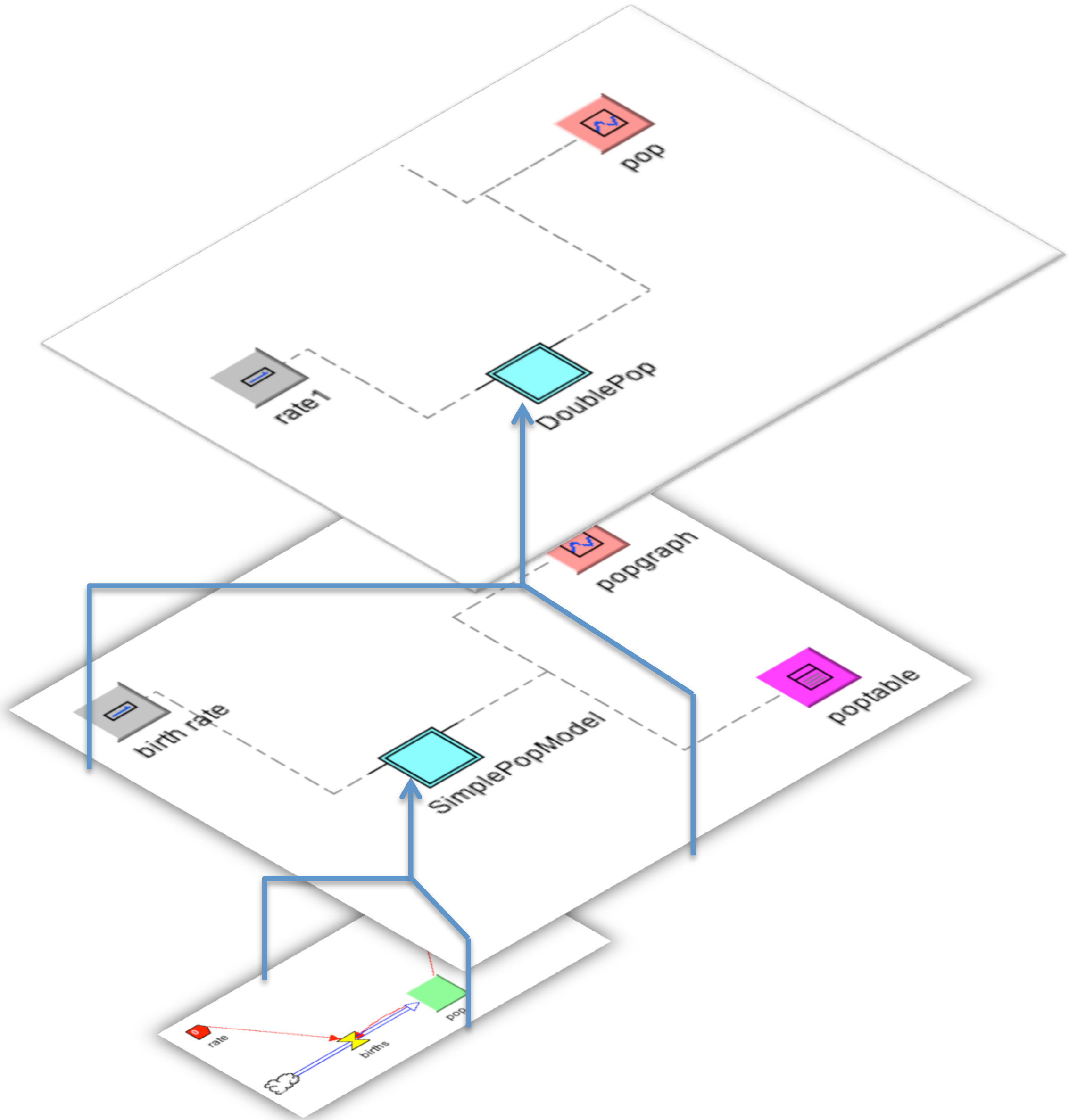
Capsule with Pins



Chip



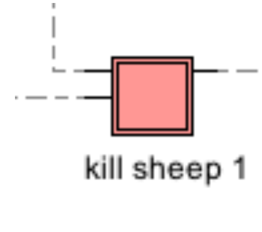
rate
Population Ou
births
birth rate
popgraph
SimplePopModel
poptable


