

In D. Abrahamson (Organizer), U. Wilensky (Chair), & R. Lesh (Discussant), *Learning Complexity: Agent-based modeling supporting education research on student cognition in social contexts*. Paper presented at the annual meeting of the American Educational Research Association, Chicago, IL.

Agent-Based Modeling as a Bridge Between Cognitive and Social Perspectives on Learning

Dor Abrahamson, *University of California - Berkeley*

Uri Wilensky, *Northwestern University*

James Levin, *University of California - San Diego*

This paper is a proof-of-existence empirical paper. That said, it is also a methodological paper. The methodology is agent-based modeling, a computer-supported mode of inquiry into complex phenomena, such as weather fronts, market fluctuations, or participation patterns in a middle-school mathematics lesson. We have previously shown that agent-based simulation can express theoretical models of learning (Abrahamson & Wilensky, 2005; see also Smith & Conrey, 2007). In that paper, we claimed that a promising attribute of simulation-based research into learning is that it fosters scholarly critique and engaging collaboration. It is that claim that is herein proven to stand. The proof lies in the collaborative criticism offered by the third author, Jim Levin, to the first two authors, Dor Abrahamson and Uri Wilensky, concerning their agent-based simulation of learning, which was first presented at the 2005 annual meeting of the Jean Piaget Society and was then made available online, along with its underlying computational procedures that were laid out for scrutiny. The 2005 paper explicitly invited fellow researchers to critique the simulation and possibly modify it so as to accommodate their own perspectives and possibly enable their own investigations. Indeed, the critique received from Levin took a unique form—he improved the computer-based model such that it better simulates the target constructs. Such co-constructive critique, we argue, is a hallmark of the promise of agent-based modeling. So we submit that this proof of existence, if anecdotal validation, may be a harbinger of a new mode of research in the learning sciences and beyond—a mode that builds on constructionism (Papert, 1991): *constructionist collaboration*.

Background

We have argued (Abrahamson & Wilensky, 2005) that ABM has potential to contribute to the advancement of theory in at least three major ways: (a) explicitizing—ABM computational environments demand an exacting level of clarity and specificity in expressing a theoretical model and provide the tools, structures, and standard practices to achieve this high level; (b) emergence—the computational power of ABM enables the researcher to mobilize an otherwise static list of conjectured behaviors and witness any group-level patterns that may unfold through multiple interactions between the agents who implement these conjectured behaviors; and (c) intra/inter-disciplinary collaboration—the *lingua franca* of ABM enables researchers, who otherwise use different frameworks, terminology, and methodologies, to understand and critique each others' theory and even challenge or improve the theory by modifying and/or extending the computational procedures that underlie the model. It is the latter attribute of ABM that enabled the third author to readily engage in discourse with the first two authors, following their initial presentation of the “I’m Game!” Piagetian–Vygotskiiian model (Abrahamson & Wilensky, 2005).

In this paper, we begin by recapping the Abrahamson–Wilensky paper and model (hence, A–W) and then present Levin’s proposed improvement on the A–W model as a case study of the A–W call for model-based constructive critique. We end with a cautionary remark on the limitations of ABM, the importance of recognizing these limitations, and the relevance of this caution for the field’s understanding, and prospective incorporation, of ABM as a viable means of propelling research into the mechanisms of learning.

The A–W 2005 Paper: ABM as a Methodology in the Service of Theory-of-Learning Research

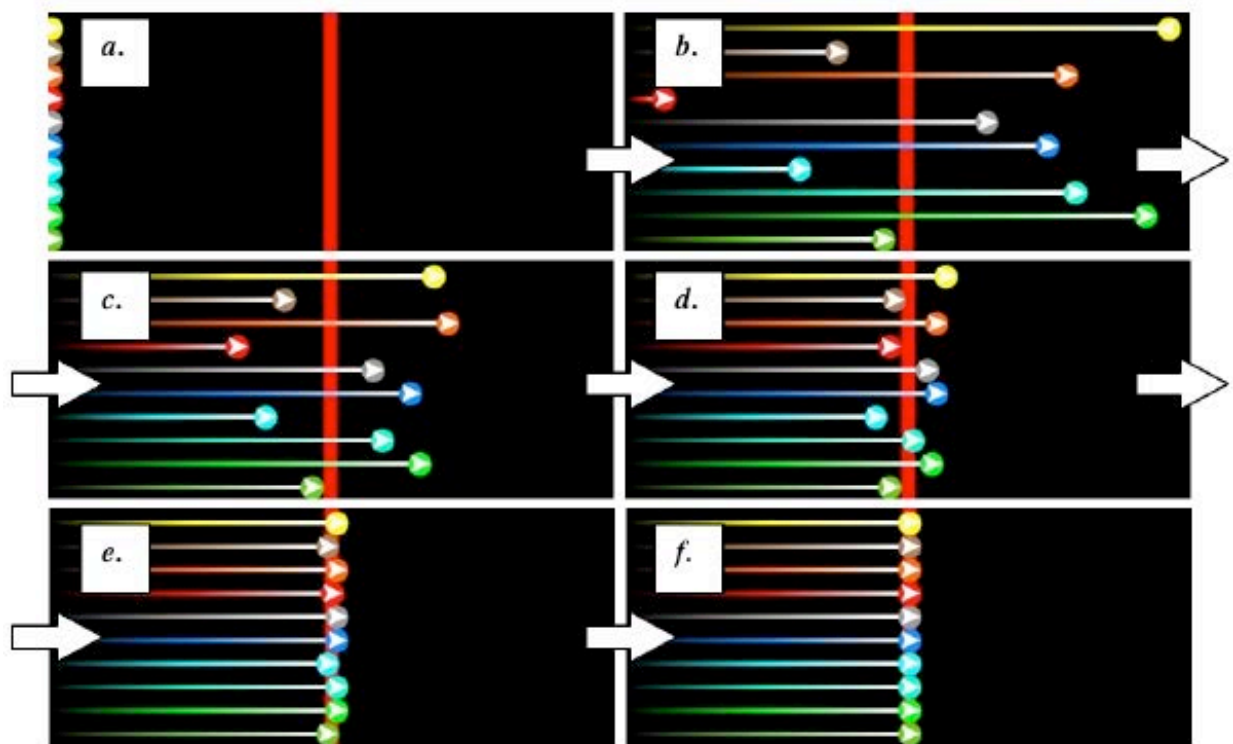


Figure 1. Six snapshots from successive experimental runs in the “I’m Game!” model.

