What dimensions do students notice through computational modeling and data analysis?: An investigation using [Anonymous]

Abstract: This paper draws on a larger project in which we design for students to iteratively engage in scientific practices of computational modeling and data analysis. Here, we report on two sixth grade science classes' work in a unit about how ink diffuses through hot and cold water. Using interaction analysis, we analyzed what dimensions students attended to as they analyzed data, constructed computational models, and compared the two to validate their models. Our analysis led to three findings: 1. Visual cues from video data were salient to students who heavily drew on them to iterate on their models.; 2. Programming computational models raised questions about the behavior of the individual particles in the phenomenon.; and, 3. The visual data made salient the contrasting conditions being modeled. However, instead of developing a single model that explained diffusion in both hot and cold water, students programmed distinct behaviors for each condition. The findings illustrate how visual data and modeling together can help students generate explanations to account for scientific phenomena, and show evidence that students need explicit supports for thinking about models as providing an explanation for a range of related conditions in the system.

Introduction: Subject & Problem

Scientists go back and forth between building theories using models and collecting and analyzing data to investigate phenomena (MacLeod & Nersessian, 2013). For students in science classrooms, modeling offers opportunities to engage in theory building to account for scientific phenomena. Exploring real-world data holds students accountable for their claims regarding scientific phenomena.

Domain-specific modeling has emerged as a generative tool to engage students in computational modeling (e.g., Wilkerson, Wagh & Wilensky, 2015). The domain-specific nature of programming can align with students' existing ways of thinking and provide a language to articulate their scientific ideas. It also provides accessible entry points and supports students' developing conceptual understanding and mechanistic reasoning (Wagh & Wilensky, 2018). In particular, it helps students reason about the properties and behaviors of individual entities in the phenomenon (e.g., Krist et al., 2019).

Researchers have also integrated computational modeling with real-world data in order to highlight — rather than dismiss — discrepancies between models and data, bringing noise, uncertainty, and intrinsic differences between them to the forefront (Blikstein et al., 2016). Other research efforts introduce uncertainty into student investigations through the use of experimental data to create resistances for students to consider, tackle, and possibly resolve (Gouvea & Wagh, 2018). Tackling such uncertainties together with the use of data is intended to amplify disciplinary ideas while supporting progress along a modeling trajectory.

The current study is part of a larger project in which we integrate computational modeling with data analysis to support students' sense-making of scientific phenomena (DRL-*XXX*). In this paper, we investigate the specific dimensions that students attend to and act on as they navigate between computational modeling and visual real-world data to explain a scientific phenomenon. The computational environment we use, [Anonymous], is a web-based domain-specific modeling environment that juxtaposes student models with real-world data analysis for comparison and validation (Authors, 2022a; Authors, 2022b).

We are guided by the following research question: What dimensions do students orient to as they navigate back and forth between real-world video data and computational modeling?

Methods

Design of the unit

[Anonymous] is a web-based environment that juxtaposes computational modeling with real-world data for comparison and validation. The environment includes a modeling area and a real-world data area (Figure 1). The modeling area consists of a coding workspace where learners program their models using a library of domain-specific blocks. This is a custom block-based coding tool built on Google's Blockly library. It also includes a simulation workspace, generated using the NetLogo engine (Wilensky, 1999), in which students can run and manipulate the model they built in the coding workspace. On the far right of the environment, the real-world data area displays visual data such as images or videos. In this study, we implemented a unit on diffusion called *How Ink Spreads in Water* (Authors, 2022b).