pop
rate1
DoublePop
popgraph
birth rate
SimplePopModel
poptable
rate
births
pop

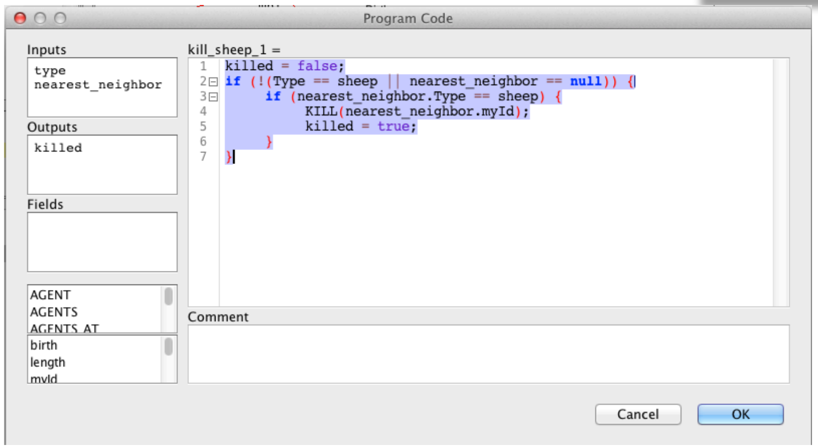
Containers (Aggregators)

- CellMatrix
  - 2 dimensional array of capsules
  - facilitates interaction among cells on a Cartesian grid
- NodeNetwork
  - An array of capsule *nodes* connected by a set of weighted links (equiv. to a mathematical graph)
  - facilitates transmission of data through the network.
- AgentVector
  - Agent = Capsule + location and trajectory parameters
  - AgentVector is 1-dimensional array of agents
  - AgentVector manages a set of agents in a common space
    - spatial position
    - births/deaths
- SimWorld
  - CellMatrix + AgentVector
  - Agent space corresponds to Cell topology
  - facilitates interaction between agent and cell environments
- NetWorld
  - NodeNetwork + AgentVector
  - Agent space correponds to Network topology
  - facilitates interaction between agent and node environments

Code Chips



- Contains code implementing a computational method
- Easy to implement multiple instances
- Easy to export/import into new model



Clocked Chip

- Attach a clock to chip so that each “tick” of the host model corresponds to a complete “run” of the encapsulated model.

Plugins

- API for creating new components
- Visualization
- Other useful extensions

Nova Online

- A visual Nova model is "captured" into a script (NovaScript) before it is executed on the Nova runtime engine.
- A Javascript implementation of this runtime has made possible a browser-based runtime using HTML5 graphics:
  – Nova Online
- Currently under construction: automatic creation of Nova Online Website.
- Also under construction: server-side NovaScript runtime for multi-core and high-performance execution.

