



Modeling Commons

Reuven M. Lerner, reuven@lerner.co.il

Center for Connected Learning and Computer-Based Modeling, Northwestern University

Sharona T. Levy, stlevy@edu.haifa.ac.il

Faculty of Education, University of Haifa

Uri Wilensky, uri@northwestern.edu

Center for Connected Learning and Computer-Based Modeling, Northwestern University

Abstract (style: Abstract title)

Our research project, the “Modeling Commons,” is designed to make the agent-based NetLogo modeling environment (Wilensky, 1999) not only an effective tool for creating models, but also for sharing them with others and collaborating during the modeling process. It is also meant to help modelers organize their work into “families” of related models. In this demonstration of constructionist media, we will show the basic features of the Modeling Commons, a Web application that allows NetLogo users to share models with one another, collaboratively write and edit models, discuss models, and categorize models with social tags. We point to the ways in which we have designed and implemented the Modeling Commons as a constructionist environment, from the implementation of a version-control system accessible to non-programmers, to making questions prominent and visible to all users, to the permission system that lets modelers keep their work private until they consider it ready for public consumption. We then describe planned future features for the Modeling Commons, including support for NetLogo extensions, a Wiki for each model, and cross-model discussions.

Keywords (style: Keywords)

Constructionism, Collaboration, Modeling, Social interactions, CSCL, World Wide Web, NetLogo

Introduction

Modeling plays an important role in many fields of research, including science, mathematics, engineering, and (most recently) social science (Morrison & Morgan, 1999; Axelrod, 2005; Epstein, 2008). Models provide a bridge between theory and practice, offering a form of reified theory that can be manipulated, prodded, and poked in various ways, to test its responses to various inputs and scenarios. While there are different types of models, ranging from equations to physical mock-ups, modern-day models are often built in software, in the form of a computer-based simulation (Jonassen, 2006).

One technique for creating software simulations is known as “agent-based modeling.” In such a system, the modeler creates software entities, known as “agents,” and then gives those agents rules to follow. Such rules can describe how they move around the virtual world and how to interact with other agents they might encounter. For example, to model the spread of disease in a population, each agent might represent a person; an infected person would pass the disease onto another based on a number of pre-programmed criteria, such as the distance between the two agents, the health of each agent, and a random factor.



A popular environment for creating agent-based models is NetLogo, which has long supported constructionist learning (Wilensky, 1999). NetLogo's users range from middle-school students through university researchers. The software comes with several-hundred sample models, on subjects ranging from the aforementioned spread of disease in a population, to the "Monty Hall" problem, to the way in which leaves change color in the autumn, to the way in which people separate when participating in an exercise class. NetLogo is exploratory and constructionist by design, encouraging users to create and run their own models, learning about a particular subject domain in the process.

A growing body of theory and evidence point to the importance of social interactions when among learners (Vygotsky, 1978; Lave & Wenger, 1991). Papert himself, when he defined constructionism (Papert, 1980), indicated that it involves not only building, but also sharing with others.

From interviews with NetLogo users of varying skill levels, we know that NetLogo models are often built through a process of collaboration, sometimes with other modelers and at other times with domain experts. However, NetLogo lacks any built-in tools for facilitating such "genuine interdependence" (Salomon, 1992). Modelers have thus created their own tools and processes for sharing models with one another, gathering feedback from peers, and including multiple people in the modeling process.

Our research project, the Modeling Commons, has two separate but related goals: First, to facilitate a variety of types of interactions among modelers, to offer them a platform that allows them to cooperate, collaborate, and share with one another. A second goal is to better understand the ways in which modelers communicate and interact. By interviewing modelers before and during their use of the Modeling Commons, and by examining the logfiles generated from its use, we have already gained perspective into the ways in which people work together, and are using such information to improve the experience and the modeling environment.

While the Modeling Commons has been under development for several years, it was announced and open to the public in early 2012. Several hundred new users have already registered, and several dozen new models have been shared.

For the Constructionism 2012 conference, we would like to demonstrate the Modeling Commons, reviewing its features, including a number of the design considerations that went into its creation not only as a tool for Web-based collaboration, but as an environment in the constructionist tradition. We hope that by sharing the Modeling Commons with other researchers, we will help to improve the development not only of technologies among constructionists — who need no convincing about the benefits that such technologies can provide — but of technologies that encourage a variety of social interactions.