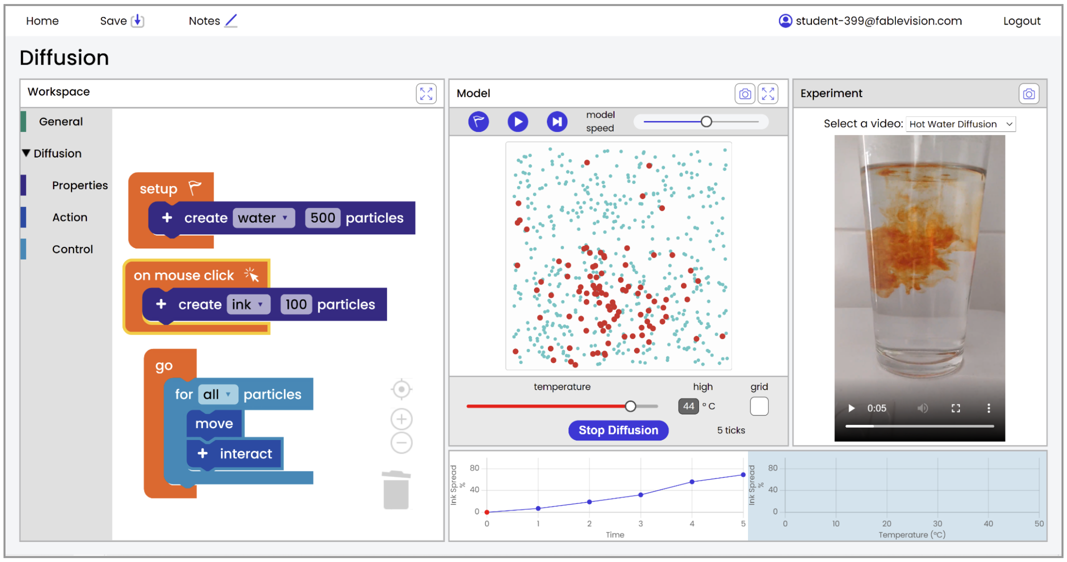


Figure 1: Screenshot from the *How Ink Spreads in Water* unit

This diffusion unit took place over eight class periods. Students first conducted an experiment to compare the rate of ink spread in hot and cold water. They ran the experiment three times and recorded their data. They then developed a paper model to explain the difference in the rate of spread in the two conditions. Students were then introduced to [Anonymous] in which they programmed computational models to explain their observations. Students shared their paper and computational models with the class to get feedback from their peers and teacher. On the last day, students discussed the validity of their model and watched a video that explained diffusion.

Participants and setting

The study was conducted at a private school in California. The unit was implemented in two class sections, consisting of 10 and 8 students. Of those, a total of 16 students consented to participate in the study.

Data collection & analysis

Throughout the 8-day-long unit, two researchers were present in class to support the teacher and students as well as take field notes and collect data. One researcher took detailed field notes in each class session. A pre/post test was administered before and after the unit, and 5 students were interviewed after the unit. A total of four pairs were video recorded through Zoom as they engaged in small group work with programming their computer models and analyzing data. Any whole class share-outs and discussions were also video recorded.

In the initial rounds of analysis, we analyzed field notes and Zoom recordings of 3 pairs. This involved content logging the video- and screen-recordings to document student activity and thinking. To study the dimensions students attended to in their data and modeling activities, we focus our analysis in this paper on the work of one student pair during the last two days of the unit when they had access to both the modeling and the data through [Anonymous]. Video data of this pair was transcribed and analyzed using interaction analysis (Jordan & Henderson, 1995). Two researchers collectively analyzed the transcript to identify salient themes in the data as well as the dimensions that were being inspected by the students.

Findings

Our analysis led to three themes: #1. Visual cues from the video data were salient to students who heavily drew on them to iterate on their models. These moves could be productive or not in terms of progressing towards an explanation of the phenomenon; #2. Programming and refining their computational models raised questions about behaviors of the individual particles in the phenomenon.; and, #3. The two videos made salient the contrasting conditions that students were modeling for. However, instead of developing a single model that explained both conditions, students programmed distinct behaviors for each condition. We exemplify these themes using three chronologically sequenced episodes from A and O's last two days on the unit.

<u>Theme #1: Orienting to visual cues from the video data</u>

This episode comes from A and O's collaborative [Anonymous] work on Day 6. A asks to watch the video for hot water diffusion and O, who is the driver, plays the video.

1 A: *(Plays video for hot water)* I want to see does it just goes straight down or does it like move. It sort of goes side to side, but how do we like show that?

2 O: Ink, ink, ink, ink. *(O points to ink drops in the simulation)*

3 A: Because right now they are just falling straight down

4 A: Hey, do you want to do like hot water moves faster, cold water moves slower? Ok, I think you do "ask ink particles". *(O moves cursor over to the coding area and moves blocks)*

In this episode, A attends to two aspects of the movement of the ink in the data: its path (line 1, 3) and that the ink spreads quickly in the hot water leading her to conclude that the water moves faster (line 4). Going forward, they begin to add behaviors to their model code for the ink particles to follow as seen in the following excerpt.

