“Can Molecules Change their Color?” Exploring Students' Non-Canonical Ideas while Programming a Model of Diffusion

Tamar Fuhrmann, Teachers College, Columbia University, tf2464@tc.columbia.edu
Cassia Fernandez, Escola Politécnica, Universidade de São Paulo, cassia.fernandez@usp.br
Paulo Blikstein, Teachers College, Columbia University, pb2755@tc.columbia.edu
Roseli de Deus Lopes, Escola Politécnica, Universidade de São Paulo, roseli.lopes@usp.br

Abstract: This study investigates students' non-canonical ideas while creating computational models. Four 5th-grade students individually observed a diffusion experiment and created a model to explain the experiment using domain-specific programming blocks. Results illustrate that translating ideas into code reveals various non-canonical ideas. We argue that a limited number of blocks does not interfere with students expressing their ideas. However, creating domain-specific blocks for modeling needs to be a thoughtful and carefully designed process.

Introduction

Designing computer models is a promising approach to science learning. It combines the advantages of traditional modeling with computational literacy, opening new possibilities for inquiry-based learning (Weintrop et al., 2017; Wilkerson, 2015). In the past decade, many new environments have been designed to allow kids to create their own models using block-based (as opposed to script-based) programming languages and other innovative user interfaces such as NetTango (Horn et al., 2014), Deltatick (Wilkerson et al. 2015), StarLogo Nova (Klopfer et al., 2009), ViMap (Sengupta et al., 2015), or the phenomenological Gas Particle Sandbox (Aslan et al., 2020). Domain-specific block-based programming environments often use a small library of blocks based on canonical scientific ideas, thereby allowing students to focus on a particular aspect of a phenomenon. In science classes, these modeling environments are often used to confirm a theory rather than as an inquiry tool, so students don't have the opportunity to use models to test and explore their own hypotheses. Students are given a final, canonical model that scientists have developed over numerous years, and little time is spent showing them the evidence for the model or allowing them to construct models by themselves (Krajcik et al. 2012). In some cases, students are asked to use or manipulate a model from existing blocks/elements that present a “canonical" scientific explanation of a phenomenon. In doing that, students may not understand that scientific inquiry is necessary to design models and that model building is an iterative process that builds on the results of inquiries verified by data. With this context in mind, we designed domain-specific blocks of the scientific phenomenon of diffusion as an extension to the visual programming environment Scratch (Fernandez et al., 2021).

Diffusion is a fundamental concept in many fields, including chemistry, biology, and physics (Friedler, Amir & Tamir 1987). However, during the last 30 years, several studies have shown that it is extremely difficult for students to master the concept of diffusion (Odom, 1995). In this paper, the term "idea" is used to refer to the understanding that students have developed about the phenomena of diffusion of ink in hot and cold water. For the purposes of this paper, we define "canonical ideas" as science ideas that are typically found in textbooks, and "non-canonical ideas" as explanations for scientific phenomena that are not commonly used in textbooks or in traditional lab activities; this includes students’ initial ideas that are different from normative scientific concepts. In this sense, designing a model with domain-specific blocks can be a valuable way to disclose students' non-canonical ideas about diffusion while students are still in the midst of an inquiry activity. In this paper we present students' non-canonical ideas on the topic of diffusion while they design a model using a block-based computer modeling environment. These alternative ideas can be a starting point for teachers, curriculum designers and designers of educational environments to design activities and environments that allow students to test and develop their ideas. Our paper asks: What non-canonical ideas do students have about diffusion? How do they express these ideas while designing models using a computerized modeling environment?

Methods

The Scratch scientific diffusion model using domain-specific blocks

We created nine domain-specific blocks related to diffusion as a Scratch extension for younger audiences who lack programming knowledge (Fernandez et al., 2021). Each block was designed to embed a set of commands that perform a specific key procedure related to diffusion. The idea was that, rather than spending time dealing with the complexity of the code, students could focus on testing and refining their ideas about the mechanics of diffusion while designing a model. While we felt it was important to have specific blocks that allowed learners to engage easily with the big ideas related to diffusion we defined, we also wanted learners to be able to represent and test their own ideas about the phenomenon rather than being restricted to only the scientific ideas we had previously considered.

Participants, settings, and instructional sequence

The first two authors recruited seven US 5th-grade students (10-11 years old), four boys and three girls, to participate in this study. Five out of seven students (71%) were familiar with Scratch before the study. Each student participated individually in a one-hour Zoom call with the authors, wherein the authors instructed the diffusion session for the present study. We analyzed data from the four students (two boys and two girls) who had the richest sessions in terms of detailing their thoughts. The hour-long online session was based on the Bifocal Modeling approach (Blikstein, 2014; Fuhrmann et al., 2018) and was split into four "mini activities" followed by a short reflection interview. In the activities, students: (1) watched two diffusion videos with food coloring in water at different temperatures, described what they observed, and created a hypothesis to explain the phenomenon observed; (2) were presented with several examples of models and were asked to define and explain the function of a scientific model; (3) were introduced to the Scratch extension and the domain-specific blocks and explored the environment in an open-ended way; (4) were asked to work on three models (of cold water particles, of ink spreading in water, and of ink spreading in cold and hot water) that scaffolded them to design their final model of diffusion.