Collaboration

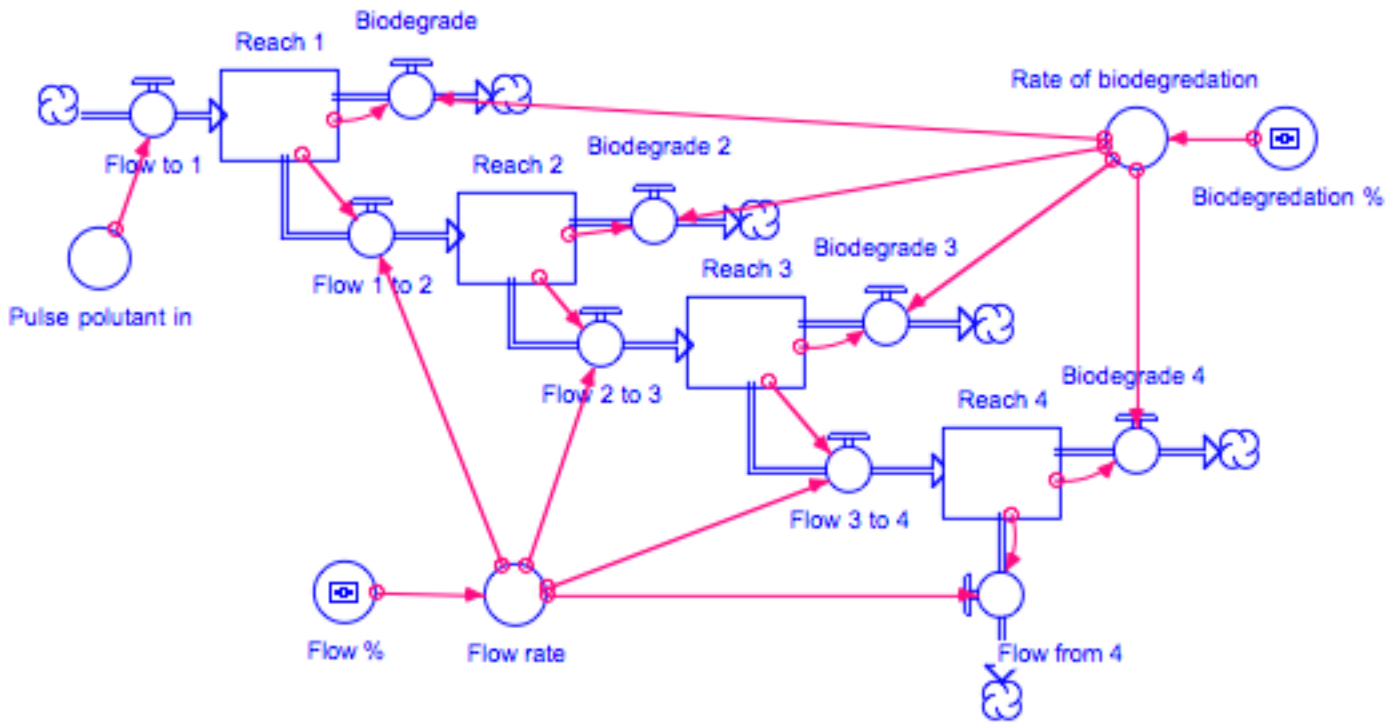
- Sharing of submodels.
- Sharing of codechips.
- Sharing of plug-ins.

- Interaction with R, GIS

- Combining submodels, codechips and plug-ins into a "kit" for a particular application area.

- Nova Website to serve as an archive and marketplace for shared components.

