THE JOURNAL OF THE LEARNING SCIENCES, *10*(3), 323–363

Affordances of Collaborative Software Design Planning for Elementary Students' Science Talk

Yasmin B. Kafai
Graduate School of Education & Information Studies
University of California at Los Angeles

Cynthia Carter Ching
College of Education
University of Illinois, Urbana–Champaign

Although educational research and practice has found many benefits of long-term and complex design activities, an issue of growing concern is that students might lose sight of science learning while diverting their attention to design aesthetics, collaborative management, and technology. A question is whether or not science is actually separate from these aspects; it may be that science permeates the design environment and is thus contexted within these other activities. To investigate this possibility we followed a classroom of 33 students, divided into 7 teams, and we examined their science discussions as they planned for creating instructional software designs. Specifically, we investigated which conversational contexts gave rise to science talk. We found that a focus on the fine-grained details of the instructional science designs themselves and the contributions of more design-experienced students played an important role in the sophistication of the science content in the planning discussions. In examining less productive contexts for science talk, we found that a conversational focus in planning discussions on collaboration and software issues, as well as the science focus of the software designs, impacted the quality of science integration. In our discussion, we address the issue of which design contexts afford opportunities for richer discourse and the implications for other project-based design activities.

In the past decade, project-based science activities have gained increasing acceptance among educational researchers and practitioners (e.g., Baumgartner & Reiser, 1998; Brown & Campione, 1994; Hmelo, Holton, & Kolodner, 2000; Linn

---

Correspondence and requests for reprints should be sent to Yasmin B. Kafai, 2331 Moore Hall, Box 951521, UCLA Graduate School of Education & Information Studies, Los Angeles, CA 90095–1521. E-mail: kafai@gseis.ucla.edu

& Hsi, 2000; Penner, Lehrer, & Schauble, 1998; Roth, 1998; Scardamalia & Bereiter, 1994). This approach to science learning engages students in long-term projects, in which they collaboratively develop research questions, conduct research and experiments, and document and discuss their inquiries (Blumenfeld et al., 1991). In some versions of project-based learning, students design and build working artifacts; in others, they design and create research reports, multimedia software, or communal databases; in others' students' activities focus on generating and evaluating explanations around given scientific problems or contributing to ongoing debates. In all versions of such curricula, the implementation of these projects requires teachers and students to make significant changes in classroom interactions, learning strategies, and assessment practices.

An issue of growing concern has been that students might lose sight of science while diverting their attention to the other ongoing demands of the project task. Researchers have studied how computer-based support and scaffolding (e.g., Guzdial, 1994; Linn & Hsi, 2000) or constrained design tasks (e.g., Cuthbert & Hoadley, 1998; Hmelo et al., 2000) can guide students in their science discourse and inquiry. Our approach to this issue examines how a given project design task, creating multimedia software for science instruction, affords opportunities for students to engage in science inquiry. Designing instructional software, as it is defined here, fosters students' expressions of science questions and ideas in a concrete computational artifact—an appropriate approach for elementary students. Such an approach provides affordances that allow students to "reformulate" (Harel & Papert, 1990) their approach to science concepts. The concept of reformulation addresses students' individual and collaborative efforts to connect their own learning to their previous understanding, with the help of their teachers or peers.

In particular, we believe the collaborative activities involved in creating software presentations have powerful affordances for such reformulation. To best study that collaborative process we examined data from our software design project, in which student teams created computer software artifacts via programming and multimedia design. Unlike many science classrooms where students engage in science activities together with the teacher, in these software design projects students within a team decide on which research questions to ask, how to integrate them within their software designs, and how to make connections between different science topics. We analyzed video data that captured students' collaborative planning sessions at the beginning (Week 1), middle (Week 5), and end (Week 9) of the 10-week-long design project. We chose these planning sessions because they represented an opportunity to examine students' science discussions in a more private forum, free from some of the social pressures found in public classroom discussions with their teacher (Abrams, 1997). Our results reveal to what depth student teams verbally address science content during design planning and its development over the course of the project. Furthermore, we paid attention to which design contexts proved to be particularly fruitful for situating science discourse and in which ways students contributed to initiating and expanding science talk. In addition, we also examined less productive contexts and individual student contributions. In our discussion we address the issue of which design contexts afford opportunities for richer discourse and the implications for other project-based learning activities.

