1 Encouraging Collaborative Constructionism: Principles Behind the Modeling Commons

Reuven M. Lerner (style: Author and Name), ***reuven@lerner.co.il***

Department of Learning Sciences, Northwestern University, Evanston, Illinois

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

Faculty of Education, University of Haifa, Mount Carmel 31905, Israel

Uri Wilensky, uri@northwestern.edu

Departments of Learning Sciences and Computer Science, Center for Connected Learning and Computer-Based Modeling, Northwestern Institute on Compex Systems (NICO), Northwestern University, Evanston, Illinois

Abstract

Constructionism argues that learning occurs best when constructing a public artifact (Papert 1983). NetLogo (Wilensky, 1999a) is an agent-based modeling language that has successfully been used in a variety of constructionist contexts. However, NetLogo lacks built-in support for making artifacts public, or for creating models collaboratively. Our research focuses on a Web-based tool known as the "Modeling Commons," which is designed to make NetLogo not only an effective tool for creating models, but also for sharing them with others and collaborating during the modeling process.

From our efforts and research spent creating the Modeling Commons, we have identified a number of design principles, which would appear to encourage the creation of collaborative online communities. These principles have provided us with guidelines during the design process, and have helped to focus our questions during design-based research interviews and analysis.

Briefly, our principles are: Focus on artifacts, provide multiple entry points, be forgiving, maximize findability, provide flexible permissions, and keeping users informed.

In this paper, we describe these design principles and explain why we believe that these principles are important and how they have affected our design of the system.

Keywords (style: Keywords)

Keywords: Modeling; Constructionism; Collaboration; Design-Based Research;

Computer-Supported Collaborative Learning

1 Introduction

Constructionism (Papert, 1980) argues in favor of learning through the creation and sharing of artifacts. NetLogo (Wilensky, 1999a) an agent-based modeling environment, has long supported individual constructionist learning (Jonassen, 2006; Reisman & Wilensky, 2006). However, NetLogo lacks built-in support for sharing models, let alone collaboratively developing them. Recent theory and evidence demonstrate the central role that social interaction plays in learning (Vygotsky, 1978; Lave & Wenger, 1991; Wenger, 1998) in general, and when modeling in paritcular (de Aennle, 2009, Bollen et. al., 2002).

In order to support and encourage interactions among modelers, we have created a Web-based complement to NetLogo, known as the Modeling Commons (Lerner, Levy, & Wilensky, 2010b). The Modeling Commons (see Figure 1) makes it possible for NetLogo modelers to share their work with others via a central Web site. The author of a model may optionally allow others to edit and update the model, providing a mechanism for collaborative modeling. Users may attach one or more supporting files to a model, such as input data, curricular suggestions, research papers, and PowerPoint presentations. Those users who visit the model in the Modeling Commons may participate in a discussion about the model, apply one or more social tags to it, or recommend it to their friends.

Anecdotal evidence indicates that many NetLogo users already share models with one another, and even collaborate with their peers while modeling. However, we believe that such interactions are less frequent than could be if collaborative tools were more easily accessible. For example, the netlogo-users e-mail list is the primary forum for discussing the NetLogo language. Over a period of seven years, containing 9,696 messages sent to this list, only 5.7% were found to contain models. NetLogo's Community Models library, hosted on the same Web site as NetLogo, has added only 274 models over the same time period.

At this time, the Modeling Commons is still being tested, with an expected launch in the coming months. It has already undergone three rounds of design research (Brown, 1992; Collins, Joseph, & Bielaczyc, 2004) with 72 subjects, and it has been used in courses at three American universities in the last 18 months.