Theme #2: Orienting to individual level behaviors when developing their computational model

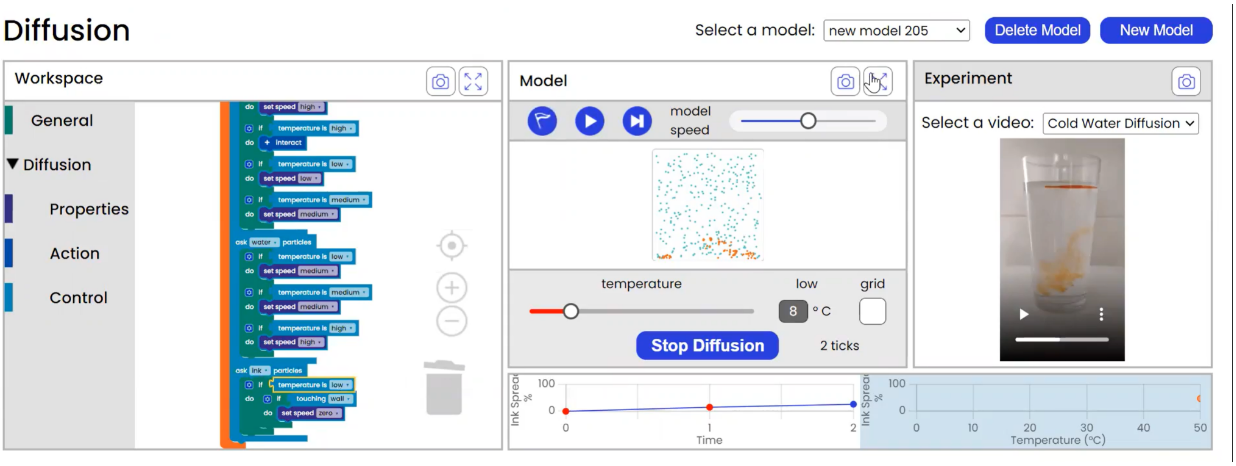
As A and O program behaviors for ink particles, they decide to build out their initial explanation for this phenomenon: that particles attach to each other, causing the ink to spread. A and O drag in an "attach" block in their code, but it freezes the model. They seek help from a researcher. In their conversation with the researchers, some questions that arise include: *When* do the ink particles attach? *What* do they attach to - other ink particles, water particles, or both?

When they finally have a revised model, A and O reset their simulation and run the model. O drops ink particles into the model, and they watch the simulation closely:

5 A: Are they attaching to water particles? How do you tell if they are… do you want to make this bigger? *(O expands the simulation workspace)*

6 A: Oh yeah they are attaching! Look, there's two attached.

In the episode above, A and O want to verify that the simulation actually follows the code they have entered into their model (line 5). They run their model, make the simulation workspace larger so they can observe more closely and follow the particles to verify that they attach as encoded in the program code (line 6). Developing the code for their computer model and then watching the resulting simulation run helps A and O attend to the behaviors of individual ink and water particles in the modeling environment. In other words, building out their models leads students to reason about how individual entities in this system behave.



```
    do  set speed high
    if  temperature is high
    do  + interact
    if  temperature is low
    do  set speed low
    if  temperature is medium
    do  set speed medium
ask water particles
    if  temperature is low
    do  set speed medium
    if  temperature is medium
    do  set speed medium
    if  temperature is high
    do  set speed high
ask ink particles
    if  temperature is low
    do  if  touching wall
        do  set speed zero
```

Figure 2: The final location of ink particles in the cold water video

Theme #3: Specifying distinct code blocks for each condition

Later in the same class period, the teacher stopped by their table. She listened to A and O describe their model and compare it with the hot water condition. The teacher then asked, "What happens in cold water?," A smiled and did not reply. In the conversation that followed, the teacher asked them four times about the cold water condition before they actually played the video for cold water. On playing the video, they notice that the ink particles "go down to the bottom and bounce back up." This outcome, the final location of the ink particles through spread, is different from in the hot water condition (Figure 2). A and O spend much of the rest of the class time and time in the next class period trying to replicate this positioning using the model they had developed for the hot water condition.

