CTSiM: A Computational Thinking Environment for Learning Science using Simulation and Modeling

Gautam Biswas
Dept. of EECS/ISIS
Vanderbilt University, Nashville, TN. USA.
gautam.biswas@vanderbilt.edu
http://www.vuse.vanderbilt.edu/~biswas (www.teachableagents.org)

Acknowledge Collaborators
Ningyu Zhang, Satabdi Basu, John Kinnebrew, Brian Sulcer, Naveed Mohammed
Doug Clark, Pratim Sengupta, Ashlyn Karan
Acknowledge Funding
NSF Cyberlearning Grants: # 1124175 and #1441542

Integrating CT with the K-12 curricula

- Growing consensus that all children need to be offered experiences with CT in their K-12 years
- In order to reach every student
  - Computing education must be introduced as part of a curriculum
  - Integrated with existing curricula
- Our approach (Sengupta, et al., 2013)
  - Integrate CT with existing middle school science curricula
  - Goal: Synergistic learning of science and CT concepts

(NRC 2010; Basu, et al., 2017; Navlakha, & Bar-Joseph, 2011; Ioannidou, et al., 2010; Weintrop et al, 2016; Wilensky, Brady & Horn, 2014)

What is Computational Thinking?

- General, analytic approach to:
  - Problem solving
  - System design
  - Understanding human behavior
- Concepts fundamental to computing & computer science
  - Algorithm design & structure
  - Decomposition & Composition
  - Modularity
- Practices central to STEM modeling, reasoning, and problem solving
  - Problem representation
  - Abstraction and decomposition
  - Simulation and prediction
  - Verification

Barr & Stephenson (2011)
Guzdial (2008)
Wing (2006, 2008, 2010)

Distinguishing characteristics of our research

- Emphasis on integrating CT with existing middle (& high) school science curricula
  - Simple enough for use by science teachers with no programming experience
- Understanding challenges typically faced by students working in such CT-based environments
  - Focus on synergistic integrated learning
- Developing and evaluating an adaptive scaffolding framework based on an assessment of students' modeling strategies and performances
  - Students solve complex, open-ended problems; provide scaffolding that helps them learn and succeed
- Use of multiple modes of assessment for studying students' science and CT learning and characterizing students' learning processes
  - Analyze students' learning performances and behaviors

Outline of Talk

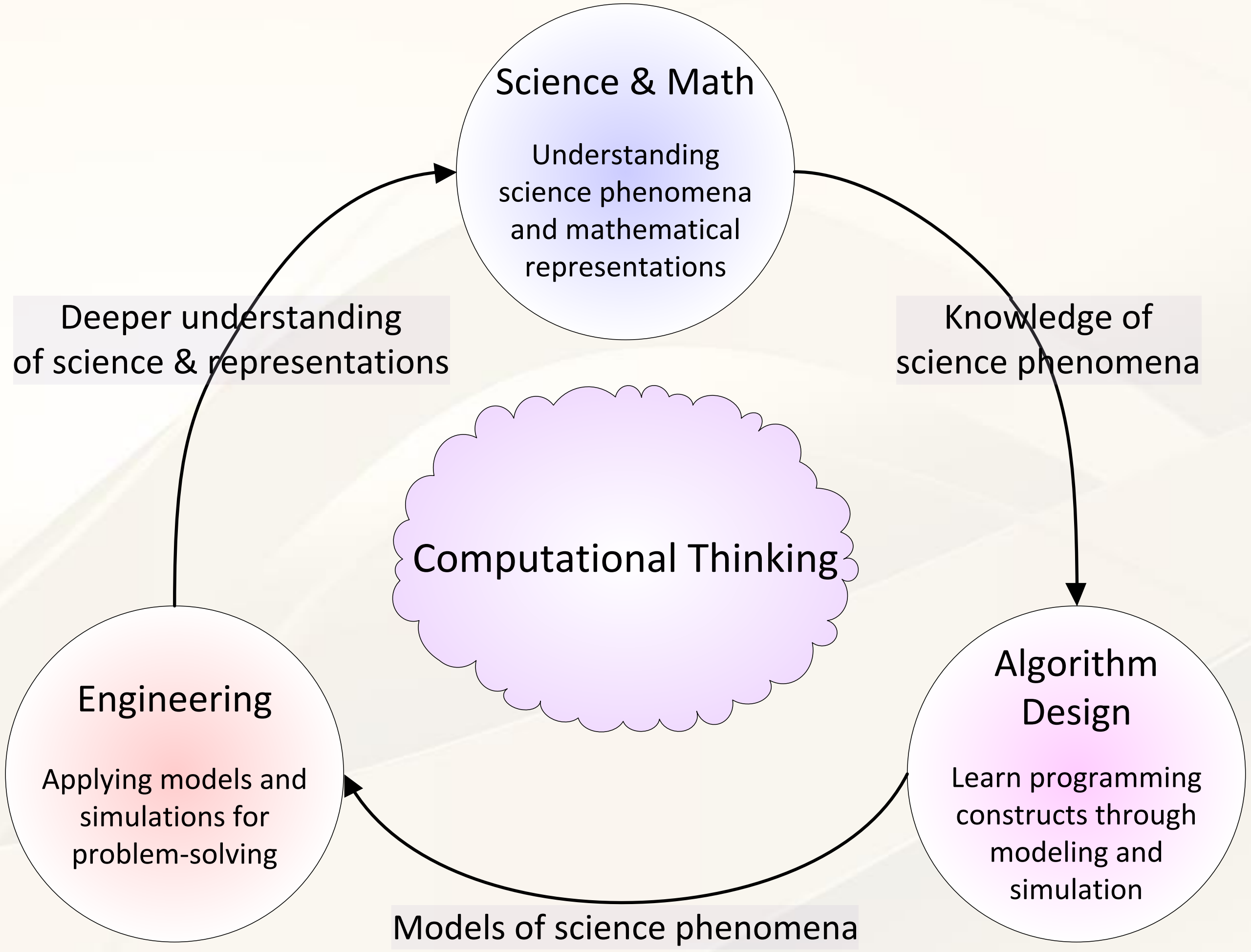
- Designing Open Ended Learning Environments that focus on synergistic science and CT learning
- The CTSiM system
- Early Studies
  - Understanding Students' Difficulties
  - Provide better scaffolding and adaptive feedback
- Recent studies
  - Focus on synergistic learning and students' learning behaviors
  - Effectiveness of adaptive strategy support
- Discussion and Conclusions
  - On going work in developing units for middle and high school curricula

OELEs Developed by our group

- Open-ended Learning Environments (OELEs)
  - Students are provided with specific goals
    - Build model of an airdrop from a moving aircraft – Learning by Modeling
    - Solve a problem (How long will the fish survive in my fish tank?)
  - Set of tools to scaffold their information acquisition, solution construction, and solution assessment tasks
    - Resources
    - Model Building Representations & Interfaces
    - Verification Tools
  - But they are free to go about developing their solutions as they like
- Example systems: Betty's Brain; CTSiM, C^3STEM, C^2STEM

CTSiM: Design Principles
(Sengupta, et al., 2013)

- **Low threshold**: easy to learn
- **High ceiling**: advanced modeling & programming possible
- **Wide walls**: range of artifacts (*e.g.*, science phenomena, animations, & games)
- Scaffolding
  - Algorithm visualization
  - Debugging support
  - Feedback from virtual agents



CTSiM Pedagogy

- Learning by modeling: Students build simulation models of complex science topics
- Agent-based, visual programming approach using a domain specific modeling language (DSML)
  - Agent based modeling leverages intuitions about individual agents to help understand emergent system behaviors
  - A DSML helps contextualize programming constructs in domain concepts and emphasize the generality of CT constructs across domains
- Tools provided to acquire information relevant for model construction
- Tools provided to test and verify models as agent-based simulations
- Tools for problem solving

Modeling Support

Build models at different levels of abstraction

- Modeling using two separate but linked representations
- Conceptual modeling
  - Organize the domain in terms of its agents, environment elements, their properties and behaviors
  - Describe agent behaviors as sense-act processes
- Computational modeling
  - Drag and arrange blocks from a provided computational palette to describe agent behaviors
  - Availability of blocks in the palette for an agent behavior dependent on conceptualization of sense-act processes for the behavior
- An example of recent interface changes based on previous observations
  - Students previously used a lot of trial and error while selecting and arranging blocks
  - Students had problems identifying entities and their interactions

Learning by Modeling

Build Model

Conceptual Modeling

Construct Domain models as
- Agents
- Agent Properties
- Agent Behaviors

Environment Elements

Computational Modeling

Construct Agent Behaviors
- Visual, Block-structured language
- Sense & Act Behaviors

Test your Model

Execute Model as Agent-Based Simulation
- Can set parameters
- Can single step through computational model