SETTING THE CONTEXT

Students' problems with science learning have been well documented in the research literature. Coming to understand science is not an easy process for most students. From the informal knowledge gathered from everyday experiences that students bring into the classroom and must then reconcile with scientific knowledge, to students' difficulties with systematically conducting science research and analyzing evidence, facilitating science understanding poses a significant problem for curriculum design. Various curricular and technological innovations have been created to support students in all of the following: communicating scientific findings to an audience (Songer, 1996), emulating scientific discourse by examining and making arguments (Linn & Hsi, 2000), or facilitating and enriching discussions through peer interactions (Brown & Campione, 1994). Other activities have provided tools that support students in explicitly addressing their informal knowledge (e.g., Hunt & Ministrell, 1994), developing fruitful research questions (e.g., Scardamalia & Bereiter, 1992), or visualizing and representing data (Gordin, Polman, & Pea, 1994; Jackson, Krajcik, & Soloway, 2000).

These studies have greatly contributed to our understanding of how students can engage in various aspects of science practice with the support of argumentation templates, task constraints, or data representation tools. The approach used in the current project draws on this research but with two significant differences. First, rather than providing students with educational software designed by content specialists and programmers, here students themselves are placed in the role of instructional designers, as they collaboratively create and implement software to help younger students learn about neuroscience concepts (Kafai, Ching, & Marshall, 1997). As part of this process they have to decide the focus of their science software designs, learn programming skills, deal with work and resource allocation issues, and develop appropriate representations and explanations of science phenomena. Consequently, students' science inquiry is situated within all of these activities. For example, although students are thinking about neurons and dendrites, they have to make decisions about their graphic representation, whether to animate particular aspects or parts, and who is going to work on different tasks. The goal is to allow students not only to cover general science concepts but also to gain familiarity with particular concepts that interest them—thus providing a context that engages students in essential science inquiry skills, such as framing questions that can be answered, conducting research to get answers to those questions, and expressing what they have learned to others via their instructional software designs. Such science context and inquiry skills seem particularly appropriate for elementary students as an introduction and preparation for later more formalized science activities (see Abrams, 1997).

A second important difference refers to the learning-through-design approach developed for this classroom project (Kafai, 1995, pp. 6–17). When students are asked to develop instructional software designs to explain science concepts, they are engaged in what could be described as an artistic or architectural view of design as a process. The core feature of such a design process is finding issues that need to be addressed and coming up with plans for addressing them (e.g., Harel, 1991). Such a view of design is different from an engineering view of design where one designs and constructs artifacts or working models and iteratively works toward a solution to a design challenge by trying something out, seeing and explaining where its deficiencies lie, and revising (e.g., Kolodner, Crismond, Holbrook, & Puntambekar, 1998). Although both approaches result in students creating artifacts for learning, a crucial difference is that artifacts generated under an "engineering" model of design provide direct feedback about the appropriateness of the design implementation. For example, in the process of designing an elbow, students actually can test the functioning of the elbow using the design artifact (Penner, Lehrer, & Schauble, 1998). The design of an instructional presentation does not provide such direct feedback, because an "architectural" design process produces good or bad solutions, not right or wrong ones. In our case, feedback is provided by the intended audience (described in Method section as "usability testing") and by student peers in planning discussions. Such a design approach to science inquiry promotes students' ability to express their ideas and interests while integrating them within a science context.

