ResearchGate

Scaffolding Complex Learning: The Mechanisms of Structuring and Problematizing Student Work

Article *in* Journal of the Learning Sciences · January 2004

DOI: 10.1207/s15327809jls1303_2

1 author:

Brian J Reiser
Northwestern University

THE JOURNAL OF THE LEARNING SCIENCES, *13*(3), 273–304

Scaffolding Complex Learning: The Mechanisms of Structuring and Problematizing Student Work

Brian J. Reiser
School of Education and Social Policy
Northwestern University

There has been much interest in using software tools to scaffold learners in complex tasks, that is, to provide supports that enable students to deal with more complex content and skill demands than they could otherwise handle. Many different approaches to scaffolding techniques have been presented in a broad range of software tools. I argue that two complementary mechanisms can explain how a diversity of scaffolding approaches in software act to support learners. Software tools can help structure the learning task, guiding learners through key components and supporting their planning and performance. In addition, tools can shape students' performance and understanding of the task in terms of key disciplinary content and strategies and thus problematize this important content. Although making the task more difficult in the short term, by forcing learners to engage with this complexity, such scaffolded tools make this work more productive opportunities for learning. I present arguments for these mechanisms in terms of the obstacles learners face, and I present several brief examples to illustrate their use in design guidelines. Finally, I examine how the mechanisms of structuring and problematizing are sometimes complementary and sometimes in tension in design, discuss design tradeoffs in developing scaffolded investigation tools for learners, and consider the reliance of scaffolding on a classroom system of supports.

There is much interest in education reform in using technology to support learners. One aspect of the argument for technology has been that software can be used to help learners succeed in more complex tasks than they could otherwise master (Davis &

Correspondence and requests for reprints should be sent to Brian J. Reiser, School of Education and Social Policy, Northwestern University, 2120 Campus Drive, Evanston, IL 60208. E-mail: reiser@northwestern.edu

Linn, 2000; Edelson, Gordin, & Pea, 1999; Guzdial, 1994; Quintana, Eng, Carra, Wu, & Soloway, 1999; Reiser et al., 2001). Researchers have invoked the notion of *scaffolding,* a construct originally crafted to characterize how more experienced peers or adults can assist learners. As defined and used in early research, scaffolding is said to occur when a more knowledgeable person helps a learner succeed in tasks that would be otherwise beyond their reach (Wood, Bruner, & Ross, 1976). In the last two decades of learning sciences research, scaffolding has become increasingly prominent. Scaffolding is a key strategy in cognitive apprenticeship, in which students can learn by taking increasing responsibility and ownership for their role in complex problem solving with the structure and guidance of more knowledgeable mentors or teachers (Collins, Brown, & Newman, 1989).

Many different approaches to scaffolding have emerged from the design research on interactive learning environments, and a variety of design guidelines or principles have been proposed (Edelson et al., 1999; Guzdial, 1994; Kolodner, Owensby, & Guzdial, 2004; Linn, 2000; Reiser et al., 2001). To engage in principled development and empirical study of design guidelines requires greater clarity concerning what is meant when one says that a tool has scaffolded learners, and requires a model of how the tool has benefited learners. In particular, it is important to characterize the mechanisms by which a software tool can provide scaffolding for learners. Developing a common system of design guidelines for scaffolded software requires such a model of mechanisms that explain why a tool reflecting these guidelines would benefit learners.

In this article, I present an analysis of two general mechanisms to characterize how scaffolded tools can support learning. I describe how these dual mechanisms can address the challenges learners face by structuring tasks to make them more tractable and to shape tasks for learners in ways that makes their problem solving more productive. I develop the argument for these mechanisms by first considering how tools affect the experience of tasks for learners. Then I review some of the critical challenges learners face in complex domains such as science and mathematics learning. In describing each mechanism, I present brief examples of software environments to illustrate the mechanisms in practice. Finally, I consider how the mechanisms can interact and discuss issues of the embedding of tools in classroom contexts.

TRADITIONAL APPROACHES TO SCAFFOLDING

To consider how software tools can scaffold learners, I first review the source of the scaffolding metaphor. The term *scaffolding* has traditionally been used to refer to the process by which a teacher or more knowledgeable peer assists a learner, altering the learning task so the learner can solve problems or accomplish tasks that would otherwise be out of reach (Collins et al., 1989; Wood et al., 1976). The central component of this definition is that another person intervenes at times appropriate for that learner in that context, and what the learner can accomplish increases with these interventions. For example, a teacher may help a child in a board game by reminding him or her of the rules or by suggesting strategic steps if the child is stuck. The conception is associated with Vygotsky's (1978) notion of the zone of proximal development, which characterizes the region of tasks between what the learner could accomplish alone and what he or she could accomplish (and master) with assistance (Rogoff, 1990).

The idea of scaffolding is now in increasing use in educational design. In these contexts, the intention is that the support not only assists learners in accomplishing tasks but also enables them to learn from the experience. The use of the notion of scaffolding has not always been explicitly limited to learning settings. For example, one might consider an adult providing support to a child for some task (such as observing an animal at the zoo) in which there is no intention that the child learns to perform the task in the future more effectively. For educational settings, it is important to stress the dual aspects of both (a) accomplishing the task and (b) learning from one's efforts, that is, improving one's performance on the future tasks in the process. If learners are assisted in the task but are not able to understand or take advantage of the experience, the assistance will have been local to that instance of scaffolding but will not have provided support for learning. Thus, scaffolding entails a delicate negotiation between providing support and continuing to engage learners actively in the process (Hogan, Nastasi, & Pressley, 1999; Merrill, Reiser, Merrill, & Landes, 1995). Lepper, Woolverton, Mumme, and Gurtner (1993) described this as maintaining an "optimum" level of challenge for learners. I return to the need for balancing assistance with ensuring the work on the task is productive in later discussions of the two scaffolding mechanisms.

Recent design research on interactive learning environments has adapted the notion of scaffolding (Davis & Linn, 2000; Edelson et al., 1999; Guzdial, 1994; Quintana et al., 1999; Reiser et al., 2001). This vision of scaffolding refers to ways the software tool itself can support learners rather than only teachers or peers. As applied to software, scaffolding refers to cases in which the tool changes the task in some way so that learners can accomplish tasks that would otherwise be out of their reach. Software scaffolding provides some aspect of support that helps make the learning more tractable for learners. For example, the software might provide prompts to encourage or remind students what steps to take (Davis & Linn, 2000), graphical organizers or other notations to help students plan and organize their problem solving (Quintana et al., 1999), or representations that help learners track what steps they have taken (Collins & Brown, 1988; Koedinger & Anderson, 1993). In all these cases, the software provides additional assistance beyond what a simpler, more basic tool would have provided to allow learners to accomplish more ambitious tasks. Sherin, Reiser, and Edelson (this issue) argue that software scaffolding should be characterized in terms of the differences the scaffolding creates in comparison to some presumably more difficult reference version of the task.

This work on scaffolded software tools has been very encouraging, and scaffolding promises to be an important benefit in integrating technological tools into classrooms. However, there has been a wide range of approaches to designing software scaffolds and many kinds of design principles. Integrating different scaffolding approaches into a common framework requires an analysis of how scaffolding can occur in the interactions between learners and software tools. How can researchers characterize the mechanisms by which software scaffolding assists learners?

Although there have been many different design principles proposed, I argue that underlying these principles are some common assumptions about how to make more productive learning experiences for students. The focus of this article is to consider design arguments and principles that have been proposed for software scaffolding and to characterize the common mechanisms by which these strategies achieve benefits for learners. Such an analysis of mechanisms is needed to clarify and evaluate what types of scaffolding are effective. In this work, I build on the framework developed by Quintana et al. (this issue). This framework consists of a set of design guidelines and specific design strategies that synthesize design ideas across a range of software tools and grounds these strategies in the types of obstacles learners need to overcome. The scaffolding mechanisms proposed here are meant to explain why these design guidelines work to support learners in mastering complex tasks.

To construct this argument, I focus on scaffolding in the discipline of science. Much of the work on scaffolding tools has taken place in this domain, and there is a rich literature on the obstacles learners face. Furthermore, tools to access and interpret data are a central part of the practices of scientific investigation, so this domain is a productive context in which to explore the design of scaffolded tools.

NEEDS OF LEARNERS

A principled analysis of the manners in which tools can influence learning must begin with an analysis of the needs of learners and the ways that shaping the tool can affect the ability of learners to overcome these challenges. In this section, I briefly consider the challenges of learners in the discipline of science to delineate the opportunities for a software tool to help learners overcome these challenges.

Instructional approaches in science emphasize learning by engaging in knowledge construction practices. In the case of science, this entails learning science through investigation and argumentation (Olson & Loucks-Horsley, 2000). In project-based science, students learn general principles in the context of investigating particular problem scenarios such as learning introductory chemistry by analyzing the quality of air in the local community (Blumenfeld et al., 1991; Edelson, 2001;

Hmelo, Holton, & Kolodner, 2000). In addition to constructing conceptual understanding, students need to acquire new disciplinary strategies to guide reasoning in the domain (Schauble, Glaser, Raghavan, & Reiner, 1991; Tabak, this issue).

These approaches to learning through inquiry, although providing the potential to connect knowledge more effectively to real-world contexts, also pose particular challenges for learners. Quintana et al. (this issue) consider the challenges learners face and organize them around three constituent processes involved in learning through scientific investigation—sense making, process management, and articulation and reflection. Each type of process is challenging for learners.

Sense making entails constructing and interpreting empirical tests of hypotheses. Students need to coordinate their reasoning about experiments or data comparisons with the implications of the findings for an explanation of the scientific phenomena. This coordination and mapping task is complex and requires rich subject matter knowledge to design data comparisons and interpret findings in light of hypotheses (Klahr & Dunbar, 1988; Schauble et al., 1991).

