International Society of the Learning Sciences

Opening the Black Box: Student Motivations in an Unpackable-Block-Based Computational Modeling Environment

Adelmo Eloy, University of São Paulo, Brazil and Transformative Learning Technologies Lab, Columbia University, USA/Brazil, adelmo.eloy@usp.br
Tamar Fuhrmann, Columbia University, USA, tamarrf@gmail.com
Roseli de Deus Lopes, University of Sao Paulo, Brazil, roseli.lopes@usp.br
Paulo Blikstein, Columbia University, USA, and I. Tecnológico de Monterrey, Mexico, paulob@tc.columbia.edu
Aditi Wagh, Massachusetts Institute of Technology, USA, awagh@mit.edu

Abstract: Engaging with computational models is central to both scientific and computational learning. A promising approach to "lower the floor" and make computational modeling more accessible is the development of domain-specific and block-based environments, which reduce programming complexity while leveraging students' intuitions about scientific ideas. To balance usability and expressiveness in these environments, we develop the feature of "unpacking" blocks, allowing users to open and modify high-level blocks into the simpler constituent elements that define them. In this study, we analyze high school students' models, screen recordings, and artifact-based interviews to investigate their motivation for modifying domain-specific blocks for eutrophication in aquatic ecosystems. We found that unpacking and modifying blocks supported students in both exploring scientific ideas and addressing specific goals of computational modeling, providing insights on how unpacking domain-specific blocks can support both computing and science learning.

Introduction

Engaging with computational models is a central practice that benefits both scientific and computational learning (e.g., Sengupta et al., 2013; Weintrop et al., 2016). However, balancing usability and expressiveness in domain-specific, block-based environments remains challenging (Repening & Amabach, 1997). To navigate this tension, we introduce the concept of *"unpacking"* domain-specific, high-level blocks into more general-purpose computational elements. This approach enables learners to read and interpret the underlying structure of a procedure. Once familiar with this internal logic, learners can modify the definition of a block for various purposes. Using a design-based research paradigm (Barab & Squire, 2004) to examine learners' engagement with unpackable blocks in a domain-specific, agent-based modeling environment, this paper investigates the following research question: *What motivates students to modify unpackable blocks as they represent and program scientific models in a domain-specific and block-based modeling environment?* We designed a curricular unit using MoDa (Fuhrmann et al., 2023; Wagh et al., 2022), a web-based computational modeling agent-based environment, to support students in modeling eutrophication, the process by which excess nutrients lead to algal blooms and a decrease in oxygen in aquatic systems. The unit included six "unpackable" blocks that students could modify as they worked through challenges to construct computational models of eutrophication. We analyzed data from a pilot implementation with students in Brazil, examining the various motivations behind students' modifications, highlighting how unpacking facilitated both the exploration of representations of scientific ideas and the achievement of specific goals through computational modeling modeling.

Background

We build on prior research on domain-specific modeling environments that facilitate engagement with specific domains by providing tailored primitives and syntax (e.g., Hutchins et al., 2020; Kahn, 2007; Wagh & Wilensky, 2018; Wilkerson et al., 2015). While those interfaces offer custom-designed primitives closely aligned with curricular topics (Aslan et al., 2020), they also have potential drawbacks, such as limiting engagement with more complex computing concepts by restricting usability for expressiveness (e.g., Repenning & Ambach, 1997). In this study, we introduce "unpacking" as a mid-level approach between high-level blocks and text-based languages. Unpacking enables students to enhance and modify selected domain-specific blocks, exposing their inner computational structure in a finer-grained yet still block-based format.

Methods

Our focal participants were five high school students from a public school in a suburban area of Southeast Brazil. In addition to their regular classes, they were enrolled in a vocational track in Informatics, similar to AP courses in computer science in the U.S. The participants had prior exposure to text-based programming languages but no experience with block-based programming. Students worked in two pairs and one independently across a four-class instructional unit to construct MoDa computational agent-based models using domain-specific blocks. Six key action and interaction blocks were unpackable: <breathe>, <create>, <decompose>, <die>, <move>, and <reproduce>. Data sources comprised students' final models, three-hour screen recordings, and artifact-based interviews (Haduong & Brennan, 2020) conducted one week after the last session. We analyzed all students' final models after each session, identifying 49 modifications in unpackable blocks. The first two authors then employed inductive coding (Saldaña, 2021) on the data from screen recordings and interviews to classify the motivations behind those modifications. Any discrepancies were discussed to establish a common interpretation. The findings section details these motivations with illustrative examples.

Findings

We provide a summary of the motivations provided by students in Table 1. The four cases (8%) in which students could not recall or articulate an explanation (e.g., "We do not recall") were not included. In addition, Figure 1 includes what those modifications look like at the block level for Motivations (1) (Figure 1.1) and (2) (Figure 1.2). An example for 'motivation' (3) was not included due to page constraints.