This design context for science learning addresses several challenging issues also found in other project-based approaches: those of the nature of science discourse, the particular contexts in which such discourse occurs, and collaborative contributions to discursive events. In terms of the nature of science discourse, it has been recognized that the development of scientific understanding is more than just learning science concepts and conducting experiments; it is an induction into particular discourse practices and activity structures that are part of the scientific community. Studies that analyze classroom discourse illustrate the methods by which students and teachers co-construct what counts as "science" and what does not (e.g., Carlsen, 1992; Lemke, 1990). Research on actual classroom scientific practices compared to those portrayed in textbooks paints a picture of highly contextualized activities (e.g., Roth & McGinn, 1997). Within our software design project, we can examine how students contextualize science and negotiate through different types of discourse while managing the other equally important issues of collaboration, resource allocation, and tool mastery. The issue of “standard science talk” versus more colloquial forms (Gallas, 1995; Lemke, 1990) is of particular interest here, because our analysis focus is not on classroom sessions mediated by the teacher but on planning sessions, in which each team of students collaboratively plans their software science designs.

We also examine the different software design contexts in which science discourse occurs. Our approach here compliments and builds on the work of other researchers who have focused on the role of mediational tools (Jones et al., 1999) or inscriptions (Roth & McGinn, 1998) in classroom interactions. Our approach draws from this research but emphasizes the particular role the software artifact can play in shaping students’ collaborative science design and inquiry. To better articulate this process, Harel and Papert (1990) introduced the concept of *knowledge reformulation* to describe the role software products play in mediating a translation between digital and nondigital media forms, such as graphic fraction representations and Logo program code. Here we expand the concept of knowledge reformulation by examining the translation processes between different forms of expression, or how the vocabulary of science can be integrated within the grammar of software designs. Student designers have to think about how to present the science they are learning on the screen in a format that younger students will understand and connect with. In addition, the designers themselves are students; therefore, they need concrete contexts that connect with their own experience for helping them wonder about and search for solutions to their own science questions.

The collaborative nature of software design also raises important issues related to research on students’ interactions and contributions. The research literature on collaboration is rich with debates on how to constitute teams, how to orchestrate interactions, what kinds of tasks to assign, and what sorts of rewards to offer for success (see Webb & Palincsar, 1994). Yet most of this research has taken place in the context of short-term and well-defined tasks. Teams that work on more open-ended problems in a long-term context face very different challenges, as Cohen (1994) pointed out. It also appears that highly scripted interactions, as found for example in reciprocal teaching (Salomon & Globerson, 1989), might not be genuine or appropriate for tasks such as software design. More recent studies of small team collaborations over extended time periods present a complex picture of how students’ interactions and contributions are configured (Anderson, Holland, & Palincsar, 1997; Artzt & Lamour-Thomas, 1991; Bianchini, 1997; Daiute & Dalton, 1993; Fuchs et al., 2000; Hogan, 1999; Hogan, Nastasi, & Pressley, 2000; Lumpe & Staver, 1992; Richmond & Striley, 1996). A consistent finding in all of these studies is that student team conversations are on-task, but students’ participation varies considerably across teams. At the same time, intervention studies designed to provide students with discourse-related, collaborative practice reveal less promising findings: although students were able to master the techniques, many of them did not implement them in their actual team sessions (Hogan, 1999) or focused in their interactions on procedural and surface science issues (Bianchini, 1997).

Of particular interest to our study are observations that address the nature of students' productive participation in collaborative interactions. Although considerable research has pointed out the benefits of grouping together students of varying abilities (e.g., Webb, 1983), this has been mostly discussed in terms of generalized academic skill differences. Fuchs et al. (2000) found that students' previous collaborative experience (irrespective of status) had a significant positive impact on team work productivity in solving a complex, mathematical task. Hogan et al. (2000) compared teacher-directed with peer-directed discourse in small team scientific discussions over 12 weeks and found that peer-directed teams offered more opportunities for generative and elaborative science discourse, whereas teacher-directed teams focused more on conceptual aspects. Even when no particular collaborative experience is present, researchers have argued that student "novices can be masters" in collaborations (Daiute & Dalton, 1993).