The “I’m Game!” model (see Figure 1, above), built in the NetLogo agent-based modeling environment (Wilensky, 1999), implements sketches of “Piagetian” and “Vygotskiiian” interpretations of human learning into a single model and examines learner–agents’ performance under “Piagetian,” “Vygotskiiian,” and combined “Piagetian–Vygotskiiian” conditions. “Players” (circled arrows) stand in a row (Figure 1a). They each roll a marble toward a target line. Some players undershoot the line, some overshoot it (Figure 1b). Players collect their marbles, adjust the force of their roll—based either on their own performance (“Piagetian”) or on their neighbors’ (“Vygotskiiian”)—and, on a subsequent trial (Figure 1c), improve on their first trial—they have “learned” as individuals. Gradually, the group converges on the target line (see Figure 1d-1f, for three later attempts).

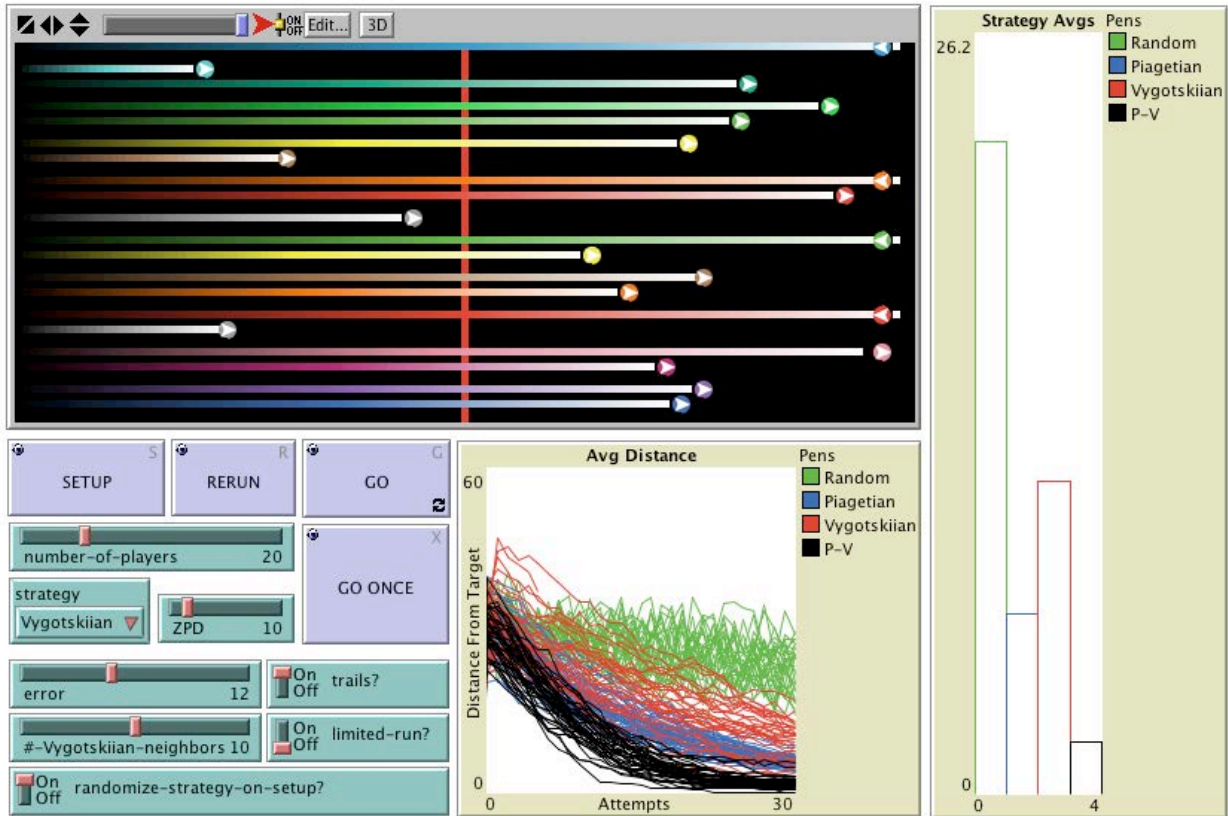


Figure 2. Interface of the “I’m Game!” NetLogo model, with results from multiple runs.

We have run the model under a wide range of conditions so as to evaluate its capacity to simulate reliably core features of the embedded theoretical model. For example, Figure 2 (above) shows results from running the simulation within a particular parameter space (see the sliders and switches) under the three experimental conditions and a fourth, control condition (“random”). Group mean performance (distance from target) was ranked as “Piagetian–Vygotskian” (nearest, so best), “Piagetian,” “Vygotskian,” and “Random” (furthest, so worst). We thus express one possible interpretation of the complementarity of the Piagetian and Vygotskian perspectives (e.g., Cole & Wertsch, 1996).

