Social Interactions Among Modelers

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

NetLogo (Wilensky, 1999) 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 project, the "Modeling Commons," 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. In this study, we interviewed NetLogo modelers about their interactions —sharing, cooperation, and collaboration — with others during the modeling process. We found examples of all three interactions, but also saw that modelers collaborate separately, and differently, with both programmers and domain experts. In this paper, we describe these interactions, including the distinction between "code collaboration" and "domain collaboration." We describe changes we intend to make to the Modeling Commons to provide domain experts with additional affordances for collaboration.

Keywords

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

Introduction

Constructionism (Papert, 1980) argues in favor of learning through the creation and sharing of artifacts. NetLogo (Wilensky, 1999) is an agent-based modeling environment that has long supported individual constructionist learning (Jonassen, 2006; Reisman & Wilensky, 2006).

The artifacts created by NetLogo users are models — software simulations, typically in the domains of science, mathematics, and social sciences. Models have long been used by scientists and engineers, in a wide variety of domains, and for many different reasons (Morrison & Morgan, 1999; Epstein, 2008). Software simulations can be used for understanding, exploration, and prediction to test plausible explanations for phenomena, discover new relationships from multiple runs of models at different settings, and predict future events based on past trends and complex systems principles.

NetLogo has many thousands of users, ranging from middle- and high-school students learning about science, modelling, and complexity, through university and corporate researchers. However, NetLogo lacks built-in support for sharing models, let alone interacting during the modeling process. 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 particular (Bollen, Hoppe, Milrad, & Pinkwart, 2002; de Aennle, 2009). Moreover,

Papert's original description of constructionism (Papert, 1980) describes not only building, but also sharing with others, as a critical part of the process.

For example, among software developers, "pair programming" (Beck, 2000) has become a popular method for collaboration, one which may be appropriate for at least some NetLogo modelers. Studies indicate that pair programming results in higher-quality software, but also in a feeling among developers that they have learned much from one another (Cockburn & Williams, 2001).

In this study, we are interested in the social interactions that take place among modelers in order to better design a supporting platform we have created, the Modeling Commons. In this paper, we focus on three forms of interaction: *collaboration*, *cooperation*, and *sharing*.

Researchers distinguish between collaboration and cooperation when discussing interactions: "In cooperation, partners split the work, solve sub-tasks individually and then assemble the partial results into the final output. In collaboration, partners do the work 'together' " (Dillenbourg, 1999). In a collaborative project, no participant can continue on his or her task without input, advice, and assistance from one or more partners, what has been called "genuine interdependence" (Salomon, 1992). By contrast, cooperation describes a situation in which the main task is split into parallel, somewhat independent sub-tasks. We see sharing — i.e., showing an artifact to others after the creation process, rather than while it is taking place — as a third type of personal interaction. In designing the Modeling Commons, we wished to support all three of these forms of interaction.

Our research project, the Modeling Commons (Lerner, Levy, & Wilensky, 2010b), is a Web platform for social modeling, both facilitating interactions among modelers and providing insights into modelers' interactions and learning. Using a Web browser, members of the Modeling Commons can share, discuss, categorize, and collaboratively author NetLogo models. As a design-research project (Brown, 1992; Collins, 1992), insights gained from user experiences are used to change and improve the Modeling Commons software, to better encourage improved interactions — and, we hope, improved models, as well.

The Modeling Commons was formally announced to the NetLogo community in January 2012. As of April, more than 70 new users have registered for the Modeling Commons. Preliminary versions were used by university courses on constructionism, modeling, and complexity, as well as by individual researchers and modelers. Feedback from these initial trials, as well as analysis of the system's logs (Lerner, Levy, & Wilensky, 2010a), helped us to improve the system, as well as to understand unique and various ways in which the Modeling Commons may be used.

