Learning Integrated STEM Using Tangible Agent-Based Modeling

Gokul Krishnan & Pratim Sengupta
Mind, Matter & Media Lab, Vanderbilt University – Peabody College
Email: gokul.krishnan@vanderbilt.edu, pratim.sengupta@vanderbilt.edu

Abstract: We investigate how the integration of visual agent-based programming and computationally augmented physical structures can support curricular integration across STEM domains for elementary and middle school students. We present ViMAP-Tangible, a learning environment, which integrates ultrasonic sensors with the ViMAP visual programming language using a distributed computation infrastructure, and report preliminary findings from studies conducted with 3rd - 5th grade students.

Introduction and Background

Integrating agent-based programming and modeling with computationally augmented physical structures can support students in authentic scientific inquiry and allow students to develop and investigate their own scientific hypotheses (Blikstein & Wilensky, 2007). In a bifocal modeling approach, students connect computational behavior in virtual simulations with phenomena detected by physical sensors or produced in the physical world by motors or other output devices. In this paper, we develop this line of research further by investigating how agent-based visual programming can be seamlessly integrated with physical computation in order to support both learners self-expression and learning of canonical concepts and representational practices (e.g. programming, mathematical reasoning, and scientific modeling). We present an Integrated STEM (Sanders & Wells, 2006) curriculum, in which learners develop engineering concepts and practices, which in turn leverage the natural connections between two or more STEM domains. To support such a learning environment we use a technological infrastructure called ViMAP-Tangible (Sengupta, Krishnan & Wright, 2014). The central engineering practice that learners develop in our curriculum is User-centered design (UCD) (Norman, 1998). In this poster, we investigate how our pedagogical approach supports the development of specific forms of representational practices that are central to learning engineering, science and math. In particular, we present an illustrative case of how the distributed computational infrastructure, and the emphasis on User-centered design supported students' learning.

The Learning Environment

Three key elements of ViMAP-Tangible are: (1) Computer Programming using ViMAP (Sengupta, Farris & Wright, 2012): Students generate algorithms by selecting commands from the library for each agent using a drag-and-drop interface in the ViMAP programming language; (2) Tangible and gestural representation of digital information: Students can use gestures or place physical structures in front the sensors to control different agent-variables (e.g., speed, color, position) within the ViMAP algorithm; (3) Simulation of agent behaviors: Students can dynamically visualize NetLogo (Wilensky, 1999) simulations of the behavior of each computational agent.

The learning activities consisted of three phases (Sengupta et al., 2014). During the first phase, students were introduced to agent based programming. During this phase, they learned to generate open and closed shapes (e.g. squares, circles, spirals) using ViMAP. In the second phase, students used these shapes as models of Newtonian mechanics (constant speed and acceleration). The third phase used an engineering design context in which students worked in dyads and were presented with a consequential design challenge. In both the studies, the goal during the third phase was for students to design a physical machine, linked to a ViMAP program, as well as accompanying user guides, using which *other* users – i.e., people besides the group members - would be able to accomplish specific tasks such as drawing geometric shapes using turtle graphics (see Figure 1). Initially, students were encouraged to designed machines so that they could be operated by meaningful, intuitive gestures in front of the sensors (i.e. raising and lowering of hand over the sensor would result in increase or decrease in readings). However, through user testing, they progressively developed more refined physical structures for greater reliability in the performance of their machines. For example, instead of using hand gestures directly in front of the distance sensor to generate a reading, students designed simple machines – such as Lego plates controlled by pulleys and wearable flat surfaces – in order to control the readings more reliably. These activities introduced students to User-Centered Design, which in turn encouraged and supported them to develop design thinking, and to be more articulate about the underlying rules of operation of their computational and physical artifacts.

The Studies & Preliminary Findings

We conducted two pilot studies in a classroom on the campus of a large private university in a metropolitan city the mid-southern USA. Each study was conducted in the form of an enrichment program for elementary and middle school children, and classes met once a week (9:00 am to 11:30am) on six consecutive Saturday mornings. None of the students in this course had any prior programming experience. In Study 1, there were 16 participants, out of which eight students were in 4th grade and eight students were in 5th grade. In Study 2, there were 16 participants, out of which eight students were in 3rd grade, and eight were in 4th grade. The data for the cases were in the form of in-depth interviews with the participants, student work (ViMAP programs, written user guides, and physical artifacts), and field notes. Here, we briefly present a case from Study 2 that also illustrates some of the themes we found in Study 1. A detailed analysis of student work can be found in Sengupta et al. (2014).

