Of Black and Glass Boxes: Scaffolding for Doing and Learning

Cindy E. Hmelo
Mark Guzdial
College of Computing
Georgia Institute of Technology
Atlanta GA 30332-0280
{ceh, guzdial}@cc.gatech.edu

ABSTRACT: Constructivist approaches to education advocate situating learning in real-world problems-- but these problems are complex. Providing scaffolding allows learners to deal with a problem's complexity and successfully solve and learn from these kinds of problems. In this paper, we describe a theory of learning-by-doing and suggest several ways that the "doing" can be supported without forgetting the learning. We use the metaphors of black-box and glass-box scaffolding to suggest how scaffolding might be used to support learning and performance. *Black-box scaffolding* is scaffolding that facilitates student performance. Black-box scaffolding performs a task in place of the student performing that performance goal, usually because learning to perform that goal is determined to be unimportant for the learning goals of the activity. *Glass-box scaffolding* is scaffolding that facilitates performance and learning. It is important for the student to understand what glass-box scaffolding is providing because we want the student to be able to take on the functions that the glass-box scaffolding is providing. Finally, we illustrate how glass-box and black-box scaffolding have been used in two educational software programs.

Introduction

Constructivist approaches to education advocate situating learning in complex, real-world problems [CTGV, 1993; Collins, Brown, & Newman, 1989]. These problems are motivating and more likely to produce transfer than typical end-of-the chapter school problems. However, solving these kinds of problems is hard. Such problems involve multiple subgoals and require a wide range of knowledge and skills for successful solution [CTGV, 1993; Guzdial, et al., 1996]. Traditional schooling has often simplified the problems that learners are asked to solve, however some researchers argue that this is the wrong approach. Instead, they advocate supporting students in dealing with complexity [Spiro, Feltovich, Coulsen, & Anderson, 1989]. This kind of support is known as *scaffolding*. In this paper we propose a theory of scaffolding that makes a distinction between two kinds of support that can be provided for learners. By scaffolding, we mean the help that is provided to learners so that they can accomplish a task they could not otherwise accomplish without that help [Collins et al., 1989]. We use the metaphors of black boxes and glass boxes to refer to scaffolding that supports performance and scaffolding that supports learning.

The problem of supporting students in learning while engaged in authentic problem-solving activity is an example of the general notion of learning-by-doing. How does one support students in learning as part of some activity? In addition, how, in this setting, can one ensure that the learning is not forgotten in the process of completing a task or solving a problem [Bereiter & Scardamalia, 1989]? In this paper, we address first a theoretical framework for learning through doing and introduce the concepts of *scaffolding* in general, and *software-realized scaffolding* in particular. We relate this framework to the problem of facilitating learning through doing by providing some examples of black-box and glass-box scaffolding.

Theoretical Framework for Learning-by-Doing

Constructivist and sociocultural theories inform our views of learning and the ways in which we might provide support for learning [e.g., Bereiter & Scardamalia, 1989; Collins et al., 1989; Lave & Wenger, 1991]. In these views, learning occurs through active construction of knowledge, growth of metacognitive strategies, and an enculturation process. Thus, learning is socially mediated by a community of practice with common tools, discourse, and belief systems. In this view, learning occurs through participating in the authentic activities of the community. There are two approaches that researchers take with respect to describing and realizing learning-by-doing:

- **Doing for learning:** In this perspective, students are engaged in activities that help them to achieve learning goals. Ng & Bereiter [Ng & Bereiter, 1995] discuss the kinds of goals that students take on in this perspective. Typically, students have task completion goals (just get it done with as little learning as possible), instructional goals (do it to learn what is expected), and knowledge-building goals (do in order to learn for one's own agenda). The problem with knowledge-building goals is that they are artificial – it is not at all the norm (outside of academia) for people to learn without regard to the relevance of the learning to some activity or performance goal [Ram & Leake, 1995; Sperber & Wilson, 1986]. Indeed, Ng & Bereiter [1995] found that task completion goals and instructional goals were most common and knowledge-building goals were quite rare. School activities often engender task-completion and instructional goals so that students' goals are to complete the worksheets and other activities; learning, if it occurs, is incidental to completing these tasks [Bereiter & Scardamalia, 1989].

