

## Designing to Improve Biology Understanding Through Complex Systems in High School Classrooms: No Simple Matter!

Susan Yoon, Joyce Wang, University of Pennsylvania, 3700 Walnut Street Philadelphia, PA 19104

Email: yoonsa@gse.upenn.edu, joycew@gse.upenn.edu

Eric Klopfer, Josh Sheldon, Daniel Wendel, Ilana Schoenfeld, Massachusetts Institute of Technology, 20 Ames Street, 02139

Email: klopfer@mit.edu, jsheldon@mit.edu, djwendel@mit.edu, ilanasch@mit.edu

Hal Scheintaub, Governor's Academy, 1 Elm Street Byfield, MA 01922

Email: HScheintaub@govsacademy.org

David Reider, Education Design INC, 7 Gibson Road, Newtonville, MA 02460

Email: david@educationdesign.biz

**Abstract:** This symposium seeks to illustrate and discuss the salient elements of project design and design decisions of a two-year implementation effort in high school science classrooms focused on improving knowledge and skills in biology content, complex systems, and computational thinking using StarLogo TNG simulations. We present design challenges that emerged in the areas of 1) Professional Development and Workshop Design; 2) Designing for Computational Thinking Through Computational Modeling; 3) Curriculum Design and Development; 4) Issues in the Design of Learning Progressions; and 5) Designing Assessments for Larger Scale. Through interactive discussion with the CSCL audience, we hope to share experiences that will enable similarly oriented design researchers to build successful programs in real-world educational systems.

### Symposium Focus

A core activity of learning science researchers is to design interventions to improve learning and implement those interventions in real-world educational environments. As Kolodner (2004) writes, “If we want to understand how learning happens in complex situations, then we should study learning as it is occurring in those environments—with all the messiness of the real world and requiring methodologies that can nonetheless extract trends and descriptions” (p. 6). To do this, we turn to design research, which has gained momentum in the learning sciences for several reasons: its emphasis on accommodating a wide range of variables that may have contextual importance (Confrey, 2006); the dynamic process of assessment and evaluation of system states toward the goal of higher levels of educational improvement (Reimann, 2010); and the grounded-in-practice nature of the research, which increases the likelihood that interventions will be successful and sustained in the real world (Bielaczyc & Collins, 2007). Although design research is popular as a methodology for iterative improvement, few studies have discussed how design trajectories unfold. As Puntambekar and Sandoval (2009) write, “The learning sciences could benefit from clear examples of research trajectories that explicate how microcycles of analysis inform macrocycles and how iterated macrocycles build new knowledge” (p. 325). In other words, we need better descriptions and theories about how design teams measure, make decisions about and redesign system variables to produce the desired system-wide outcomes.

This symposium seeks to illustrate and discuss the salient elements of project design and design decisions as they emerged and changed over a two-year time frame. We report on implementation activities of a large-scale US National Science Foundation project called *BioGraph: Graphical Programming for Constructing Complex Systems Understanding in Biology*. This project is relevant to the CSCL community in that our goal is to develop curricular and instructional strategies to help teachers and students improve knowledge of biology content through computational modeling tools, emphasis on complex systems concepts, and a hypothesized curricular learning progression. Below, we provide a brief review of educational literature that highlights the need for this research followed by the curriculum and instruction conceptual framework that underpins the project design and activities.

### Misconceptions in the Biological Sciences

Recent advances in the biological sciences that include the mapping of the human genome and the ability to manipulate atoms and molecules at the nanoscale have yielded unprecedented opportunities for humans to fashion our own evolutionary pathway. As Venter and Cohen (2004) write, “If the 20th century was the century of physics, the 21st century will be the century of biology” (p. 73). Yet, despite the enormous contemporary saliency, studies in science education have revealed robust misconceptions about concepts and processes in high school biology that directly impact students’ abilities to understand these recent advances. For example, student misunderstandings have been found across the scale of atoms (Taber & Garcia-Franco, 2010), to cells and genes (Garvin-Doxas & Klymkowsky, 2008), to organisms and ecology (Gotwals & Songer, 2010), as well as in the

relationships between these various scales (Sewell, 2002). Education researchers have speculated that these problems exist due to a lack of understanding of the complex systems realms in which these entities exist and interact (Chi, 2005). Thus, educational agencies in the US have urged science curriculum and instruction to emphasize systems content (AAAS, 2009; NRC, 2011).