Compare your Model Behaviors

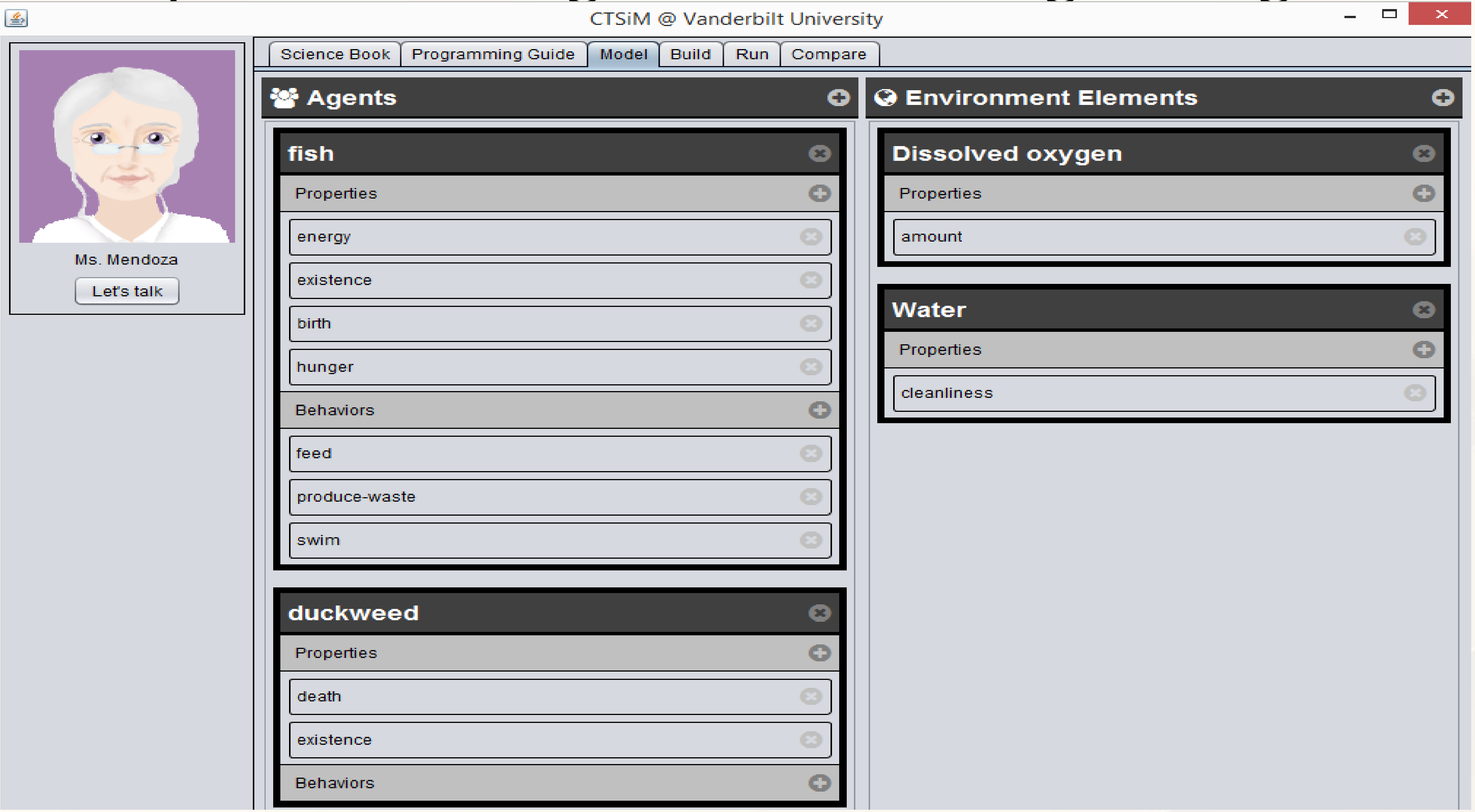
Against Expert Model
- Set parameters
- Test in parts