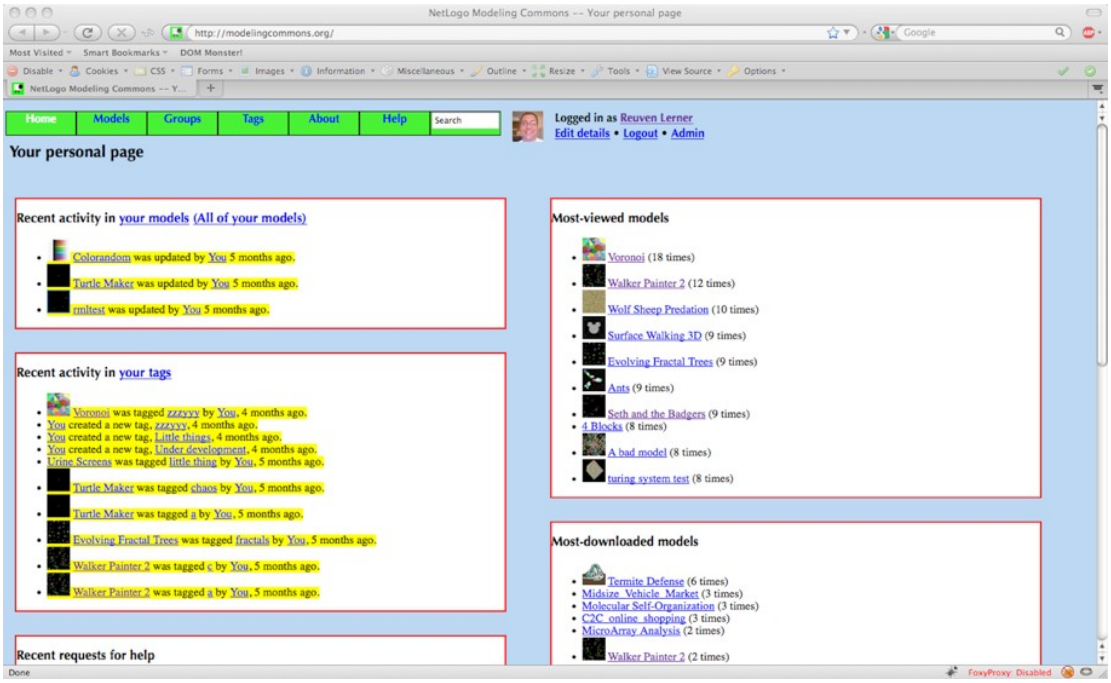

Figure 1: User's home page in the Modeling Commons

We believe that these trials have provided us with many insights into the fostering of public sharing and collaboration among NetLogo modelers. These principles may well be of interest to others creating similar collaborative constructionist environments (e.g., (Monroy-Hernández, 2007)). Our list was inspired by Bruckman (Bruckman, 1997), who enumerated principles that guided her design of the MOOSE language. By publicizing our list, we hope to engage in a discussion with researchers implementing similar environments, with an eye toward a set of "best practices" that encourage constructionist collaboration. There are already many descriptions of how to create an effective online community, and we have drawn upon many of their ideas, as well (Powazek, 2002; Spolsky, 2009).

1 Focus on artifacts

First and foremost, the social interactions are structured around artifacts – and more specifically, agent-based computer models. Models not only provides a focus for the user interface, but are also the core items to which users may attach documents (e.g., curricula, research papers, and NetLogo extensions written in Java), and serve as the foundation for both discussions and social tags (Smith, 2008). This model-centric focus reflects the constructionist philosophy of the NetLogo software. It also takes into account social learning theory (Lave & Wenger, 1991), and specifically Communities of Practice (Wenger, 1998), in which a shared repository of knowledge sits at the core of the community. By working on models in a variety of ways, new participants join the community.

Our experience to date indicates that this approach has been successful (Lerner et. al., under review). In the study, we have found that when looking at all communication taking place in a community, the models serve to create a fully connected social network.

The recent introduction of several oft-requested features — e.g., collections of related models ("projects"), and "families" of models derived from a common ancestor — has posed at least one challenge to this principle. For example, once there are 20 variations on the Fire model (Wilensky, 1998), should discussions be applied only to a single model? Or should they apply to all models with a common ancestry? We are currently considering a number of solutions to this problem.