Proof of Existence: LCHC Responds to CCL’s Call for ABM-Based Theory-of-Learning Collaborative Research

As part of an effort to study distributed learning, members of the Laboratory of Comparative Human Cognition (LCHC) at the University of California, San Diego began using NetLogo as a modeling environment for learning and examined the Abrahamson–Wilensky (A–W) model of Piagetian and Vygotskian learning. LCHC draws heavily on cultural historical activity theory developed by Vygotsky and others, and so they were especially interested in the expression of Vygotskian learning embedded in the A–W model. The A–W model was introduced at a weekly laboratory meeting in several ways. First, the model was shown to the whole group while it ran through a series of simulations of learning. The parameters made available through the interface were explored. The underlying code was examined, but the amount of code and the relative unfamiliarity with NetLogo code by most of the Lab limited the utility of this examination. A printout of a subset of the code was distributed to the Lab members, with the core code

expressing the theory highlighted. Even though members of LCHC by and large did not know how to interpret NetLogo statements, a textual explanation of the core code was given by two members of the Lab who were familiar with NetLogo.

During the discussion, it became clear that the A–W implementation of the concept of “Zone of Proximal Development” was “simplex” – that is, learning in the ZPD embodied in the model depended on changes by the less skilled member of a pair of learners without any changes by the more skilled member. Several members of LCHC pointed out how the dynamic construction of a ZPD is “duplex,” that it involves both the learner and the “teacher” (the more skilled person). It occurred to Jim Levin that a relatively simple change to the model would implement a way in which teachers change their behavior depending on their knowledge of the level at which their students are performing. In this way, the members of LCHC were accepting the challenge in the Abrahamson & Wilensky (2005) paper describing this model: “We would welcome a critique that uses the existing model as a basis for expressing these constructs” (p. 28).

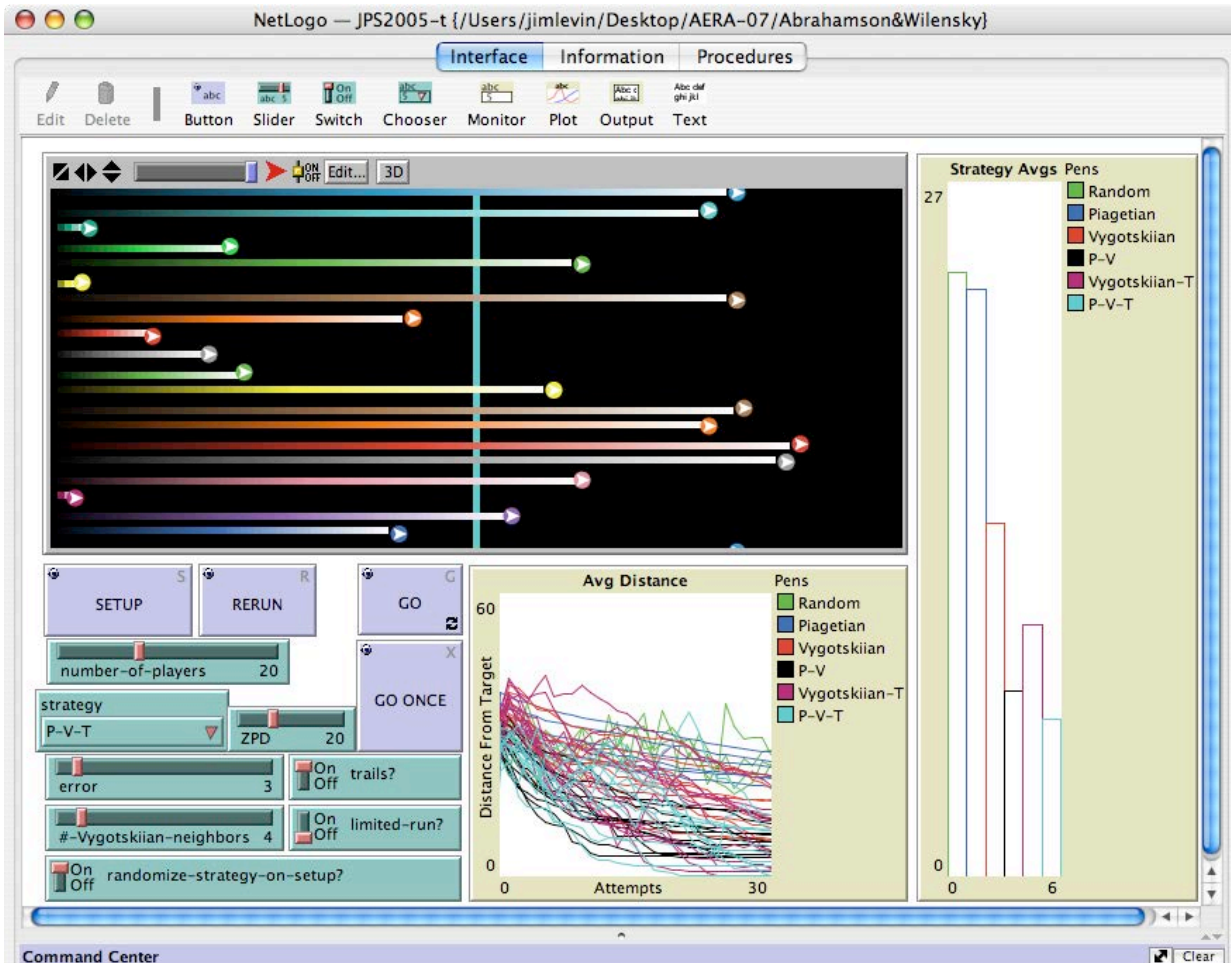


Figure 3: Interface of the “I’m Game!” A-W model as modified by Jim Levin to improve the Vygotskian learning.