Hypertext Resources

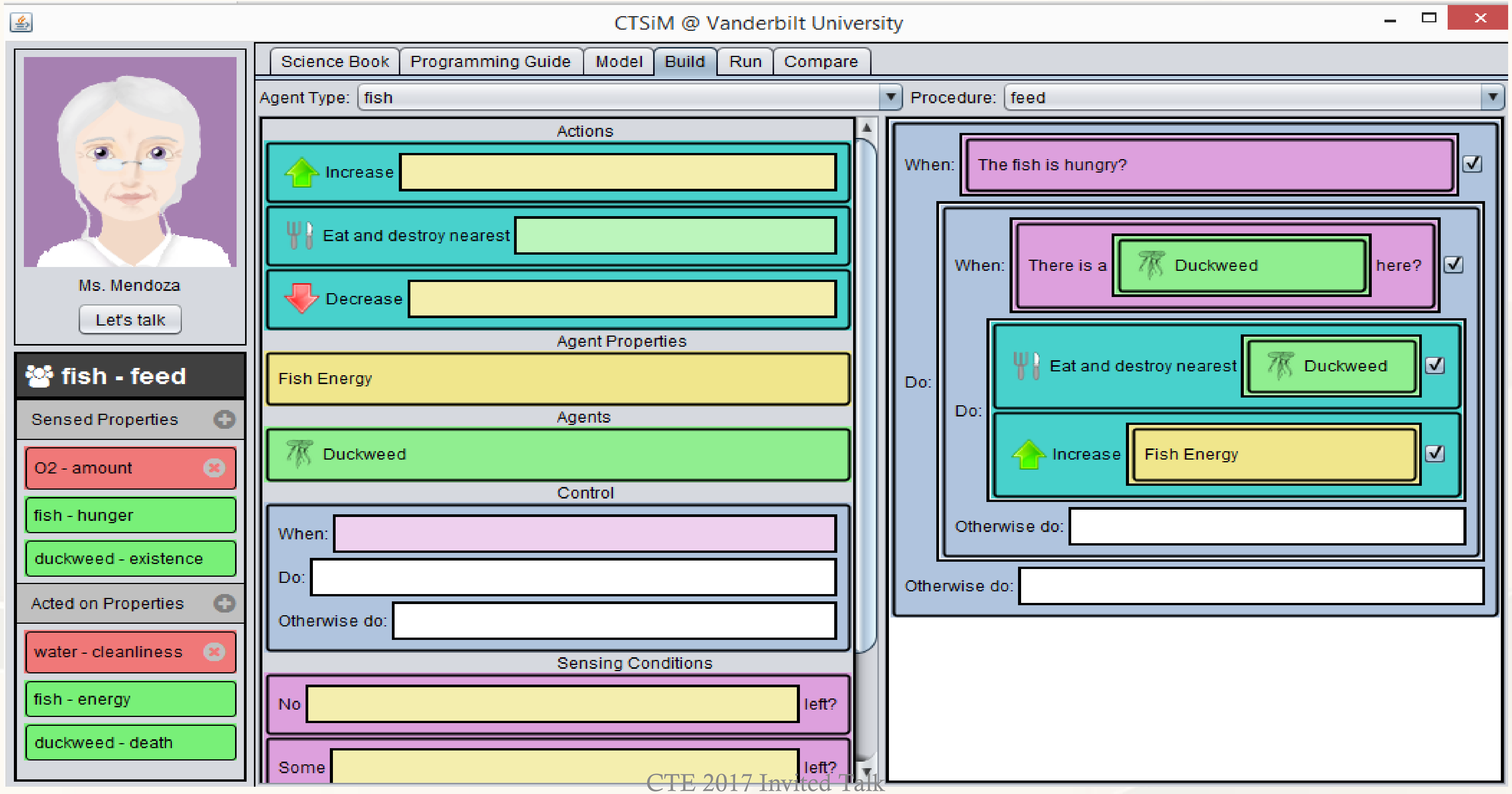
Formative Quizzes

Refine Model

The conceptual modeling interface for organizing the domain



The linked conceptual-computational representation for modeling agent behaviors



The ‘Run’ interface for observing model behaviors

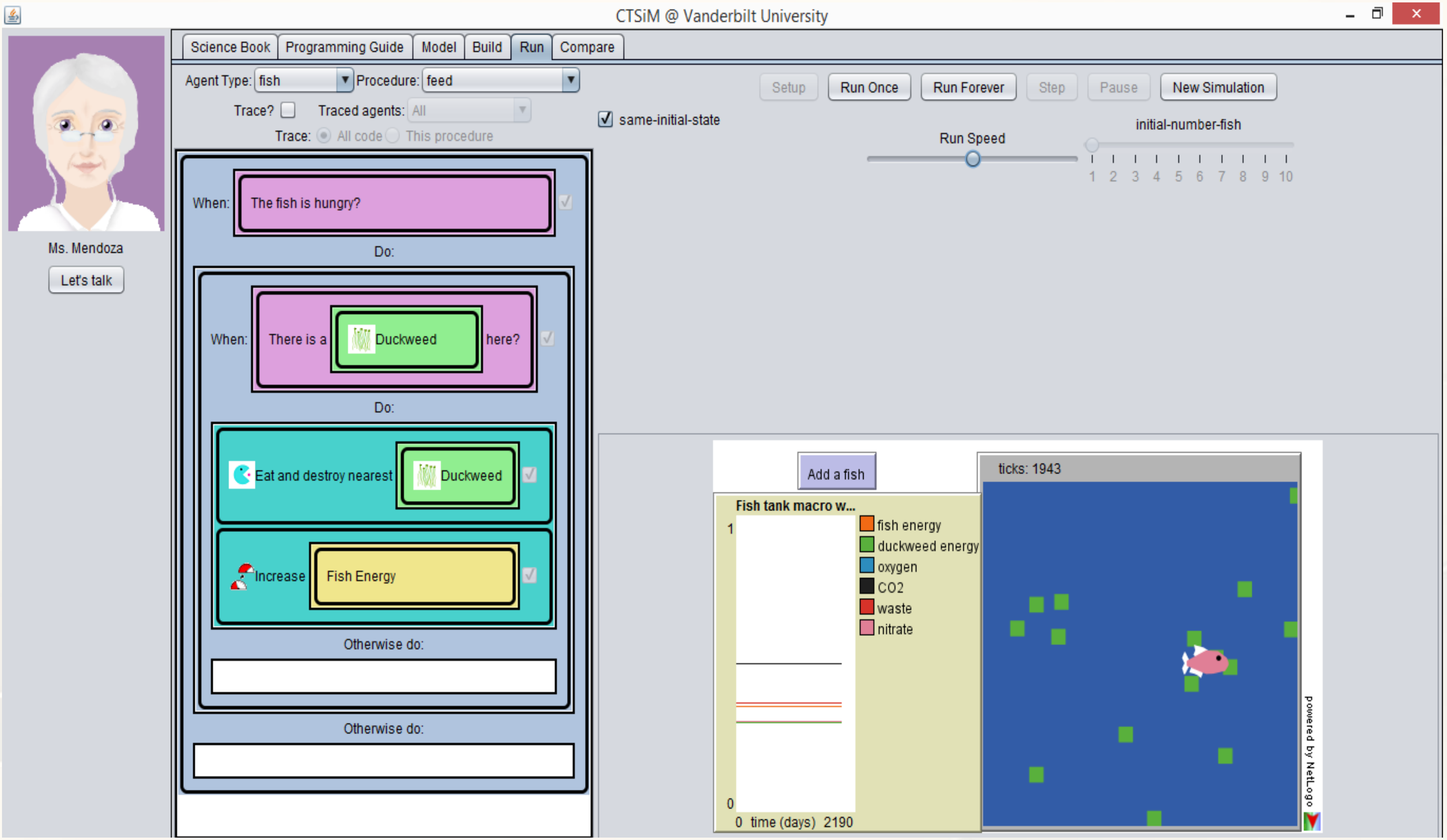
CTSiM DSML visual primitives

⇩

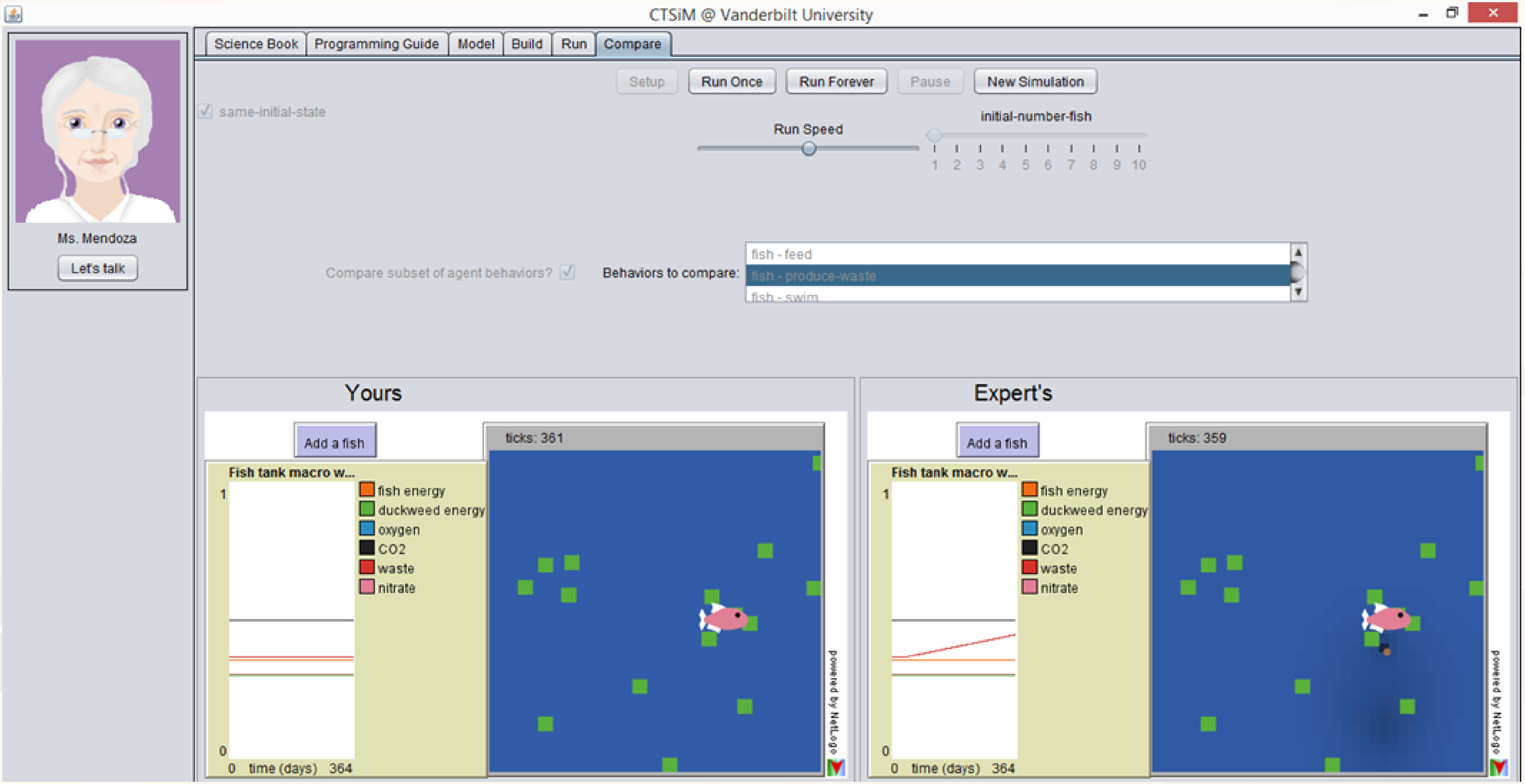
Intermediate language computational primitives