2 Provide multiple entry points.

Constructionism has often used the example of "samba schools" (Papert, 1980; Zagal & Bruckman, 2005), in which participants take on a role within a community event. Part of the power of the samba-school example is its acknowledgement that there is no one, single path to learning and community participation. Rather, by providing a multiplicity of entry points, newcomers may choose a role that best suits them — or from a number of different roles, until they find something that seems appropriate.

We have tried to design the Modeling Commons such that it encourages users with different skills and interests to participate in ways that are most appropriate for them. They may browse or search through the list of models, reading about and experimenting with models that interest them. They may ask or answer a question, or simply post a comment, about a model. They may apply one or more social tags, such that other users will more easily find the model. They may contribute a document, providing additional documentation or curricular materials to accompany the model. Or they might read and modify the NetLogo code, enhancing the model itself.

One practical lesson that we learned during our design research was that many initial users wanted to be able to browse through the Modeling Commons without having to register or log in. Until that point, we had required registration, in part for easier tracking of user interests and activity. Removing this barrier to entry meant that browsing, or simply stumbling upon models in the Commons via a Google search, provided yet another flexible entry point.

Some initial findings (Lerner, Levy, & Wilensky, 2010a), showed that users did indeed take advantage of different facilities. The resulting social network from each of these individual communication links was a subset of the fully connected graph. Only after combining all modes

of communication was the graph fully connected, demonstrating that different users preferred to use different aspects of the Modeling Commons.

3 Be forgiving.

A design principle of this same name appears in Bruckman (Bruckman, 1997), but we give it a different meaning: Whereas Bruckman designed the MOOSE language to forgive syntactic errors, we have designed the Modeling Commons to forgive editing and uploading errors. We did this by implementing a system similar to that used by Wikipedia (Wikimedia Foundation), storing all old versions and allowing users to download or revert to any of those versions.