NetLogo modelers have been interacting for years without the benefit of the Modeling Commons. To be effective, our design research must incorporate existing practices among NetLogo modelers, without the Modeling Commons. However, our ultimate goal is not just to facilitate existing interactions, but to allow for the creation and development of new paradigms, such that interacting via the Modeling Commons will be more effective than even face-to-face collaboration could be (Hollan & Stornetta, 1992).

This paper describes a study that we conducted in order to better understand how NetLogo users currently interact, without benefit of the Modeling Commons. The research question for this study was: What types of interactions and organizational structures currently exist among NetLogo users, without the Modeling Commons? Answering this question will not only help us to understand current practice, but also how we can detect and categorize interactions among users of the Modeling Commons.

Methods

Even before the study began, we had strong anecdotal evidence – from our interactions with NetLogo users, including many students learning NetLogo as part of a university class – that the question was not whether there were social interactions among modelers, but rather what form they took, and how the modeling process was affected as a result.

Participants: We aimed to recruit up to 20 people, experienced with NetLogo but unfamiliar with the Modeling Commons, to describe the ways in which they develop models, and the interactions they have when doing so. Our main source for subjects was NetLogo-users, a public e-mail list that with nearly 4,000 users that has served for more than a decade as the chief method for peer-to-peer communication and support within the NetLogo community. Via private e-mail, we asked approximately 15 of the most active recent participants in NetLogo-users to participate in our study. Following that invitation, we asked for volunteers among all members of NetLogo-users to agree to participate in our study. Additional subjects were recruited via the snowball method, as well as by initiating contact with specific people whom we knew to be active in the NetLogo community. We recognize that there was some selection bias, in that we specifically indicated in our recruitment messages that we would be asking about collaboration and personal interactions. Our subjects may well have been more likely to collaborate with others than the average NetLogo user.

These efforts resulted in nine interviews with 10 different subjects. (One interview was with a pair of subjects who often work together.) All were adult researchers, and nearly all either had a PhD or were working toward one. Only one had used the Modeling Commons before the interview took place, but all were experienced NetLogo users, with several of them having worked with NetLogo for several years, on a number of significant models.

Research tools and paradigm: Interviews were conducted by telephone or Skype in the clinical style (Ginsburg, 1997). Interviews were recorded, transcribed, and coded for topics having to do with social interactions and modeling.

Results

Interview subjects were all asked whether they worked with others when creating NetLogo models. The answer was uniformly positive, with all saying that they work with others at some point during the modeling process. However, the subjects reported engaging in widely divergent types of interactions with others, including all three types mentioned above: Sharing, cooperation, and collaboration.

Sharing: All of the subjects reported that they have shared NetLogo models — that is, shown a model to others after having reached at least one significant milestone. Most reported having shared models with a small number of people, such as a doctoral committee. One subject reported having shared his model with the readers of a journal article he wrote; when asked how he shared the model, he said that he had used the Modeling Commons to do so.

In some cases, subjects said that they used NetLogo's "applet" feature to share a model via a Web site, allowing others to view and use the model without having to learn or install NetLogo. As one said, "It provides a painless way of them seeing it working, and then if they want to get into it further they've got the NetLogo file to download. They can go and get themselves a copy of NetLogo." Another subject reported that his organization has a Web site on which they publicize models on a regular basis. A small number reported having submitting to NetLogo's community models page, which allows for sharing but neither collaboration nor cooperation, or to the

OpenABM.org site sponsored by the Open Modeling Consortium.

Code collaboration: Most subjects also reported having collaborated with other modelers. One said that he explicitly engaged in pair programming, "Yeah, we won't be passing the file back and forward, or putting it on a web site and saying, 'Hey, you have a go now. Let me know when you're finished or anything like that.' We'll actually just be sitting at the screen together. That would be the most common mode for me, anyway."