⇩

NetLogo Commands

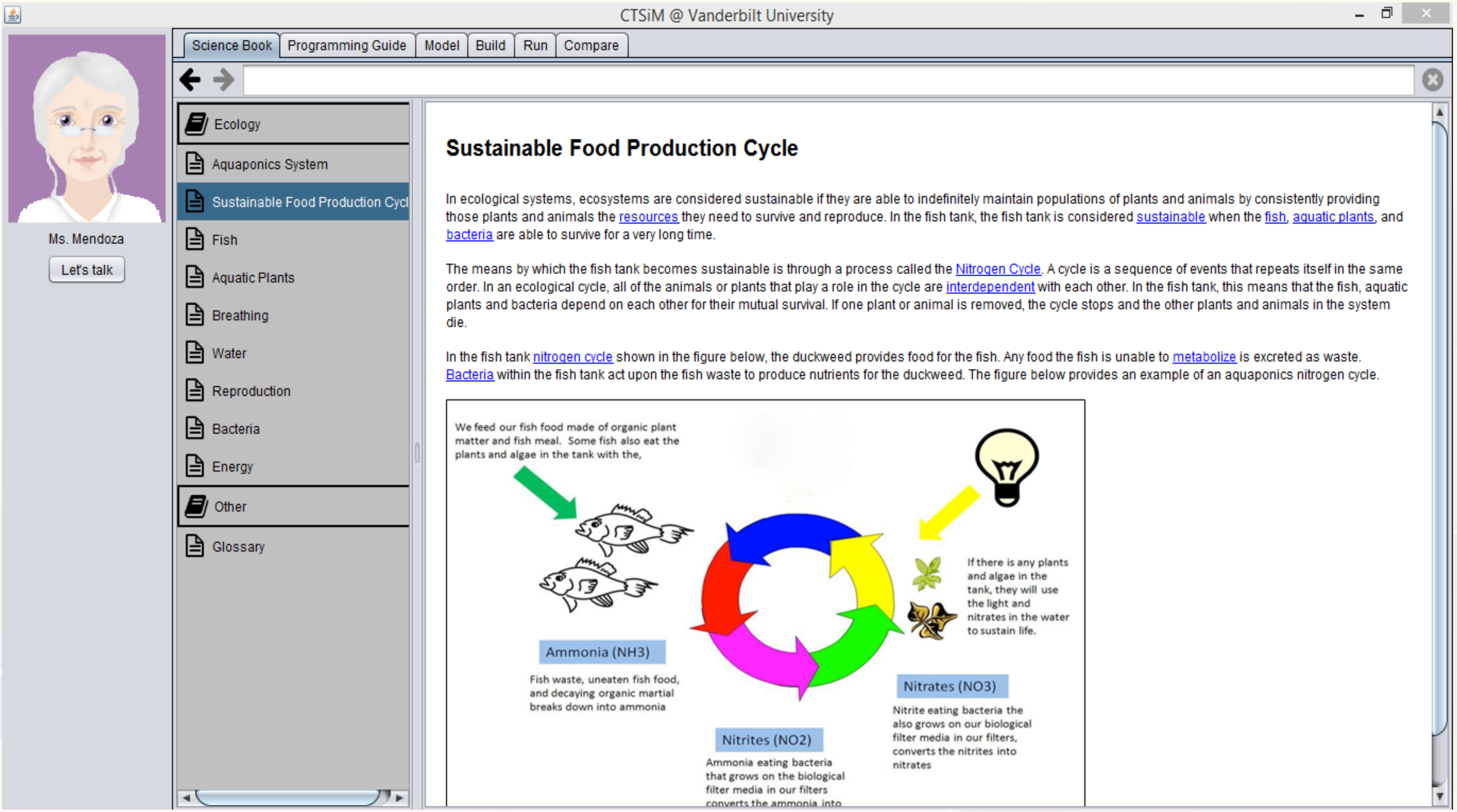


The model behavior 'Compare' interface



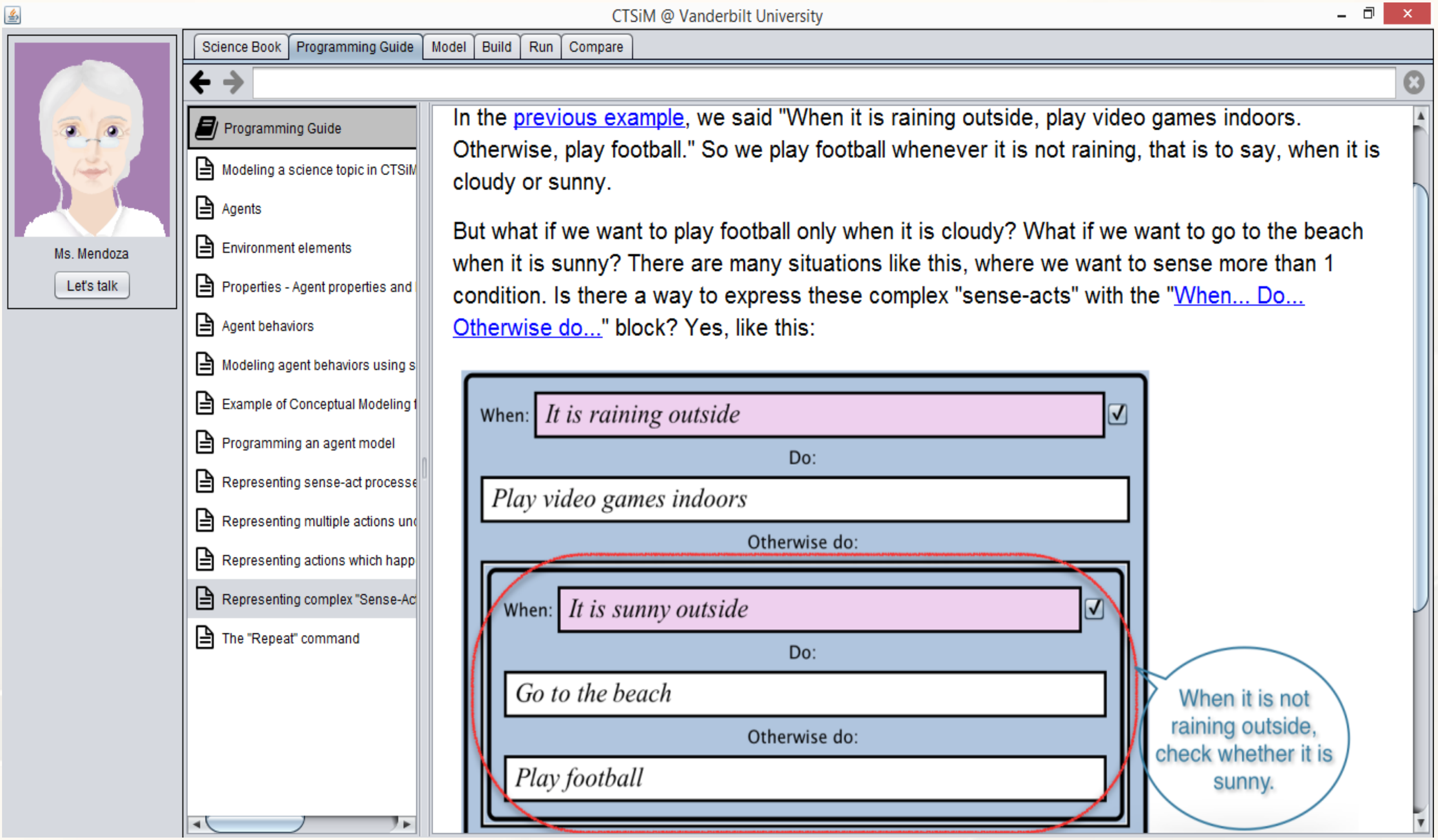
The domain resources or 'Science book' interface

Students are provided with resources containing relevant information about the science domain they are modeling

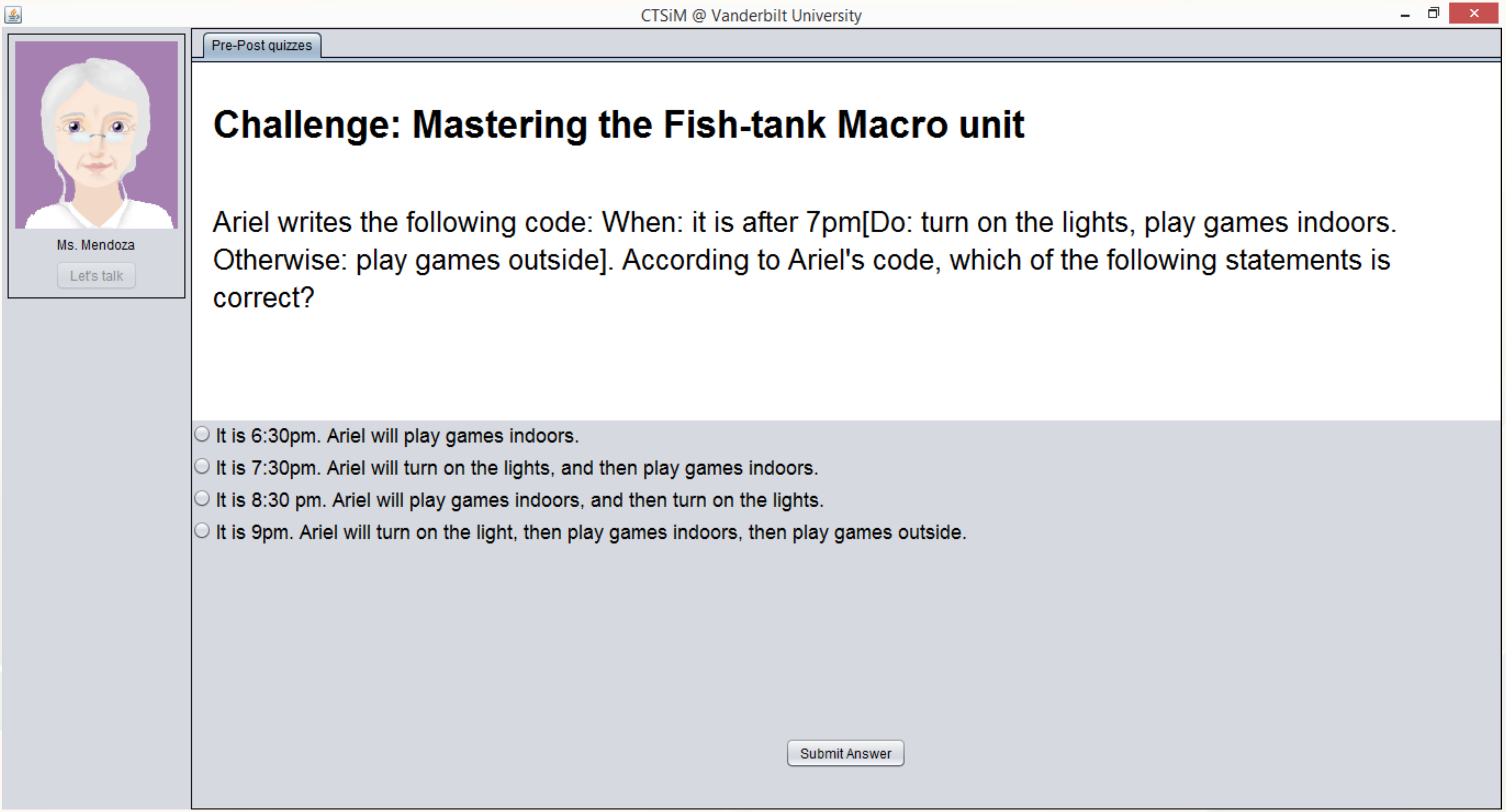


The CT resources or 'Programming guide' interface

Students are also provided with resources explaining and providing examples of agent-based modeling and computational concepts used in CTSiM