### **Learning about Complex Systems and Learning Progressions**

For more than a decade, knowledge of how students develop an understanding of complex systems has been an important theme in learning sciences research (Hmelo et al., 2000; Jacobson & Wilensky, 2006; Wilensky & Reisman, 2006; Yoon, 2008; 2011). Complex systems provide a framework through which one can explain and understand how patterns emerge across scales, that is, macro scale phenomena emerge from micro scale individual interactions. Complex systems scientists and educational researchers speculate that students have a hard time understanding the mechanisms that drive the emergence of large scale global phenomena from smaller scales of interacting agents (Chi, 2000). Explanations for how patterns emerge require integrating and matching explanations across scales. For example, while local environmental conditions can impose hard limits on where species can live and thereby impose large distribution patterns, the interactions of individuals within and between species contributes substantially to pattern development, influencing biodiversity and even evolution (Levin, 1999). While a coherent understanding of complex systems presently eludes most students (Jacobson, 2001), a biology sequence that is grounded in concrete examples as a starting place can tap into student's intuition about such systems and help build a deep understanding about complex systems as applied to biology, and even more generally. Recent learning progressions research offers a systematic approach in structuring such learning sequences (Alonso & Steedle, 2006; CPRE, 2009). For example, Mohan and colleagues (2009) identified levels of increasing sophistication in students' perception of carbon-transforming events (e.g., combustion, respiration) in complex socio-ecological systems. These ordered descriptions represent a research-informed framework for structuring the learning of core scientific ideas (NRC, 2007; Songer et al., 2009).

### **Computational Modeling**

In addition to identifying a learning progression, educational computational modeling software and associated curricula including StarLogo, NetLogo, Biologica, and handheld Participatory Simulations (Colella et al., 2001; Gobert, 2005; Klopfer et al., 2005; Stieff & Wilensky, 2003; Wilensky & Reisman, 2006) have been created for school age students to learn about and visualize systems. Agent-based programs like StarLogo and NetLogo reveal how simple rules for interaction ascribed to individual agents with varying traits can produce emergent population scale patterns such as flocking behavior in birds, slime mold aggregation, or ant colony organization. Despite the promise of and need for these computational tools, widespread adoption in classrooms has not happened, and there are few studies that provide conclusive objective evidence of their benefit on learning. With respect to adoption by teachers, we know that the incorporation of technologically advanced curricular material into classrooms is met with many well documented challenges, including teacher time constraints, teachers' understanding of technology, teacher confidence levels in terms of computer programming, access to technology, and the lack of supporting curricular materials (Fishman et al., 2004; Yoon & Klopfer, 2006).

### **Project Activities and the Curriculum and Instruction Framework**

The project entails building a curricular and instructional sequence in four high school biology units – Chemistry of Life, Population Ecology, Community Ecology, and Evolution that promote student learning of complex systems for implementation in high school biology classrooms. The project was funded in September of 2010. We report on project activities that ensued for two years until November 2012, which included: building biology simulations through the StarLogo TNG software that combines a graphical blocks-based programming with a 3-D game-like experience; development of classroom curricular materials, e.g., student lessons and teacher guides; construction of a summer teacher professional development workshop; and project implementation in classrooms. We worked with our first pilot cohort of four teachers between August 2011 and June 2012. Work with our second full cohort of 10 teachers began in August 2012 with the summer PD workshop. In order to address the aforementioned educational needs we constructed a curriculum and instruction (C and I) framework to inform the design and implementation of project activities. The C and I framework emerged from a legacy of design work (cf. Klopfer & Begel, 2005), classroom testing (Yoon & Klopfer, 2006), and educational learning research (Klopfer, 2008; Klopfer & Yoon, 2005) that spans more than a decade. As seen in Figure 1, the C and I framework has four main components: 1) Curricular relevance to ensure that project materials will be implemented and useful for students and in the classroom; 2) Cognitively-rich pedagogies to build on relevant understanding of best practices in learning theories; 3) Tools for learning and teaching to scaffold computational and curricular experiences; and 4) Learning progressions to structure sequences that will enable optimal understanding of project goals in computational thinking, biology content and complex systems concepts. In the remaining sections, we describe the design and development of five major project activities that will be discussed in more detail in the symposium: 1) Professional Development and Workshop Design; 2) Designing for Computational Thinking Through Computational Modeling; 3) Curriculum

Design and Development; 4) Issues in the Design of Learning Progressions; and 5) Designing Assessments for Larger Scale.