Levin implemented a modification to the A–W model in which the better performing player modified their next move to be within the ZPD of the less well performing play, making a play

that was worse than they knew how to make, in order to help the less well performing player learn to play better. The modification is called the “-T” version (“T” for “teacher”), and a screenshot of the model after about 50 runs is shown in Figure 3, above.

While members of LCHC who viewed this modified model at a later meeting then raised further concerns about the model, especially about the potential for oversimplification with this simulation, they were also fairly impressed with the ease of modification. For researchers who are new to modeling-based inquiry, it is clearly easier to modify an existing model than to construct one.¹ And it is even easier to explore the parameter space of an existing model.

A close examination of Figures 2 and 3 will reveal that not only does Figure 3 have two additional learning strategies but it also has different parameter values for three of the four parameters that take a range of numeric values, “ZPD,” “error,” and “#-Vygotskian-neighbors”. If we set those three parameters to have the same values as in Figure 2, we see quite different results, as shown in Figure 4, below.

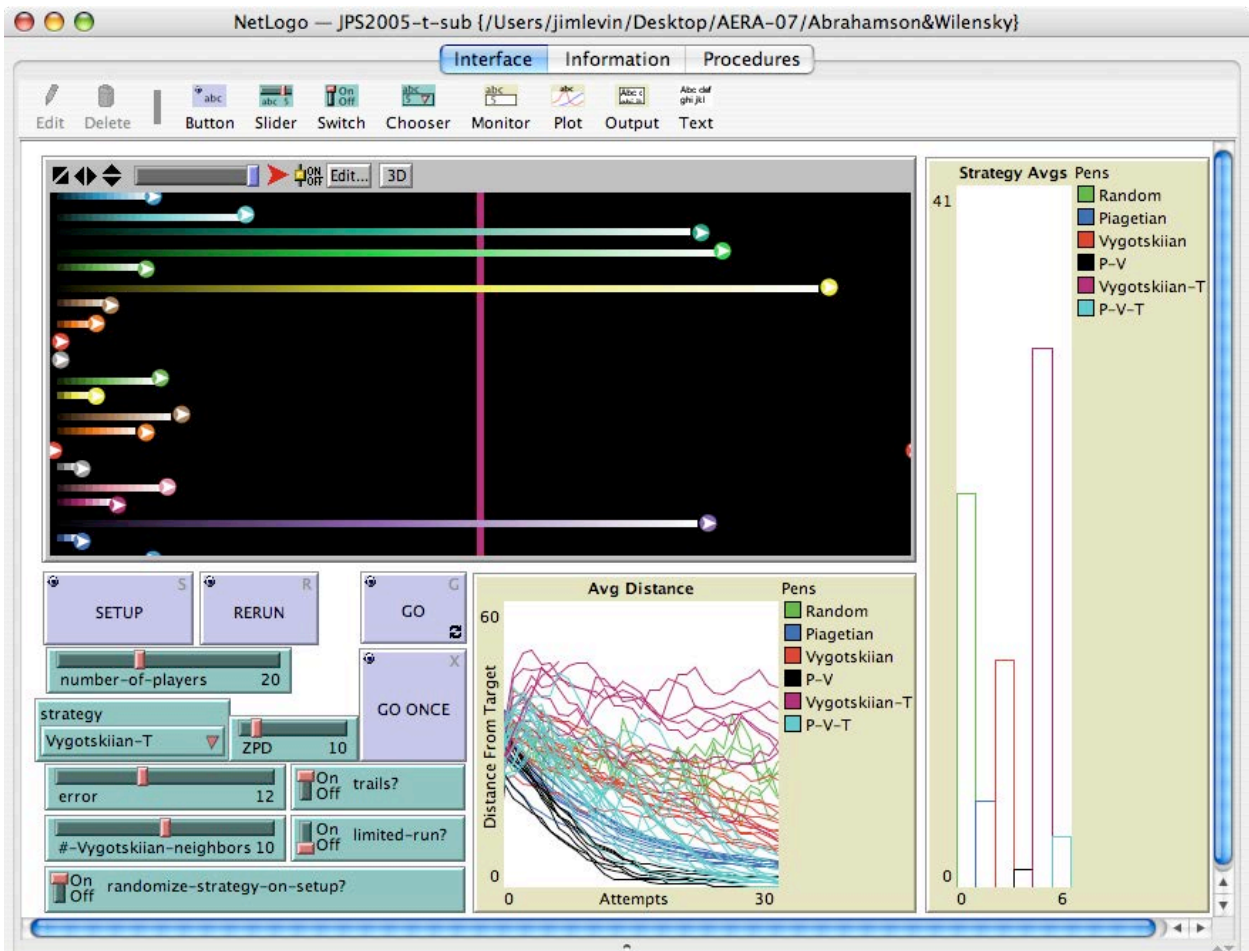


Figure 4: Interface of Levin’s modified A–W model, with a different set of numeric parameters.

¹ NetLogo was designed so as to enable even children to be able to easily construct their own models. In the pedagogy developed at the CCL, we have found that an important step in the process of learning to construct one’s own model is to modify an existing model.

Note that in this run of the modified A–W model, the Piagetian strategy does well, but not as well as the combined P-V or P-V-T strategies. The Vygotskiiian strategy does less well, and the Vygotskiiian-t version does worse than the “Random” strategy. So it matters quite a bit which set of parameters are chosen in comparing these learning strategies. The Abrahamson & Wilensky paper (2005) explores some of the multi-dimensional parameter space, using NetLogo’s *BehaviorSpace* research tools (Wilensky, 2001; Tisue & Wilensky, 2004). But it is not surprising that the figures in that paper, presented to the Jean Piaget Society, show the model operating with parameters in which the “Piagetian” strategy does well, while the parameters shown in Figure 3 are chosen for the Levin & Cole (2007) paper which is part of this symposium.