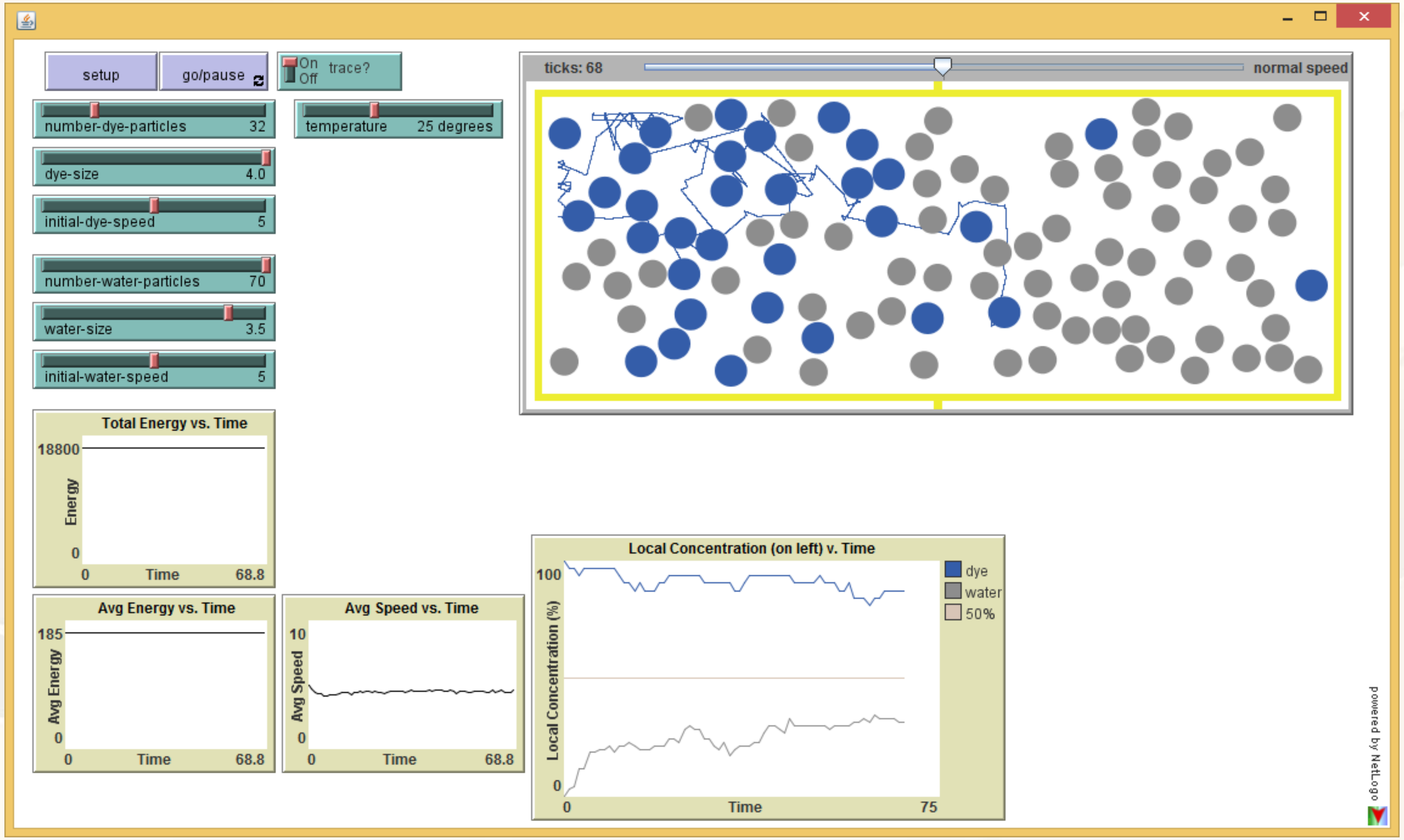


Example of formative assessments for checking science and CT understanding



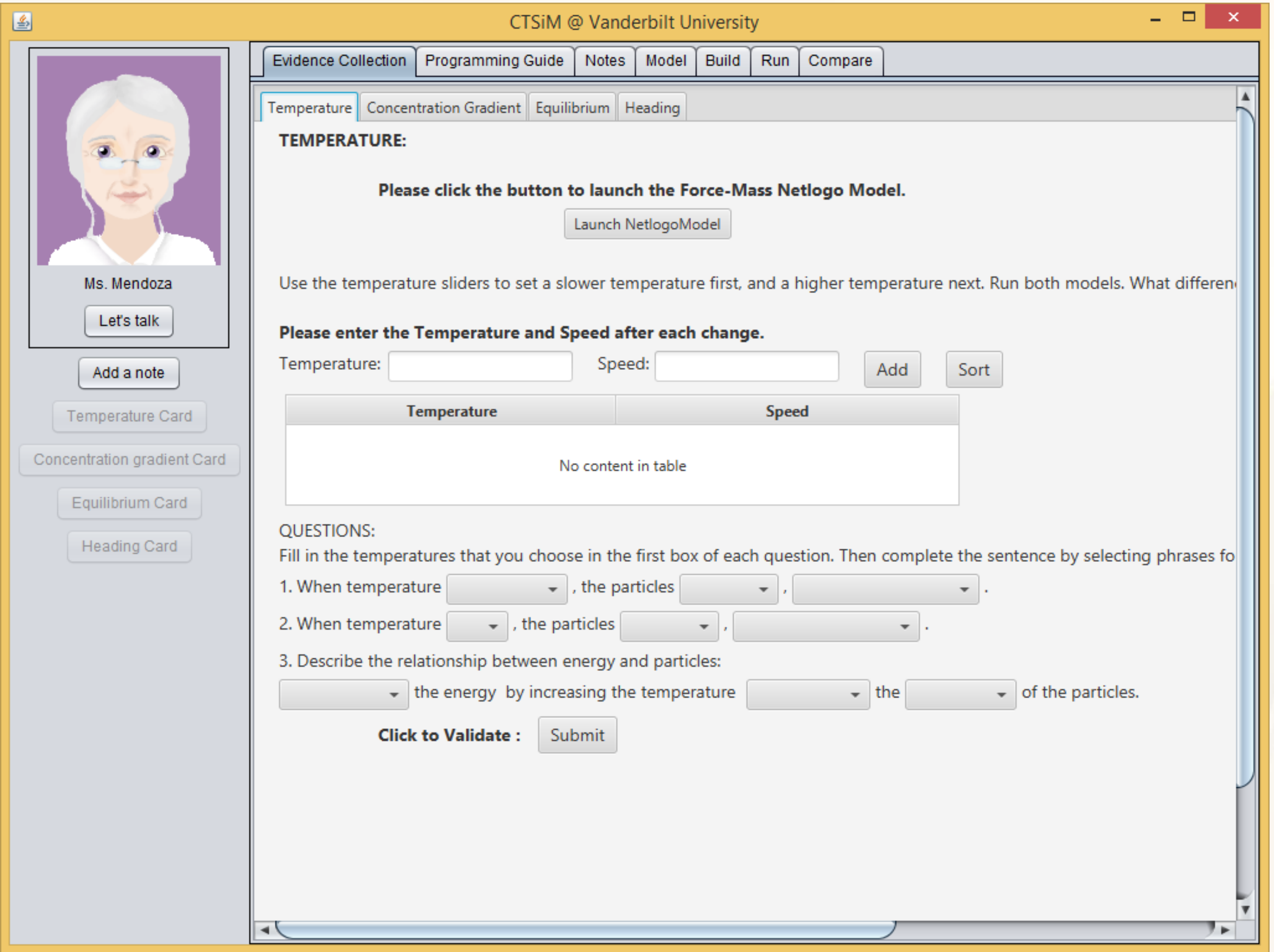
Recent work: Supporting Critical Thinking Skills

- Evidence Collection by Simulation – Diffusion Unit



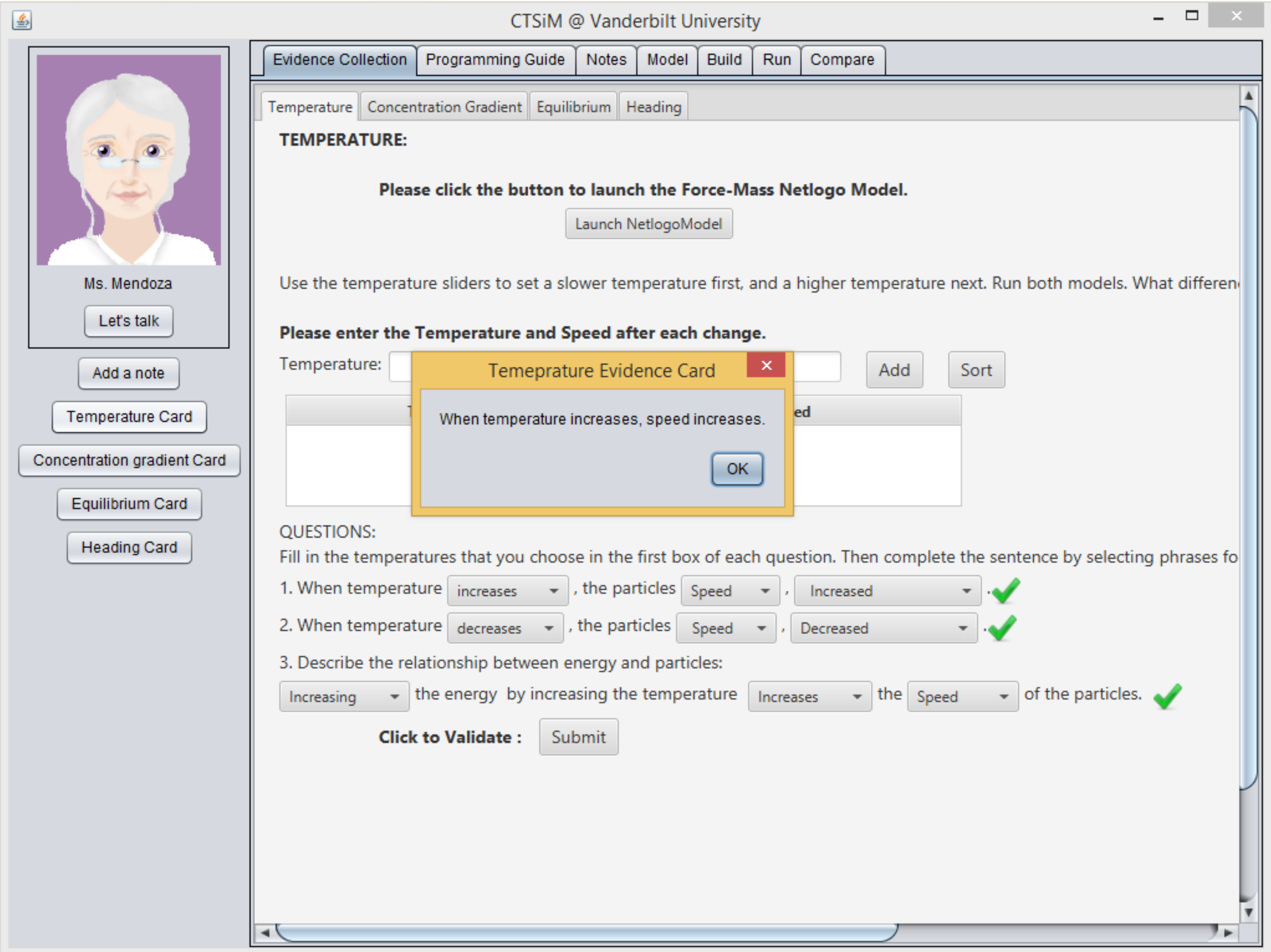
- Sliders to control variables
- Visualizations of the aggregated variables
- Tracing the motion of an individual molecule

Evidence Collection



- Guided construction of important relations
  - Temperature, Concentration Gradient, Equilibrium, and Heading

Collect & Use Evidence Cards



- Acquire an evidence card that summarizes the learning construct of an evidence collection subtask
- Enabled when the corresponding questions are correctly answered
- Learners can click on a card during model building activities to reference the learning construct;

Computational concepts and practices fostered in CTSiM

- Concepts:
  - Algorithmic notions of flow of control: serial execution, conditional logic, iterations
  - Variables to define agent properties and behaviors
  - user inputs to study different scenarios
- Practices:
  - Structured problem decomposition using an agent-based framework
  - Abstraction and modularizing
  - Being incremental and iterative – combining modeling representations
  - Testing and debugging