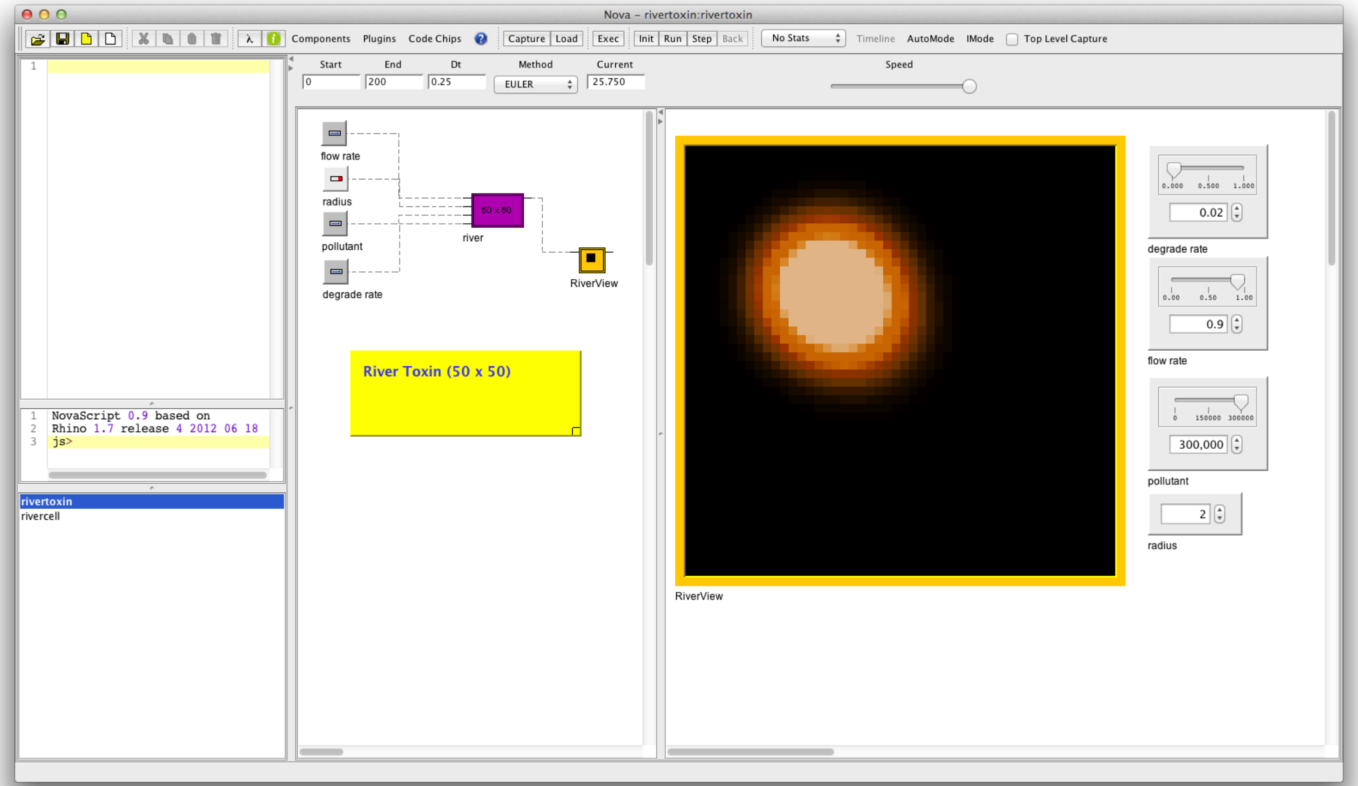
Example 1: River Toxin Advection



STELLA Version

River Toxin: Nova Version

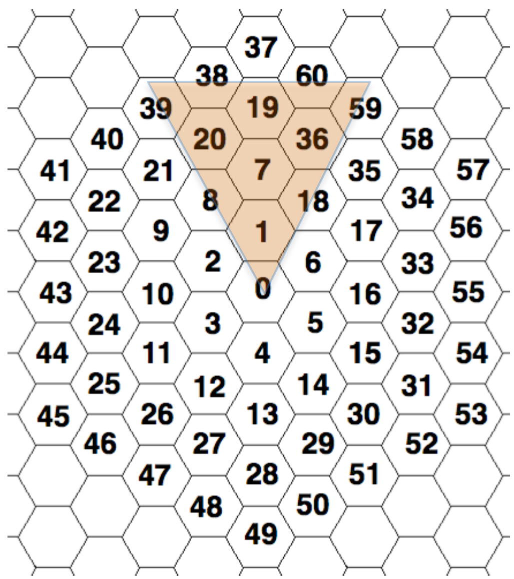
- Model spatially as a grid of cells

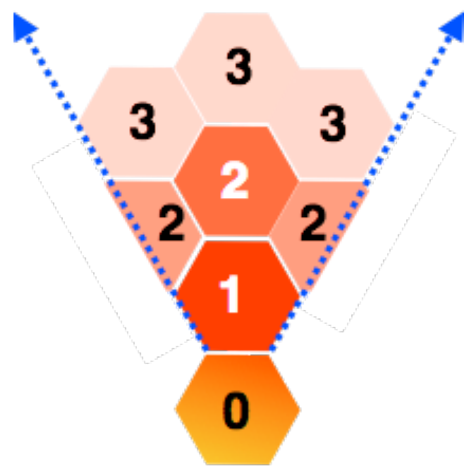


Oberlin Modeling Initiative

Example 2: Hexagonal Grazing (Getz)

- “World” is a hexagonal grid of cells.
- Agents are animals consuming food from cells.
- Cells contain food for consumption.
- At each time step agent must decide to either
  - Eat in current cell
  - Move to an adjacent cell
  - Decision governed by weight parameters: $q_1, q_2, q_3, \ldots$