So an important lesson is that parameterized models such as these are open to many different interpretations, depending on which regions of the parameter space you elect to explore. Further enhancements to ABM environments could facilitate this exploration by offering users interactive tools for navigating in the conceptual space expressed in the combinatorics of the parameters selected for modeling (for a very first sketch of possible development directions, see Appendix A for the entire code of the “-T” model, with print-based orientation coding to help constructionist collaborators navigate through procedures).

Conclusion

We have presented a case study—a proof of existence—of an agent-based model that enabled constructionist collaboration of two research groups in the refinement of a theoretical model. The following attributes of agent-based modeling enabled and supported the collaborative research. Agent-based modeling:

- a. compels clear articulation of conceptual models and, *inter alia*, catalyzes personal refinement of these models (Papert’s “debugging principle,” 1980);
- b. clearly circumscribes the theoretically known—the unknowns are expressed in the form of “patchy” procedures that hard-code emergence into the simulation;
- c. introduces modularity into the theoretical model, i.e., the program’s procedures isolate each of the modeler’s conceptual assumptions. Thus, reviewers can readily locate each of the assumptions and implement modifications to the modeling code such that it agree with their own theoretical assumptions;
- d. constitutes a *lingua franca* that transcends and cuts across the ever-ramifying niches of academic expertise (Jacobson & Wilensky, 2006);
- e. stimulates and facilitates collaboration, including focused critique and modification (Levin & Cole’s “distributed research” principle, this symposium). In particular, ABM invites researchers who have not collaborated before to partake in a distributed constructionist project.

Finally, the exteriorization of the model as a conceptual object with a felt agency of its own (“It’s alive!”) is perhaps more conducive to collaboration than are idiosyncratic models. Thus, ABM invites co-construction but not at the cost of constructive criticism.

Our point of departure was that learning is a complex phenomenon and is therefore suitable for study through agent-based modeling. And yet, the emergence of a modeling *lingua franca* for

studying learning can itself be studied through the lenses of complexity studies. In particular, ABM is an inherently interdisciplinary form of inquiry, because the meta-principles of diverse phenomena can be articulated from the perspective of numerous agents acting locally with more-or-less common rules (Jacobson & Wilensky, 2006). Therefore, we see a game-theoretical advantage for multiple researchers within and between disciplines to share a mode of inquiry. Collaborative research, a form of learning, is itself complex, and we perceive agent-based modeling as advantageous to all researcher–agents.

The Illusion of Agency: On What Agent-Based Modeling Is and What It Is Not

In closing, we wish to submit the following remark that may help in further clarifying our perspective on the nature of agent-based modeling vis-à-vis more traditional forms, especially with regards to its standing as embodying a form of inquiry. This remark should be taken as preemptive—it responds to anecdotal, rather than well-articulated, sentiments toward ABM that we have encountered as we share our work.

From the view-point of pre-ABM, Graham Cairns-Smith, a chemist focusing on evolution and consciousness, for whom modeling is an essential *modus operandi*, supposes that any educated consumer of models can distinguish between what a model is and what it is not:

Good analogies are *like* in some respects that are clearly understood; and *unlike* in other respects that are clear also. The organic chemist’s models, for example, are thought to correspond to real molecules with respect to distances between atoms and so on, but not literally in all respects, and usually the distinctions are clear enough. Maybe the little sticks are made of steel and they go rusty, but there is no danger at all of this being taken as a blinding new insight about the nature of molecules....Of course you should never *believe* models, they only have an ‘as if’ status and they are likely to let you down any time” (Cairns-Smith, 1996, pp. 46-47).

Any agent-based model is primarily no more and no less than a model. Granted, ABM incorporates avatars that often bear greater iconic resemblance to their phenomenal correlates than do static diagrams; agents are dynamic and interactive such that they induce a compelling anthropomorphic sense of agency; and agents operate fast, in parallel, and iteratively such that an uninformed model user is likely overwhelmed by the magnitude of perceptual information—it is as though the agent-based model takes on a life of its own, as though it is not an ‘as-if’ but ‘the thing itself.’ Indeed, we have observed many middle-school students cheering as a green triangle and a blue triangle compete against each other in a stochastic race across the computer interface. Students’ suspension of disbelief, projection of self, and syntonic embodiment of the stark geometrical forms appears vastly facilitated by the self-propelled motion (see Papert, 1980, on syntonicity). It needn’t take a Tamagochi reality pet to induce reality—blobs moving across the screen can do it, too. Thus, the naïve phenomenology, if not the critical rationalization, of viewing a “run” of an agent-based model appears to share much in common with viewing that which is modeled therein.

However, no matter how great the verisimilitude, the complexity, and unpredictability of a model; however transfixing its dynamism; however enticing, engaging, and immersive its syntonic narrative...it is still a model. Indeed, one important reason that ABMs have proved so

powerful for learning is this verisimilitude. But this verisimilitude can have a cost of making people forget that they are working with a model, essentially the same kind of beast as an equation. Just as a quadratic equation can model ballistic motion but would not ever be expected to describe the texture of the projectile, so one should not expect an ABM to model the entire richness of a social interaction, only the aspects chosen to be modeled. As such, agent-based models—like any other model employed in the sciences and social sciences—is an expression only of carefully selected aspects of a phenomenon—aspects that are assumed to operate individually and in concert so as to factor into the key phenomenon of interest. Beyond that, appearances are meant only ostentatiously—they tag the avatar as ‘a model of X’—and this ostentation may cause confusion or frustration when the avatar does not deliver the richness of what X does in reality. This beguiling property of models is to be understood by the model user. The modeler certainly does not purport to offer a virtual animistic fetish—a Golem or Gollum—it is the viewer, and not the modeler, who invests the avatar with life and so expects this life to live to the brim. We hope this clarification will assist in fostering a critical yet productive use of agent-based modeling in collaborative scientific discourse.