EARLY STUDIES WITH CTSIM

Classroom Study with CTSiM

- Refs: Basu, et al., 2014 (CSEDU), 2015 (ICCE, ICLS)
- Quasi-experimental design
  - 26 5th grade students (average age = 10.5)
  - Study supervised by science teacher assisted by a graduate research assistant (Basu)
  - Study run daily during science period (45 minutes/day) for 15 days over a period of 3 weeks
  - Students worked individually on all activities
  - Pre-test (Day 1) → Kinematics units (Shapes + Roller Coaster: Days 2-7) → Post-test: Kinematics + CT (Day 8) → Ecology units (Macro + Micro Fish tank: Days 9-14) → Post Test: Ecology + CT (Day 15)

Multiple measures for assessing student learning

1. Summative science and CT tests (pre-post design)
2. Accuracy of students' conceptual and computational models & temporal evolution of the models
   – Distance metrics
3. Average Resource Reading Time

Summary of Results

- Significant learning gains in both science and CT concepts
  - Learning gains, i.e., pre- to post test differences, in science and CT, $p < 0.001$
- Models compared against expert models
  - Correctness, Incorrectness & Distance wrt expert model

Measures	Roller Coaster unit	Fish-macro unit	Fish-micro unit
Final model distance	.39 (.09)	.30 (.23)	.24 (.37)
Number of model edits	155.0 (63.9)	232.2 (87.7)	134.3 (62.9)
Effectiveness of edits	.38 (.08)	.52 (.07)	.58 (.11)
Consistency of edits	.70 (.15)	.87 (.19)	.86 (.17)

  - Model accuracy strong predictor of pre-post learning gains
- Resource Reading Time
  - CT decreased successively from one unit to the next – from ~ 1221 sec to 34 sec
  - Science book reading time $\propto$ difficulty of unit
    - Model accuracy $\propto$ reading time

Challenges students face when building models

- Basu, et al. (2016) RPTEL Journal
- Domain Knowledge Challenges
  - Incomplete or Incorrect Domain Knowledge
    - e.g., Acceleration always increases speed; non zero speed ⇒ acceleration; lack of knowledge of waste cycle in fish tank ecosystem
- Modeling / Agent Based Thinking Challenges
  - Identifying interactions among entities; modeling initial conditions correctly; systematic checking; lack of verification strategies
    - e.g., relation between steepness and acceleration; relation between fish hunger, swim to food, and energy gain (swimming decreases energy; eating increases energy)
- Programming / Computational Challenges
  - Modeling conditionals, choosing the conditions correctly; creating correct nested loops
    - e.g., nested conditionals for roller coaster motion; generality of certain procedures, e.g., eat, breathe – therefore, they can be reused

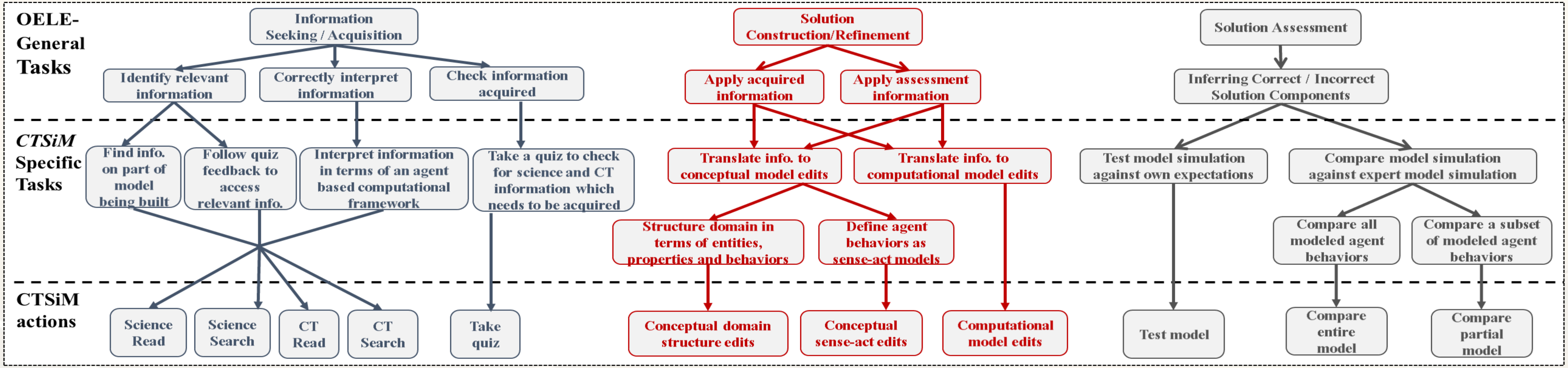
RECENT WORK ON UNDERSTANDING STUDENTS' LEARNING BEHAVIORS IN CTSIM

Online assessment of learner behavior and performance for adaptive scaffolding

- Open-ended nature of CTSiM tasks
  - Freedom to choose from a variety of tasks and combine them in different ways
  - Difficult to infer student plans & strategies for achieving task goals

- Our approach:
  - A task and strategy based modeling framework along with ‘effectiveness’ and ‘coherence’ measures to combine students’ behavior and performance information in the system

The CTSiM task model



Kinnebrew, Segedy, & Biswas (2017) – IEE TLT; Basu, Biswas, & Kinnebrew (2017) -- UMUAI

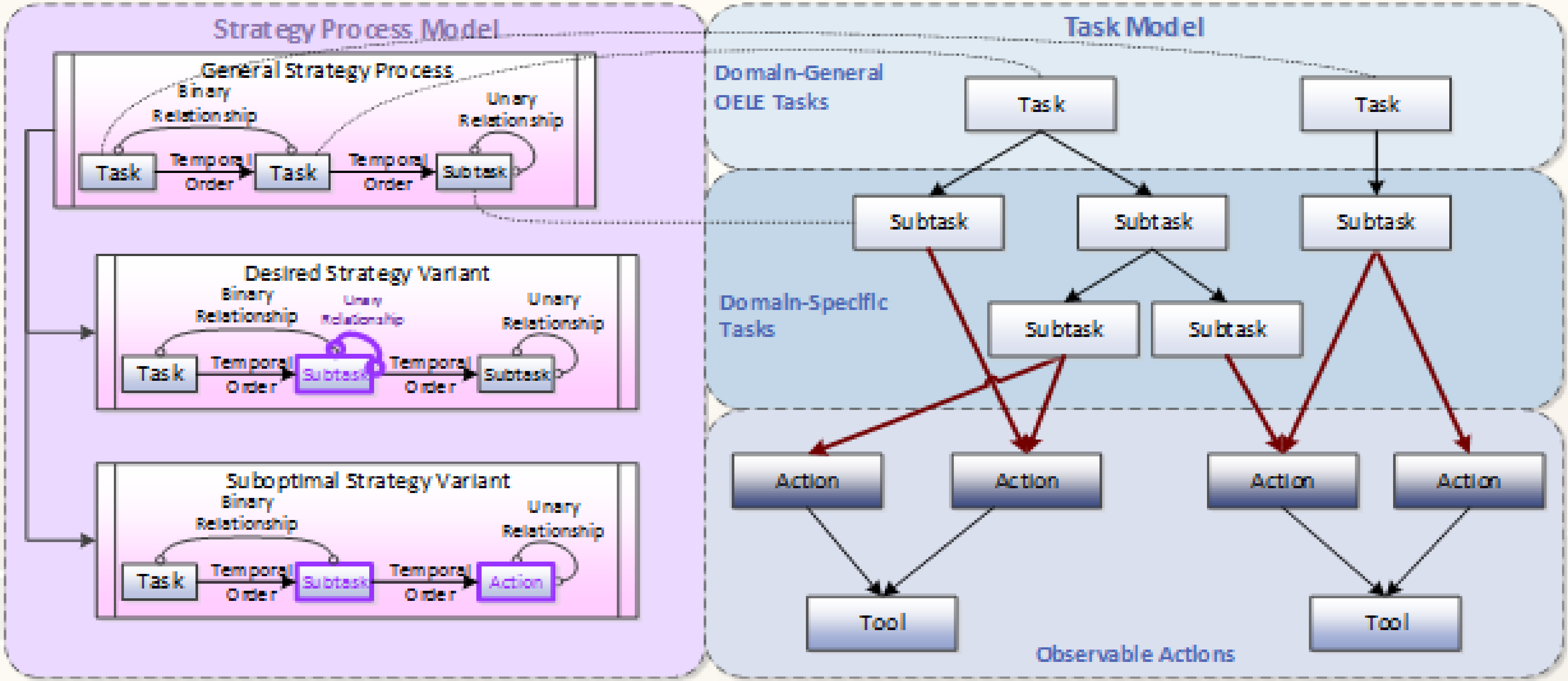
Effectiveness and coherence relations defining strategies

- Kinnebrew, Segedy, & Biswas (2017) – IEE TLT; Segedy, Kinnebrew, & Biswas (2015) – JLA

- Effectiveness of actions:
  - Actions are considered effective if they move the learner closer to their corresponding task goal

- Coherence between actions:
  - Two temporally ordered actions ($x \rightarrow y$), i.e., x before y, exhibit the coherence relationship ($x \Rightarrow y$) if x and y share contexts, and the context for y contains information contained in the context for x
  - The context for an action comprises specific information about the action, such as the specific resource pages read, the particular conceptual or computational components edited, or the agent behaviors compared

Our task and strategy based modeling framework



Adaptive Scaffolding: Combining strategy and performance information to assess and scaffold learners

- Scaffold suboptimal strategies when
  - Modeling performance is below par (Ineffective SC actions) & Incorrect agent behaviors not bring assessed in SA actions

- Also, scaffold desired strategies that lack coherence or lead to low modeling performance (i.e., does not match expert model)
  - e.g., a desired (*SC => Science Read*) strategy is ineffective *if* the *Science Read* not coherent with *behavior blocks created in SC*, or *Read* corresponds to *part of model* that student has already verified to be correct