Strategic guidance is critical to manage the complex investigation process. Investigations require an iterative processes of designing an investigation, collecting data, constructing and revising explanations based on data, evaluating explanations, and communicating arguments (Olson & Loucks-Horsley, 2000). This requires strategic knowledge to plan, conduct investigations, and make decisions about next steps based on interim results. These require both discipline-specific processes and content knowledge that may be new to learners.

Finally, investigations require the complementary processes of reflection and articulation as students monitor and evaluate their progress, reconsider and refine their plans, and articulate their understanding as they proceed. These communicative and metacognitive skills pose additional challenges for learners.

Thus, learners face challenges at several levels. Students must master conceptual knowledge, domain process skills, domain-specific strategies, and more general metacognitive processes. In addition to cognitive and metacognitive challenges, these practices include a social dimension, as investigations involve working together in teams, planning and negotiating within a group, communicating, and debating with peers about scientific interpretations (Brown & Campione, 1994). These social practices and the discourse practices they entail are potentially unfamiliar, and pose additional social interaction challenges for learners (Webb & Palincsar, 1996). Next, I consider specific obstacles that arise as learners grapple with these challenges.

Unfamiliar Strategies

Sophisticated problem solving relies on strategies for planning and guiding reasoning. These heuristic strategies in science are needed to plan investigations, select data comparisons, and synthesize findings. These strategies involve general strategies for scientific inquiry (Klahr, 2000; Krajcik et al., 1998; White & Frederiksen, 1998) and discipline-specific explanatory frameworks (Passmore & Stewart, 2002; Reiser et al., 2001; Sandoval & Reiser, in press; Tabak, Smith, Sandoval, & Reiser, 1996). A key challenge is that this knowledge is typically tacit for more experienced reasoners and may be taken for granted. Instruction often fails to make these strategies explicit for learners. Learners experience challenges in using general strategies for designing empirical tests of hypotheses (Klahr, Fay, & Dunbar, 1993) and in using specific domain knowledge to plan and guide investigations (Schauble et al., 1991).

Nonreflective Work

Learners tend to focus on products rather than on explanatory and learning goals (Perkins, 1998; Schauble, Glaser, Duschl, Schulze, & John, 1995). For example, they focus on achieving desired results rather than on understanding the principles behind the results and become distracted by superficial aspects of the products they need to construct (Krajcik et al., 1998). The difficulty in managing investigations leads to insufficient attention devoted to reflection and reevaluation (Loh, 2003; Loh et al., 2001). Lack of content knowledge further complicates the process of evaluating the progress of an investigation. Another aspect to this challenge is that learners may need assistance in generalizing appropriately from their work on specific problem scenarios. For learning through investigation to succeed, students must not only construct solutions to the particular scenario but must connect the explanations or arguments they construct to more general disciplinary frameworks (Williams, 1992).

Fragile and Superficial Understanding

Learners tend to focus on superficial details and have difficulty seeing the underlying structure that is visible with more experience (Chi, Feltovich, & Glaser, 1981). They may have difficulty mapping between their intuitive understandings and more precise scientific constructs and to formal representations that are the medium for representing work in the domain (Reif & Larkin, 1991; Sherin, 2001). Furthermore, they may be too quick to decide interpretations are warranted without sufficient evaluation of alternatives (Klahr, 2000; Kuhn, Amsel, & O'Loughlin, 1988). Learners are not always effective in analyzing whether they have understood and may be overconfident in their self-assessments (Chi, Bassok, Lewis, Reimann, & Glaser, 1989; Davis, 2003).

Unfamiliar Social Interaction Practices

Scientific investigations require practices that include both cognitive and social interaction components. These include constructing scientific arguments to persuade peers, receiving questions and critiques, and improving explanations based on feedback (Bell & Linn, 2000; de Vries, Lund, & Baker, 2002; Driver, Newton, & Osborne, 2000; Kuhn, 1993; Sandoval & Reiser, in press). These processes are typically conducted in classrooms in teams, requiring collaborative planning, negotiation, and self-assessment. This complex collaborative work presents social interaction challenges such as weighing opinions and keeping track of alternatives proposed by all group members, ensuring participation from all group members, and learning to offer and receive critiques (Coleman, 1998; Webb & Palincsar, 1996). The success of group work can be compromised by difficulties in these social interactions (Barron, 2003; Kurth, Anderson, & Palincsar, 2002).

Unfamiliar Discourse Practices

The third component of practices, in addition to cognitive strategies and social interactions, consists of characteristic discourse practices. Discourse plays a particularly central role in the practice of scientific inquiry and places demands on learners (Lemke, 1990). Scientific practices are implemented in particular discourse practices and uses of language—for example, for expressing hypotheses, arguing from evidence, critiquing an idea, and so on—which may be unfamiliar (Rosebery, Warren, & Conant, 1992). The linguistic and cognitive aspects of these practices are interrelated, with important constructs signaled by language and precise language used to communicate about scientific processes (e.g., "support," "argue," "falsify," etc.). There may be a disconnect between some learners' understanding of the practices underlying language use and the scientifically accepted practice (Moje, Collazo, Carrillo, & Marx, 2001; Reif & Larkin, 1991). Learners may need support in using language appropriately and in connecting the language with implementation in scientific practices.

In summary, the task demands of engaging in scientific investigations reveal a system of challenges for learners. Difficulties arise from the cognitive complexity of the practices as well as from new social interaction and discourse challenges. The cognitive complexity arises from unfamiliar general and discipline-specific strategies that are required in sense making and process management. Learners may not articulate their ongoing understanding, focusing instead on pragmatic goals of creating required products. They may need to be prompted to be more reflective and focused on understanding rather than performance and to go beyond superficial solutions to problems. Students may need to become proficient in new practices for social interaction and discourse associated with knowledge building. The focus on public knowledge building and the specialized uses of language it requires pose an ongoing challenge during the investigation. They may need assistance in scientific discourse to move beyond description and communicate a scientific argument.

In the next section, I consider how software tools influence the nature of tasks for learners. With that as a foundation, I then propose specific mechanisms to explain how software scaffolding can assist with the specific challenges identified here.

HOW CAN TOOLS HELP LEARNERS?

Traditional views of scaffolding have focused on interactions with teachers or peers as the source of assistance, articulating how a more knowledgeable person can provide assistance in the context of a task (Hogan & Pressley, 1997; Wood et al., 1976). The focus of the last two decades of research on the learning sciences issues in technology design has illuminated ways in which technological tools may provide some types of scaffolding functions. Instructional designers have investigated how to create tools that can help learners accomplish complex tasks. In considering how to design effective scaffolded tools, it is important to reconceptualize the learning problem from that of an individual working on tasks, perhaps with assistance of another more knowledgeable person, to a consideration of the context in which the people are acting, the tools they use, and the knowledge embedded in this context. Rather than considering what the individual can accomplish, this view of distributed cognition focuses on what a person or group working with tools can accomplish as a system (Hollan, Hutchins, & Kirsh, 2000). The structure of a tool shapes how people interact with the task and affects what can be accomplished. I consider how the nature of the tool can alter the task facing learners.

Tools Can Distribute Work and Reduce What Is Required of Learners

One clear way that tools change the nature of tasks for learners is in automating aspects of tasks and thereby limiting the part of the task the learners need to perform, potentially enabling them to focus on more productive parts of the tasks (Salomon, Perkins, & Globerson, 1991). Salomon et al. considered how the partnership of person and tool can accomplish tasks together that extend beyond what the person could accomplish alone. This is the perhaps the most straightforward sense of scaffolding. For example, calculators can offload simple computations, allowing people to focus on other parts of the data manipulation tasks such as considering what calculations to compose together to solve a problem. Word processors with spelling checkers can allow writers to focus more on the construction of their prose rather than devoting time to checking spelling in dictionaries. The result of offloading aspects of the task may be to reduce the overall complexity. If offloading these aspects of the task allows learners to focus more effectively on the conceptually important aspects and thereby learn from their experience, the tool has scaffolded that learning.

Tools Can Transform Tasks

Tools can have even more dramatic effects in the way they transform the nature of the tasks. In fact, the tools people use can be a critical factor in how people envision and engage in the tasks they perform (Hutchins, 1995; Norman, 1991). This is particularly true when tasks involve accessing, manipulating, storing, or reasoning about information. Norman (1991) described *cognitive artifacts,* or tools that are used to represent and manipulate information in a task. Cognitive artifacts can change the task in fundamental ways. Users translate their intentions into actions to be taken in the tool and construct an understanding of the state of the external world from the representation the tool provides. The tool provides an inscription or encoding of information about the world. In this way, the tool provides the representation of external states and the vehicle for operating on the environment. Because of the central role of tools in effecting actions in information domains, the task cannot be defined independently of the tools that people use in the practice of that task (Bannon & Bødker, 1987).

The framework of distributed cognition has been used to describe these transformative tools (Collins, 1991; Hollan et al., 2000; Pea, 1992; Pea & Gomez, 1992). In this framework, tools affect and may extend what users can do. Tools may make some types of computation unnecessary for users (such as symbolic calculators) or may create inscriptions that encode information in a more usable form. Because cognitive artifacts mediate between people and the world, such tools can transform the task in the way in which they represent and allow people to manipulate information. If the inscription provided by the tool enables useful inferences more effectively, it can extend the range of what users can do. Visualization tools that provide conceptually meaningful representations are designed to help users form deep models of an underlying system (Hollan, Bederson, & Helfman, 1997). For example, studying atmospheric sciences phenomena using tools to access and construct sophisticated visualizations of primary data enables scientists to construct and test conjectures about the phenomena. These specialized inscriptions are designed to enable users to notice patterns in data and fundamentally alter the kinds and level of scientific reasoning they can do about the material (Edelson et al., 1999). Similarly, direct-manipulation interfaces allow users to control a process by appearing to act on it directly through a visual metaphor (Hutchins, Hollan, & Norman, 1986).

The tools can also affect the nature of interactions between collaborators. For example, the structure of the tool may influence the design of the task into constituents, influencing how roles are defined. The structure of the tool may also influence the focus of conversation between collaborators. For example, through its organization of functionality, the tool may focus conversation on particular choices or may highlight certain relations and influence the course of decision making by providing a vehicle for students to articulate aspects of their understanding (Scardamalia & Bereiter, 1994).