Another indicated that while he and his colleagues often worked from home, they would also work from a shared physical office, which allowed them to discuss issues they encountered in their respective models. "So we had a management team in place and whatnot. But we couldn't have done that if we couldn't get together and hash things out or argue them or whatnot. It would have been a huge impediment towards progress, you know." When asked to clarify what he meant by "hash things out," he pointed to the interdependent nature of his collaboration, in that they would debate and discuss the most appropriate way to implement the model. "A lot of the meetings were just figuring out the best way to do it, what sort of formula. How you take a simple situation where you have two agents[1] meet, and one has one opinion, and the other has another opinion. And you say well, how does each change the opinion of the other, you know?"

Cooperation: There were also examples of cooperation — that is, parallel development tracks that were combined toward the end of the project, without a large degree of interdependence during model development. In several cases, this meant using NetLogo's ability to read "NLS files," which make it possible to break a single model into a number of separate files. Using such files make it easier to reuse functionality across multiple models and to split tasks among multiple people, in addition to improving the readability of the NetLogo code.

One subject described how he begins with a mock "dummy" model, which then loads and executes the NLS files that the groups are suppose to develop: "And so the big model will say initialize and it then calls initialize for each of the pieces in those NLS files and then it will say go forth and behave […] And as long as those definitions stay the same I can take my NLS file within the context of these other dummy ones and just edit my one file." In other words, this subject uses the NLS-file capability of NetLogo to allow modelers to work in parallel, even though there are some things that cannot be parallelized.

Domain collaboration: The above description does not reflect the variety of collaborative styles in which subjects worked, or with whom they collaborated. All of the subjects reported that they also collaborated with domain experts, who could verify the accuracy of the model that they had developed, but who didn't work directly on the model. These domain experts often had little or no understanding of modeling, let alone of NetLogo — but their expertise made it possible to write and write better models. One subject said that he tried to let everyone focus on the thing that they do best: "Let's just work together and I'll do my thing and you'll do your thing and together we'll have a peanut butter and jelly sandwich that's delicious with my peanut butter and your jelly."

Another subject described his experience collaborating with a domain expert. The modeler would write the NetLogo code, while the expert would describe the structure of the model and the rules that determined how the various agents interacted. The modeler distinguished between

[1] In NetLogo, the basic elements are "agents," computerized objects that can be given commands. These commands can range from changing their location to visual attributes (e.g., color or size), to what rules they should follow when they encounter another agent.

programming in NetLogo and the model, saying that the expert's comments weren't "specifically related to any correction we got in the NetLogo — it was more about the modeling."

Several subjects reported the crucial role played by those who could bridge the gap between the modeler and the domain. One indicated that it was significantly easier to work with domain experts who had at least a basic understanding of programming and computers: "You know, we talked to a sociologist or the economist or something, and they'd give general outlines. We were very lucky to have a sociologist with a mathematics background first of all. That's an important point, because he thought in computational terms."

Another subject described that when his students are given a modeling assignment, one is typically assigned the role of programmer, while the others learn about the subject and become the domain experts. He added that creating a successful model requires more than just technical skills: "We're not just looking for people to demonstrate technical proficiency. We're wanting a question and a model that's built to address that question."

Face-to-face interaction: In all of these cases — sharing, cooperation, and collaboration — subjects largely preferred to work face-to-face with others, but remote access and interactions were not uncommon. One subject said that he normally collaborates in person but that "once a model is reasonably mature, we might go to separate places and pass it back and forward a bit via email." In several cases, subjects met in person with domain experts because such experts were less familiar with computer-based modeling in general, and with NetLogo in particular. Several subjects described sharing models with others via e-mail, but because running a NetLogo model requires the installation and use of the entire NetLogo environment, this was seen as a barrier to entry. One subject said, "I think it's painful in the distance. I mean, that's a problem," adding, "I think it was best to be there and work with him because that's my way of expression."