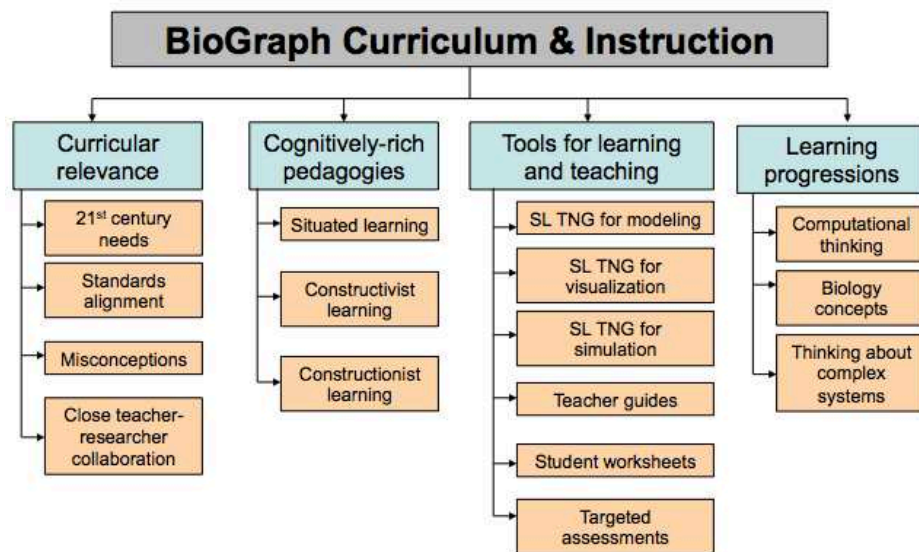


Figure 1. BioGraph Curriculum and Instruction Framework.

Since we are interested in describing the design parameters and trajectories, each description includes:

1. Initial ideas about the design of the activity as they are related to the curriculum and instruction framework and why and how they factored into the design.
2. How the activities as they unfolded did or did not address what we had envisioned to support the curriculum and instruction framework.
3. The rationale, decisions made, and steps taken in the redesign of the activity.

### Professional Development Workshop Design

In this talk we will focus on the major design elements of the professional development activities that were constructed for our teacher participants. The four teachers in the pilot year came from three Cambridge and Boston area schools with an additional ten teachers in the second cohort coming from seven schools spanning eastern Massachusetts. Professional development activities for the pilot year teachers consisted of a one-week summer workshop followed by two half-day school year sessions.

Initially, the summer workshop was designed with the following goals: to introduce the topic of complex systems to teachers; to involve teachers in the co-design of curriculum activities (which were still in the process of being written and refined); and to introduce and begin helping teachers to become comfortable teaching computational modeling as part of their BioGraph instruction. The scope and sequence of activities was focused on gaining teachers' interest and engagement in adopting the C and I framework particularly in the way that computational modeling as a tool for learning could be used to encourage student inquiry and discovery in biology and related complex systems topics. In alignment with professional development research and the components needed for high-quality participation and enactment (e.g., Garet et al., 2001), we wanted to provide hands-on activities for teachers to learn about complex systems through participating in off-computer and on-computer simulations as their students would in inquiry-based activities. Additionally, we assumed from past experience that interacting with computer models and programming with teachers would be time-consuming. Thus, we made key design decisions to accommodate the finite amount of workshop hours which included decreasing the amount of time focused on learning about complex systems theory, fewer moments for reflection, and relatively little focus on classroom management and potential instructional issues in the implementation of project activities. Initial findings from the pilot teachers in workshop reviews, classroom observations, and follow-up interviews indicated a need to revise the professional development activities.

The pilot teachers indicated that they did not fully understand which core complex systems concepts were being illustrated in the StarLogo models and did not feel confident in teaching those to their students. Although they appreciated the active engagement and exploration of the models, they felt they were left to infer the meanings of terms. In classroom observations, teachers rarely made connections between the modeling activities and the higher-level complex systems concept being modeled e.g., self-organization, randomness, or decentralization. In the subsequent workshop in 2012, greater emphasis was placed on acquiring a more robust theoretical understanding of complexity. Teachers were provided several readings in advance of the workshop, were shown a number of videos and illustrations of different models and asked to reflect on the core similarities,

and importantly in order to make time for this theoretical investigation, two more professional development days were added prior to the summer workshop.