References

- Abrahamson, D., & Wilensky, U. (2005). *Piaget? Vygotsky? I'm game!/: Agent-based modeling for psychology research*. Paper presented at the annual meeting of the Jean Piaget Society.
- Cairns-Smith, A. G. (1996). *Evolving the mind: On the nature of matter and the origin of consciousness*. New York: Cambridge University Press.
- Cole, M., & Wertsch, J. V. (1996). Beyond the individual-social antinomy in discussions of Piaget and Vygotsky. *Human Development, 39*(5), 250-256.
- Jacobson, M. J., & Wilensky, U. (2006). Complex systems in education: Scientific and educational importance and research challenges for the learning sciences. *Journal of the Learning Sciences, 15*(1), 11-34.
- Levin, J. A., & Cole, M. (2007). Simulations as mediators for distributed research activity. Paper presented at the American Educational Research Association meetings, Chicago, IL.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. NY: Basic Books.
- Papert, S. (1991). Situating constructionism. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 1-12). Norwood, NJ: Ablex Publishing Corp.
- Smith, E. R., & Conrey, F. C. (2007). Agent-based modeling: A new approachy for theory building in social psychology. *Personality and social psychology review, 11*, 87-104.
- Tisue, S., & Wilensky, U. (2004, October). *NetLogo: Design and implementation of a multi-agent modeling environment*. Paper presented at the Agent 2004 conference, Chicago, IL.
- Wilensky, U. (2001). *BehaviorSpace* [Computer Software]. Evanston, IL: Center for Connected Learning and Computer Based Modeling, Northwestern University.
<http://ccl.northwestern.edu/netlogo/behaviorspace>.
- Wilensky, U. (1999). *NetLogo*. [Computer Software]. Evanston, IL: The Center for Connected Learning and Computer-Based Modeling, Northwestern University.
<http://ccl.northwestern.edu/netlogo/>.

Appendix A – NetLogo Procedures of the “I’m Game” Model “-T,” With Proposed Print-Based Orientation System Supporting Constructionist Collaboration

```

;; Coded for core vs. infrastructure
;; Core theory content coded for high confidence, normal confidence, and less confidence

;; Modifications to the model by Dor Abrahamson & Uri Wilensky added by J A Levin 5 Feb 06
;; Further mods added by J A Levin 27 Apr 06
;; Original model available at:
;; http://ccl.sesp.northwestern.edu/research/conferences/JPS2005/JPS2005.nlogo
;; This modification is available at:
;; http://tpepserver.ucsd.edu/~jlevin/JPS2005-t.nlogo

globals [ target ;; the target line in the middle
  max-dist ;; the maximum distance to that wall
  runcount ;; the number of times the ball has been thrown since the start
  ticks-left ;; the number of runs left of the current strategy
  pavg vavg pvavg ravg vavgt pvavgt ;; lists of each run's distance halfway through ticks-left
  ;; ex. pavg = [ 2 3 17 ... ]
  ;; for comparison

  current-strategy
  result
]

turtles-own [ max-moves ;; the max-moves of the current throw
  best-max-moves ;; the best throw of the agent (or, if no memory, the current throw)
  best-max-moves-private ;; best throw, even if current throw different for teaching reasons (JAL)
  moves-left ;; the number of moves left until the ball comes to rest
  score ;; the current score (how far from target... lower is better)
  best-score ;; best score (or, if no memory, current throw)
]

to setup
  ca
  set runcount 0
  set-default-shape turtles "circle-arrow"
  crt number-of-players
  ask turtles [ set color color + 0.1 ] ;; to make trails a little easier
  set pavg [0] set vavg [0] set pvavg [0] set ravg [0] set vavgt [0] set pvavgt [0]
  ;; initialized with zeros so i can use mean from the start
  rerun
end

to rerun
  cp
  set runcount runcount + 1
  ;; random strategies lets you keep the simulation running picking from the available strategies
  ;; at random
  if ( randomize-strategy-on-setup? ) [
    let new-strategy random 6
    if ( new-strategy = 3 ) [ set strategy "Random" set current-strategy 3 ]
    if ( new-strategy = 2 ) [ set strategy "Piagetian" set current-strategy 2 ]
    if ( new-strategy = 1 ) [ set strategy "Vygotskiiian" set current-strategy 1 ]
    if ( new-strategy = 0 ) [ set strategy "P-V" set current-strategy 0 ]
    if ( new-strategy = 4 ) [ set strategy "Vygotskiiian-T" set current-strategy 4 ]
  ]
end