Thus, the design of the tool itself is an important constituent in defining tasks. The nature of the task emerges from the interactions of people, subject matter, and tools. The nature of the tool clearly affects how tractable the performance of a task is for learners.

Norman (1991) described the challenges of designing effective tools for complex tasks. Because tools mediate users' interactions with the environment, users need to map between understanding, a tool's representation, and the world it represents. Difficulties in this mapping create challenges and can lead to extra steps or levels of indirection in reasoning. The goal of human–computer interaction design research is to make that mapping between a tool's representation and what it represents "transparent." That is, users should be able to "see" the core meaning in a representation (such as noticing which regions are hotter by comparing red and blue colors on a temperature map) rather than getting bogged down in reasoning to translate from the representation to its underlying meaning. Simplifying this translation can reduce the complexity of the task, minimize errors, and extend what is possible with the tool (Norman, 1993).

In the design of scaffolded tools, instructional designers can use this mapping to an advantage. Rather than striving only for transparency between the representation and the world it represents, designers can bend that representation to instructional purposes. Cognitive artifacts provide a lever for designers to shape how learners think about tasks. In the next section, I consider how cognitive artifacts can be used to instructional advantage.

MECHANISMS OF SCAFFOLDING: STRUCTURE AND PROBLEMATIZE

I have discussed how changing the nature of tools people use can fundamentally transform the task, determining the cognitive and social interaction demands of a task. Now I consider the kinds of transformation involved in tools that are designed to scaffold learners.

How can software tools provide scaffolding? First, I revisit the definition of scaffolding as applied to science. There are two critical notions in scaffolding: (a) learners receive assistance to succeed in more complex tasks that would otherwise be too difficult, and (b) learners draw from that experience and improve in process skills and/or content understanding. Focusing on the representational properties of tools provides a dimension on which to consider how the specific design of the tool can support learning tasks.

I propose two complementary mechanisms of scaffolding in software tools—it can help *structure the task* of problem solving, and it can *problematize subject matter,* and thus provoke learners to devote resources to issues they might not otherwise address. I describe each mechanism and the nature of the scaffolding influence on learners. To illustrate each mechanism in action, I present an example learning environment and its associated design guidelines that rely on that manner of support for learners. I then discuss some of the design tradeoffs and tensions that exist in utilizing these two mechanisms.

The roots of these ideas of structuring and problematizing are in traditional notions of scaffolding as developed with parents, tutors, and teachers. However, although both of these ideas are implicit in the underlying designs of software tools developed in design research, the function of problematizing subject matter has not received much attention in the design arguments and theoretical accounts of software scaffolding. My goal in this article is to help make explicit both of these mechanisms in theoretical accounts of software scaffolding.

Structuring the Task

The first sense of scaffolding is the most common in design research on scaffolding, and it is the most straightforward. If reasoning is difficult due to complexity or the open-ended nature of the task, then one way to help learners is to use the tool to reduce complexity and choice by providing additional structure to the task. For example, this may be done by providing structured work spaces to help learners decompose a task and organize their work or prompts to help learners recognize important goals to pursue.

The notion of supporting problem solving by providing more structure has been a central constituent of scaffolding theories beginning with the analyses of Wood et al. (1976). Wood et al. characterized scaffolding as involving "reduction in degrees of freedom" and support to help learners "maintain direction." The core idea is that by providing structure or constraints, perhaps in the form of explicit direction or by narrowing choices, the complexity facing the learner is reduced and the problem solving is more tractable.

Influence of Structuring Student Work

There are several types of learner challenges for which this approach to structuring work can help learners including decomposing tasks, focusing effort, and monitoring.

Decomposing complex tasks. The tool's interface can be organized to help learners decompose open-ended problems. For example, Model-It™, a tool for helping learners construct qualitative models (represented as causal maps), uses functional modes to structure the task of modeling into plan, build, and test processes and organizes relevant functions in each of those modes (Jackson, Krajcik, & Soloway, 1998). This can help guide what actions to take, their order, or necessary aspects of work products. In this way, the software tool helps overcome some of the obstacles of unfamiliar strategies by helping the learners attend to important goals. It may also help nonreflective work by helping learners keep track of what goals have been addressed and what aspects of the task are pending. Several inquiry support systems use checklists or diagrams to help learners identify and implement aspects of the process they may otherwise neglect to perform (Davis, 2003; Davis & Linn, 2000; Linn, Bell, & Davis, 2004; Quintana et al., 1999). This assistance with decomposition can be a resource groups can use to organize work among themselves.

Focusing effort. Restricting the problem space, for example, by narrowing options, preselecting data, or offloading more routine parts of the task, can help learners focus resources on the aspects of the task more productive for learning. For example, visualization software for learners may include specialized tools that automatically access anchoring reference information (such as names of countries or cities) not typically represented in scientific tools for experts (Edelson et al., 1999). Functionality can be organized to restrict options to those relevant to the learner's current goals, again helping learners focus resources in productive ways (Quintana et al., 1999). Like decomposition, these design strategies can help learners overcome obstacles of unfamiliar strategies by reducing the overload experienced in handling strategic decisions concurrent with managing the implementation of a plan. The support in the tool that helps focus effort may provide a resource to help learners work together more effectively. For example, guidance in structuring the work may help reduce options and ambiguities that groups face when negotiating possible directions (Barron, 2003; Krajcik et al., 1998).

Monitoring. Explicit structures such as prompts, agendas, or graphical organizers can help learners keep track of their plans and monitor their progress. This type of structuring is characteristic of a number of scaffolding approaches. Guidance embedded in software can remind learners of important goals to pursue or criteria to apply to their work. In the Knowledge Integration Environment, prompts can remind learners of important criteria they should apply to their work (Davis, 2003; Davis & Linn, 2000). In these cases, the scaffolded tool addresses the tendency toward nonreflective work by helping learners construct their plans, consider the possible actions relevant to each stage of the process, monitor the plan, and tie in relevant disciplinary ideas as they make sense and communicate about their data.

Examples of Structuring Student Work

The following example illustrates how a software environment can support learners through the scaffolding mechanism of structuring their work. I describe a learning environment and the design guidelines it represents and consider how the guideline acts to structure student work. The goal is to illustrate this mechanism rather than to present an extensive review of types of design guidelines that utilize the mechanism (see Quintana et al., this issue, for a design guidelines framework that synthesizes a broad range of design approaches from the field).

One approach for structuring student work is represented by the design guideline "facilitate ongoing articulation and reflection during the investigation" ("Guideline 7" in Quintana et al., this issue). To see this guideline and its realization of structuring, consider the Biology Guided Inquiry Learning Environments (BGuILE) software tool ExplanationConstructor (Reiser et al., 2001; Sandoval, 1998, 2003; Sandoval & Reiser, in press). ExplanationConstructor is a computer-based science journal in which students construct their scientific explanations. While working on an investigation, students record their research questions and new subquestions as they emerge, construct candidate explanations and associate them with their research questions, and record evidence for each assertion (see Figure 1). Students use ExplanationConstructor in concert with other applications that contain data or a simulation, and it provides a structured work space where they can record the sense they are making of their findings.

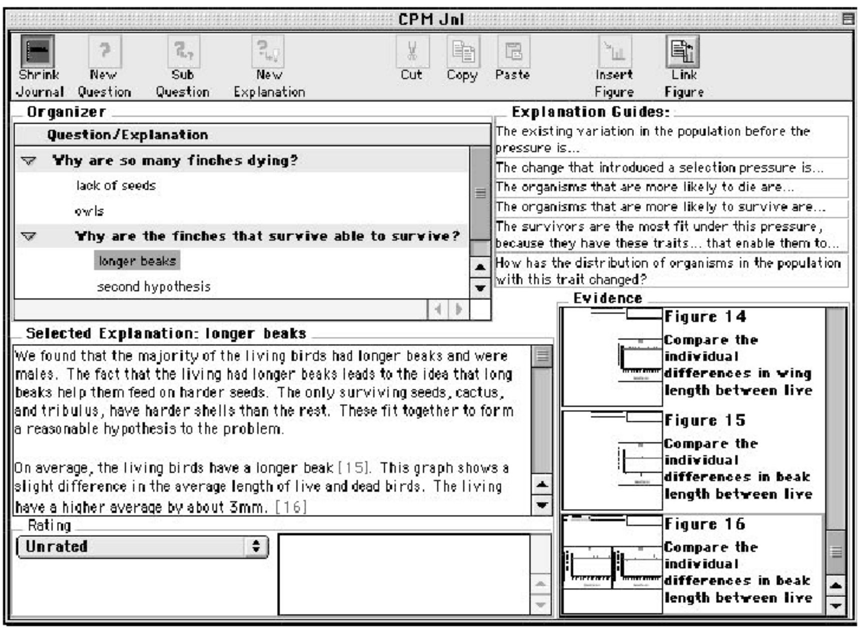

FIGURE 1 The ExplanationConstructor used to articulate questions, explanations, and backing support. The outline of students' questions, subquestions, and explanations is shown in the upper left Organizer panel; Explanation Guides specific to the explanatory framework selected are shown in the upper right.

ExplanationConstructor embodies two kinds of structuring that embody scaffolding strategies for articulation and reflection: provide guidance for planning and guidance for monitoring (Strategies 7a and 7b in Quintana et al., this issue). The environment contains tools geared to help students articulate their research questions and the links between candidate explanations and these questions. Encouraging students to record pending questions and to be explicit about the relevance of their findings for those questions is intended to provide needed structure, encouraging students to record their overall plans and continuously monitor their progress. These are elements of articulation and reflection that students frequently omit in the rush toward producing final products.

Another aspect of the structuring is apparent in the explanation guides that serve as prompts for critical constituents of an explanation. These reflect an additional scaffolding strategy of highlighting epistemic features of scientific products (Strategy 7d in Quintana et al., this issue). These prompts are intended to help students structure their interpretations in light of relevant disciplinary frameworks.

