

Behavior Patterns: Bridging Conceptual Models and Agent-Based Simulations in Interactive Learning Environments

Swaroop S. Vattam, Ashok K. Goel and Spencer Rugaber
Design & Intelligence Laboratory, College of Computing
Georgia Institute of Technology, Atlanta, Georgia 30308, USA
{svattam, goel, spencer}@cc.gatech.edu

Abstract—We describe a technique that takes conceptual, declarative Structure-Behavior-Function (SBF) models of a complex system, and simulates the behavior of the system in NetLogo, an agent-based simulation environment. Our technique uses a library of behavior patterns, where a behavior pattern is a parameterized generic abstraction over classes of SBF models. Given an SBF model constructed by a student, our technique first recognizes the model as an instance of a behavior pattern, then uses the parameters of the pattern to set up the NetLogo simulation, and finally runs the simulation to produce an animation that acts as feedback on the conceptual model.

Keywords: *Complex systems; educational technology; functional models; modeling and simulation.*

I. INTRODUCTION

Modeling is a very familiar idea in science, science education, and educational technology, e.g., [1]. A model is a student's interpretation of a target system (or phenomenon), and proposes mechanisms underlying the system (or phenomenon). For the last several years, we have explored the construction of Structure-Behavior-Function (SBF) models [2] of complex systems for helping middle school students learn about the functions and behaviors of such systems [3]. We have developed interactive learning environments (called ACT for Aquarium Construction Toolkit) that support construction of SBF models of classroom aquaria [4]. We have found that learning about SBF models in middle school science classrooms scaffolded in part by ACT results in statistically significant improvement in the identification of the invisible functions and behaviors of aquaria [4].

Simulation is another familiar idea in science, science education, and educational technology, e.g., [1]. A simulation enables a student to set values of selected system variables and observe the temporal and spatial evolution in the values of the system variables. A student may evaluate or experiment with a model by simulating it. The results of the simulation may result in model revision. In no small part because of legacy software [3], ACT uses the NetLogo platform for generating simulations of classroom aquaria [5,6].

This juxtaposition of conceptual, declarative SBF models and agent-based stochastic NetLogo simulations has raised some interesting issues. Figure 1 illustrates the core problem: how to take a conceptual SBF model (Figure 1a), and produce a dynamic, graphical NetLogo simulation (Figure 1b). In this paper, we describe a computational technique

that bridges the conceptual declarative SBF models and the agent-based stochastic NetLogo simulations. The technique is based on the notion of *behavior patterns*: a behavior pattern is a parameterized generic abstraction over classes of SBF models. Our technique uses a library of such behavior patterns. Given an SBF model generated by a student, our technique first recognizes the model as an instance of a behavior pattern, then uses the parameters of the pattern to set up the NetLogo simulation, and finally runs the simulation to produce an animation.

II. THE ACT LEARNING ENVIRONMENT

Complex systems not only have many interconnected components, but also have causal processes at multiple levels of abstraction such that a causal process at one level emerges out of component interactions at a lower level. SBF models offer important affordances for learning about complex systems. *Structure* in SBF models refers to components of a system (e.g. fish, filters and air pump). *Function* refers to the outcomes of a system or subsystem (e.g. the function of filter is to remove waste in the aquarium ecosystem). *Behavior* refers to the causal mechanisms that result in the functional outcomes (e.g. the bacterial mechanism of conversion of ammonia to less toxic chemicals). The SBF model enables students to consider a complex system such as a classroom aquarium in terms of its structures, behaviors, functions, and the connections among them.

ACT includes an SBF modeling tool called SBFAuthor that provides a visual notation for SBF modeling. SBFAuthor partitions the SBF modeling space into three views: Structure, Behavior and Function views, as illustrated in Figure 2. Each view contains a palette of icons for constructing a model. Once a student creates an SBF model using SBFAuthor, she can simulate that model in NetLogo. The simulation provides an animated plot of the values of the different parameters of the model (see bottom of Figure 2). The simulation can be stopped and resumed at any time.

III. THE ACT ARCHITECTURE

ACT's technique for simulating SBF models using NetLogo introduces two computational constructs: (1) The SBF-NetLogo compiler, and (2) Behavior patterns that are abstractions over classes of SBF models. Figure 3 depicts the relationships between these constructs. Students create SBF models. An SBF model selected for simulation and its associated behavior pattern from the behavior patterns library are inputs to the compilation process that results in the code that is executed by NetLogo.