```

```

if( new-strategy = 5 ) [ set strategy "P-V-T" set current-strategy 5]
]
set ticks-left 30
; set ticks-left 16 ;; we use this one to stop each run half-way through. (see graph) This gives the info we need
;; , because the histogram represents values from half way through the run. For presentation, though, use "30"

set max-dist ( world-width )
setup-plots
setup-target
setup-turtles
display
end

to setup-plots
set-current-plot "avg distance"
set-plot-y-range 0 max-pxcor ;; most will fall within this range (all after a couple of steps)
set-current-plot-pen strategy ppu plotxy -1 0 ppd

set-current-plot "strategy avgs"
if length ravg > 1 [set-current-plot-pen "Random" plot-pen-reset plotxy 0 precision mean butlast ravg 2]
if length pavg > 1 [set-current-plot-pen "Piagetian" plot-pen-reset plotxy 1 precision mean butlast pavg 2]
if length vavg > 1 [set-current-plot-pen "Vygotskiiian" plot-pen-reset plotxy 2 precision mean butlast vavg 2]
if length pvavg > 1 [set-current-plot-pen "P-V" plot-pen-reset plotxy 3 precision mean butlast pvavg 2]
if length vavgt > 1 [set-current-plot-pen "Vygotskiiian-T" plot-pen-reset plotxy 4 precision mean butlast vavgt 2]
if length pvavgt > 1 [set-current-plot-pen "P-V-T" plot-pen-reset plotxy 5 precision mean butlast pvavgt 2]

end

to-report get-strategy-color
ifelse ( strategy = "Random" )
[ report green ]
[ ifelse ( strategy = "Piagetian" )
[ report blue ]
[ ifelse ( strategy = "Vygotskiiian" )
[ report red ]
[ ifelse ( strategy = "Vygotskiiian-T" )
[ report magenta ]
[ ifelse ( strategy = "P-V-T" )
[ report cyan ]
[ report grey ]
]
]
]
]
end

to setup-target
;; the target is the line in the center of the screen, 2 patches thick
set target patches with [ abs ( pxcor ) < 1 ]
ask target [ set pcolor get-strategy-color ]
end

```

```

to setup-turtles
  ask turtles
  [
    ;; spread out the turtles evenly
    set size ( world-height / count turtles )
    setxy min-pxcor ((who * world-height / count turtles)) + 2
    set score 100000
    set best-score score
    set heading 90

    ;; their max-moves are randomly distributed over the length of the playing field
    set max-moves random max-dist
  ]
end

to go
  ;; the score is their distance from the target
  ask turtles [ set score evaluate-score ] ;; the score is the distance of the turtle from the target line (evaluate-score is a
  reporter that provides this distance)
  ask turtles [ adjust ] ;; act according to the strategy, e.g., in Vygotskiiian, you compare your scores to a neighbor and
  possibly update your score

  reset

  ;; move all the turtles forward to their max-moves spot.
  ;; we can't just say "fd max-moves" because we want them to bounce off the wall
  ;; + leave a dissipating trail
  ask turtles [ set moves-left limit-legal-distance max-moves ]
  let moving-turtles turtles with [ moves-left > 0 ]
  while [ any? moving-turtles ] [
    set moving-turtles turtles with [ moves-left > 0 ]
    ask moving-turtles [
      move-x
      if (trails?) [ set pcolor (color - 5) + ( 10 * ( max-moves - moves-left ) / max-moves ) ]
    ]
  ]

  do-plots
  ;;if sum (sentence length pavg length vavg length pvavg length ravg) >= 5 [stop]
  if (ticks-left < 0) and limited-run? [ display stop ]
  if (ticks-left < 0) and not limited-run? [ display rerun
    if ( strategy = "Random" ) [ if length ravg > 1 [ set result precision mean butlast ravg 2 ] ]
    if ( strategy = "Piagetian" ) [ if length pavg > 1 [ set result precision mean butlast pavg 2 ] ]
    if ( strategy = "Vygotskiiian" ) [ if length vavg > 1 [ set result precision mean butlast vavg 2 ] ]
    if ( strategy = "P-V" ) [ if length pvavg > 1 [ set result precision mean butlast pvavg 2 ] ]
    if ( strategy = "Vygotskiiian-T" ) [ if length vavgt > 1 [ set result precision mean butlast vavgt 2 ] ]
    if ( strategy = "P-V-T" ) [ if length pvavgt > 1 [ set result precision mean butlast pvavgt 2 ] ]
  ]

  set ticks-left ticks-left - 1

end

```

```

to move-x
  set moves-left moves-left - 1
  fd 1
  if ( pxcor >= (max-pxcor - 1) )
    [ set heading 270 fd 2 ]
end

to-report evaluate-score
  report ( distancexy-nowrap 0 ycor ) ; the target has x-coordinate of 0, so this gives the horizontal distance of this turtle
  to the target
end

to-report limit-legal-distance [ val ]
  report ( min ( list ( max-dist - 1 ) max ( list 0 val ) ) )
end

to do-plots
  set-current-plot "avg distance"
  let curr-mean mean values-from turtles [ evaluate-score ]
  plot curr-mean
  ;; we sample after 15 steps. later than that, we lose information
  if (ticks-left = 1) [
    if strategy = "Piagetian" [ set pavg fput round curr-mean pavg ]
    if strategy = "Vygotskiiian" [ set vavg fput round curr-mean vavg ]
    if strategy = "P-V" [ set pvavg fput round curr-mean pvavg ]
    if strategy = "Random" [ set ravg fput round curr-mean ravg ]
    if strategy = "Vygotskiiian-T" [ set vavgf fput round curr-mean vavgf ]
    if strategy = "P-V-T" [ set pvavgf fput round curr-mean pvavgf ]
  ]
end

to reset
  cp
  setup-target
  ask turtles [ set heading 90 set xcor min-pxcor ]
  display
end

to adjust
  if strategy = "Random" [
    r-adjust
    set max-moves best-max-moves
    stop
  ]
  if strategy = "Piagetian" [ p-adjust ]
  if strategy = "Vygotskiiian" [ v-adjust ]
  if strategy = "Vygotskiiian-T" [ v-adjust-t ]
  if strategy = "P-V" [ pv-adjust ]
  if strategy = "P-V-T" [ pv-adjust-t ]

  if (strategy = "Vygotskiiian") or (strategy = "Vygotskiiian-T") ;; or (strategy = "P-V-T")
  [
    set max-moves limit-legal-distance
    ( best-max-moves ± ( random-normal 0 ( error * best-score / max-dist ) ) )
  ]