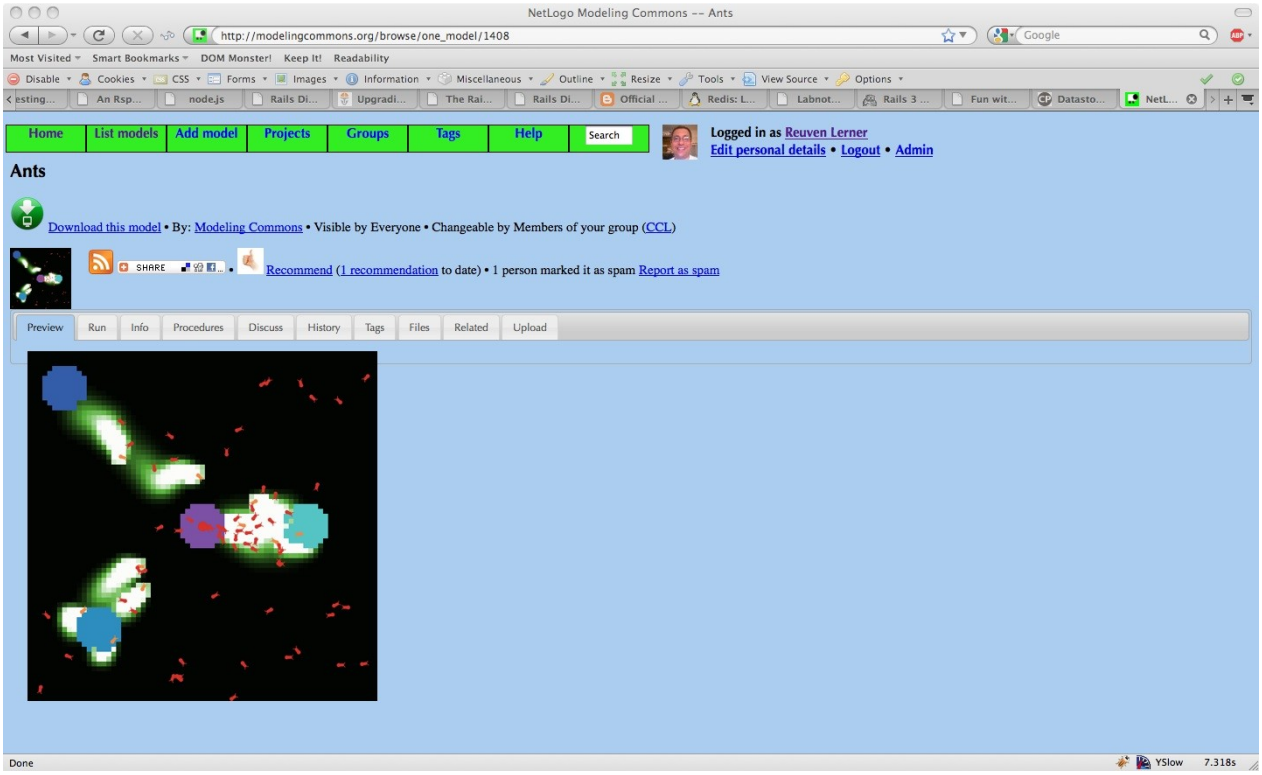

Figure 2: One model (Ants) in the Modeling Commons

Our design research found that even most experienced programmers, whose version-control systems provide similar facilities, failed to use such tools when modeling in NetLogo. By providing such capabilities, the Modeling Commons encourages users to experiment, without having to fear that previously saved changes will be lost. Experimentation is at the heart of constructionist learning, and we aim to offer modelers a safe environment for experimenting with their ideas.

4 Maximize findability.

Morville (2005) claims that information needs to not only exist, but be easily findable by someone interested in using it. Because many users of the Modeling Commons are new to NetLogo modeling, we aimed to structure the site for maximum findability, both from within the system and from without.

Within the system, we created a search system (see Figure 3) that looks at the model name, its authors' names, and the social tags that are associated with it. In addition, we made it possible to easily find models associated with a group of users, on a specific project, or to which a particular social tag had been applied.

Separately, we are employing techniques known collectively among Web professionals as “search engine optimization,” or SEO, to increase the chances of outside users discovering and then using the Modeling Commons. Already, before the official launch of the Modeling

Commons, we have found that some users have found and downloaded models after searching for a specific model name in Google.