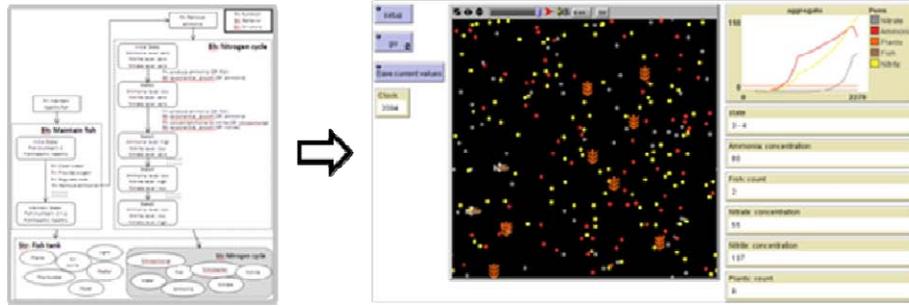


Figure 1. (a) Sample SBF model, (b) corresponding simulation in NetLogo

A. Behavior patterns

To mitigate these challenges due to the context sensitivity of the compilation process, we assume that the mapping rules and the required mathematical equations can be fixed beforehand for a class of SBF models that share deep similarity. That is, if the mapping rules and the equations for a particular class of models are known, any SBF model which is an instance of this class can be simulated by instantiating the class' mapping rules and equations.

instance SBF models that students create. It is our assumption that such patterns can be identified and codified by experts and made part of ACT's library of behavior patterns. In the domain of fish ecosystem, we have identified two behavior patterns: "production-consumption-cycle" and "population dynamics" patterns.

The role of behavior patterns is to encode knowledge common to a set of similar SBF model instances. Therefore, behavior patterns set up expectations for SBF model instances that belong to that pattern. They suggest what information a model belonging to a specific pattern should contain. For example, the "production-consumption-cycle" pattern suggests that there must be one or more substances that are being produced and consumed (e.g., ammonia, nitrite, nitrate, and food in the ammonia cycle model). This pattern suggests that for every substance there has to be a producing agent and a consuming agent (e.g., fish consumes food and produce ammonia, nitrosomonas bacteria consume ammonia and produce nitrite, etc.). Finally, this pattern also suggests that there has to be a mathematical relationship governing the rates of production and consumption of various substances.

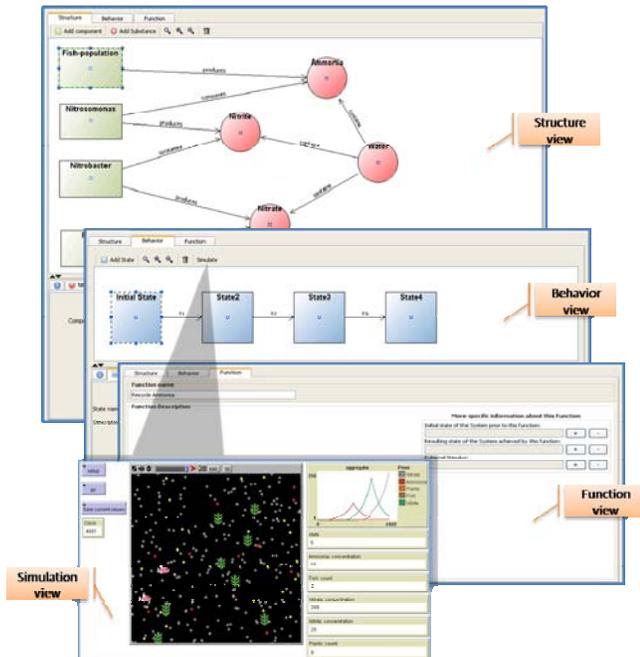


Figure 2. The ACT environment

Behavior patterns are generic abstractions over many SBF models that share a common deep structure. For instance, models of ecosystem containing the carbon-dioxide cycle and the nitrogen cycle are both instances of the "production-consumption-cycle" behavior pattern. Every domain that is the target of students' learning contains one or more such behavior patterns that can account for most of the

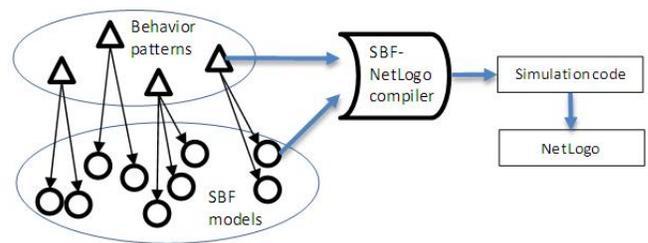


Figure 3. The ACT architecture

Figure 4 shows a simple "production-consumption-cycle" behavior pattern. This pattern shows that four substances are being produced and consumed (S_1, S_2, S_3, S_4) and that four agents are responsible for producing and consuming them (A_1, A_2, A_3, A_4). It also shows which agents are responsible for producing and consuming which substances. Further, it captures the behaviors of the substances as simple transitions between two states (e.g., $S_i(t)$ and $S_i(t+1)$ refer to states quantified by the amount of S_i at times t and $t+1$ respectively). State changes in

substance behavior are governed by the following simple equation:

$$S_i(t + 1) = S_i(t) + amountOf(A_p) * productionRate(A_p, S_i) - amountOf(A_c) * consumptionRate(A_c, S_i)$$

where S_i is the substance, A_p is the agent that produces S_i , A_c is the agent that consumes S_i , and $productionRate$ and $consumptionRate$ are the rates at which A_p produces S_i and A_c consumes S_i . Note that this pattern is just one generalization of the cyclic behavior. It is not necessarily the best nor the most accurate. It merely reflects a reasonable causal model.