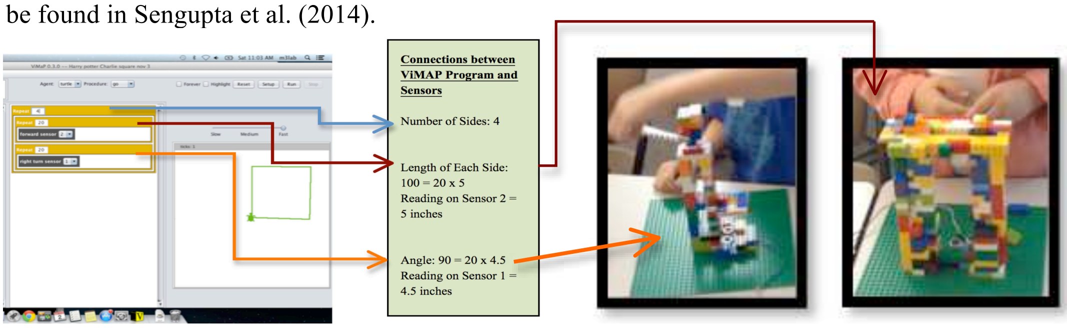

Figure 1. Chuck & Jerry's physical machines and their ViMAP program for generating a square

Across both the studies, we found that a) the distributed computing infrastructure resulted in creating positive interdependence (Antle & Wise, 2013) among students, as members of each group made coordinated, non-redundant contributions towards a common goal; and b) a focus on UCD resulted in students iteratively developing refined physical and computational representations towards their design objectives, which in turn resulted in deeper scientific and mathematical reasoning. For example, in Study 2, during Phase 3, a pair of students, Chuck and Jerry, built two separate machines, connected together by a ViMAP program (Figure 1). One machine consisted of a flat plate that could be lowered or raised above a distance sensor, using a pulley mechanism, in order to control the angular turn of the computational agent. The other machine also comprised of a flat plate, which could be lowered or raised above a distance sensor, using a manually operated crank lift, in order to control the speed of the computational agent. Both of these machines were communicating to the same ViMAP program (see Figure 1). In this program, students designed a nested loop in order to control the number of sides and angles, where the number of each loop was determined based on the relevant sensor reading, using multiplicative reasoning (explained in Figure 1). In addition to building these structures, the students created a set of rules that would provide the users instructions on how to use the physical setups to draw specific geometric shapes. As a result of user testing, these rules became more refined in order to facilitate better coordination users. The physical structures also improved in order to provide better control for users as they manipulated the flat plates. These observations suggest that UCD afforded two key benefits: first, UCD supported the development of design thinking; second, UCD grounded in an Integrated-STEM pedagogy supported learning through making and doing through an iterative process of designing, building, testing, revising and evaluating models.

References

Antle, A. N., & Wise, A.F. (2013). Getting down to details: Using learning theory to inform tangibles research and design for children. *Interacting with Computers*, 25, 1.

Blikstein, P., & Wilensky, U. (2007). *Bifocal modeling: a framework for combining computer modeling, robotics and real-world sensing.* In annual meeting of the American Educational Research Association (AERA 2007), Chicago, USA.

Norman, D. A. (1998). *The design of everyday things.* New York.

Sanders, M. & Wells, J.G. (2006). *Integrative STEM education*. Retrieved from www.soe.vt.edu/istemed/

Sengupta, P., Krishnan, G., & Wright, M. (2014). Integrated STEM in Elementary Grades Using Distributed Agent-based Computation. To appear in: *Proceedings of the Sixth International Conference on Computer Supported Education*.

Sengupta, P., Voss Farris, A., & Wright, M. (2012). From Agents to Aggregation via Aesthetics: Learning Mechanics with Visual Agent-based Computational Modeling. *Technology, Knowledge & Learning*, Vol. 17 (1 -2), pp. 23 - 42.

Wilensky, U. 1999. *NetLogo.* Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston, IL. URL: http://ccl.northwestern.edu/netlogo.