Iterations and versions: Several subjects indicated that when they are modeling, they work in small increments, making improvements in each iteration. Along the way, they create many versions, which they typically save in files containing a manually determined version number, allowing them to revert to previous versions as necessary. One said, "I've written a lot of derivations of the same model. Now you know, by the time you get to the end, it doesn't look like anything that in the beginning. So I guess from start to finish I have probably done 50 different versions of three different models." Several subjects also mentioned that they often create a family of related models, each of which is a slight variation on the theme.

Discussion

From the beginning, our work on the Modeling Commons has been driven by an interest to better understand the ways in which NetLogo modelers interact with others during the modeling process, and to offer a platform that both facilitates existing interactions and encourages new ones. The interviews validated many of our findings and design decisions to date, indicating that our work on the Modeling Commons does seem to answer many of the needs of NetLogo modelers. However, the data also suggest additional forms of interaction that could benefit social interactions within the Modeling Commons.

On the specific subject of collaboration, the Modeling Commons software seems to offer many solutions to problems and issues that the subjects described: It provides an easy-to-use mechanism for sharing NetLogo models, either privately or publicly. It supports, and even encourages, the rapid creation of many iterations of models, as well as different related versions of the same model, both of which were cited by a number of subjects as part of their development

process. It allows modelers to ask questions of one another, and to collaboratively edit models, selectively granting and revoking permissions to particular individuals if the model is not yet ready to be released publicly.

Certain elements of NetLogo development are not currently supported by the Modeling Commons, and these interviews pointed to several of these elements that have now been given higher priority. One of these is the use of NLS files, which make it possible for multiple users to work in parallel with one another, while avoiding the need to cut-and-paste code from one user's computer to another.

But perhaps the most important finding from these interviews, was what we have termed "domain collaboration," between modelers and domain experts. Our work to date, as well as research literature, sees "collaboration" as a single type of activity. However, our interviews found, time and again, that modelers collaborate with two different types of colleagues, and have different types of interactions with them: Fellow modelers, with whom they may develop and improve a model, and domain experts, who provide feedback on the validity of the model based on existing theories and evidence. Some people are certainly both modelers and domain experts, but according to our interviews, this occurs in the minority of cases. Most of the time, domain experts are interested in seeing the model succeed, and in helping it to develop and grow — but they are not involved in the day-to-day development of the model.

If we see collaboration as "interdependence," to use Salomon's term, then the person implementing a model is interdependent with a domain expert. The domain expert cannot create a model alone, but neither can the modeler create the model without a domain expert. Each needs the other, and the modeling process consists of many iterations of development followed by feedback from the expert.

In this way, we see that "sharing" is not merely one type of interaction that a modeler has with his or her peers, but an activity that occurs between iterations of a model's development. Sharing a model with a code collaborator offers the chance to improve the model's implementation but without changing the theory that drives the model. Sharing a model with a domain collaborator will almost certainly result in code changes, but only inasmuch as the theory requires.

It would thus seem that merely referring to "collaboration" does not adequately reflect the distinct types of interactions that we can expect to see in modelers' interactions. We have begun, in our own work, to distinguish between "code collaboration" and "domain collaboration" as two distinct types of interaction, each requiring its own form of support.

We could have used a term such as "expert verification" to describe interactions between the modeler and the domain expert. Given the deep, extensive, and intertwined nature of the work done by modelers and domain experts, we believe that it is fair and appropriate to use the term "collaboration" in our description.

From the interviews, as well as from previous design iterations on the Modeling Commons software, we believe that the Modeling Commons offers adequate support for code collaboration. If we were merely interested in facilitating the development of software, then that might be sufficient. But as researchers working to encourage and advance the state of modeling, we believe that support for domain collaboration will expand the potential audience of the Modeling Commons. Just as the Modeling Commons software offers a variety of communication channels — forums, tags, and collaborative modeling — for code collaboration, it must also provide tools for domain collaboration.

