



NetLogo and Multi-Agent Simulation (in Introductory Computer Science)

Matthew Dickerson

Middlebury College, Vermont

dickerso@middlebury.edu

Supported by the National Science Foundation

DUE-1044806

<http://ccl.northwestern.edu/netlogo/>

Motivation (in Brief)

Recruitment/Retention in CSCI

Computational Thinking Across STEM

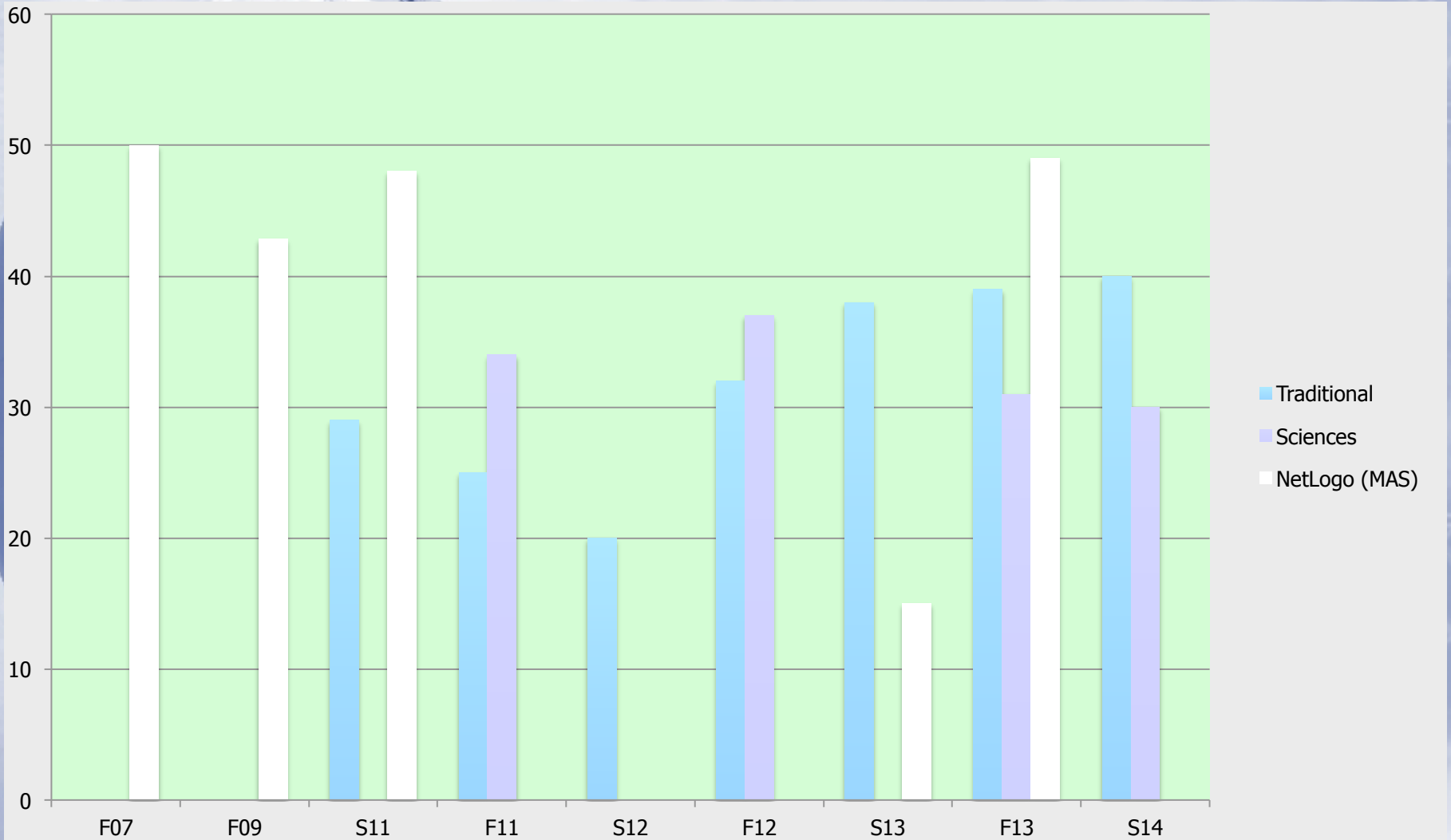
CS0/CS1

- ✧ Both course enrollments and number of majors in computer science at Middlebury College have more than tripled in the past 4 years
- ✧ Total enrollments 08/09-13/14: 164, 188, 221, 289, 453, 509
- ✧ Fall enrollments 08/09-14/15: 92, 93, 99, 126, 197, 230, 276* [315]

Nonetheless... Approaches emphasizing interdisciplinary activities and relevant applications to problem domains other than computer science are increasingly seen as important and effective, especially with respect to recruiting and retaining females. [LiBl96],[FMM97],[NSF07], and [Ast09].

Also... Teaching computational thinking to scientists.

% Females in CS0/CS1



Introduction and Overview

Rest of the Presentation

- ✧ **Agent-Based & Individual-Based Modeling**
- ✧ **NetLogo**
- ✧ **The Course**

Computer Modeling and Simulation

Computer Modeling is representing a phenomena (system) on a computer (or computers) through the use of (mathematical) models. A model is a *simplification or abstraction of a [real] system.*

Simulation is modeling over time. Simulation “involves the generation of an artificial history of a system, and the observation of that artificial history to draw inferences concerning the operation characteristics of the real system.” (Banks, et al.)

What is an agent?

definition (From Honovar): “An **agent** is any entity that can be viewed as *perceiving* its *environment* through *sensors* and *acting* upon its environment through *effectors*”