Sandoval and Reiser (in press) analyzed the use of ExplanationConstructor in conjunction with the BGuILE investigation environment Galápagos Finches (Reiser et al., 2001; Tabak et al., 1996) in a high school biology classroom. The Galápagos Finches enables learners to investigate changes in populations of plants and animals in an ecosystem and serves as a platform for learning principles of ecosystems and natural selection. Students can analyze a population across time or according to some dimension of interest such as comparing male to female or young to adult. Students used the Galápagos Finches to analyze data to understand what killed many of the finches in the a population of finches during a crisis period and why some finches were able to survive. They used ExplanationConstructor to help manage the investigation, record their questions, construct explanations, and link in evidence for their claims.

Sandoval and Reiser (in press) described the use of several aspects of the system as productive structure for planning and self-monitoring in the investigations. For example, the list of explanation guides ("existing variation," "change in environment," etc.) and the students' own generated subquestions both served as external aides to help students track what issues their explanation needs to address. These provided an anchor to which students returned to evaluate their in-progress explanations. Students continued to refer to their list of research questions to remind themselves which questions were accomplished and which were pending—for example, "we are still answering that question [pointing to question in journal]." They negotiated what tasks to work on next by revisiting their questions—"How are we gonna answer that?; Where are we gonna go look?" They returned to questions and discovered "we don't know that yet," helping them determine where they needed to focus next in the data. In general, the explanation facilities seemed to provide a structure students could use to construct a representation of their plans, and they used this representation to guide and monitor their own investigation.

Problematizing Aspects of Subject Matter

The second mechanism for scaffolding is to make some aspects of students' work more "problematic" (Hiebert et al., 1996) in a way that increases the utility of the problem-solving experience for learning. That is, the software tools help students see something as requiring attention and decision making that they might otherwise overlook. Rather than simplifying the task, the software leads students to encounter and grapple with important ideas or processes. This may actually add difficulty in the short term, but in a way that is productive for learning.

This notion of problematizing as supporting learners has its roots in the original analysis of scaffolding presented by Wood et al. (1976). Wood et al. characterized part of the functions of scaffolding as focusing attention by "marking critical features" of a task, which may involve highlighting "discrepancies" between what a child might produce and "correct production." The scaffolding function of "maintaining direction" mentioned by Wood et al. may also serve as problematizing, focusing learners on aspects of the task not yet performed. In a way, problematizing is the flip side of structuring—although a structure may be helpful, it also may introduce a cost in reworking one's ideas in terms of the presented framework.

The importance of the mechanism of problematizing has been highlighted in recent studies of teachers' support of inquiry classrooms. In a study of a community of learners classroom, Engle and Conant (2002) found that making subject matter problematic and connecting students' work to disciplinary frameworks are key aspects of a productive discussion-based classroom. Similarly, the teaching strategy of inducing "cognitive conflict" involves drawing attention to problems in students' work in progress (Webb & Palincsar, 1996). In their analyses of expert tutors, Lepper et al. (1993) pointed out that tutors modulate their support to target an optimal level of difficulty. Tutors appear to seek a balance in eliciting the learner's active engagement with the problem and providing enough support to prevent frustration and nonproductive floundering. They refrain from providing too much help that would remove productive complexity or prevent errors that are learning opportunities (Lepper et al., 1993; Merrill et al., 1995). These studies suggest a role for guiding learning not only by simplifying but also in encouraging learners to face some of the complexity of the domain in productive ways.

Problematizing in a learning situation consists generally of several characteristics. First, it involves focusing students' attention on an aspect of a situation that needs resolution. Second, it involves engaging students—eliciting students' commitment of attention and resources to reasoning about an aspect of a problem. This may involve creating a sense of dissonance or curiosity. Finally, it may involve an affective component—creating interest in some aspect of a problem or getting students to care about understanding or resolving an issue.

How can software tools achieve some of these same ends? I discussed earlier how tools can shape users' conceptions of a task. A tool's interface can shape the concepts and language students use to identify actions. This may provide useful structure, but it may also act to problematize important disciplinary constructs. For example, in Belvedere, a software environment in which students create arguments represented as a graph of claims and evidence, students must decide whether each assertion is a claim or evidence and must indicate the relations between claims and evidence in their representation (Cavalli-Sforza, Weiner, & Lesgold, 1994; Toth, Suthers, & Lesgold, 2002). This can focus students on worrying about how a particular idea should be classified, something that simply expressing an argument in the fuzziness of natural language may gloss over.

Despite some suggestions of the importance of engaging students with complexity in early accounts of scaffolding with human teachers, accounts of scaffolding in software have tended to emphasize the structuring aspect of support, making tasks more tractable by reducing difficulty, reducing overwhelming options, or embedding prompts or guides to help learners focus their efforts. There are some discussions in these accounts of providing an optimal level of such support and fading that support when appropriate (Collins et al., 1989; Soloway, Guzdial, & Hay, 1994). However, the notion of problematizing goes beyond avoiding giving too much help or fading the help as learners develop increasing expertise. This core of this approach is to guide the learner into facing complexity in the domain that will be productive for learning, for example, by connecting their work on a problem to disciplinary frameworks.

The general goal of increasing learners' engagement with complex disciplinary ideas is a key aspect of the last few decades of education reforms. Reform ideas such as conceptual understanding versus rote learning, engaging in authentic practices, and learning by doing all create a need to focus learners on unfamiliar but productive disciplinary ideas (American Association for the Advancement of Science, 1990; Bruner, 1966; Cohen, 1988). Although software tools are often designed with the goal of eliciting deeper engagement and reasoning, little attention in theoretical definitions of scaffolding or design arguments has been focused on the problematizing aspect. The mechanisms of structuring and problematizing are an attempt to characterize how software tools can help learners grapple with more ambitious learning goals.

This focus of resources inherent in problematizing can address the problems raised earlier of nonreflective work and superficial analyses, encouraging and requiring students to address critical questions and ideas in the discipline. In these cases, requiring students to connect their thinking to disciplinary issues can provoke deliberations, debate, and decisions that are productive for students as they make sense of the findings of their investigation and manage its progress.

By leading students to encounter particular ways of thinking, scaffolding can provoke students, "rocking the boat" when they are proceeding along without being mindful enough of the rich connections of their decisions to the domain content. It may challenge their self-assessments that they have really addressed aspects of the research question appropriately (Davis, 2003). This provocation may occur as the tools force them into decisions or commitments required to use the vocabulary and machinery of the interface. This type of scaffolded tool may create short-term costs, preventing students from rushing through their work in a problem without being mindful of the subject-matter issues that are the goal of the instruction. Although this may be a short-term challenge rather than directly assisting with more expeditious solutions, such a tool may make the students' efforts in the problem a more productive learning opportunity.

The social context of collaborative problem solving is often integral to the problematizing nature of the tool. Students must make their understanding public when using tools that represent conceptual distinctions explicitly. Such tools require students to discuss disciplinary ideas to effect actions in the tool. In this way, the artifact students use to examine data and the external records they create of their interpretations become a vehicle for negotiation of understanding about the disciplinary ideas and their application to the task at hand. The pressure to be explicit in a shared external representation can serve as a catalyst for negotiation of ideas (Pea & Gomez, 1992; Roschelle, 1992; Teasley & Roschelle, 1993).

Influence of Problematizing Student Work

There are several ways that problematizing tools can help address learner difficulties.

Elicit articulation. The tools can lead learners to be more explicit about their reasoning by providing a restricted representation that makes important distinctions explicit. This can help counter the tendency toward superficial and nonreflective work. For example, Computer-Supported Intentional Learning Environments (CSILE) requires students to indicate the rhetorical connection of their comment to an ongoing discussion (Scardamalia & Bereiter, 1994). The explicit representation in the artifact also provides concrete evidence of the work that can enable later reflection and revisiting of the history of the investigation (Collins & Brown, 1988; Loh et al., 2001). The explicit representation may also aid the group dynamics. For example the inscription of the work so far can make explicit the consensus that has been reached and thereby help the group avoid reinventing solutions to decisions already constructed. Productive discussions in community of learners classrooms require connecting the discussion to disciplinary norms (Engle & Conant, 2002). Software tools may provide a context for more productive discussions within a group if they can be designed to encourage these connections and thereby focus discussion to deal with these critical aspects (de Vries et al., 2002; Sandoval & Reiser, in press).

Elicit decisions. Software tools can employ explicit representations that are more precise than natural language or students' paper-and-pencil work. Requiring students to select from limited options can encourage them to grapple with decisions they might otherwise overlook such as classifying the way evidence connects to positions in an argument (Bell & Linn, 2000; Toth et al., 2002). This may help students face the challenges of unfamiliar strategies as they engage in and document the decision making elicited in interactions with the tool.

Surface gaps and disagreements. The explicitness required in constrained artifacts can lead students to discover and address disagreements that may be beneath the surface in their individual interpretations of the group's work. Articulating interpretations in a joint product, particularly within a constrained representation, helps make clear where there is disagreement and need for resolution (Kyza, Golan, Reiser, & Edelson, 2002; Teasley & Roschelle, 1993). Barron (2003) argued that collaborative work in investigations requires a convergence of social and discourse practices with the cognitive practices of constructing and exploring a joint problem space. Productive classroom discussions require not just participation but active engagement in particular learning processes including uncovering divergent positions and resolving them through fair and moderated argument (Barron, 2003; Jimenenez-Aleixandre, Rodriguez, & Duschl, 2000; Phelps & Damon, 1989; van Zee & Minstrell, 1997). Software tools have the potential to aid teams of learners by surfacing disagreements critical to disciplinary goals.