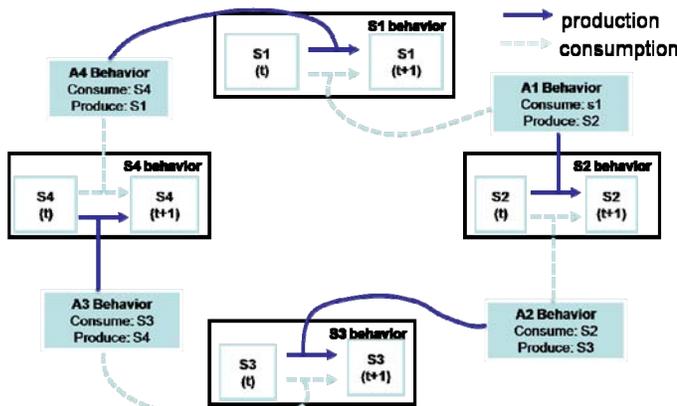


Figure 4. A sample behavior pattern

In addition to capturing the conceptual relationships in a class of models, behavior patterns are also ideal for capturing domain-specific visual, temporal and spatial constraints that translate to compilation rules that determine: (1) which components are mapped to agents and which are mapped to breeds (e.g., the component “ammonia” is mapped to a breed), (2) which components have movement and which do not (e.g., “ammonia” in the nitrogen cycle behavior has movement), (3) if a component has movement in a particular behavior, whether that movement is random (e.g., ammonia) or it follows a path (e.g. electricity through a wire).

IV. CONCLUSIONS

In this paper, we described a computational technique for integrating conceptual SBF models with agent-based NetLogo simulations. This coupling was mediated by behavior patterns that are abstractions over SBF model instances. A behavior pattern sets up expectations and

encodes model knowledge needed by the simulator. A compiler then translates the model elements in the SBF language to the language executed by the NetLogo simulator based on the semantics of the behavior pattern. The advantages of this integration are that students can potentially (1) visualize the behavior of the system they modeled, (2) get feedback about their models by comparing the behavior of the system they modeled with either an expert’s simulation or a real-life version of the system, and (3) get feedback at intermediate stages during the development of their models. The ability to simulate a conceptual representation combines the advantages of conceptual representations (global view, hierarchical decomposition, explicitness, explanatory power) with the benefits of simulations (generality, generativity, precision, predictive power). It also potentially affords learners with timely and immediate feedback on the models of the world they generate.

ACKNOWLEDGEMENTS

Cindy Hmelo-Silver and Rebecca Jordan at Rutgers University have been our research partners on the ACT project, and this work has benefited from several discussions with them. We are also grateful to the US Institute for Education Sciences (Grant # R305A090210) for their support of this work.

REFERENCES

- [1] Clement, J. (2008). *Creative Model Construction in Scientists and Students: The Role of Imagery, Analogy, and Mental Simulation*. Dordrecht: Springer.
- [2] Goel, A., Rugaber, S., & Vattam, S. (2009) Structure, Behavior & Function of Complex Systems: The Structure,, Behavior, Function Modeling Language. *AI for Engineering Design, Analysis and Manufacturing (AIEDAM), Special Issue on Developing and Using Engineering Ontologies*, 23:23-35.
- [3] Hmelo-Silver, C., Jordan, R., Demeter, M., Gray, S., Liu, L., Vattam, S., Rugaber, S., & Goel, A (2008). Focusing on Function: Thinking Below the Surface of Complex Natural Systems. *Science Scope*, Summer 2008.
- [4] Vattam, S., Goel, A., Rugaber, S., Hmelo-Silver, C., Jordan, R., Gray, S., & Sinha, S., (2010) Understanding Complex Natural Systems by Articulating Structure-Behavior-Function Models. *Educational Technology & Society, Special Issue on Creative Design*, 14(1): 166-181.
- [5] Wilensky, U. (1999) NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.
- [6] Wilensky, U., & Resnick, M. (1999). Thinking in levels: A dynamic systems approach to making sense of the world. *Journal of Science Education and Technology*, 8, 3-19.