Milo and J-Mole: Computers as Constructivist Teachable Agents

Blair, Kristen P. & Schwartz, Daniel L.
Stanford University School of Education, 485 Lasuen Mall, Stanford, CA 94305
kpilner@stanford.edu

Understanding mathematical symbols requires more than procedural fluency. It also involves understanding the communicative work that symbols need to accomplish (Greeno & Hall, 1997). Typically, students receive a formula and apply it narrowly. This practice subverts opportunities to consider whether and why the formula appropriately captures and communicates the situation at hand. For example, students often fail to understand when one model is preferable over another or why. One solution is to have students develop models that others will have to interpret (e.g., Kordaki & Potari, 1998). By evaluating the extent to which their model communicates the relevant properties, students may begin to appreciate the qualities that make a model more or less useful.

We have developed "Teachable Agent" (TA) software (Biswas et al., 2001) where students teach an agent and see how it performs as a result. In this work, we examined the potential benefits of teaching a "constructivist" agent that needs to interpret the messages it receives. We investigated the value of constructivist TA's for early learning in two mathematical domains. In the statistical application **Milo**, students explicitly try to teach the agent to recreate a distribution. In the multi-player application **J-Mole**, students try to discern which of several players is an agent based on their respective interpretations of communicated models.

With **Milo**, students receive a statistical distribution and create a code using an on-screen calculator (e.g., Code: Largest Value – Smallest Value = 9; N = 500). They send the code to Milo, who uses a genetic algorithm to generate distributions consistent with the code, and then Milo selects a subset to show his possible "interpretations." Students can overlay the original distribution against Milo's interpretations to see how well they communicated. If Milo's interpretations are too diverse, students modify their code (for example, by adding parameters or a shape assumption) until they are satisfied that Milo's interpretations are similar to the original distribution. We conducted a brief pre-post study with 39 high school seniors who used Milo for one class period. After using Milo, students significantly improved in their appreciation of how multiple parameters reduce ambiguity and increase specificity in a model. They also improved in their ability to recognize and construct less ambiguous models.

J-Mole involves three student teams at separate computers. One team is the "code-maker" and the others are "code-guessers." The code-makers receive a picture (or symbol). Using the symbols (or shapes) in their toolbar, they create a code. The other students receive the "code" and try to recreate what the original must have looked like. Once the guessers submit their interpretations, all teams see the original referent, all the guesses, and the submission from J-Mole. They know J-Mole is always wrong (his responses are seeded with common misconceptions), and they have to guess which of the four representations comes from J-Mole. The J-Mole application, used in the domain of early fractions, revealed positive benefits in a two-day study with 50 5th-graders separated into J-Mole and control conditions. On a posttest, the J-Mole students were more likely to anticipate how a code could be ambiguous and posit the misinterpretations people might make.

In both studies, creating models/representations that a computer agent interpreted appeared to help students appreciate the communicative aspect of representations. Clearly, these are small studies and more empirical work needs to be done, including the design of assessments that can evaluate students' abilities to appreciate underlying assumptions that determine the value of a symbolic model. Additionally, we think it is valuable to develop a general framework for creating constructivist teachable agents and to evaluate their usefulness.

References

Biswas, G., Schwartz, D. L., Bransford, J. D., & TAG-V. (2001). Technology support for complex problem solving: From SAD Environments to AI. In K. Forbus & P. Feltovich (Eds.), *Smart machines in education (pp. 71-98).* Menlo Park, CA: AAAI/MIT Press.
Greeno, J. G., & Hall, R. P. (1997). Practicing representation learning with and about representational forms. *Phi Delta Kappan*, *78*, 361-367.
Kordaki, M., & Potari, D., (1998). Childrens' approaches on area measurement through Different Contexts. *Journal of Mathematical Behavior,* 17, 303-316.