Explanation: *Three continuously performed functions/activities* [Hayes-Roth]:

- 1. “perception of dynamic conditions in the environment”*
- 2. “reasoning to interpret perceptions, solve problems, draw inferences, and determine actions.”*
- 3. Perform actions (in and on the environment and other agents)*

Agents are fundamentally algorithmic.

Introduction to Properties of Multi-agent Simulation?



- 1. Multiple agents, *acting independently***
- 2. Agents make decisions *based on a local environment***
- 3. Agents effect/change their (local) environment**
- 4. *Emergent behavior:*** The properties of the system as a whole are studied and understood as they ***emerge*** (via simulation) from the behaviors of the individuals.
 - a) Based on computational experiments rather than differential calculus
 - b) Field/laboratory studies still necessary.

Individual-Based Models (IBMs)

A step further than multi-agent simulation...
(From Grimm and Railsback:)

1. “Individuals grow and develop, changing in ways over their life cycle.”
2. “Individuals reproduce and die.”
3. “Individuals differ from each other, even within the same species and age.”



Grimm & Railsback (cont' d)

Getting Beyond *Agents* to *Individuals*

✧ *“In ecology, individuals are not atoms but living organisms. Individual organisms have properties an atom does not have. Individuals grow and develop, changing in many ways over their life cycle. Individuals reproduce and die, typically persisting for much less time than the systems to which they belong.”*

✧ *“Because individuals need resources, they modify their environment.”*

✧ *“Individuals differ from each other, even within the same species and age, so each interacts with its environment in unique ways.”*