- **Learning for doing:** The alternative is to establish a performance or *doing* goal first, and to arrange the context or goal such that learning subgoals are generated from that task and that the learning subgoals are a necessary part of achieving or significantly enhancing the performance. This is how researchers in cognitive science believe that learning goals most commonly are generated in everyday life – from the need to achieve some performance or activity goal [CTGV 1993; Ram & Leake, 1995]. Schank and the researchers at ILS have a methodology for designing *goal-based scenarios* in which learning subgoals related to curriculum objectives arise in the natural course of achieving the performance goal [Schank, Fano, Bell, & Jona, 1994]. In the anchored instruction approach, developed at Vanderbilt [CTGV, 1993], the video-based scenario provides the goal for the students to learn and use mathematical problem-solving techniques. In both these cases, the goal is not just knowledge-building but it is *knowledge-building for action.*

For the students, the difference between these is whether the primary goal is focused on learning versus doing. The former is what school is traditionally about, but the latter is what real work is traditionally about [Soloway, Guzdial, & Hay, 1994]. In school, however, without the right kinds of support, the learning is often forgotten and just getting the problem solved becomes the goal [Doyle, 1983; Newstetter & Hmelo, 1996]. The goal in the latter approach is for students to grapple with an authentic problem, learn through the process of solving this problem, and then develop a solution that is commensurate with one that an expert might develop. A great many performance goals (problems) afford a rich assortment of potential learning subgoals. It is not necessary to introduce simulated or inauthentic performance goals. For example, one can learn math through studying music, with the appropriate supports that instill the learning goals to analyze rhythms, patterns of notes, chords, and so on. While not all problems will support all learning objectives, authentic problems can be found for many learning objectives.

The challenge is to develop supports for open, complex problems that arise in real practice which (1) lead to successfully solving the problems and (2) inculcate the appropriate learning goals to meet objectives. Given a problem, what we can structure for the students (in order to facilitate learning-for-doing) is the process, the supports provided to the student, and the context in which the problem-solving occurs. Students often do not know what process to use, or use an idiosyncratic process [Jeffries, Turner, Polson, & Atwood, 1981; Lave & Wenger, 1991], and unless students perceive that the problem is solvable (with a good process and enough support), a complex problem can diminish motivation more than augment it [Blumenfeld et al., 1991]. The classroom teacher is involved in setting the context but software can help as well. *Software-realized scaffolding* can provide the supports needed to enable students to be successful at learning-for-doing. These are supports provided to a student which define the process such that the student successfully solves the problem and learns in the process.

Software-realized scaffolding

Software-realized scaffolding is based on the educational concept of *scaffolding*. Scaffolding is support which (1) enables a student to succeed where he or she might not without it and (2) facilitates learning to succeed even without the support [Collins et al., 1989; Rogoff, 1990; Wood, Bruner, & Ross, 1975]. Scaffolding has been used to describe how tutors help students problem-solve, how parents teach their children, and how masters teach apprentices a craft [Lave, 1988; Lave & Wenger, 1991]. In general, scaffolding is meant to *fade*: disappearing over time (sometimes returning at points) so that the learner can succeed without the support. Software-realized scaffolding is a set of technological techniques which provide similar kinds of support to students for learning-by-doing.

The research on cognitive apprenticeship discusses three kinds of scaffolding [Collins, et al., 1989; Guzdial, 1993, 1995; Hmelo & Guzdial, 1995]:

> •*Communicating process* has two parts: (1) structuring and sometimes simplifying the process; and (2) presenting the process to students (e.g., modeling it, telling them the steps, etc.) Structuring the process means defining the stages of an activity whereas presenting it involves explicitly providing the students with the stages of an activity. This can be accomplished by means of checklists, or in software, by menus or screen design.
>
> •*Coaching* is providing guidance to the student while they are performing a task. In a traditional apprenticeship, it is the feedback and suggestions that the master provides to the apprentice as s/he is performing the task. This can be accomplished by highlighting critical steps of the process as the student is working on a problem.
>
> •*Eliciting articulation* is asking the student to explain (to their self or to others) in order to encourage reflection. Encouraging students to reflect helps prepare them to be able to transfer the knowledge and skills they are learning. This may involve asking the students questions or by having the students work in collaborative groups where they need to discuss the project they are working on.

Software-realized scaffolding provides scaffolding in computer-based learning environments. We talk about two kinds of software-realized scaffolding: Black-box and glass-box. *Black-box scaffolding* is scaffolding that facilitates student performance (more than learning), but which may not fade during use of the environment. Black-box scaffolding performs a task in place of the student, usually because learning to perform that goal (or even to understand that the goal is being met) is determined to be unimportant for the learning goals of the activity. For example, a structured editor for a programming language can be black-box scaffolding if it is determined that generating language syntax is not an important learning goal. Black-box scaffolding fades if the student ever performs the activity without the environment. It simplifies the process but does not increase the student's understanding of it.