In our software design project student teams are comprised of more and less experienced students. We propose a distinction based on students' experiential level (i.e., participation in a previous software design project). This approach resembles models of cognitive apprenticeship (Collins, Brown, & Newman, 1989), with the important distinction that not adults but students with previous experience are here configured as the more able participants. In accordance with Lave and Wenger (1991), we term the experienced students *old-timers* and the other students *newcomers,* who are new to the software design project. A focus in our analysis will be in which ways, if at all, old-timers and newcomers differ in their contributions to science planning discourse.

These three issues then—the nature and quality of students' science discourse, the ways in which particular design contexts afford a focus on science, and the nature of collaborative contributions and effects of students' experience levels on science talk—constitute the framework for our analyses of how elementary students contextualize science within instructional design planning.

METHOD

Participants

The participants in this study were 31 upper elementary students from one self-contained multigrade classroom in a school in metropolitan Los Angeles. The class was composed of 16 fourth graders and 15 fifth graders, with 14 boys and 17 girls whose ethnic backgrounds are representative of the distribution found in the State of California: 47% of the students are White, 19% are Hispanic, 12% are

Asian, 14% are African American, and 8% are of mixed background. All of the students used computers as part of other classroom activities, such as word processing and Internet searching, or with specifically selected educational software.

Eleven of the 31 students had participated in a similar research project the previous year. The group of experienced students, or “old-timers,” consisted of 10 fifth-grade girls and 1 fifth-grade boy. Before that year, the same classroom teacher had already participated in two learning-science-by-design projects with other students, one on astronomy software and the other on ocean environments, from which the 11 old-timers originated. Students were divided by the classroom teacher and the researchers into teams of 4 or 5 students each; each team contained at least 1 experienced designer, a mix of fourth and fifth graders, and a mix of both genders. Students worked in these teams for the entire 10 weeks to collaboratively create their software designs. The current analysis looks at all seven of the teams involved in software design.

Classroom Setup

The classroom area consisted of different areas: a “rug” area in front of a whiteboard and seven table clusters, each of which had Macintosh PowerPC workstation connected to the Internet. In addition, there were four other computers in the classroom against the walls, one of which was used mostly by the teacher to demonstrate projects or activities on a large-screen TV display, while the other served as a scanner station. All the computers contained Microworlds™ software, which students used to program their instructional software. Each team had also a foldable planning board that consisted of three connected panels onto which students pasted their instructional screen ideas, computer work schedules, printouts of Internet searches, and other items.

Classroom Activities

The science intervention and related design project lasted 10 weeks. The class spent 75 min, 4 days per week, on science and software design-related activities described later in more detail. A thorough description of the curriculum students experienced is necessary to understand the context in which students’ science talk and design planning discussions occurred (see also, Galas, 1997–1998). The introduction to the science unit starts out with an all-class discussion in which students generate all the questions they have about the science topic of neuroscience. Over the course of the next 3 weeks, students revisit their questions and decide on which ones will become their own research question. Some examples of individual research questions are: “What controls our dreams?” “How do your eyes see?” “How does your brain know to turn around the picture you see?" or "How does memorizing things and memory work?"

A central information source for the neuroscience simulation project is a World Wide Web site[1] called "Neuroscience for Kids" developed by Eric H. Chudler, University of Washington. This site contains a variety of resources: background information in text and graphics, on-line memory and vision experiments, and game activities. All the computers in the classroom have Internet access; thus, students visit this site not only when the teacher introduces topics in class sessions, but also when they are in their teams to conduct research for their research questions. Halfway through the project, some students also write e-mail questions to Eric Chudler if they cannot find pertinent information on his Web site. In addition, students visit the neuroscience laboratory on the university campus, the students conduct a dissection of a sheep eye, and a brain surgeon (a father of one of the students) comes to talk about his work and brings a human and cow brain for comparison purposes.