Data sources and analysis

Data included four hours of video recording. Students and parents signed consent forms for the Zoom sessions to be recorded. Videos were transcribed and analyzed with a focus on students' ideas regarding the process of diffusion that were expressed and captured during their programming session. We scanned the transcripts and selected the non-canonical explanations regarding diffusion.

Results and Discussion

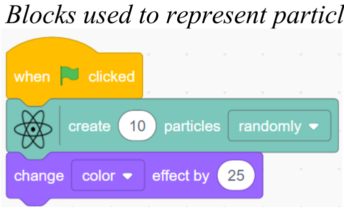
We illustrate five different non-canonical ideas expressed by students while designing a model for diffusion using the Scratch Scientific Modeling Extension. For each idea, we present examples of how students stated it and how (if at all) they expressed it with blocks from the environment. All names are pseudonyms.

1. Water particles change their color when touching ink particles

Another non-canonical idea expressed by one student was that the phenomenon of diffusion can be explained by the propensity of water particles to change their color when touching an ink particle. Alice was designing her model of diffusion in hot water when she was looking for a new block that would make the color of a particle change, explaining "*I am trying to change the color of the particles. I think that since the color is dispersed...this will be the change in hot water.*" Her hypothesis was that when an ink particle touches a water particle, it "infects" it with its color, so the water changes color to become the same color as the ink. However, she could not translate her idea into a working code, and thus she couldn't test her hypothesis (Figure 1).

Figure 1

Blocks used to represent particles changing their color when touching



2. Ink and water particles move at different speeds

One student, Alice, expressed the idea that ink and water particles do not move at the same speed when mixed in one receptacle, saying "*I think that water will be a little slower than the food coloring.*" To represent this idea in the model, she used the "set speed" block to set different speeds for the two particles (ink particles with high speed, and water particles at medium speed). When she ran the model and watched the results, she changed her mind regarding this explanation, saying that "*Actually, they will be the same speed because I think that they [food coloring particles] are mixing with the water [particles].*" She then adapted her model, setting the speed of all the particles to high and concluding "*yes...they are all the same speed and mixing together.*" In this case, the existence of a block to change the particles' speed enabled Alice to test her ideas using the model. Thus, she shifted from a non-canonical idea about ink and water particles moving at different speeds to the canonical idea that both particles move at the same speed, even without explicit instruction from an external source.

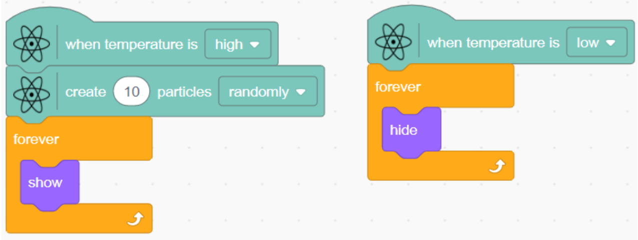
3. Temperature is related to the "weight" of the particles

Introducing the variable of weight to the system, Tonia thought that the particles in hot water were lighter than those in cold water and floated on the surface: "*I guess the hot water particles are maybe lighter and they float to the surface faster.*" She mentioned that the behavior observed and the density of the particles is related to the bubbles in the hot water: "*I don't know, but this relates in my mind to the bubbles... I don't know why.* " Tonia expressed these ideas verbally without translating them into a model.

4. Temperature is related to a particle's appearance and disappearance

This non-canonical idea was that in high temperatures more particles are appearing, and that they disappear as the temperature gets lower. When designing his model of cold and hot water, Owen related the behavior of particles at different temperatures to their appearance or disappearance. When researchers asked him to further explain his idea, he only said, "*What I want to do is this: when the temperature is high my costume shows and when the temperature is low my costume does not show.*" However, he faced challenges while coding his model and could not test his ideas since he was not able to create a working code (Figure 2).

Figure 2
Blocks used to represent particles appearing and disappearing



5. Temperature is related to the number of particles

Three students expressed the non-canonical idea that hot water and cold water contain different numbers of particles, and that that explains why ink behaves differently in the two temperatures.

Alice was designing the diffusion of ink in cold water when she suggested that "cold water particles move slower and there are less of them." When asked to explain her idea, she stated that "in cold water, they are all bunched up. In hot water, they are all mixing and stuff. In the hot water there are more particles because they disperse more." To adapt her model to express this idea, she said: "For the cold water, I will change the speed and make less amounts of particles." and changed the number of particles in her model from 30 to 20.

Mark expressed this same idea, relating it to a "chaotic movement". While designing his model, he reasoned about ways to make it more similar to the experiment and said, "I would change the water molecules to be really, really chaotic. And I would change the food coloring molecules to be about as chaotic as the water molecules." To make the particles more "chaotic," he started by changing the number of particles: "To make it more hectic I can make this one [ink] 20 and this one [water] 35." After changing the number of particles in the code and observing its effects, he confirmed his idea that more particles make the water movement more chaotic: "Yes, that's way more hectic. (...) If I wanted it to be less hectic, this [ink] would be 10 and this [water] would be 15." In this case, his non-canonical idea resulted in a model that behaved similarly to the experiment.