Examples of Problematizing Student Work

To illustrate scaffolding through problematizing, I consider another scaffolding strategy associated with the guideline of facilitating articulation and reflection—"highlight epistemic features of scientific practices and products" (Strategy 7d in Quintana et al., this issue). As students interact with the tools and create work products, they need to use the important epistemic features of the discipline and may need to reframe their own ways of thinking in terms of these features. For example, as mentioned earlier, Belvedere requires students to clearly indicate the relation (supporting or disconfirming) between evidence and claims (Toth et al., 2002). Similarly, CSILE requires students to indicate the rhetorical relation between their contribution and other entries in an online discussion such as elaboration, new question, disagreement, and so on (Scardamalia & Bereiter, 1994).

Transparency in cognitive artifacts that people use can affect the ease with which they can achieve their desired ends through actions in the interface (Hutchins et al., 1986; Norman, 1988, 1991). Here, the goal, rather than solely being only the ease of mapping between learners' intentions and the representations and language of the tool, is the fit of the interactions with the tool with the type of thinking needed. Tools can provide structure and focus learners on important constituents of tasks such as argument structure. However, the need to use the structures in the tool may uncover complexity in the domain. This can help learners avoid solutions that are too superficial and lead them to focus resources on productive issues. The thinking required to make decisions in authoring an argument in Belvedere or participating in a CSILE discussion goes beyond what may be required to produce more traditional artifacts such as verbal essays in which the research behind the essay and the deeper argument structure embedded in the essay may be very difficult to discern. Interacting with a more general tool like a word processor may not create the same pressures to face the difficult decisions entailed in these scaffolded systems.

The following example, drawn from Sandoval and Reiser (in press), demonstrates the problematizing nature of the software tools. In this example, three high school biology students are working in a group using Galápagos Finches and ExplanationConstructor to investigate changes in the finches population. In this excerpt, the students are deliberating which explanatory framework of explanation guides to select. One student's suggestion to use the natural selection framework provokes debate on one of the key ideas in the domain, the nature of traits. In considering whether this framework fits their interpretation of the problem, the group disagrees about whether food choice qualifies as a trait. In the course of this debate, the group brings in key ideas about physical traits and the relation between structure and function.

Evan: (Reading prompt on framework) "Environment causes …"
Janie: No!
Evan: Yeah, "to be selected for …"
Janie: Yeah, but that means like …
Evan: // what food they eat //
Janie: … organism with these trait
Evan: // the trait being the food
Franny: Yeah, that's right.
Janie: No, because like, if my trait is to eat steak, and there's no steak, I'm immediately gonna go to something else.
Evan: If you're only a vegetarian and you only eat … you don't eat meat, you're not gonna eat meat. Well, that depends …//
Janie: Are you insane!?
Franny: OK, OK. Don't think of people. Think of these guys (the finches). If they only eat one type of seed with their beaks and that seed is gone then they can't live anymore.

This example demonstrates aspects of discourse that the tool is designed to catalyze and support and suggests a way that tools can help problematize subject matter. Having to structure the analysis of their findings in terms of the theoretical framework embedded in the tool required students to frame their understanding in terms of principles of the domain. Rather than just writing down their explanation, the tool forces them to consider how to express their hypothesis and its support within a disciplinary framework such as natural selection. Decisions about the use of the artifact became the context for negotiation between the students of these important disciplinary ideas such as the nature of a trait and the difference between a specie's characteristic traits and learned behaviors. The tool was a context for useful conversations that helped students overcome the limitations of unfamiliar strategies and helped them avoid superficial solutions not connected to disciplinary ideas. In this case, both sense making (interpreting the observed differences in individuals as candidates for traits supporting differential survival) and articulation were supported by the problematizing nature of the tool.

It is important to note that the tool was a support for these practices, but its effective use relies on other factors such as the inclination of the students to engage sufficiently in handling the complexity uncovered in their interactions with the tool as well as expectations and classroom norms fostered by the teacher (Tabak, this issue). I return to the role software scaffolds can play in the larger classroom system in the Discussion section.

Another example of problematizing reflects the scaffolding strategy "make disciplinary strategies explicit in the artifacts learners create," associated with the guideline to "organize tools and artifacts around the semantics of the discipline" (Strategy 2b and "Guideline 2" in Quintana et al., this issue, respectively). The BGuILE environment Animal Landlord is designed to provide tools to analyze animal behavior that help students understand the strategies and constructs in the discipline (Smith & Reiser, 1998). In the Animal Landlord, students study examples of animal behavior to isolate and analyze the key components of complex animal behavior. Students analyze digital video clip examples of behavior such as hunting or eating and deconstruct the complex sequence into what they see as the important causal events. Students identify significant events by selecting frames from a clip, categorizing them from a behavioral taxonomy, and annotating them with their interpretations (see Figure 2). In this way, students build an annotated storyboard that describes the progression of critical events in a behavior. Students use the tool to compare and contrast examples in the corpus such as different patterns in lion hunting or how different animals obtain food (Golan, Kyza, Reiser, & Edelson, 2001; Smith & Reiser, 1998).

The type of artifact students create is designed to provide useful structure for the students' investigation and to focus attention and make problematic disciplinary distinctions. The structure of the artifact, a sequence of labeled and annotated frames, clearly represents the structure of the process they use—decomposing complex behavior into its critical constituents, classifying and interpreting each constituent. In this way, the inscriptions students create to record their analyses are a relatively transparent representation of the processes students need to use—decomposition and categorization. Furthermore, the analyses are organized in these inscriptions into two distinct categories—*observations* refer to what one can see directly in the data, and *interpretations* represent what one infers from these observations. This is a

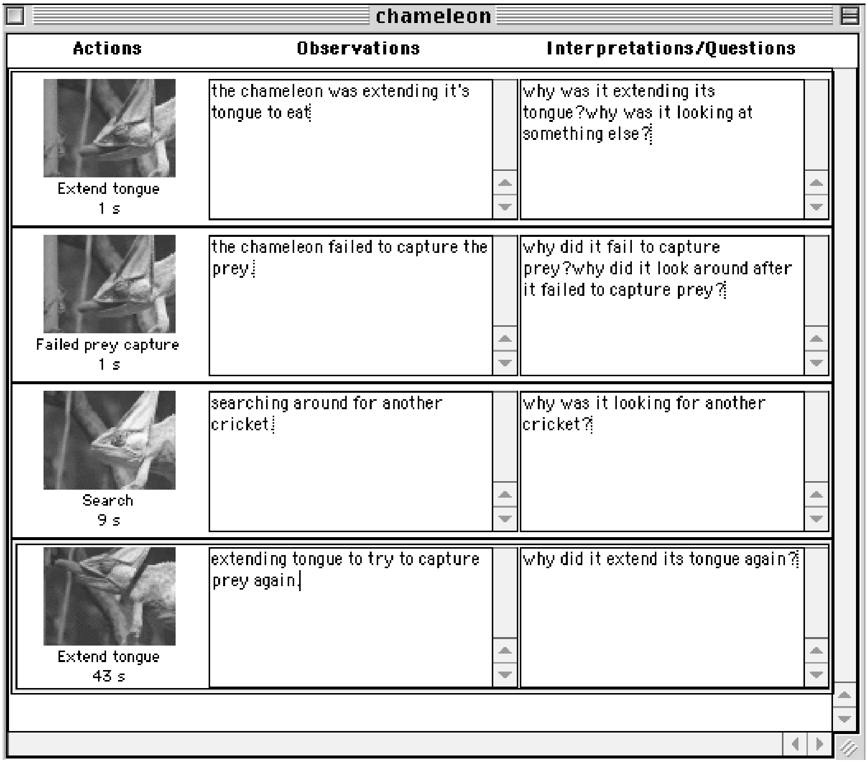

FIGURE 2 Artifacts constructed in the Animal Landlord. Students decompose complex behavior into its constituents, categorize each constituent, and record their observations and interpretations.

key epistemic distinction in the scientific practice students are learning; thus, this aspect of the tool also reflects the scaffolding strategy "highlight epistemic features" mentioned earlier in the ExplanationConstructor example.

Interacting with the tool to create these artifacts can also help problematize key aspects of the practice. Requiring students to categorize their observations in terms of disciplinary frameworks pushes students to articulate their understanding and represent it in the artifact. The explicit distinction between observation and interpretation is intended to elicit and support discussions geared at understanding the relation between the two.

The following episode illustrates how the structure of the artifacts can lead students to grapple with disciplinary content (Golan et al., 2001). This debate was recorded from a group of three seventh-grade students who were watching a clip featuring two Golden Lion Tamarin monkeys eating a grape. One of the monkeys (the male) had the grape first. The female took part of the grape away from him, and then jumped away to another branch. Students had selected frames from the clip to annotate that included the point at which the female jumped. The students were arguing about this event, and disagreed on their interpretation. Two of the students believed that the event was an instance of "mount" behavior, whereas the third student did not agree. (The term *mount* was one of 15 built-in labels available; others included sniff, rest, move, take food, follow, etc.) One student (Chandan, who was currently controlling the keyboard), had labeled the clip as "mount" and was bringing up what to type into the observation annotations associated with that event. In essence, this argument was how to define the behavior mount—as merely contact between two animals or as a specialized kind of contact.

Chandan: What did we observe as mount? (reading the prompt from the observation field)
Danny: No, that one is yours because I totally disagree with you guys!
Chandan: Good for you! Come on man, you see in the clip it just looks mounting, they got on top of each other.
Danny: No, he jumped *over* her.
Dennis: She, she jumped over him …
Danny: Whatever, she jumped over him.
Chandan: I know, but still, the contact …
Danny: She jumped over him, doesn't matter, contact is not what you guys are talking about. Shoot, you are talking about like getting on top of each other and staying on top for a couple of minutes.
Dennis: No, no, no. We are not talking about that!
Chandan: No, no, not that, no, no, that's not. See, look, watch this contact. (replaying the clip beginning at the selected frame) Boom! Look at that!
Danny: Jumping *over*! *Over!*

This argument was finally settled by one of the researchers clarifying what the behavior mount means in the domain of animal behavior. This discussion surfaced discrepancies in students' implicit definitions of the behavior. Converging on clear and explicit definitions is an important step in developing and applying a categorization scheme. Making these decisions as part of their analysis and clearly representing these distinctions in the artifacts they create provoked these and similar arguments about the meaning or indicators of behavior, surfacing disagreements and eventually refining students' definitions of these behaviors.