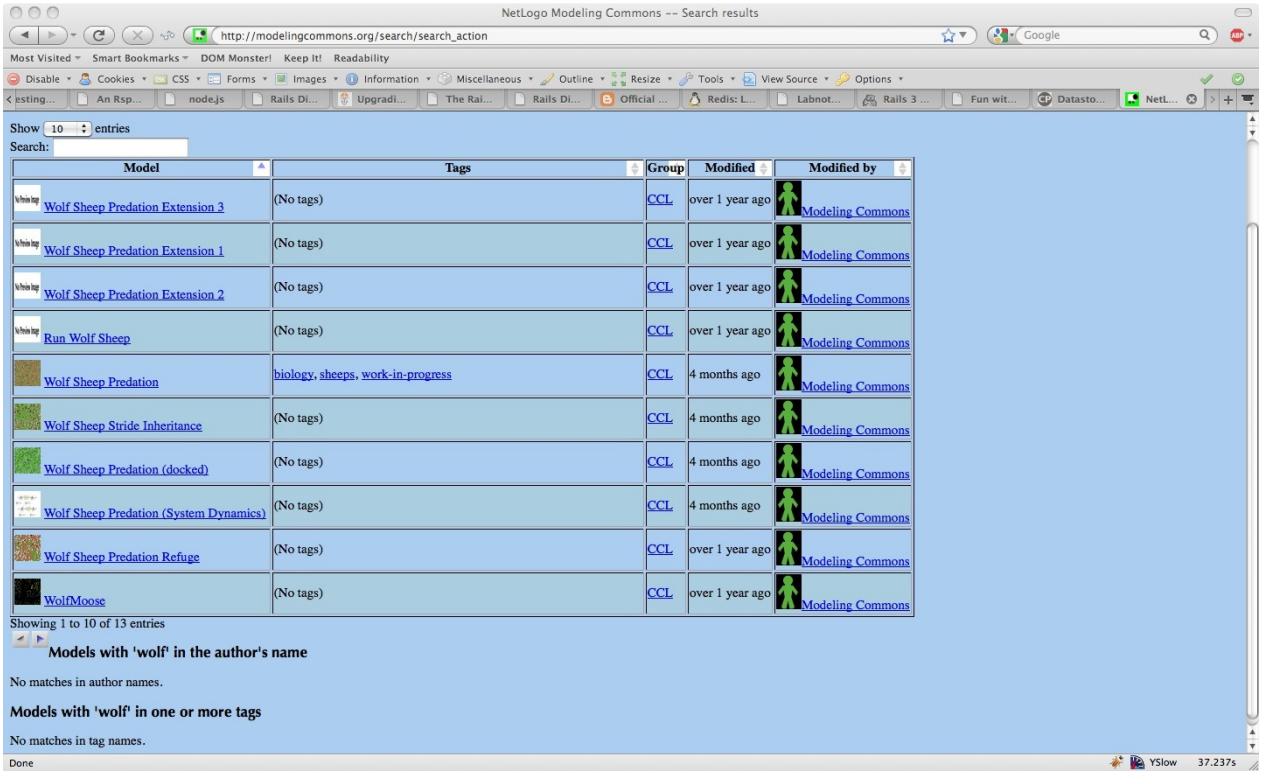

Figure 3: Search results in the Modeling Commons

5 Provide flexible permissions.

The original design of the Modeling Commons called for all models to be publicly readable and writable by all other users. Interviews revealed that many potential users were uncomfortable with such a policy, either because the models were still under development, or because the modelers were students, interested in sharing only with their classmates. Teachers were similarly nervous about using the Modeling Commons in their courses, if it meant that a students' classwork assignment could be found and discussed, let alone potentially modified, by someone from outside of the class. At the same time, interviews showed that modelers were interested in eventually revealing their models and making them available to others in the Modeling Commons.

Protecting users's privacy and security, while ensuring flexibility and allowing users to change those permissions down the road, thus became a key design principle. We distinguish between the "read" and "write" permissions that a user applies to a model (see Figure 4).

We should note that the question of permissions was complicated somewhat by the introduction of collaborators. If a user creates a model, and a second user modifies it, then should the original author be allowed to shut out the second one, by changing the permissions? We decided that any collaborator should be granted author status — meaning that once someone has contributed to a model, it is impossible to shut them out completely. Changing a model's permissions such that it is only visible to the author effectively allows all authors to still see it, as well as to modify it.

The issue of permissions is an important and sensitive one. We expect that it will require additional attention over time, and are curious to see how users will take advantage of these permissions, and how often they will indeed make previously closed models visible to the general public.