The development of research questions and design of first software screen ideas continues on each team's planning board. After the generation of wonderment questions, all the teams meet at their work stations to begin discussing their software ideas, the distribution of computer time, and other work issues. To begin these sessions, the teacher often asks one team to showcase how they would conduct their planning session. Students met every 3 weeks for 15 to 25 min to formally discuss their project planning; these sessions are videotaped. These taped sessions are not the only opportunities for team planning, however. Often one can observe team members informally assemble in front of the planning board and discuss ideas, or individual team members decorating and expanding on planning board ideas that have been posted previously. We have discussed in a separate article the changing functions of the planning board within these teams (see Marshall & Kafai, 1998). When teams are not explicitly engaged in planning, students work individually on their software designs or participate in science lessons.

Parallel to the generation of research questions, all teams participated in a 3-hr introduction to basic graphic and software design functionalities of their programming environment, Logo Microworlds™. The main project activity of the introduction is to design a simple animation of muscle movement. In this context, students learn about basic graphic tools and actions such as cutting and pasting, hypermedia functions such as creating pages and linking them via buttons, and turtle programming such as repeated movements. In each team, the old-timers are asked by the teacher to help the newcomers learn the basic function while creating their first animation. In the following weeks, there are no explicit times set aside for programming instruction; it is rather a "learning on

---

[1] The URL is: http://weber.u.washington.edu/~chudler/neurok.html

demand" model in which students express an idea of what they want to program and ask either their team members, other teams, or their teacher for assistance.

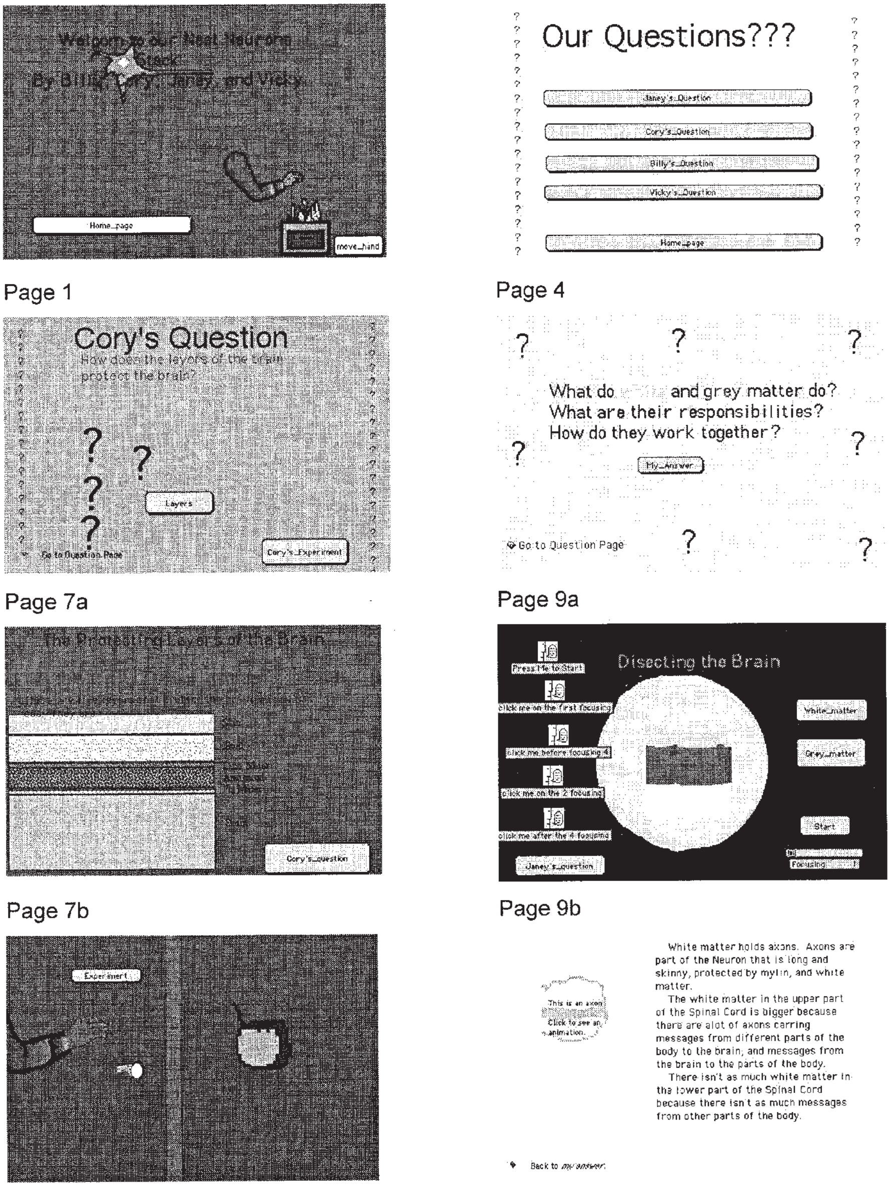
To provide an illustration of a team's final software project, we have included a printout of selected screen pages in Figure 1. We selected this project for two reasons: because it contained a rich mix of various central science concepts and illustrations, and because we use this project in the Results section in a comparative analysis of team performances. The whole project of Team 4 consists of 22 pages designed by Jamie[2] (old-timer), Val (old-timer), Bob (newcomer) and Calvin (newcomer).