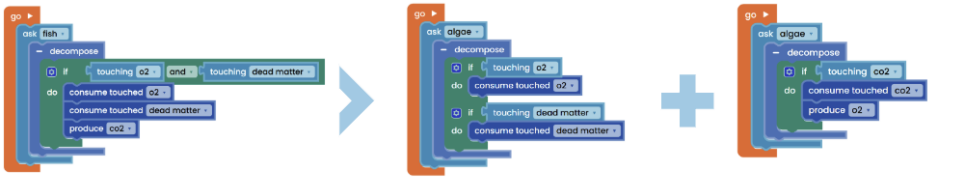
Table 1

Summary Information on Motivations Provided by Students

Motivation	Cases	Definition	Sample student explanation
(1) meeting the challenge	16 (33%)	Fulfill the goals set by the instructors	*"We were not using CO2, so we took it out. And we were testing a new format for the decompose."*
(2) exploring scientific ideas	26 (53%)	Represent scientific concepts related to the unit learning goals	*"Decomposition is just feeding organisms. I used it as a 'synonym' for feeding algae through nutrients. I put a higher value on energy because they are the primary producers of energy in a food chain."*
(3) Motivations (1) and (2) combined	3 (6%)	The two previous motivations combined	*"We followed a rationale of the food chain. So fish would reproduce less and require more energy to reproduce." [...] "Also, there were a lot of fish, so we want to reduce [the population]."*

Figure 1

Modified Versions of the <decompose> Block



Original <decompose> block.

(a) Modified <decompose>, with two independent if-conditions and no <produce CO2> block.

(b) Modified <decompose> block, with algae doing photosynthesis.

Discussion and implications

Our preliminary findings reveal that students' motivation for modifying unpackable blocks was significantly influenced by the instructional challenges presented in the unit and their efforts to explore and understand scientific concepts. This has led to a diverse range of computational models created by students. MoDa's block-based nature supports variability in students' models, while making key blocks unpackable adds an extra layer of expressiveness. This diversification enhances the variety of questions students can pose and representations they can create of the same phenomenon. In future work, we plan to investigate the complexity levels in block modifications to deepen our understanding of what students can achieve with unpacking and how this relates to their motivations. Additionally, considering our small sample size and the prior computer science backgrounds of students, we aim to expand the study to include a larger and more diverse student population. This will allow us to further explore how unpacking influences engagement with computational modeling in science education.

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 Proceedings of the Interaction Design and Children Conference (pp. 299-310).

Barab, S. A. , & Squire , K. (2004). Design- based research: Putting a stake in the ground. *The Journal of the Learning Sciences, 13*(1), 1-14.

Fuhrmann, T., Wagh, A., Rosenbaum, L. F., Eloy, A., Wilkerson, M., & Blikstein, P. (2023). How can computational modeling help students shift their ideas towards scientifically accurate explanations?. In Blikstein, P., Van Aalst, J., Kizito, R., & Brennan, K. (Eds.), *Proceedings of the 17th International Conference of the Learning Sciences - ICLS 2023* (pp. 441-448). International Society of the Learning Sciences.

Haduong, P. & Brennan, K. (2020). Talking in Pairs: Learning From and With Teachers Through Artifact-Based Dyadic Interviews. In Gresalfi, M. and Horn, I. S. (Eds.), *The Interdisciplinarity of the Learning Sciences, 14th International Conference of the Learning Sciences (ICLS) 2020, Volume 4* (pp. 2369-2370). Nashville, Tennessee: International Society of the Learning Sciences.

Hutchins, N. M., Biswas, G., Zhang, N., Snyder, C., Lédeczi, Á., & Maróti, M. (2020). Domain-specific modeling languages in computer-based learning environments: A systematic approach to support science learning through computational modeling. *International Journal of Artificial Intelligence in Education*, *30*, 537-580.

Kahn, K. (2007). Building computer models from small pieces. In *Proceedings of the 2007 Summer Computer Simulation Conference* (pp. 931-936).

Repenning, A., & Ambach, J. (1997). The agentsheets behavior exchange: Supporting social behavior processing. In *CHI'97 Extended Abstracts on Human Factors in Computing Systems* (pp. 26-27).

Saldaña, J. (2021). *The coding manual for qualitative researcher*s.Thousand Oaks, CA: Sage

Sengupta, P., Kinnebrew, J. S., Basu, S., Biswas, G., & Clark, D. (2013). Integrating computational thinking with K-12 science education using agent-based computation: A theoretical framework. *Education and Information Technologies, 18*, 351-380.

Wagh, A., Fuhrmann, T., Eloy, A. A. da S., Wolf, J., Bumbacher, E., Blikstein, P., & Wilkerson, M. (2022). MoDa: Designing a Tool to Interweave Computational Modeling with Real-world Data Analysis for Science Learning in Middle School. In *Proceedings of Interaction Design and Children*, 206–211.

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.

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of science education and technology, 25*, 127-147.

Wilkerson, 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.

Acknowledgments

This study is supported by funding from the National Science Foundation under Grant No. 2010413. This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001. We thank the partner teachers and focal students for engaging with this unit, and Lucas Coletti for software development.