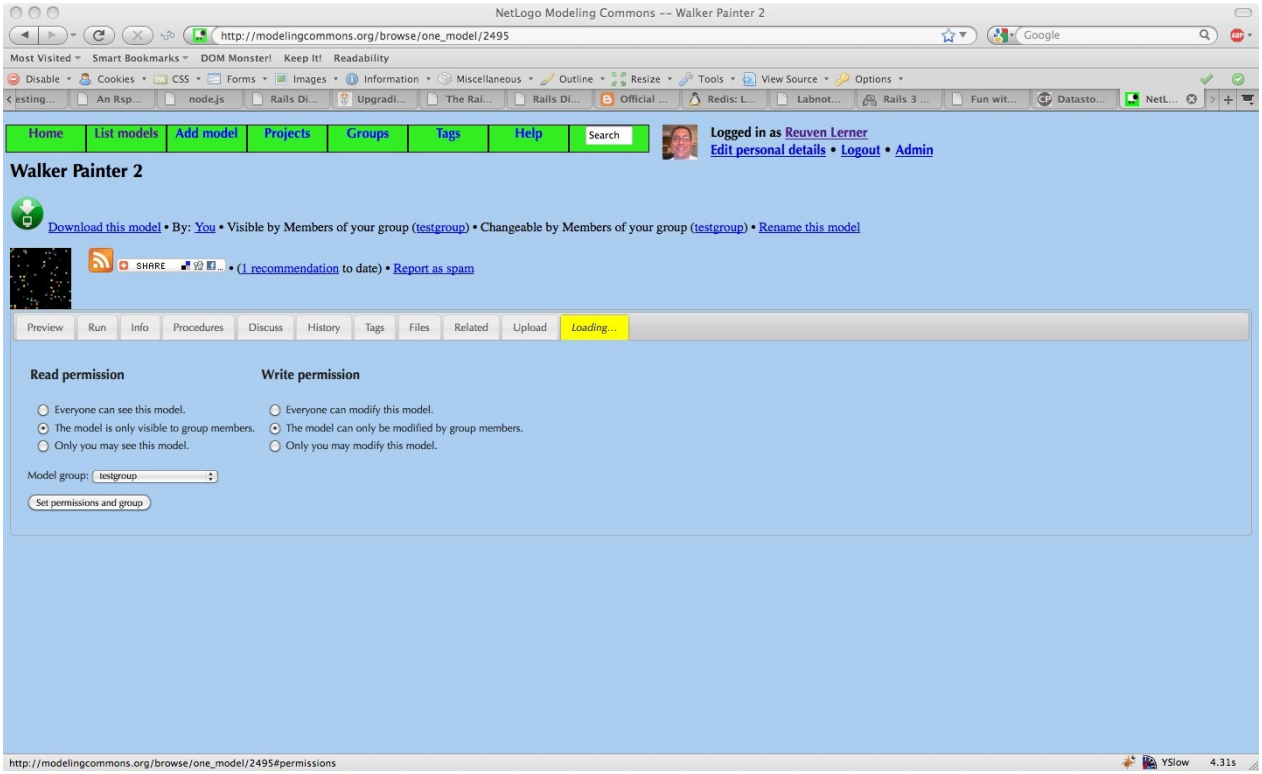

Figure 4: Permissions selection in the Modeling Commons

6 Keep users informed.

Collaboration requires participation, and participation requires an awareness of others in a community are doing. The Modeling Commons tries to keep users aware of what activities others are engaged in, both while they are using the site and when they are not. Many of these features were developed in response to users' feedback; each round of interviews provided additional ideas and suggestions for new features that should be added in order to make the system more social.

In the Modeling Commons itself, the user's default home page provides a bird's-eye view of the latest updates to the site, from the user's perspective. Every user sees nearly the same list of recently updated models, applied tags, and most-downloaded models. (The "nearly" reflects the fact that models to which the user lacks read permissions are removed from the list.) The user's home page also indicates what changes have taken place in the models to which the user has contributed, as well as those social tags that the user has created or applied.

The model pages also attempt to foster inclusion and participation. Users may indicate that they like a model, or recommend a model to their friends via a number of online networks and services, such as Facebook and Twitter. Users may participate in a discussion about a model; each comment in a discussion displays not only the contributor's name, but also their picture, much like blogs and recent social-networking sites.

Users may opt to receive updates about a particular model via "RSS," a popular protocol used by "feed readers," commonly employed by blogs. In this way, a user may "subscribe" to one or more models, learning about updates as they happen, but without cluttering their inbox.

If community members need to explicitly visit the site to find out what is happening, participation will be less frequent than if it is “pushed” to the users in some way. For this reason, an e-mail message is generated and sent to each of a model’s authors when a tag, discussion posting, or new model version is submitted. A user thus knows when other users users are improving or adding to a model. Each e-mail message contains a hyperlink back to the referenced model, discussion, or tag, making it easy to participate (and perhaps respond) immediately.