Adaptive scaffolding in CTSiM

- Principles guiding the feedback
  - Help students only when they have recurrent problems with a task or use of a strategy
  - Feedback contextualized by student's current activities and information available to the student
  - Conversational, mixed-initiative dialog initiated by the mentor agent
  - Suggest useful strategies and where to focus attention
  - Never provide *bottom out hints* (unlike ITS, especially Cognitive Tutors)

Strategies and their suboptimal variants

- Strategies monitored were not exhaustive (Basu & Biswas, 2016 –UMUAI Journal)
  - Were based on students' difficulties observed in previous studies
- Picked five strategies to monitor and provide feedback and hints
- S1: Solution construction followed by relevant information acquisition strategy (SC → Science Read)
  - *Suboptimal S1: (ineffective SC → Science Read), incoherent action sequence*
- S2: Solution assessment followed by relevant information acquisition strategy (SA → Science Read)
  - *Suboptimal S2: (effective SA detecting incorrect agent behaviors → Science Read), incoherent action sequence*
- S3: Information acquisition prior to solution construction or assessment strategy (Science Read → SC|SA)
  - *Suboptimal S3: lack of a Science Read action or an incoherent Science Read action before an effective SA action detecting incorrect agent behaviors*

Strategies and their suboptimal variants

- S4: Test in parts strategy (Effective comparison by isolating erroneous parts or separating erroneous parts)
  - *Suboptimal S4: ineffective Compare action*
- S5: Coherence of Conceptual and Computational models strategy (Sense-act specification → Computational build)
  - *Suboptimal S5: incoherent (Sense-act build → Computational build) action sequence or lack of the action sequence*

Recent experimental study using CTSiM

- Ninety-eight 6th grade students (4 sections)
  - Two conditions: Control (No adaptive scaffolding) versus Experimental (adaptive scaffolding)
  - Students from two sections assigned to control condition (n=46) and students from the other two sections assigned to experimental condition (n=52)
  - Study run daily during science period (1 hour slot for each section) over a period of 3 weeks
  - Students worked individually on all activities

Assessing the effectiveness of CTSiM & our scaffolding approach

- **Synergistic learning gains** in science topic and CT
  - Pre-post tests
  - Reduction in difficulties over time, and reduction in errors made across multiple units
  - Correlation between modeling accuracy versus pre-post learning gains
  - Evolution of modeling accuracy
  - Transfer task – modeling skills and ability in pencil-and-paper test
- **Advantages of coupled representations** (with supporting feedback)
  - Modeling accuracy
  - Change in model building behaviors
- **Effectiveness of adaptive feedback**
  - Model building accuracy
  - Use of strategies
  - Change in amount of feedback received across time

Learning activity progression

- Kinematics Unit (single agent)
  - Teaches relations between speed, acceleration and distance; mathematical representations of motion
  - Introductory practice activity: Draw simple shapes followed by growing and shrinking spirals to understand the relations between constant acceleration, speed, and distance
  - Activity 1: **Model motion of a roller-coaster** on different segments of its track

- Ecology Unit (multiple agents)
  - Teaches notions of balance and interdependence amongst species in an ecosystem
  - Activity 2: **Build a macro-level semi-stable model** of the behavior of fish and duckweed in a fish-tank
  - Activity 3: **Build a micro-level model** of the waste cycle in the fish-tank with bacteria

Summary of Results

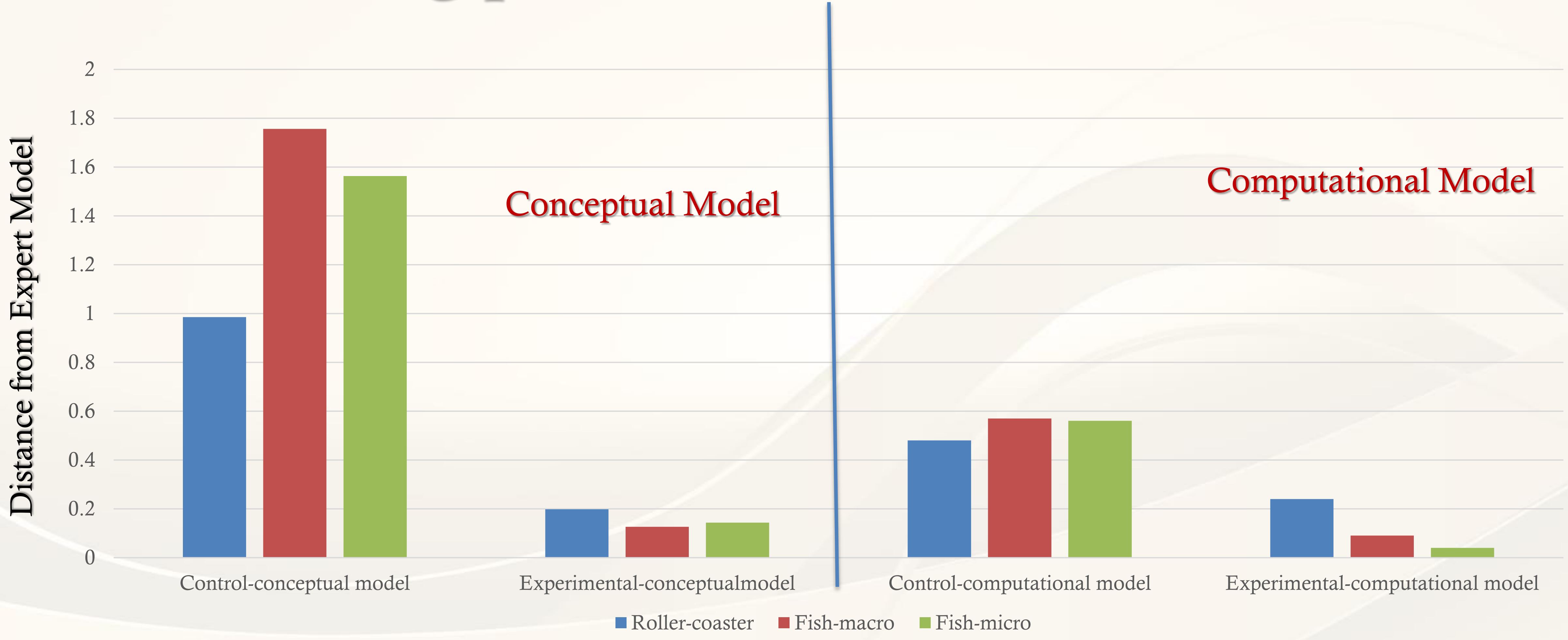
- Experimental vs Control group
  - Performance
    - Significantly higher synergistic learning gains (Kinematics, Ecology, CT)
    - More accurate models (conceptual and computational)
    - Modeling effectiveness, trends, and consistency better
  - Learning Behaviors
    - CT Practices: Better able to build and test models in parts; more coherence when switching between two modeling representations; consistency in model building actions
  - Feedback
    - Effective, Showed fading effect
  - Good transfer of approach and practices
    - Computer to paper and pencil task

Science and CT pre-post learning gains

		Pre	Post	Pre-to-post gains	Pre-to-post *p*-value	Pre-to-post Cohen's d
Kinematics (max = 45)	Control	12.52 (6.32)	15.55 (5.72)	3.03 (4.78)	<0.0001	0.55
	Experimental	16.65 (6.61)	22.38 (6.39)	5.72 (5.62)	<0.0001	0.88
Ecology (max = 39.5)	Control	7.40 (3.90)	16.19 (8.35)	8.78 (7.17)	<0.0001	1.35
	Experimental	9.39 (4.47)	27.91 (6.70)	18.53 (6.31)	<0.0001	3.25
CT (max = 60)	Control	16.49 (5.68)	22.53 (5.70)	6.04 (5.44)	<0.0001	1.06
	Experimental	22.72 (7.68)	32.24 (5.86)	9.52 (5.23)	<0.0001	1.39

- All students gained on science and CT from pre to post test
- Experimental group students had higher gains
  - ANCOVAs factoring out effects of pre-test scores
    - Kinematics: $F = 18.91$, $p < 0.0001$, $\eta_p^2 = 0.17$
    - Ecology: $F = 52.29$, $p < 0.0001$, $\eta_p^2 = 0.36$
    - CT: $F = 40.69$, $p < 0.0001$, $\eta_p^2 = 0.31$

Modeling performance across conditions



Experimental Group built more accurate models – both conceptual & computational

Evolution of students' models in an activity

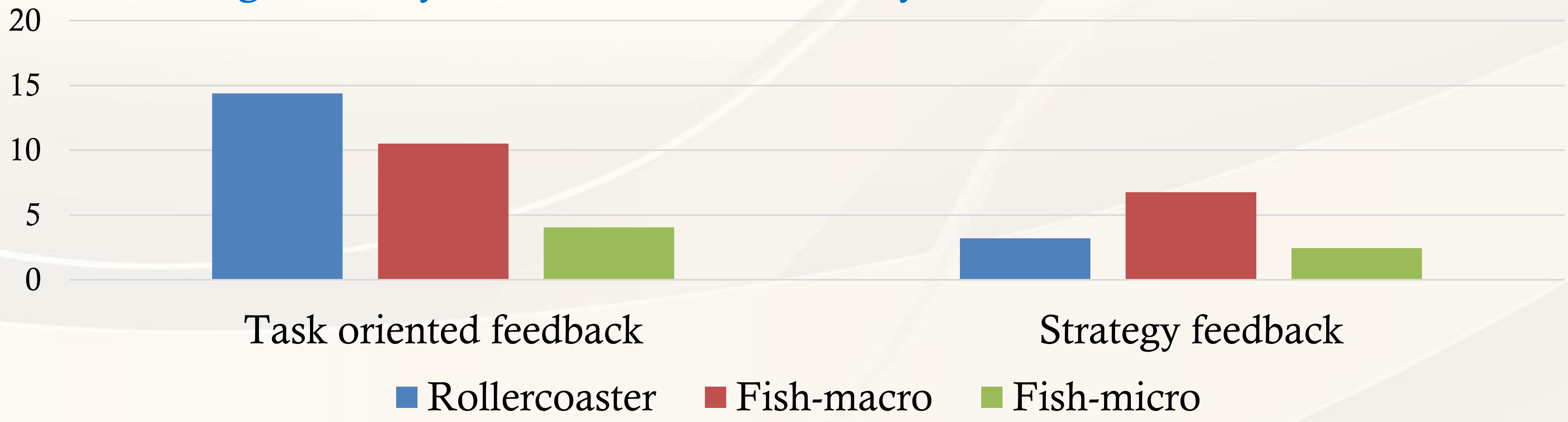
- *Effectiveness* - the proportion of model edits that bring the model closer to the expert model
- *Slope* – the rate and direction of change in the model distance as students build their models
- *Consistency* – How closely the model distance evolution matches a linear trend.
  - For all three measures experimental group outperformed control group, $p < 0.05$ or better