Glass-box scaffolding is scaffolding that facilitates performance and learning but is meant to fade during use of the environment. It is important for the student to understand what glass-box scaffolding is providing because we want the student to be able to take on the functions that the glass-box scaffolding is providing. Glass-box scaffolding can include several different kinds of supports including prompts for self-explanations, performance supports that are meant to fade (e.g., a structured editor might fade in favor of a traditional program editor if learning syntax is an objective), collaborative environments, intelligent agents (as guides and coaches), and representations (e.g., column and row headings, prompts for representation elements, etc.). Glass-box scaffolding is provided to allow students to focus on one set of learning goals before dealing with other, lower-level subgoals. Table 1 provides examples of black-box and glass-box scaffolding that communicate process, provide coaching, and elicit articulation.

Glass-box scaffolding leads to, what Salomon, Perkins, and Globerson [Salomon, Perkins, & Globerson, 1991] refer to as, *effects of* technology-- that is the learner has a "cognitive residue" from the experience. By observing the model of a knowledgeable coach, seeing implicit processes made explicit, and being encouraged to reflect on those processes the learner is able to construct new knowledge structures and modify existing structures. Black-box scaffolding provides effects with technology– the learner can do things that could not otherwise be accomplished but once the support is removed, the problem has been solved, but the learner has not learned anything about the process. Cognitively, this allows the individual to offload part of the task and may afford higher level accomplishments [Salomon, 1993]. For example, by having a computer perform a repetitive but complex calculation, the learner can allocate cognitive resources to understanding why the calculation is being performed.

It is a critical point that the choice between black-box and glass-box scaffolding is a *curricular* and design decision, not a technical concern. One uses black-box scaffolding to limit students to an efficient path, one where the search space is constrained and there are few unproductive paths. Glass-box scaffolding limits students to a shortened path for efficiency's sake, but with the goal of more exploration later.

	Glass-box Scaffolding	**Black-box Scaffolding**
Communicating Process	• Apple Guide: Explains what to do to accomplish a task and why.	• Menu systems: Do not make evident why items are disabled.
Coaching	• Critics that explain rationale or support argumentation/extension [Fischer, Lemke, McCall, & Morch, 1991].	• Wizards in Excel, Clarisworks, etc.: Accomplish tasks under your direction, but do not tell you how to do them *without* the coaches – the scaffolding is opaque and not meant to fade.
Eliciting Articulation	• CSILE: Students choose the metacognitive prompts which they will then use to label their notes--students understand that this is to help them understand the role of their notes in their learning [Scardamalia et al., 1989].	• Summary prompts in Microsoft Word: No explanation for why a summary (and other articulations) are being requested.[1]

Table 1. Examples of glass-box and black-box scaffolding

Supporting Learning by Doing: Examples of Software-realized scaffolding

In this section, we describe examples of how black-box and glass-box scaffolding have been implemented in two pieces of software. One example is a clinical simulation in critical care medicine, and the other is an example of helping students program simulations in physics.

In *CLINSIM* [Hmelo, 1985], the learner is in the position of managing a critically ill child and had to troubleshoot various problems during the course of a computer simulation. Students had two learning goals: (1) to learn the clinical management techniques for a patient with a head injury and (2) to learn how to troubleshoot the life support equipment that is used in the management. The performance goal was to maintain or improve the patient's condition. Both the learning and performance goals involved iterative sequences of information gathering and decision-making or troubleshooting actions.

> •*Example of glass-box scaffolding in CLINSIM:* Understanding how to make the decisions involved was important for the learning subgoals so glass-box scaffolding was provided. If the learner had no idea of what to do at a particular decision point, s/he could call upon a friendly software "supervisor." The supervisor provided advice about a systematic way to determine what the problem was. For example, if the problem was a leak in the equipment, the software supervisor would suggest "Start

[1] Summary articulations can be used in searches, but there is no explanation or support to facilitate the fading.

with the patient end of the circuit and follow it all the way back to the humidifier and finally bypass the humidifier." An explanation of this follows, thus it is *glass box scaffolding*. The relevant process is communicated to the learner by a coach (in a manner that is very similar to what happens in clinical apprenticeships in the health professions). Similarly, if the learner did not have a clue as to how to begin gathering information, s/he could request help from a supervisor who could help coach and communicate the commonly accepted information gathering process.