It is interesting that this disagreement had been brewing for some time in the group, but it had not yet been addressed. Here we see the potential for software to problematize aspects of the subject matter. Had the group merely been asked to report a summary of their observations or turn in a simple text report, it is possible this discussion would not have occurred. Requiring students to articulate their understanding and represent it explicitly using menu item labels and the observation and interpretation structure finally surfaced the differences in interpretation and provoked these productive discussions. For several reasons, students may be inclined to downplay the type of confusion or disagreement that finally surfaced—for example, a tendency to accept solutions that appear on the surface sufficient and the inclination to downplay the significance of difference of opinion to avoid dissension. Yet when required to commit their analysis in the precise representation (identifying each important event as an example of a small number of behavioral categories) as required by the software tool, this disagreement came to the surface and became the focus of conversation in the group.

DISCUSSION

I have argued that software scaffolding can help learners by providing needed structure and by problematizing important subject matter. I presented brief examples of learners' encounters with software environments to help illustrate these mechanisms. The aim of scaffolding design is to find ways to use the nature of learners' interactions with the tool to help shape their thinking in productive ways. I discussed how the explicitness of representations and ability to represent important conceptual aspects of a discipline in tools can play a role in structuring and problematizing learners' engagement with the subject matter.

In this section, I consider a number of general issues about the scaffolding mechanisms of structuring and problematizing. First, I consider how the two mechanisms of structuring and problematizing are related to one another. I discuss design tensions that arise in trying to design learning environments that provide structure and lead students to problematize subject matter. Finally, I consider other aspects of the context that are important for scaffolding in software tools to influence learning practices.

Complementarity and Tensions

The two mechanisms of structuring and problematizing are often complementary. I described earlier how representations in tools can be designed to provide needed structure that students can use to help guide and organize their work. The structure can help students decompose tasks, monitor the performance of component tasks, and encode the results of their work in useful forms. I suggested that the features of a software tool that provide useful structure may also elicit attention to critical issues in subject matter, for example, calling attention to important distinctions such as observation and interpretation. Thus, as shown in the examples with ExplanationConstructor and Animal Landlord, the same characteristic of a tool designed to help structure students' engagement in a task, for example, by providing guiding prompts or making explicit a set of subtasks, may also problematize the disciplinary ideas by requiring students to make sense of the options and connect their own work to these disciplinary ideas.

Yet the goals of supporting these two mechanisms may also be in tension—the goals of simplifying the problem space may be somewhat in tension with encouraging learners to grapple with distinctions or disciplinary frameworks that are likely to require more deliberation. For structure to be useful, learners must be able to recognize and use the distinctions presented. Calling students' attention to and requiring use of unfamiliar strategies may work against the system's usefulness for guiding students' investigations. It may require additional reasoning steps that work counter to the structures intended to be useful. Or, if the strategies are unfamiliar enough and students cannot make the connections to their owns ways of thinking, they may use the systems' structuring improperly or superficially. For example, despite careful crafting of prompts to guide students' work, students may treat the software environment as "just another worksheet" and ignore the fine distinctions in the system's attempt to structure the reporting of their work, or may enter minimal answers rather than carefully considering what is needed.

The proper balance is tricky to achieve. Vygotsky's (1978) notion of zone of proximal development is important in characterizing the sense of balance needed in scaffolded software tools. The technical tools of a scientist make extensive use of disciplinary ideas, yet these tools are often inappropriate for learners. They are not supportive enough to engage learners in grappling with the complex ideas in the domain that they represent. Introducing overwhelming complexity without needed structure does not lead to problematizing subject matter. On the other hand, a tool that strips away the complexity, guiding students in lockstep, may not engage them in grappling with the complexity and reasoning through the solutions they are led to construct. Thus, achieving both functions requires a careful balance.

Design Tensions in Crafting Scaffolded Tools

Considering the tensions between attempting to problematize and provide structure suggests several important design tensions in designing scaffolded tools.

Support intuitive strategies versus problematize disciplinary content. Designs have to strive for an optimal balance between connecting with students' intuitive strategies on one hand and requiring students to work within disciplinary frameworks on the other. Attempts to provide structure may focus attention and highlight critical features, but the problematizing is only effective if the students can make the connections in bridging from their own intuitive strategies to the structures enforced by the tool. Quintana et al. (this issue) review a number of scaffolding strategies that take on this challenge—using authentic complex representations, but tailored to better connect with learners' intuitive understandings. For example, Model-It™ enables learners to construct dynamic systems models but provides an intuitively compelling representation of a concept map and enables learners to simply link two factors with qualitative types of influence rather than getting bogged down in cumbersome mathematical relations between two variables (Jackson et al., 1998).

Generality versus specificity. General guidance may be very helpful for structuring work. For example, a general structure for posing questions, designing investigations, and interpreting data may provide useful structures to enable learners to organize their investigation. It may have the benefit of introducing learners to a small set of very powerful disciplinary ideas such as controlling variables or graphically representing data. These ideas would potentially be applicable to broad range of new problems. Yet to problematize subject matter, more specific guidance may be needed such as the disciplinary frameworks embedded in ExplanationConstructor. Students may have difficulty acting on such general guidance, for example, knowing counterevidence is important in an argument but having difficulty figuring out what types of evidence to collect or evaluate (Sandoval, 2003; Sandoval & Reiser, in press). Designers have to find the right balance for the particular target learners in the tension between broadly applicable guidance and guidance more tailored to subject matter.

Student responsibility and control versus more constrained choices. Ultimately the goal is for students to be able to direct their own investigations and to be involved in defining a problem, planning a solution, and conducting empirical investigations. In scaffolded inquiry approaches, the learner may be engaged directly in only some subset or in narrowed versions of these components. For example, often in project-based science, the general problem is provided to students, although great care is taken to contextualize that problem in students' experiences and to create a sense of ownership as students explore their own solutions to that problem. However, the risk of this narrowing to provide helpful structure is that it may lead to learners just "going through the motions" rather than being reflective about what is being required and why. In contrast, problematizing by requiring important steps or distinctions may take some of the control for their own investigations away from students.

Studies of teaching approaches exhibited by tutors and classroom teachers have revealed a careful balance between providing help of various sorts to assist problem solving and help avoid difficulties on one hand while not providing too much support that curtails learners' active engagement or circumvents learners' involvement in the target reasoning practices (Hogan & Pressley, 1997; Lepper et al., 1993; Merrill et al., 1995). Similarly, design of effective software scaffolding must negotiate such a balance between these tensions. For example, tools may attempt to link more intuitive representations with scientific language, provide prompts that continually act to operationalize important strategies, or help students document the connections between strategic decisions and the results in the investigation (Quintana et al., this issue), thereby attempting to bridge learners to expert practice using representations that are elaborated and connected to learners' starting conceptions.

An important limitation in finding an optimal balance between these tensions between structuring and problematizing is the limited ability of most scaffolded tools to individualize their support. The notion of tailoring support to the needs of particular learners and learning situations has been an important aspect of scaffolding in its application to teaching and tutoring (Webb & Palincsar, 1996; Wood et al., 1976). Whereas reasoning about the states of individual learners has been explored in intelligent tutoring systems for more procedural domains (Anderson, Corbett, Koedinger, & Pelletier, 1995), the approach in the scaffolded cognitive tools has been to embed support within the system as prompts or as representations in the structure of the tool itself. Such tools are adaptable, to a limited extent, under the control of the learners who can explore additional prompts or assistance available, attempt to follow or work around the system's advice, or perhaps under control of teachers who may tailor messages or functionality provided. However, the sense in which tools can scaffold learners under such conditions are clearly quite different than expert teachers who can tailor their advice to an assessment of the individual learner state. Of course, there are other pressures on classroom teachers who must deal with the needs of many learners simultaneously, so their ability to scaffold dozens of learners simultaneously faces other challenges.

Scaffolding Requires a System

A final caution to be discussed in exploring models of scaffolding in software tools is that learners, tools, and teachers work together as a system, and it is an oversimplification to consider how tools can scaffold learners without considering the other aspects of this system. Learners come with attitudes and expectations toward the subject matter and toward learning, and these expectations are shaped in part by the classroom culture created by teachers. Tools cannot force learners to reason about an idea or to use kinds of language. Rather, tools can provide support that in the right context may influence the directions and practices of learners and teachers. Thus, scaffolded tools can create opportunities, but whether learners capitalize on these opportunities depends on the expectations and practices established in the classroom. The nature of the scaffolding is that it may act to provoke or catalyze, but of course the software tools cannot require that learners mindfully engage with these opportunities.

A first critical factor is that the ways of thinking the tool is designed to support must be threaded through all aspects of the classroom system—in the curricular activities that surround the tool, in teachers' support working with individual groups, and in the teachers' structuring and guidance of whole class discussions. Although software may provoke attention to an idea, such as the distinction between observation and interpretation, the classrooms practices need to take up this idea in classroom discourse for these ideas to take on real meaning (Hogan et al., 1999; Hogan & Pressley, 1997; Lemke, 1990; Tabak & Reiser, 1997). Tabak (2002, this issue; Tabak & Reiser, 1997) has characterized this as a synergistic process in which teachers capitalize on and reinforce helpful structure in tools, and the influence of tools relies on how teachers cultivate their use. Software tools may influence the focus of attention, and teachers can then capitalize on and reinforce in their questioning and guidance of students (Tabak, 2002). For these distinctions in the tool's representations to be taken seriously and treated mindfully by learners, teachers need to structure their interactions with students around the same framework. The software tool provides a way to reinforce a way of talking and thinking about the data, but it is only part of the classroom system in which teachers frame the way learners think about material. Teachers and software tools may work in concert, with the software tools providing a concrete representation of distinctions that teachers have brought into classroom discussions (Kemp, Tzou, Reiser, & Spillane, 2002; Reiser et al., 2001; Tabak & Reiser, 1997; Tzou, Reiser, Spillane, & Kemp, 2002). Thus, software scaffolds can influence learning by supporting more productive problem-solving conversations among learners and among learners and teachers (Teasley & Roschelle, 1993).