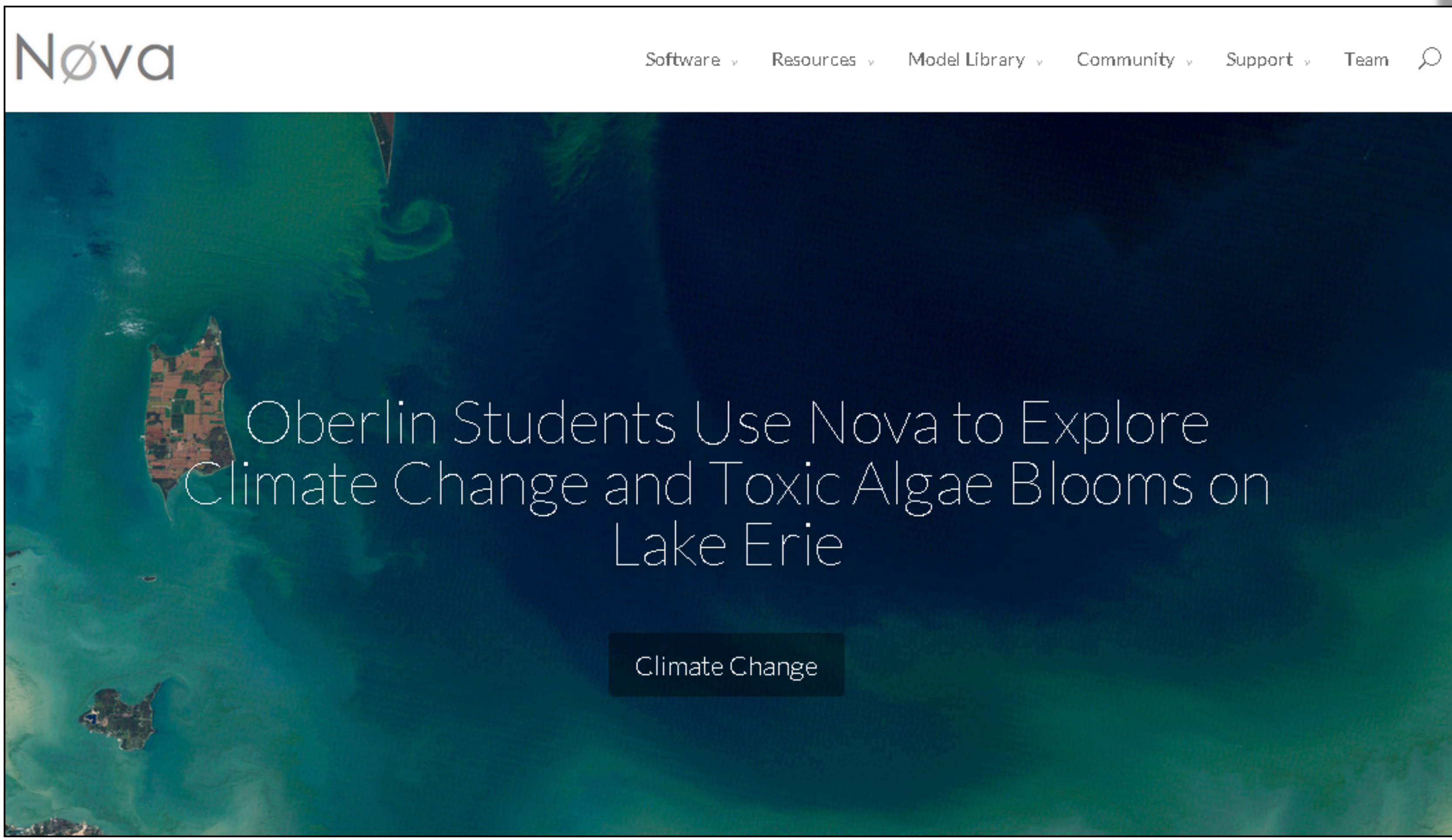
$$A_1 = q_1 a_1 + q_2\,((a_8 + a_{18})/2 + a_7) + q_3\,(a_{19} + a_{20} + a_{36})$$

Oberlin Modeling Initiative

Example 3: Florida invasive snail -- *Pomacea maculata*

- Model depicts a 25 square meter area with patches of size 10^{-2} sq m. Four snail "types" shown:
  - Males (blue)
  - Unfertilized Females (pink)
  - Fertilized Females (red)
  - Juveniles (yellow)
- Once fertilized, female lays an eggcase with up to 1000 eggs every 14 days (laying action depicted as enlarged purple agent token). Eggs hatch in 14 days with a 10% survival rate.
- Juveniles mature to adult status in 120 days (size of juvenile agent token grows with age).
- Separate juvenile/adult movement and consumption rates used.
- Attraction of males to unfertilized females is modeled.
- Carrying capacity is proportional to current biomass.
- Five year timespan modeled with seasonal variation of biomass growth.
- Snail aestivation occurs in December and January.
- Actual GIS-derived terrain is depicted.

www.novamodeler.com



Oberlin Modeling Initiative