One such channel does already exist, namely the discussion forum attached to a model. Since the

Modeling Commons was opened to the public, we have seen some limited use of that forum, in which users did discuss the implementation details of models. However, additional design and features aimed at domain collaborators will distinguish the Modeling Commons as a place for modeling, rather than just for programming, offering tools for where concepts and coding intersect.

We are also considering the addition of a model-specific wiki page, editable by anyone with write-permission for the model. This would provide a space in which authors can communicate, outside of the discussion forum, on the issues related to the model. Offering a free-form space in which to suggest and consider ideas, specifically for collaborators who are not programmers, may encourage deeper online collaboration than is currently possible.

Another possibility is a concept-mapping tool, which would allow domain experts to explicitly diagram, categorize, and describe the various ideas involved in a particular model, for the joint benefit of the domain expert and of the modeler who must translate such ideas into programming code.

Another option would be to provide, via the Modeling Commons, an interface to the BehaviorSpace program that comes with NetLogo. BehaviorSpace makes it possible to run a model many times, varying the parameter values with each run, to better understand how the model will run in response to different inputs. BehaviorSpace can thus provide domain experts with feedback on the results of implementing their ideas, without having to modify the code or ask a programmer for assistance.

Finally, improving the social tagging system that is currently in place will make it easier for domain experts to find models in their field, and then to help modelers to improve them.

Conclusions

We undertook this study before officially launching the Modeling Commons, in order to understand the current state of social interactions among NetLogo modelers without use of our platform. We believe that these findings confirmed our previous work, and that the design of the Modelling Commons will support many of the interactions and needs of NetLogo modelers.

That said, other findings indicated that NetLogo modelers regularly consult with domain experts, often during the modelling process itself. This "domain collaboration," as we have described it, is an important part of modelling, and helps to distinguish modelling from simple programming. These findings indicate that we need to think more about providing affordances for expert-coder interactions, such that the Modelling Commons will be useful not only for the programmer-modelers, but also for the domain experts who play a critical role.

We expect to implement a number of new features to support such interactions in the near future, and will monitor use of these features, using both the system's automated logfiles and by interviewing users, to learn if they have made it easier to collaborate with domain experts.

Acknowledgements

We acknowledge and thank the interview subjects who generously gave of their time, and provided insights into their modelling techniques and work habits. We also appreciate feedback from members of the CCL, who have provided encouragement and feedback throughout the Modeling Commons project.

References

Beck, K. (2000). *Extreme Programming Explained: Embrace Change.* New York: Addison-Wesley.

Bollen, L., Hoppe, H. U., Milrad, M., & Pinkwart, N. (2002). *Collaborative Modeling in Group Learning Environments*. Proceedings from XX International Conference of the System Dynamics Society, Palermo, Italy.

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.

Cockburn, A., & Williams, L. (2001). The Costs and Benefits of Pair Programming. In G. Succi & M. Marchesi (Eds.), *Extreme Programming Examined* (pp. 223-243). Pearson Education.

Collins, A. (1992). Toward a design science of education. In E. Scanlon & T. O'Shea (Eds.), *New directions in educational technology.* Berlin: Springer-Verlag.

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

Dillenbourg, P. (1999). What do you mean by "collaborative learning"? In *Collaborative Learning: Cognitive and Computational Approaches* (pp. 1-19). Oxford, UK: Elsevier.

Epstein, J. M. (2008). Why Model? *Journal of Artificial Societies and Social Simulation*, *11*(4), 12.

Ginsburg, H. P. (1997). *Entering the child's mind: The clinical interview in psychological research and practice.* New York: Cambridge University Press.

Hollan, J. D., & Stornetta, S. (1992). *Beyond being there*. Proceedings from CHI, New York.

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. M., Levy, S. T., & Wilensky, U. (2010a). *Encouraging Collaborative Constructionism: Principles Behind the Modeling Commons*. Proceedings from Constructionism 2010, Paris, France.

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

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.

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*.

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.

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

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