Another important role of the classroom system is the availability of resources to support the investigation. Problematizing is a process of focusing attention along productive dimensions, but naturally, it does not guarantee that this focus of attention will lead to productive results. This is particularly critical when considering the problematizing nature of software tools. Provoking productive conflict within a group requires that the group have access to resources needed to resolve the conflict such as other information resources or teacher support.

In summary, I have argued that scaffolding occurs through two mechanisms, structuring and problematizing. Most current accounts of scaffolding define support that helps students proceed through tasks by providing structure. However, given the importance of connecting students' problem-solving work to disciplinary content, skills, and strategies, it is important to provoke issues in students, veering them off the course of nonreflective work and forcing them to confront key disciplinary ideas in their work. Tools shape users' engagement with tasks. As such, tools can be designed to provide a context that can influence users' perceptions, the discourse between learners and between learners and teachers, and the ways they represent their thinking in artifacts of their work. The artifacts students use and create can be designed to map onto important disciplinary ideas and strategies, thereby problematizing these ideas as students use the tool to work through the task and represent the products of their work.

ACKNOWLEDGMENTS

An earlier version of this article was presented at the Computer Support for Collaborative Learning 2002 Conference (January 2002, Boulder, CO). This research was funded by the National Science Foundation under Grants REC–9980055 to the KDI/ASSESS project and REC–9720383 to the Center for Learning Technologies in Urban Schools and by Grant 97–57 from the James S. McDonnell Foundation, Cognitive Studies for Educational Practice. The opinions expressed herein are those of the author and not necessarily those of these foundations. For additional information about BGuILE software and curricula, see http://www.letus.org/bguile/ and http://www.hi-ce.org/iqwst/

This article builds on the design research of the BGuILE group at Northwestern University, and I am grateful for conversations with my colleagues Iris Tabak, Brian Smith, William Sandoval, Ravit Golan Duncan, and Elena Kyza about principles of scaffolding in learning environments. I am also grateful to my colleagues in the KDI/ASSESS group for discussion of these ideas—Joseph Krajcik, Danny Edelson, Elliot Soloway, Chris Quintana, and Elizabeth A. Davis—and to Bruce Sherin for many fruitful discussions. This article was greatly improved by comments from reviewers Mimi Recker and Phil Bell; and editors Elizabeth A. Davis and Naomi Miyake.

REFERENCES

American Association for the Advancement of Science. (1990). *Science for all Americans: Project 2061.* New York: Oxford University Press.

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences, 4,* 167–207.

Bannon, L. J., & Bødker, S. (1987). Beyond the interface: Encountering artifacts in use. In J. M. Carroll (Ed.), *Interfacing thought: Cognitive aspects of human–computer interaction* (pp. 227–253). Cambridge, MA: MIT Press.

Barron, B. (2003). When smart groups fail. *The Journal of the Learning Sciences, 12,* 307–359.

Bell, P., & Linn, M. C. (2000). Scientific arguments as learning artifacts: Designing for learning from the web with KIE. *International Journal of Science Education, 22,* 797–817.

Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., & Palincsar, A. (1991). Motivating project-based learning: Sustaining the doing, supporting the learning. *Educational Psychologist, 26,* 369–398.

Brown, A. L., & Campione, J. C. (1994). Guided discovery in a community of learners. In K. McGilly (Ed.), *Classroom lessons: Integrating cognitive theory and classroom practice* (pp. 229–270). Cambridge, MA: MIT Press.

Bruner, J. S. (1966). *Toward a theory of instruction.* Cambridge, MA: Belknap Press/Harvard University Press.

Cavalli-Sforza, V., Weiner, A., & Lesgold, A. M. (1994). Software support for students engaging in scientific activity and scientific controversy. *Science Education, 78,* 577–599.

Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science, 13,* 145–182.

Chi, M. T. H., Feltovich, P., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science, 5,* 121–152.

Cohen, D. K. (1988). Teaching practice: Plus que ca change. In P. W. Jackson (Ed.), *Contributing to educational change: Perspectives on research and practice* (pp. 27–84). Berkeley: McCutchan.

Coleman, E. B. (1998). Using explanatory knowledge during collaborative problem solving in science. *Journal of the Learning Sciences, 7,* 387–427.

Collins, A. (1991). Cognitive apprenticeship and instructional technology. In L. Idol & B. F. Jones (Eds.), *Educational values and cognitive instruction: Implications for reform* (pp. 121–138). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Collins, A., & Brown, J. S. (1988). The computer as a tool for learning through reflection. In H. Mandl & A. Lesgold (Eds.), *Learning issues for intelligent tutoring systems* (pp. 1–18). New York: Springer-Verlag.

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: Lawrence Erlbaum Associates, Inc.

Davis, E. A. (2003). Prompting middle school science students for productive reflection: Generic and directed prompts. *The Journal of the Learning Sciences, 12,* 91–142.

Davis, E. A., & Linn, M. C. (2000). Scaffolding students' knowledge integration: Prompts for reflection in KIE. *International Journal of Science Education, 22,* 819–837.

de Vries, E., Lund, K., & Baker, M. (2002). Computer-mediated epistemic dialogue: Explanation and argumentation as vehicles for understanding scientific notions. *The Journal of the Learning Sciences, 11,* 63–103.

Driver, R., Newton, P., & Osborne, J. (2000). Establishing the norms of scientific argumentation in classrooms. *Science Education, 84,* 287–312.

Edelson, D. C. (2001). Learning-for-use: A framework for integrating content and process learning in the design of inquiry activities. *Journal of Research in Science Teaching, 38,* 355–385.

Edelson, D. C., Gordin, D. N., & Pea, R. D. (1999). Addressing the challenges of inquiry-based learning through technology and curriculum design. *The Journal of the Learning Sciences, 8,* 391–450.

Engle, R. A., & Conant, F. R. (2002). Guiding principles for fostering productive disciplinary engagement: Explaining an emergent argument in a community of learners classroom. *Cognition and Instruction, 20,* 399–483.

Golan, R., Kyza, E. A., Reiser, B. J., & Edelson, D. C. (2001, March). *Structuring the task of behavioral analysis with software scaffolds.* Paper presented at the annual meeting of the National Association of Research in Science Teaching, St. Louis, MO.

Guzdial, M. (1994). Software-realized scaffolding to facilitate programming for science learning. *Interactive Learning Environments, 4,* 1–44.

Hiebert, J., Carpenter, T. P., Fennema, E., Fuson, K., Human, P., Murray, H., et al. (1996). Problem solving as a basis for reform in curriculum and instruction: The case of mathematics. *Educational Researcher, 25*(4), 12–21.

Hmelo, C. E., Holton, D. L., & Kolodner, J. L. (2000). Designing to learn about complex systems. *The Journal of the Learning Sciences, 9,* 247–298.

Hogan, K., Nastasi, B. K., & Pressley, M. (1999). Discourse patterns and collaborative scientific reasoning in peer and teacher-guided discussions. *Cognition and Instruction, 17,* 379–432.

Hogan, K., & Pressley, M. (1997). Scaffolding scientific competencies within classroom communities of inquiry. In K. Hogan & M. Pressley (Eds.), *Scaffolding student learning: Instructional approaches and issues* (pp. 74–107). Cambridge, MA: Brookline.

Hollan, J. D., Bederson, B. B., & Helfman, J. (1997). Information visualization. In M. G. Helander, T. K. Landauer, & V. Prabhu (Eds.), *Handbook of human–computer interaction* (pp. 33–48). Essex, England: Elsevier Science.

Hollan, J. D., Hutchins, E., & Kirsh, D. (2000). Distributed cognition: Toward a new foundation for human–computer interaction research. *ACM Transactions on Computer–Human Interaction, 7,* 174–196.

Hutchins, E. (1995). *Cognition in the wild.* Cambridge, MA: MIT Press.

Hutchins, E., Hollan, J. D., & Norman, D. A. (1986). Direct manipulation interfaces. In D. A. Norman & S. Draper (Eds.), *User centered system design: New perspectives in human–computer interaction.* Hillsdale, NJ: Lawrence Erlbaum Association, Inc.

Jackson, S. L., Krajcik, J., & Soloway, E. (1998). The design of guided learning-adaptable scaffolding in interactive learning environments. In C.-M. Karat, A. Lund, J. Coutaz, & J. Karat (Eds.), *Proceedings of CHI 98: Human factors in computing systems* (pp. 187–194). Reading, MA: Addison-Wesley.

Jimenenez-Aleixandre, M. P., Rodriguez, A. B., & Duschl, R. A. (2000). "Doing the lesson" or "doing science": Argument in high school genetics. *Science Education, 84,* 757 –792.

Kemp, E. K., Tzou, C. T., Reiser, B. J., & Spillane, J. P. (2002). Managing dilemmas in inquiry science teaching. In P. Bell, R. Stevens, & T. Satwicz (Eds.), *Keeping learning complex: The Proceedings of the Fifth International Conference of the Learning Sciences (ICLS)* (pp. 206–213). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Klahr, D. (2000). *Exploring science: The cognition and development of discovery processes.* Cambridge, MA: MIT Press.

Klahr, D., & Dunbar, K. (1988). Dual space search during scientific reasoning. *Cognitive Science, 12,* 1–48.

Klahr, D., Fay, A. L., & Dunbar, K. (1993). Heuristics for scientific experimentation: A developmental study. *Cognitive Psychology, 25,* 111–146.