Despite the great emphasis in the workshop in computer programming, during our observations none of the teachers worked extensively with their students on learning how to build different models. Due to known challenges in time constraints, and teachers' learning curves, none of the teachers went beyond interacting with pre-existing models. Realizing the time-intensive nature of computer programming and the relatively little impact on classroom practice, we decided de-emphasize its focus in the subsequent summer PD workshop, instead opting for smaller chunks of programming interspersed through the week. However, this design decision showed interesting results with the second group of teachers. It allowed more time to focus on pedagogy and complex systems and having understood those variables better, provided some cognitive space to focus attention on programming. During the August 2012 workshop, several of the teachers engaged in a fruitful discussion on their own about how to change parameters within StarLogo to support or refute hypotheses about the biological systems being studied and initial observations in at least two classrooms showed teachers and students studying the programming blocks and making changes to the code to glean deeper biological understanding (as discussed in the following talk).

Another major design decision in the first PD workshop was to focus less on classroom management and pedagogical issues. Instead, we attempted to scaffold cognitively-rich interactions into the curriculum worksheets that students would work with while using the StarLogo models. Examples of scaffolds (to be discussed in more detail in the curriculum section of the proposal) included instructions to work with a partner and to fill in the worksheet together. However, in classroom observations, we found little collaboration among students. Instead, their participation in project activities looked much more like traditional plug and chug behavior as students (although interested and engaged) answered the questions one-by-one individually. In the subsequent workshop, we made a decision to work with teachers on how to include collaboration and argumentation (McNeill & Martin, 2011) in their pedagogy in conjunction with providing more directed opportunities in the worksheets for students to collaborate and argue their findings. Initial observations with the second cohort of teachers indicate greater amounts of peer-to-peer interaction and we hope to be able to report more detailed findings on this curriculum and instruction design variable in the symposium.

### **Designing for Computational Thinking Through Complex Systems Modeling**

This talk focuses on the design decisions that influenced the construction of the simulations in the StarLogo TNG modeling tool and the computational thinking aspects implicit in the models that enable better understanding of biology content. We discuss the challenges of one of BioGraph's central goals, which is to bring computational thinking into science classrooms. Computational Thinking (CT) is increasingly understood to be a critical component of a 21<sup>st</sup> century education (NRC, 2010). However, CT is often relegated to technology classes, rarely bridging the gap to other subjects, despite its real-world prevalence in those fields.

CT is a large knowledge domain with many interpretations (NRC, 2010), but a few skills and concepts stand out as particularly relevant in a life sciences context. These are: the ability to interrogate and understand the underlying assumptions of models and simulations, and a basic paradigm for understanding and creating agent-based models. Two more general CT ideas underlie these life-science-specific skills. First is the idea that computers follow instructions literally—they do not add knowledge or interpretation beyond what is provided by the programmer. Second, combinations of these simple instructions form algorithms, which in an agent-based context lead to agent behaviors. Stated as the converse: modeling behavioral attributes requires reducing these behaviors to combinations of simple commands. In BioGraph, these CT ideas are critical for developing student understanding of complex systems, which is the cross-cutting theme throughout the curriculum, and for enabling the constructionist pedagogy of the C and I framework. For example, one cannot understand emergence without first understanding the basic rules of the individual agents. Similarly, the concepts of decentralized control and self-organization cannot simply be demonstrated by an animation, because students continue to imagine a large organizing force at work until they are able to construct a model without one.

However, there are many challenges inherent in bringing CT to the biology classroom (as discussed in the section on *Professional Development Workshop Design*). The primary challenge is the time required to integrate what is often taught as its own course into another course that is already saddled with more topics than can reasonably fit into a school year. Perhaps equally challenging is biology teachers' comfort level with teaching CT and programming. Our design efforts have focused on mitigating these challenges through intentional efforts in improving accessibility through the technology platform itself, how we embed programming activities in the curriculum, and low learning threshold activities in professional development.

StarLogo TNG, BioGraph's simulation platform, brings several technological solutions to these problems. The blocks-based visual programming interface eliminates the programming language syntax learning curve inherent to most programming languages, which can take weeks to cover in traditional computer science classes. Additionally, the 3D visualization engine lets students immediately see the effects of programming instructions, and changes to them, within an agent-based modeling paradigm. These features