Principle	Applications of this principle
Focus on artifacts	•All discussions, and social tags are attached to a model •Secondary artifacts (e.g., documents) are also attached to a model
Provide multiple entry points	•Multiple ways to participate (e.g., adding and editing models, adding documents, taking part in discussions, and adding social tags) •No need to log in, if you only want to view models •Make it possible to run a model from within the browser
Be forgiving	•Easily revert to an earlier version of a model •Compare two versions of a model •Download earlier versions of a model (without reverting)
Maximize findability	•Internal search looks at multiple model attributes •Use of SEO to draw in people via Internet search engines •Links make it possible to find all models associated with a project, user, or social tag •Navigation and graphs showing relationships among parent and children models
Provide flexible permissions	•By default, models are visible to and modifiable by all users •Users may restrict read or write access to a model to themselves, to a group that they define. •Permissions may be changed at any time •Anyone who has contributed to a model is considered an author, and thus has the same permissions as the original author
Keep users informed	•Provide context and updates on the user's home page •Provide RSS feeds, for users to receive regular updates in a separate piece of software •Send e-mail alerts when a model on which the user has worked has been modified or updated •Send e-mail alerts when a user's discussion or social tag is updated

2 Conclusion

The Modeling Commons aims to provide NetLogo modelers with a method for sharing and collaborating that goes beyond the current norm among members of this community. These design principles have served us well throughout our design tests, and have helped to focus our development efforts. We look forward to providing further reports on our design principles, as well as the Modeling Commons itself, with the research community over the coming months and years.

References

Bollen, L., Hoppe, H., Milrad, M., & Pinkwart, N. (2002). Collaborative Modelling in Group Learning Environments. In P.I. Davidsen, E. Mollona, V.G. Diker, R.S. Langer & J.I. Rowe (Eds.), Proceedings of the XX International Conference of the System Dynamics Society (pp. 53-64), Palermo (Italy), July 2002.

Brown, A. L. (1992). Design experiments: Theoretical and methodological challenges in creating complex interventions in classroom settings. Journal of the Learning Sciences, 2(2), 141-178.

Bruckman, A. (1997). MOOSE Crossing: Construction, Community, and Learning in a Networked Virtual World for Kids. Massachusetts Institute of Technology, Cambridge, MA.

Collins, A., Joseph, D., & Bielaczyc, K. (2004). Design Research: Theoretical and Methodological Issues. Journal of the Learning Sciences, 13(1), 15-42.

de Aennle, C. (2009). In Simulation Work, the Demand Is Real. The New York Times.

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.

Lerner, R., Levy, S.T., & Wilensky, U. (under review). Connected Modeling: Design and Analysis of the Modeling Commons. International Journal of E-Learning and Learning Objects (submitted to IJELLO special series of Chais Conference 2010 best papers).

Lerner, R. M., Levy, S. T., & Wilensky, U. (2010b). Modeling Commons. http://modelingcommons.org/.

Monroy-Hernández, A. (2007). ScratchR: A platform for sharing user-generated programmable media. Proceedings from IDC.

Morville, P. (2005). Ambient Findability. Sebastopol, CA: O'Reilly Media.

Papert, S. (1980). Mindstorms. New York: Basic Books.

Powazek, D. M. (2002). Design for Community: The Art of Connecting Real People in Virtual Places. Indianapolis: New Riders.

Reisman, K., & Wilensky, U. (2006). Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories -- an embodied modeling approach. Cognition and Instruction.

Smith, G. (2008). Tagging: People Powered Metadata for the Social Web. Berkeley, CA: New Riders.

Spolsky, J. (2009). Learning from StackOverflow.com. Google talks http://www.readwriteweb.com/archives/anthropology_the_art_of_building_a_successful_soci.php .

Vygotsky, L. S. (1978). Mind in society: The development of higher psychological processes. Cambridge, MA: Harvard University Press.

Wenger, E. (1998). Communities of Practice: Learning, meaning, and identity. Cambridge: Cambridge University Press.

Foundation, W. Wikipedia. Retrieved 11 November, 2007, from http://www.wikipedia.org/.

Wilensky, U. (1998). NetLogo Fire model.

Wilensky, U. (1999a). NetLogo [computer software]. Center for Connected Learning and Computer-Based Modeling, Northwestern University.

Zagal, J. P., & Bruckman, A. (2005). From samba schools to computer clubhouses: Cultural institutions as learning environments. Convergence, 11(1), 88-105.