CT Practices
Use of Linked Modeling representations

- Students in the experimental condition decompose their modeling task into more manageable chunks compared to students in the control condition. They also become better at decomposition with time
  - smaller chunk sizes and greater number of switches between conceptual & computational models
- Coherence between the two levels of abstractions in each activity is higher for experimental than the control group
  - Increases across units for the experimental group
- Better decomposition and higher coherence significantly correlated with higher science learning

Variation of feedback received over time

- Students in the experimental condition required a combination of task oriented and strategy feedback in all activities
- Fading effect on the need for scaffolds
  - The task oriented feedback required decreased significantly from the rollercoaster unit to the fish-micro unit
  - The strategy feedback needed increased from the rollercoaster to the complex fish-macro activity but then decreased significantly in the fish-micro activity



Transfer Test: Conceptual and Computational modeling skills

- Modeling a wolf-sheep-grass ecosystem on paper with all system scaffolds removed

Experimental scored significantly higher than Control

		Control	Experimental	*p*-value	Cohen's d
Conceptual modeling score	Conceptual entities (max = 5)	4.66 (0.79)	4.92 (0.39)	< 0.05	0.43
	Conceptual sense-act (max = 41)	11.54 (5.29)	20.93 (6.70)	< 0.001	1.56
	Total score (max=46)	16.21 (5.45)	25.86 (6.73)	< 0.001	1.58
Computational modeling score (max=48)		17.33 (9.23)	30.50 (8.98)	< 0.001	1.46
Total transfer test score (max=94)		33.53 13.80)	53.36 14.49)	< 0.001	1.63

Summary & Conclusions

- CTSiM helps seamlessly integrate CT with middle school science curricula, and fosters synergistic learning of science and CT concepts
- Analyzing students' actions using *task* & *strategy* models, and assessing them in terms of *effectiveness* and *coherence* measures works well
- Adaptive scaffolding based on learner performance and behavior information results in
  - Higher science and CT learning gains
  - Better CT practices
  - Better modeling performance
  - Better able to transfer modeling skills
  - More frequent use of desired strategies.

Work supported by:
NSF Cyber-learning grant #1124175 and
NSF Cyber-learning grant #1441542

Download CTSiM modules from:
http://www.teachableagents.org/downloadsoftware.php

Recent Work

- C^{2}STEM: Collaborative Computational STEM Learning

  (supported by NSF STEM+C grant)

  - run.c2stem.org
  - Directed to High School Physics curriculum in Mechanics
  - Vanderbilt lead
  - Combines instructional and model building tasks; embedded assessments (SRI); PFL assessments (Stanford); Problem solving

Work supported by:
NSF STEM+C grant #1640199

Extra Slides

Acknowledgements

- This work has been supported by
  - NSF Cyber-learning grant #1124175 and
  - NSF Cyber-learning grant #1441542

- You can download the CTSiM modules from: http://www.teachableagents.org/downloadsoftware.php

Science and CT pre-post learning gains

- All students gained on science and CT from pre to post test

Domain	Pre-test score (mean, sd)	Post-test score (mean, sd)	*p*-value 2-tailed	Effect Size
Kinematics (max score = 36.5)	13.62 (5.84)	18.38 (7.1)	< 0.05	0.34
Ecology (max score = 32.5)	5.65 (2.85)	19.69 (6.94)	< 0.0001	2.65
Computational Thinking – Post Test 1 (max score = 1 – normalized)	0.34 (0.19)	0.64 (0.14)	< 0.0001	1.80
Computational Thinking – Post Test 2 (max score = 1 – normalized)	0.34 (0.19)	0.69 (0.19)	< 0.0001	1.84

Model evaluation metrics

- Bag of words metric (Piech, et al., 2012) to compare blocks/primitives used in student model against those in the expert model

1. **weightedAverageCorrectness** $= \frac{\sum_{each\ procedure} |user \cap expert|}{\sum_{each\ procedure} |expert|}$

   Correctness score: Proportion of expert model blocks contained in student model

2. **weightedAverageIncorrectness** $= \frac{\sum_{each\ procedure} (|user| - |user \cap expert|)}{\sum_{each\ procedure} |expert|}$

   Incorrectness score: Extra blocks in student model, normalized by size of expert model

3. **Vector distance from (correctness, incorrectness) to (1,0):**

$$distance = \sqrt{incorrectness^2 + (correctness - 1)^2}$$

Distance = Vector distance from (correctness, incorrectness) vector to (1,0)

Assessing students' computational models

- Models compared against expert models
  - Correctness, Incorrectness & Distance wrt expert model

Measures	Roller Coaster unit	Fish-macro unit	Fish-micro unit
Final model distance	.39 (.09)	.30 (.23)	.24 (.37)
Number of model edits	155.0 (63.9)	232.2 (87.7)	134.3 (62.9)
Effectiveness of edits	.38 (.08)	.52 (.07)	.58 (.11)
Consistency of edits	.70 (.15)	.87 (.19)	.86 (.17)

- Model accuracy strong predictor of pre-post learning gains
  - e.g., *r(model distance fish micro final distance, Ecology gain)* = – 0.52
  - *Edit effectiveness, Consistency of edits – strong predictors of pre-post gains (p* < 0.05*)*

Reading Time: Domain & CT resources

- Time spent reading CT and domain resources

Resources	Units				
	Constant Shape Drawing	Variable Shape Drawing	Roller Coaster	Fish-macro	Fish-micro
Domain	742.9 (262.2)	508.0 (194.8)	427.6 (251.4)	1160.1 (550.7)	1045.9 (509.1)
CT	1221.6 (1359.5)	92.08 (110.3)	44.3 (86.8)	45.7 (69.8)	34.3 (82.6)

- Also, *r(model fish micro distance, Ecology reading time*) = 0.41, $p < 0.05$

Multiple measures for assessing student learning

1. Summative science and CT tests (pre-post design)
2. Accuracy of the conceptual and computational models built and the temporal evolution of the models
3. Learning transfer test when all system scaffolds are removed
4. Use of linked modeling representations to study use of CT practices like abstraction and decomposition
5. Use of desired strategies
6. Variation of feedback received over time

Use of strategies S1-S2

- Average use of strategies higher in the experimental condition
  - A higher proportion of students in the experimental condition used the strategies effectively

		RC		Fish-macro		Fish-micro	
Strategy		Percentage of students	Mean (s.d.)	Percentage of students	Mean (s.d.)	Percentage of students	Mean (s.d.)
S1. Solution construction followed by relevant science reads	Control	37%	1.33 (2.99)	54%	2.43 (4.8)	70%	1.93 (2.05)
	Experimental	63%	2.23 (4.71)	83%	4.75 (4.97)*	85%	3.4 (4.51)*
S2. Solution assessment followed by relevant science reads	Control	4%	0.07 (0.33)	26%	0.76 (1.66)	26%	0.85 (9.31)
	Experimental	38%	1.37 (2.69)**	44%	1.66 (2.29)*	44%	1.06 (0.24)

$^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$

Use of strategies S3-S5

- Average use of strategies higher in the experimental condition

Strategy		RC		Fish-macro		Fish-micro	
		Percentage of students	Mean (s.d.)	Percentage of students	Mean (s.d.)	Percentage of students	Mean (s.d.)
S3. Fraction of assessed agent behaviors which were read about before being assessed	Control	80%	.73 (.42)	93%	.5 (.33)	83%	0.89 (0.27)
	Experimental	92%	.86 (.28)	96%	.77 (.32)***	100%	0.96 (0.16)
S4. Number of partial-model comparisons	Control	0%	na	48%	2.65 (5.79)	15%	0.57 (1.98)
	Experimental	0%	na	58%	5.42 (7.16)*	19%	1.97 (3.22)*
S5. Fraction of added sense-act properties which were either removed or followed by a coherent computational edit	Control	100%	0.67 (0.27)	100%	0.69 (0.31)	98%	0.59(0.31)
	Experimental	100%	0.97 (0.1)***	100%	0.99 (0.03)***	100%	0.98 (0.06)***

*p < 0.05, **p < 0.01, ***p < 0.001

Assessing students' conceptual and computational models

- Models compared against expert models
- Conceptual model:
  - Set comparison to find expert model elements missing in student model and extra elements in student model
  - distance=missing + extra elements in student model, normalized by number of elements in the expert model
- Computational model:
  - Correctness score: Proportion of expert model blocks contained in student model
  - Incorrectness score: Extra blocks in student model, normalized by size of expert model
  - Distance: Vector distance from (correctness, incorrectness) vector to (1,0)

Use of linked modeling representations

- This helps study students' use of CT practices like decomposing a complex task, understanding relations between abstractions
- Metrics used:
  - *Total number of conceptual-computational activity chunks*: measures how many times a student switched between the two representations
  - *Average conceptual and computational chunk sizes*: number of modeling actions of one type taken before shifting to a different modeling representation
  - *Coherence between conceptual modeling actions and computational modeling actions*