•*Example of black-box scaffolding in CLINSIM:* Prior to any decision point in the simulation, the learner needs to gather information to determine the patient or equipment status. Learners could query the system for the patient information that was likely to be available in an actual clinical situation. Although the learner did not need to get all the available patient information to make a reasonable decision, critical pieces were needed. If learners did not request the critical information, it was provided anyway, with feedback noting the additions. The justification for why that information was critical was *not* provided thus it is black-box scaffolding. Learning to identify critical patient information was not a learning goal, so the information was provided to enhance performance and the scaffolding could not be faded.

Emile provides another instance of the use of glass-box and black box scaffolding. Emile was an environment to support students in learning physics and programming by building simulations [Guzdial, 1995]. The student's goal was to build interesting simulations. They referred frequently to their prospective audience as they designed and built their simulations, often adding features based on what their friends might like in their software. The learning subgoals required to meet this high-level performance goal include (1) the physics necessary to create a realistic simulation and (2) the programming knowledge necessary to realize their performance goal. Emile included extensive scaffolding to support students in achieve these goals.

• *Example of glass-box scaffolding in Emile:* Students begin programming simulations in Emile by assembling code fragments called *actions* into complete programs. Students were provided with an extensive library of actions to use in constructing their programs. Each action could be manipulated as a component: Saving, copying, moving into a desired position. Emile also supported fading the scaffolding: (1) Students could choose (by setting a preference) to create their own actions and (2) students could further choose to edit program lines of code directly rather than manipulate actions. The fading enabled students to explore and learn programming at more sophisticated levels to simulate more advanced physics than was supported by the library.

•*Example of black-box scaffolding in Emile:* The code that students actually ran was not exactly the code that they had written. Rather, the code was annotated to provide debugging features such as program tracing and stepping. However, the annotations were completely invisible to the students and could not be faded (turned off). While one could imagine an alternative version of Emile where students could inspect, manipulate, and perhaps even construct their own debugging supports in order to learn more about debugging, debugging skills were not part of the curricular goals for Emile. Thus, it was sufficient to leave the debugging supports as black-box scaffolding.

Conclusion

Our continuing research involves understanding how students learn through solving authentic problems and developing learning environments that support this constructive activity. Learning through problem solving helps motivate learners and enable them to construct deep understanding and transferable knowledge and skills [Guzdial et. al., 1996]. Trying to support the doing while not losing sight of the learning is an important issue for designers of learning environments. We tackle that issue by focusing on the kind of scaffolding support that can be provided in software by using the metaphors of glass-box and black-box scaffolding to describe ways of supporting learning and performance. Deciding what kind of support to provide depends on the learning goals being emphasized. Black-box scaffolding is used to allow the learner to complete a task and allow cognitive resources to be concentrated on higher-level goals. Glass-box scaffolding allows the learner to look inside the support being provided and understand what the support is and why it is needed. We are continuing to evolve a theory of learning-by-doing in the hopes of developing design principles for software-realized scaffolding.

References

[Bereiter & Scardamalia, 1989] Bereiter, C., & Scardamalia, M. (1989). Intentional learning as a goal of instruction. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser*, (pp. 361-392). Hillsdale NJ: Erlbaum.

[Blumenfeld et al., 1991] Blumenfeld, P., Soloway, E., Marx, R., Krajcik, J., Guzdial, M., Palincsar, A. (1991). Motivating project-based learning: Sustaining the doing, supporting the learning. *Educational Psychologist. 26*, 369-398.

[CTGV, 1993] CTGV. (1993). Anchored instruction and situated cognition revisited. *Educational Technology, 33*(3), 52-70.

[Collins et al., 1989] Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. B. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser*, (pp. 453-494). Hillsdale NJ: Erlbaum.

[Doyle, 1983] Doyle, W. (1983). Academic work. *Review of educational research, 53*(2), 159-199.

[Fischer, Lemke, McCall, & Morch, 1991] Fischer, G., Lemke, A., McCall, R., & Morch, A. I. (1991). Making argumentation serve design. *Human Computer Interaction Journal, 6*(3-4), 393-419.

[Guzdial, 1995] Guzdial, M. (1995). Software-realized scaffolding to facilitate programming for science learning. *Interactive Learning Environments, 4*, 1-44.

[Guzdial, 1993] Guzdial, M. J. (1993) *Emile: Software-realized scaffolding for science learners programming in mixed media.* Unpublished Ph.D. dissertation, University of Michigan.