```

```

stop
|

ifelse ( xcor ≥ 0 ) [
  set max-moves limit-legal-distance
  ( best-max-moves ± ( - abs random-normal 0 ( error * best-score / max-dist ) ) )
] |
  set max-moves limit-legal-distance
  ( best-max-moves ± ( abs random-normal 0 ( error * best-score / max-dist ) ) )
|
end

to p-adjust
;; if your score is better, that's your new best, otherwise stick with the old
if ( score < best-score ) [
  set best-score score
  set best-max-moves max-moves
]
end

to v-adjust-t ;; modified from v-adjust by JAL
if (abs((max-pxcor / 2) - best-max-moves) > abs((max-pxcor / 2) - best-max-moves-private)) [set best-max-moves best-max-moves-private] ;; restore best knowledge
let fellow nobody
while [ fellow = nobody or fellow = self ] [ set fellow turtle (who ± ( - (#-Vygotskiiian-neighbors / 2) ) + random (1 ± #-Vygotskiiian-neighbors ) ) ]
;; look randomly to one of your neighbors

ifelse ( best-score > best-score-of fellow ) and ( best-score - ZPD <= best-score-of fellow ) ;; if other is better and within ZPD
[
  set best-score best-score-of fellow
  set best-max-moves best-max-moves-of fellow
]
[
  set best-score score
  set best-max-moves max-moves
]

;; below is the new code that attempts to model the "teacher's" awareness of the "student" (JAL)
set best-max-moves-private 0
if (best-score < best-score-of fellow) ;; better than other (ie. closer to the target)
[
  set best-max-moves-private best-max-moves ;; save own knowledge of own best move, since we're going to move suboptimally for teaching purposes; assumes perfect memory, which is of course questionable (JAL)
  ifelse (best-max-moves < best-max-moves-of fellow)
  [
    set best-max-moves max (list best-max-moves ((best-max-moves-of fellow) - ZPD)) ;; target is to the left
  ]
  [
    set best-max-moves min (list best-max-moves ((best-max-moves-of fellow) ± ZPD)) ;; target is to the right
  ]
]
end

```

to v-adjust ;; original Abrahamson & Wilensky v-adjust (JAL)

```

let fellow nobody
while [ fellow = nobody or fellow = self ] [ set fellow turtle (who + ( - (#-Vygotskian-neighbors / 2) ) + random (1 + #-Vygotskian-neighbors) ) ]
;; look randomly to one of your neighbors

;; if the score is better and it is within your ZPD, use their max-moves.
ifelse (best-score > best-score-of fellow) and (best-score - ZPD <= best-score-of fellow)
[
  set best-score best-score-of fellow
  set best-max-moves best-max-moves-of fellow
]
[
  set best-score score
  set best-max-moves max-moves
]
end

```

to pv-adjust-t ;; modified pv-adjust with added Vygotskian "teacher" effect (JAL)

if (abs((max-pxcor / 2) - best-max-moves) > abs((max-pxcor / 2) - best-max-moves-private)) [set best-max-moves best-max-moves-private] ;; restore best knowledge (JAL)

```

let fellow nobody
while [ fellow = nobody or fellow = self ] [ set fellow turtle (who + ( - (#-Vygotskian-neighbors / 2) ) + random (1 + #-Vygotskian-neighbors) ) ]
;; look randomly to one of your neighbors

```

;; maximize your own score and...

```

if ( score < best-score )
[
  set best-score score
  set best-max-moves max-moves
]

```

;; check it against your neighbor's score

```

ifelse (best-score >= best-score-of fellow) and (best-score - ZPD <= best-score-of fellow)

```

```

[
  set best-score best-score-of fellow
  set best-max-moves best-max-moves-of fellow
]

```

```

[

```

```

[

```

```

  set best-score score ;; present in v-adjust but not in pv-adjust? (JAL)

```

```

  set best-max-moves max-moves ;; present in v-adjust but not in pv-adjust? (JAL)

```

;; below is the new code that attempts to model the "teacher's" awareness of the "student" (JAL)

set best-max-moves-private 0

if (best-score < best-score-of fellow) ;; better than other (ie. closer to the target)

```

[

```

set best-max-moves-private best-max-moves ;; save own knowledge of own best move, since we're going to move suboptimally for teaching purposes, assumes perfect memory (JAL)

```

  ifelse (best-max-moves <= best-max-moves-of fellow)

```

```

  [

```

```

    set best-max-moves max (list best-max-moves ((best-max-moves-of fellow) - ZPD )) ; target is to the left

```

```

  ]

```

```

]

```

```

set best-max-moves min (list best-max-moves ((best-max-moves-of fellow) + ZPD )) ; target is to the right
  1
  1
  1
end

to pv-adjust ;; original pv-adjust from Abrahamson & Wilensky (JAL)
  let fellow nobody
  while [ fellow = nobody or fellow = self ] [ set fellow turtle (who + ( - (#-Vygotskiiian-neighbors / 2) ) + random (1 +
  #-Vygotskiiian-neighbors ) ) ]
  ;; look randomly to one of your neighbors

  ;; maximize your own score and...
  if ( score < best-score )
  [
    set best-score score
    set best-max-moves max-moves
  ]

  ;; check it against your neighbor's score
  if (best-score >= best-score-of fellow) and (best-score - ZPD <= best-score-of fellow)
  [
    set best-score best-score-of fellow
    set best-max-moves best-max-moves-of fellow
  ]
end

to r-adjust
  ;; random strategy changes max-moves to a random number x if it's not at the wall
  ;; where 0 < x < max-dist
  ;; if it is at the target, it stops changing.
  ifelse ( (abs pxcor) >= 0 )
  [
    set best-max-moves ( random max-dist )
  ]
  [
    set best-max-moves ( max-dist / 2 ) - 1
  ]
end

```