Koedinger, K. R., & Anderson, J. R. (1993). Reifying implicit planning in geometry: Guidelines for model-based intelligent tutoring system design. In S. P. Lajoie & S. J. Derry (Eds.), *Computers as cognitive tools* (pp. 15–45). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Kolodner, J. L., Owensby, J. N., & Guzdial, M. (2004). Case-based learning aids. In D. H. Jonassen (Ed.), *Handbook of research for educational communications and technology* (2nd ed., pp. 829–862). Mahwah, NJ: Lawrence Erlbaum Associates Inc.

Krajcik, J., Blumenfeld, P. C., Marx, R. W., Bass, K. M., Fredricks, J., & Soloway, E. (1998). Inquiry in project-based science classrooms: Initial attempts by middle school students. *Journal of the Learning Sciences, 7,* 313–350.

Kuhn, D. (1993). Science as argument: Implications for teaching and learning scientific thinking. *Science Education, 77,* 319–337.

Kuhn, D., Amsel, E., & O'Loughlin, M. (1988). *The development of scientific thinking skills.* San Diego, CA: Academic.

Kurth, L. A., Anderson, C. W., & Palincsar, A. S. (2002). The case of Carla: Dilemmas of helping all students to understand science. *Science Education, 86,* 287–313.

Kyza, E. A., Golan, R., Reiser, B. J., & Edelson, D. C. (2002). Reflective inquiry: Enabling group self-regulation in inquiry-based science using the progress portfolio tool. In G. Stahl (Ed.), *Computer support for collaborative Learning: Proceedings of CSCL 2002* (pp. 227–236). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Lemke, J. L. (1990). *Talking science: Language, learning, and values.* Norwood, NJ: Ablex.

Lepper, M. R., Woolverton, M., Mumme, D. L., & Gurtner, J. (1993). Motivational techniques of expert human tutors: Lessons for the design of computer-based tutors. In S. P. Lajoie & S. J. Derry (Eds.), *Computers as cognitive tools* (pp. 75–105). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Linn, M. C. (2000). Designing the knowledge integration environment. *International Journal of Science Education, 22,* 781–796.

Linn, M. C., Bell, B., & Davis, E. A. (2004). *Internet environments for science education.* Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Loh, B. (2003). *Using articulation and inscription as catalysts for reflection: Design principles for reflective inquiry.* Unpublished doctoral dissertation, Northwestern University, Evanston, IL.

Loh, B., Reiser, B. J., Radinsky, J., Edelson, D. C., Gomez, L. M., & Marshall, S. (2001). Developing reflective inquiry practices: A case study of software, the teacher, and students. In K. Crowley, C. D. Schunn, & T. Okada (Eds.), *Designing for science: Implications from everyday, classroom, and professional settings* (pp. 279–323). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Merrill, D. C., Reiser, B. J., Merrill, S. K., & Landes, S. (1995). Tutoring: Guided learning by doing. *Cognition and Instruction, 13,* 315–372.

Moje, E. B., Collazo, T., Carrillo, R., & Marx, R. W. (2001). "Maestro, what is 'quality'?": Language, literacy, and discourse in project-based science. *Journal of Research in Science Teaching, 38,* 469–498.

Norman, D. A. (1988). *Psychology of everyday things.* New York: Basic Books.

Norman, D. A. (1991). Cognitive artifacts. In J. M. Carroll (Ed.), *Designing interaction: Psychology at the human–computer interface* (pp. 17–38). New York: Cambridge University Press.

Norman, D. A. (1993). *Things that make us smart: Defending human attributes in the age of the machine.* Reading, MA: Addison-Wesley.

Olson, S., & Loucks-Horsley, S. (Eds.). (2000). *Inquiry and the national science education standards: A guide for teaching and learning.* Washington, DC: National Academy Press.

Passmore, C., & Stewart, J. (2002). A modeling approach to teaching evolutionary biology in high schools. *Journal of Research in Science Teaching, 39,* 185–204.

Pea, R. D. (1992). Practices of distributed intelligence and designs for education. In G. Salomon (Ed.), *Distributed cognitions: Psychological and educational considerations* (pp. 47–87). New York: Cambridge University Press.

Pea, R. D., & Gomez, L. M. (1992). Distributed multimedia learning environments: Why and how? *Interactive Learning Environments, 2,* 73–109.

Perkins, D. (1998). What is understanding? In M. S. Wiske (Ed.), *Teaching for understanding: Linking research with practice* (pp. 39–58). San Francisco: Jossey-Bass.

Phelps, E., & Damon, W. (1989). Problem solving with equals: Peer collaboration as a context for learning mathematics and spatial concepts. *Journal of Educational Psychology, 81,* 639–646.

Quintana, C., Eng, J., Carra, A., Wu, H.-K., & Soloway, E. (1999). Symphony: A case study in extending learner-centered design through process space analysis. In M. G. Williams, M. W. Altom, K. Ehrlich, & W. Newman (Eds.), *Proceedings of CHI 99 Conference on Human Factors in Computing Systems* (pp. 473–480). Reading, MA: Addison-Wesley.

Reif, F., & Larkin, J. H. (1991). Cognition in scientific and everyday domains: Comparison and learning implications. *Journal of Research in Science Teaching, 28,* 733–760.

Reiser, B. J., Tabak, I., Sandoval, W. A., Smith, B. K., Steinmuller, F., & Leone, A. J. (2001). BGuILE: Strategic and conceptual scaffolds for scientific inquiry in biology classrooms. In S. M. Carver & D. Klahr (Eds.), *Cognition and instruction: Twenty-five years of progress* (pp. 263–305). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

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

Roschelle, J. (1992). Learning by collaboration: Convergent conceptual change. *The Journal of the Learning Sciences, 2,* 235–276.

Rosebery, A. S., Warren, B., & Conant, F. R. (1992). Appropriating scientific discourse: Findings from language minority classrooms. *Journal of the Learning Sciences, 2,* 61–94.

Salomon, G., Perkins, D. N., & Globerson, T. (1991). Partners in cognition: Extending human intelligence with intelligent technologies. *Educational Researcher, 20*(3), 2–9.

Sandoval, W. A. (1998). *Inquire to explain: Structuring inquiry around explanation construction in a technology-supported biology curriculum.* Unpublished doctoral dissertation, Northwestern University, Evanston, IL.

Sandoval, W. A. (2003). Students' understanding of causal explanation and natural selection in a technology-supported inquiry curriculum. *Journal of the Learning Sciences, 12,* 5–51.

Sandoval, W. A., & Reiser, B. J. (2004). Explanation-driven inquiry: Integrating conceptual and epistemic scaffolds for scientific inquiry. *Science Education, 88,* 345–372.

Scardamalia, M., & Bereiter, C. (1994). Computer support for knowledge-building communities. *The Journal of the Learning Sciences, 3,* 265–283.

Schauble, L., Glaser, R., Duschl, R. A., Schulze, S., & John, J. (1995). Students' understanding of the objectives and procedures of experimentation in the science classroom. *The Journal of the Learning Sciences, 4,* 131–166.

Schauble, L., Glaser, R., Raghavan, K., & Reiner, M. (1991). Causal models and experimentation strategies in scientific reasoning. *The Journal of the Learning Sciences, 1,* 201–238.

Sherin, B. L. (2001). How students understand physics equations. *Cognition and Instruction, 19,* 479–541.

Smith, B. K., & Reiser, B. J. (1998). National Geographic unplugged: Designing interactive nature films for classrooms. In C.-M. Karat, A. Lund, J. Coutaz, & J. Karat (Eds.), *Proceedings of CHI 98* (pp. 424–431). New York: ACM Press.

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

Tabak, I. (2002). Teacher as monitor, mentor or partner: Uncovering participant structures involved in supporting student-directed inquiry. In P. Bell, R. Stevens, & T. Satwicz (Eds.), *Keeping learning complex: The Proceedings of the fifth international conference of the learning sciences (ICLS)* (pp. 466–472). Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Tabak, I., & Reiser, B. J. (1997). Complementary roles of software-based scaffolding and teacher–student interactions in inquiry learning. In R. Hall, N. Miyake, & N. Enyedy (Eds.), *Proceedings of Computer Support for Collaborative Learning '97* (pp. 289–298). Toronto, Ontario, Canada: Lawrence Erlbaum Associates, Inc.

Tabak, I., Smith, B. K., Sandoval, W. A., & Reiser, B. J. (1996). Combining general and domain-specific strategic support for biological inquiry. In C. Frasson, G. Gauthier, & A. Lesgold (Eds.), *Intelligent tutoring systems: Third international conference, ITS '96* (pp. 288–296). Montreal, Quebec, Canada: Springer-Verlag.

Teasley, S. D., & Roschelle, J. (1993). Constructing a joint problem space: The computer as a tool for sharing knowledge. In S. P. Lajoie & S. J. Derry (Eds.), *Computers as cognitive tools* (pp. 229–258). Hillsdale, NJ: Lawrence Erlbaum Associates Inc.

Toth, E. E., Suthers, D. D., & Lesgold, A. M. (2002). "Mapping to know": The effects of representational guidance and reflective assessment on scientific inquiry. *Science Education, 86,* 264–286.

Tzou, C. T., Reiser, B. J., Spillane, J. P., & Kemp, E. K. (2002). *Characterizing the multiple dimensions of teachers' inquiry practices.* Paper presented at the annual meeting of the American Educational Research Association, New Orleans, LA.

van Zee, E. H., & Minstrell, J. (1997). Reflective discourse: Developing shared understandings in a physics classroom. *International Journal of Science Education, 19,* 209–228.

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

Webb, N. M., & Palincsar, A. S. (1996). Group processes in the classroom. In D. C. Berliner & R. C. Calfee (Eds.), *Handbook of educational psychology* (pp. 841–873). New York: Macmillan.

White, B. Y., & Frederiksen, J. R. (1998). Inquiry, modeling, and metacognition: Making science accessible to all students. *Cognition and Instruction, 16,* 3–118.

Williams, S. M. (1992). Putting case-based instruction into context: Examples from legal and medical education. *The Journal of the Learning Sciences, 2,* 367–427.

Wood, D., Bruner, J. S., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry and Allied Disciplines, 17,* 89–100.