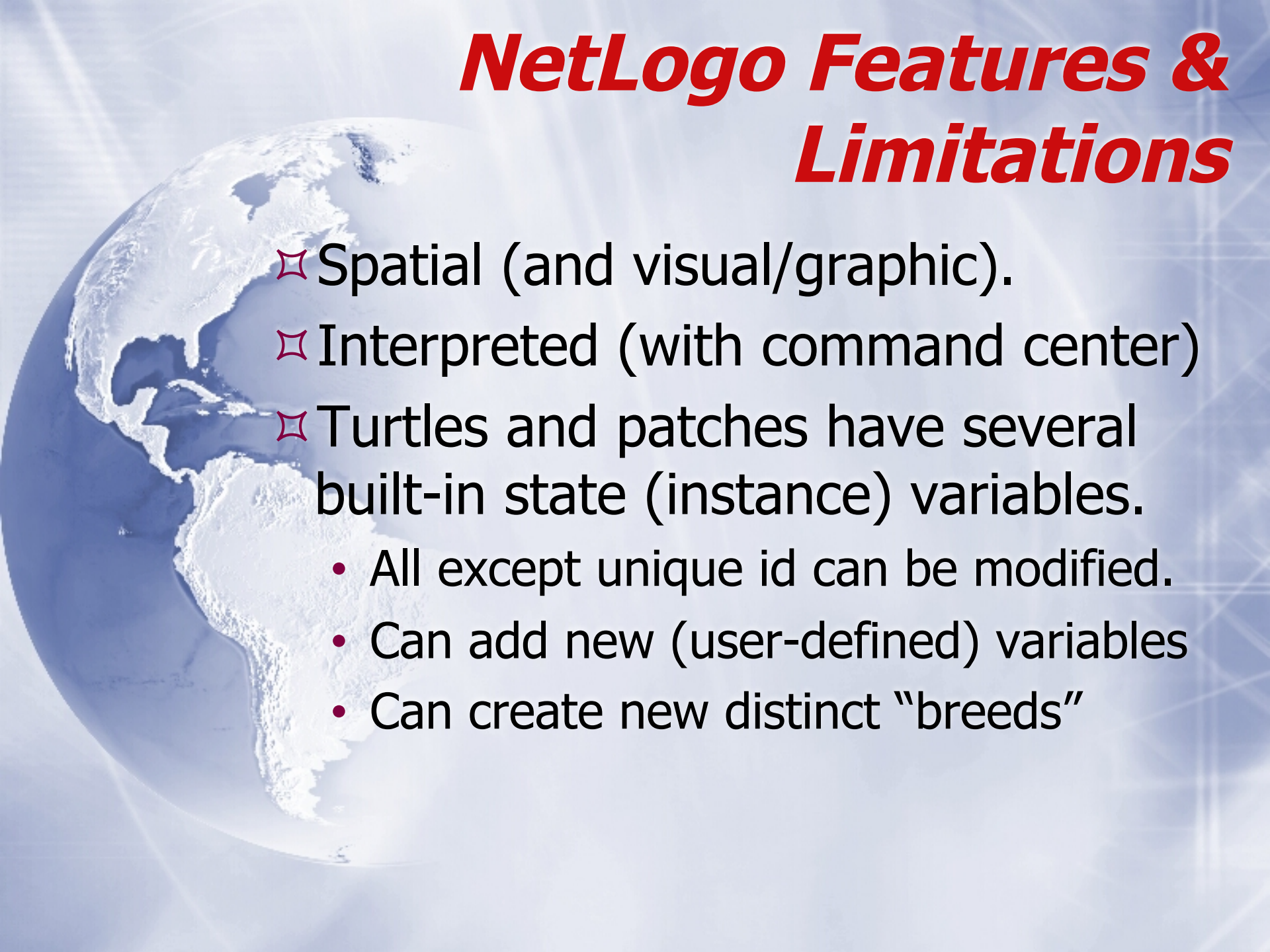
Multi-agent Simulation & NetLogo

NetLogo is a programming language and IDE (*Integrated Development Environment*) designed for multi-agent simulation.

NetLogo models represent agents as *turtles* and the environment as a collection of *patches*.

1. **Turtles act independently.** Typically, you give a command for all turtles to “act”, but each turtle may act differently based on its own condition, and also...
2. **Each turtle can find properties about local patches** and act based on those properties.
3. **Properties of the local environment** can be changed by turtles.
4. **The command for turtles to act may repeat indefinitely;** That is, we can look for **Emergent behavior** by running a simulation on the model over time.