The episode below comes from closer to the end of the class period the following day:

12 A: Ok, so I will do "if touching the wall", hmm… set speed to zero?
13 Researcher: Sure!
14 A: And then put an "ask ink particles" on the top of it?
15 Researcher: Uh-huh.
16 A: Ok, thank you! *(Researcher leaves)*
17 A: I'm pretending I know what I'm doing, but I don't know
18 *(A drags the ask particles into Go block, and then adds an "if temperature is low" above)*
19 A: Oh I know what to do… if temperature is low, if touching any wall.

In this episode, A and O try to replicate the location of the ink particles in the video of the cold water condition. They drag in a "if touching the wall" conditional into their code and set the speed to zero for the ink particles (line 12). Any ink particle that touches the wall (i.e., the bottom of the simulation) would stop moving. A then specifies that this conditional behavior be followed only in the cold water (lines 18, 19: "if temperature is low, if touching any wall"). This encoded mechanism would be applicable only to the cold water condition.

To summarize, we presented three themes for how students oriented to the data and modeling in this unit. Of these three themes, themes #2 and #3 were seen in many other student pairs as well. Several students oriented to particle behaviors primarily when using the computational modeling environment. This was important because it helped students reason about what individual entities in the system are doing - a key element of developing mechanistic explanations (e.g., Krist et al., 2019). Moreover, many students developed separate models for the hot and cold water conditions. However, theme #1 varied across pairs in how students chose to draw on the data. For instance, some students, like A and O, oriented to its path while others attended to the difference in the time it took to spread out in the two conditions.

Contributions to the teaching & learning of science

This paper reported on what dimensions students attended to when engaging in a curricular unit that integrated computational modeling with data analysis. Based on our analysis, we described three key findings. First, students attended to the visual cues such as location and path of particles in the video data to refine their models. Second, developing their computational model led students to reason through specifics about the behaviors of particles in the system. Finally, it was not straightforward for students to develop one model that explained both conditions. Students tended to construct independent models for the two conditions in the system. The findings reported in this paper contribute to our understanding of science learning in two ways. First, it shows how visual data and modeling together can help students generate explanations to account for scientific phenomena. Second, it suggests that students need explicit supports for thinking about models as providing an explanation for a range of related conditions in the system.

References

Authors, 2022a; XXX

Authors, 2022b; XXX

Blikstein, P., Fuhrmann, T., & Salehi, S. (2016). Using the Bifocal Modeling Framework to Resolve "Discrepant Events" Between Physical Experiments and Virtual Models in Biology. *Journal of Science Education and Technology*, 25(4), 513-526. https://doi.org/10.1007/s10956-016-9623-7

Fuhrmann, T., Schneider, B., & Blikstein, P. (2018). Should students design or interact with models? Using the Bifocal Modeling Framework to investigate model construction in high school science. *International Journal of Science Education*, 40(8), 867-893. https://doi.org/10.1080/09500693.2018.1453175

Jordan, B., & Henderson, A. (1995). Interaction analysis: Foundations and practice. *The journal of the learning sciences*, *4*(1), 39-103. https://doi.org/10.1207/s15327809jls0401_2

Gouvea, J., & Wagh, A. (2018). Exploring the unknown: Supporting students' navigation of scientific uncertainty with coupled methodologies. In Proceedings of the 13th International Conference of the Learning Sciences (pp. 33–40).

Krist, C., Schwarz, C. V., & Reiser, B. J. (2019). Identifying essential epistemic heuristics for guiding mechanistic reasoning in science learning. *Journal of the Learning Sciences*, 28(2), 160-205. https://doi.org/10.1080/10508406.2018.1510404

MacLeod, M., & Nersessian, N. J. (2013). Coupling simulation and experiment: The bimodal strategy in integrative systems biology. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, *44*(4, Part A), 572–584. https://doi.org/10.1016/j.shpsc.2013.07.001

Wagh, A., & Wilensky, U. (2018). EvoBuild: A Quickstart Toolkit for Programming Agent-Based Models of Evolutionary Processes. *Journal of Science Education and Technology*, *27*(2), 131–146. https://doi.org/10.1007/s10956-017-9713-1

Wilensky, U. (1999). *NetLogo*. Evanston, IL: Center for connected learning and computer-based modeling, Northwestern University.

Wilkerson-Jerde, M. H., Wagh, A., & Wilensky, U. (2015). Balancing curricular and pedagogical needs in computational construction kits: Lessons from the DeltaTick project. *Science Education,* 99(3), 465-499. https://doi.org/10.1002/sce.21157