[Guzdial et al., 1996] Guzdial, M., Kolodner, J. L., Hmelo, C., Narayanan, H., Carlson, D., Rappin, N., Hübscher, R., Turns, J., & Newstetter, W. (1996). Computer support for learning through complex problem-solving. *Communications of the ACM, 39*, 43-45.

[Hmelo & Guzdial, 1995] Hmelo, C. & Guzdial, M. (1995). Software-realized scaffolding: A case study of McBAGEL. Presented at Fourth International Workshop on Human and Machine Cognition, Seaside, FLA.

[Jeffries, Turner, Polson, & Atwood, 1981] Jeffries, R., Turner, A. A., Polson, P. G., & Atwood, M. E. (1981). The processes involved in designing software. In J. R. Anderson (Eds.), *Cognitive Skills and Their Acquisition* Hillsdale, NJ: Lawrence Erlbaum Associates.

[Hmelo, 1985] Hmelo, C. E. (1985). *A computer simulation to teach clinical problem solving skills.* Unpublished masters project. State University of New York at Stony Brook, Stony Brook NY.

[Lave, 1988] Lave, J. (1988). *Cognition in Practice.* New York: Cambridge University Press.

[Lave & Wenger] Lave, J., & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation.* Cambridge, UK: Cambridge University Press.

[Ng & Bereiter, 1995] Ng, E. & Bereiter, C. (1995). Three levels of goal orientation in learning. In A. Ram & D. B. Leake (Eds.), *Goal-Driven Learning* (pp. 354-370). Cambridge, MA: MIT Press.

[Newstetter & Hmelo, 1996] Newstetter, W., & Hmelo, C. (1996). Distributed expertise: How students don't. To appear in *Proceedings of the Second International Conference on the Learning Scienees.*

[Ram & Leake, 1995] Ram, A., & Leake, D. B. (1995). Learning, goals, and learning goals. In A. Ram & D. B. Leake (Eds.), *Goal-Driven Learning* (pp. 1-37). Cambridge, MA: MIT Press.

[Rogoff 1990] Rogoff, B. (1990). *Apprenticeship in thinking: Cognitive development in social context.* New York: Oxford University Press.

[Salomon, 1993] Salomon, G. (1993). No distribution without individual cognition: A dynamic interactional view. In G. Salomon & D. Perkins (Eds.), *Distributed cognitions*, . New York: Cambridge.

[Salomon, Perkins, & Globerson, 1991] Salomon, G., Perkins, D. N., & Globerson, T. (1991). Partners in cognition: Extending human intelligences with intelligent technologies. *Educational Researcher, 20*, 2-9.

[Scardamalia & Bereiter, 1991] Scardamalia, M. & Bereiter, C. (1991). Higher levels of agency for children in knowledge building: A challenge for the design of new knowledge media. *Journal of the Learning Sciences, 1*, 37-68.

[Scardamalia et al., 1989] Scardamalia, M., Bereiter, C., McLean, R., Swallow, J., & Woodruff, E. (1989). Computer-supported intentional learning environments.*Journal of Educational Computing Research,5*, 51-68.

[Schank, Fano, Bell, & Jona, 1994] Schank, R. C., Fano, A., Bell, B., & Jona, M. (1994). The design of goal-based scenarios. *Journal of the Learning Sciences, 3*(4), 305-346.

[Soloway, Guzdial, & Hay, 1994] Soloway, E. Guzdial, M., & Hay, K. (1994). Learner-centered design: The challenge for HCI in the 21st century. *Interactions. 1(2)* , 36-48.

[Sperber & Wilson, 1986] Sperber, D., & Wilson, D. (1986). *Relevance: Communication and Cognition.* Cambridge, MA: Harvard University Press.

[Spiro et al., 1989] Spiro, R. J., Feltovich, P. J., Coulsen, R., & Anderson, D. K. (1989). Multiple analogies for complex concepts: Antidotes for analogy-induced misconception in advanced knowledge acquisition. In A. Ortony & S. Vosniadou (Eds.), *Similarity and analogical reasoning.* (pp. 498-531). NY: Cambridge.

[Wood, Bruner, & Ross, 1975] Wood, D., Bruner, J. S., & Ross, G. (1975). The role of tutoring in problem-solving. *Journal of Child Psychology and Psychiatry, 17*, 89-100.

Acknowledgements

We thank Louis Gomez for suggesting the glass-box metaphor. Research reported here has been supported by ARPA under contract N00014-91-J-4092 monitored by ONR, ONR under contract N00014-92-J-1234, by the Woodruff Foundation's support of the EduTech Institute, and by an NSF Career Award RED-9550458 to the second author.