NetLogo Features & Limitations

- 
- ✧ Spatial (and visual/graphic).
 - ✧ Interpreted (with command center)
 - ✧ Turtles and patches have several built-in state (instance) variables.
 - All except unique id can be modified.
 - Can add new (user-defined) variables
 - Can create new distinct “breeds”

NetLogo Features & Limitations

- Supports both global and local variables, with traditional scope rules for local variables.
- Functions (*reporters*) and Procedures are both pass-by-value.
 - *! A potential limitation as there is no explicit pointer type.*
 - Agents are objects; the value passed is a reference.

NetLogo Features & Limitations

✧ Variables are not typed

- Boolean, 1 numeric type, strings.
- turtles (other breeds), patches. (**nobody** is a built-in constant.)
- turtle sets, patch sets
- lists
- links (connections between turtles)

✧ Iteration **while** **repeat n** **ask**

✧ Recursion *(Very easy sierpinski curves!)*



NetLogo Features & Limitations

✧ Interface

- configurable size and wrapping
- sliders, and switches automatically create global variables
- buttons, monitors and graphs

✧ Built-in Experiment Space

✧ Implemented in Java, extendable via Java.



NetLogo Features & Limitations

- Two levels of concurrency: **ask**
 - default is ***without*** interruption
- Can construct data structures using lists or turtles
- Easy to import graphics; supports sound

NetLogo Features and OOP

- ✧ Turtles and patches function like objects.
 - ✧ You can construct your own **breeds** which **extend** Turtles one level.
 - ✧ Patches cannot be extended (but you can add instance variables)
- ✧ Turtles, patches, and links are referenced by pointer. (Call by value passes the pointer.)
 - ✧ Lists, strings, agent sets are not.
- ✧ ***Not encapsulation***—though procedures may be specific to certain breeds, they reside in the same file.



Examples of MAS

(and a taste of NetLogo)

- ✧ **Biology: Fireflies** (Can be developed in one lab, by the 10th week of a 1st semester course)
- ✧ **Physics: Rope** (simple interface.)
- ✧ **Mathematics: Voronoi** (many variations... such as changing the distance function. Without live update, it is an easy assignment in first few weeks.)
- ✧ **Social Science: Ethnocentrism** (in honor or diversity day—read the Information section!)
- ✧ **Graphics and Recursion:** (Seeing the trees and maybe the forest also.)

A First-Year Course based on MAS and NetLogo

- ✧ No prerequisites. No assumptions.
...except high school algebra
- ✧ A “funnel” not a “sieve”.
- ✧ CS0/CS1
 - **CS0**: Environmental Studies, Biology, Economics.
 - **CS1**: At Middlebury, can lead to CS2 in Java
- ✧ Attracts...
 - + 50% female
 - Only 30% first year students (Many students already declared in economics, biology, physics, and environmental studies.)



A First-Year Course based on MAS and NetLogo

Covers Three Topics in Parallel (with adjustable balance):

1. NetLogo programming

- assignment, selection, iteration, recursion
- variables and types, scope
- procedures, functions, parameters

2. Concepts of multi-agent simulation

3. Breadth topics in computational/algorithmic thinking and computer science

- boolean logic
- data storage, analog/digital and discretization
- complexity/efficiency
- history (abstraction of data, instructions, hardware, software)
- computer organization

Could Go On... and On...

But...

- ✧ In fact, if anybody is interested, I'll send you my syllabus...
- ✧ And if you actually teach the class, I'll send some sample assignments and solutions and materials...
- ✧ More useful to conclude with a look at what students do (in one intro semester)!
 - ✧ HW4 – erosion; HW7 – Tree competition
 - ✧ Labs: Asteroids, ponds, etc.
 - ✧ Term projects