Tonia's initial thought was also that hot water had more particles than cold water. While designing her model for hot water, she stated that "In cold water I guess I will make less particles. My hypothesis is that cold water has less particles than hot water." When creating the hot water model, she said: "[In hot water] I guess there are more particles because it is hotter, but I don't know... The cold water does not have so many particles." So, she started by creating 100 water molecules and then changed it to 500 to represent hot water.

Conclusions

This study presents examples of non-canonical ideas expressed by 5th-grade students while designing a model of diffusion using a block-based programming tool. These non-canonical ideas were not imagined by the designers of the original tool. These ideas are related to phenomenological primitives, or p-prims (diSessa, 1993; Hammer, 1996) defined as relatively minimal abstractions of simple common phenomena, often hard to access and not easily put into words. Often, the p-prims, these self-contained explanations based on students' intuitive background knowledge, are non-canonical ideas. In science learning, it is important to offer students opportunities to explore these ideas, using them as epistemological resources to get to the accepted explanations (Hammer & Elby, 2003). However, designers are reluctant to present them in their environment, and teachers are concerned about using them to guide instruction. We argue that in order to incorporate these non-canonical ideas into science instruction and environments, it is essential to identify them for different reasons. Being aware of students' prior knowledge and identifying possible sources of confusion allow science educators to mediate the scientific knowledge-building process for learners; help them make sense of how scientific models are generated and validated; and scaffold learners' conceptual understandings rather than merely giving them the "correct answer." While learning how students' reason about a phenomenon, designers and researchers can create more domain-specific blocks that acknowledge students' initial ideas and sensemaking processes. As in Alice's reasoning about particles' speed, equipping students to model their non-canonical ideas can lead to their independent development toward disciplinary norms. Also, for designers, these ideas can guide the development of tools that acknowledge students' non-canonical ideas as a way to scaffold their sensemaking process based on their own hypotheses about the phenomenon. Although this study included only four students, future work could determine the degree to which these non-canonical ideas are shared among learners, helping prioritize their inclusion in block-based modeling environments.

References

Aslan, U., Lagrassa, N., Horn, M., & Wilensky, U. (2020). Phenomenological Programming : A Novel Approach to Designing Domain Specific Programming Environments for Science Learning. In *IDC'20*.

Blikstein, P. (2014). Bifocal modeling: promoting authentic scientific inquiry through exploring and comparing real and ideal systems linked in real-time. In *Playful user interfaces* (pp. 317-352). Springer, Singapore.

Fernandez, C., Fuhrmann, T., de Deus Lopes, R., & Blikstein, P. (2021). Designing domain-specific blocks for diffusion: The dialogue between pedagogical principles and design decisions. In *Interaction Design and Children, IDC 2021* (pp. 461-465).

Friedler, Y., Amir, R., & Tamir, P. (1987). High school students' difficulties in understanding osmosis. *International Journal of Science Education*, 9(5), 541-551.

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

Hammer, D., & Elby, A. (2003). Tapping epistemological resources for learning physics. The Journal of the Learning Sciences, 12(1), 53-90.

Hammer, D. (1996). Misconceptions or p-prims: How may alternative perspectives of cognitive structure influence instructional perceptions and intentions. The journal of the learning sciences, 5(2), 97-127.

Horn, M. S., Brady, C., Hjorth, A., Wagh, A., & Wilensky, U. (2014, June). Frog pond: a code first learning environment on evolution and natural selection. In Proceedings of the 2014 conference on Interaction design and children (pp. 357-360).

Klopfer, Eric, Hal Scheintaub, Wendy Huang, Daniel Wendel, and Ricarose Roque. "The simulation cycle: Combining games, simulations, engineering and science using StarLogo TNG." *E-Learning and Digital Media* 6, no. 1 (2009): 71-96.

Krajcik, Joseph, and Joi Merritt. "Engaging Students in Scientific Practices: What Does Constructing and Revising Models Look Like in the Science Classroom?: Understanding A Framework for K-12 Science Education." The Science teacher (National Science Teachers Association) 79.3 (2012): 38–41.

Odom, A.L. (1995). Secondary and college biology students' misconceptions about diffusion and osmosis. American Biology Teacher, 57(7), 409–15.

Sengupta, P., Dickes, A., Farris, A. V., Karan, A., Martin, D., & Wright, M. (2015). Programming in K-12 science classrooms. *Communications of the ACM*, *58*(11), 33-35.

diSessa, A. (1993). Toward an epistemology of physics. *Cognition and Instruction*, 10(2-3), 105– 225.

Weintrop, D., & Wilensky, U. (2017). Comparing block-based and text-based programming in high school computer science classrooms. ACM Transactions on Computing Education (TOCE), 18(1), 1-25.

Wilkerson-Jerade, M., 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.