Features of the Modeling Commons

At its core, the Modeling Commons lets people take an existing NetLogo model, written on their own computer, and upload it. Each uploaded model gets its own, unique URL and page within the Modeling Commons. After just this initial upload, going to that URL provides users with the ability to view the model's preview image (assuming that one was uploaded), run the model in the Web browser (as a Java applet), view the model's procedure definitions, and read its "info" tab, describing the model to potential users and collaborators.

We should note that the Modeling Commons is an attempt to provide a solution to multiple communities, each with its own needs. By default, uploaded models can be viewed and modified



by any registered member of the Modeling Commons. (Registration is free, and requires that the user fill out a simple, short form describing themselves.) However, we know that for some users — particularly in educational settings — users will want to keep their work private, either just to themselves or to other members of a group. For this reason, the Modeling Commons allows model authors to permissions on a model, and to define a group of people who are allowed to view and/or modify it. The default permissions are currently quite open, allowing anyone on the Internet to view a model, and also allowing any registered Modeling Commons user to change a model.

Even if someone uploads a new version of a model that breaks it, or substantially changes its behavior, it is possible to go back one version with a few clicks of the mouse: We keep the entire history of every model, such that it is always possible to revert to a previous version. This “undo” facility on models is meant to give modelers confidence, allowing them to make mistakes, debug them, and learn more about their model in the process.

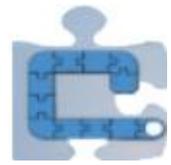
In many cases, we have found that modelers wish to create multiple variations of a model, either because they are exploring multiple, related simulations or because they had varying levels of complexity. In either event, our “model family” functionality is appropriate; when you upload a new model version, you can specify that it should replace the existing version, or that it should be a new model, but with a parent. Over time, we hope to see many families of related models, looking at topics from different angles. For collections of models that are related by topic, but which do not have an obvious parent-child relationship, we also offer “projects,” a looser and less hierarchical association than the family relationship.

The Modeling Commons also offers a forum for each model, allowing people to exchange ideas and/or questions about the model, the domain explored by the model, or NetLogo coding. Another form of communication is the “social tag,” allowing any Modeling Commons user to add one or more textual descriptions to a tag. We hope that over time, users will create a “folksonomy” for models, making it possible to find all of a models on a particular topic.

While the Modeling Commons offers modelers a variety of functions, we have learned through our testing period and since the launch that additional features are necessary: We intend to support NetLogo extensions and external files, both of which are in increasingly common use. We have also found, in our research, that providing additional support for non-programmers is important, and are thus planning to provide a Wiki for each model, as an open-ended notebook, communication device, and scratchpad for modelers. Finally, we have begun to see cracks in our model-centric design, and are planning to introduce features that will make it possible to have a discussion extend over multiple models in a family or project, rather than on only one individual model.

Conclusion

The Modeling Commons is a constructionist environment for users of a constructionist modeling tool (i.e., NetLogo). Whereas the focus behind NetLogo is the creation of new models, the Modeling Commons exists to share existing models, and to encourage people to edit, improve, and communicate with others during the modeling process. We believe that our work on the Modeling Commons will provide useful ideas for others interested in creating online environments, particularly from within the constructionist role.



References

- Axelrod, R. (2005). Advancing the Art of Simulation in the Social Sciences. In J.-P. Rennard (Ed.), *Handbook of Research on Nature Inspired Computing for Economy and Management*. Hersey, PA: Idea Group Publishing.
- Epstein, J. M. (2008). Why Model? *Journal of Artificial Societies and Social Simulation*, 11(4), 12.
- Jonassen, D. (2006). *Modeling with Technology: Mindtools for Conceptual Change* (3rd ed.). Upper Saddle River, NJ: Pearson Education, Inc.
- Lave, J., & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. New York: Cambridge University Press.
- Morrison, M., & Morgan, M. S. (1999). Models as mediating instruments. In M. S. Morgan (Ed.), *Models as Mediators: Perspectives on Natural and Social Science*. Cambridge: Cambridge University Press.
- Papert, S. (1980). *Mindstorms*. New York: Basic Books.
- Salomon, G. (1992). What Does the Design of Effective CSCL Require and How Do We Study Its Effects? *SIGCUE Outlook*, 21(3), 62-68.
- Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.
- Wilensky, U. (1999). NetLogo [computer software]. *Center for Connected Learning and Computer-Based Modeling, Northwestern University*.