Each of these students was asked to create a software "simulation" for their research question. Note that these software simulations are not simulations in a conventional sense; more often they are animated representations of a process or static representations that are sequentially linked over several pages. Students were asked to create "simulations" to encourage them to emphasize process or systemic aspects of science concepts rather than to simply display lists of facts.

The opening page introduces the team's chosen name, "The Neat Neurons," and displays an animation of a hand touching a hot plate. The team worked on this screen as part of their introduction to Microworlds Logo interface and commands. A "theme" button leads to the next page, which includes a short demonstration of the spinal cord and its different layers and a link to the brain page. The brain page was a collaborative project of all team members completed at the end of the project and displays the different parts of the brain and contains a short animation. The home page also leads to the question page (page 4) listing the research questions developed by the individual team members. Each button links to a sequence of pages that provide answers in different forms—only Calvin's and Jamie's screens are included in Figure 1. Val's question, "How do blind people dream?" includes a brief statement and a graphic illustration of the same answer. Calvin's question (page 7a) is "How do the layers of the brain protect the brain?" and his answer provides an annotated graphic display (page 7b), instructions for an "experement [*sic*]" (page 7c), and a demonstration of the experiment's results (page 7d). His split-screen animation showing how an egg breaks when shaken in a pan with no water but doesn't break when shaken in a pan with water is an example of how software design allows students to depict their knowledge in personally meaningful ways. Here Calvin has created an everyday context for explaining how layers of fluid protect the brain from jarring against the skull. Bob's question page asks "What controls what we dream about?" and provides a split screen that shows a dream and the animation in real time. In addition, Bob created an extra page that shows how the different brain parts develop in the fetus over a 9–month time line. Jamie's questions (page 9a) addressed "What do white and gray matter do? What are their responsibilities? How do they work together?" She then provided a nar-

---

[2] All student names are pseudonyms.



Page 1

Page 4

Cory's Question

How does the layers of the brain protect the brain?

Page 7a

What do [illegible] and grey matter do?
What are their responsibilities?
How do they work together?

Page 9a

Page 7b

Page 9b

Page 7d

This is an axon. Click to see an animation.

White matter holds axons. Axons are part of the Neuron that is long and skinny, protected by mylin, and white matter.

The white matter in the upper part of the Spinal Cord is bigger because there are alot of axons carring messages from different parts of the body to the brain, and messages from the brain to the parts of the body.

There isn't as much white matter in the lower part of the Spinal Cord because there isn't as much messages from other parts of the body.

Page 9c

FIGURE 1 Final software project of Team 4.

rated step-by-step description of the different parts (page 9b) in which the graphic display changes according to the verbal descriptions that can be requested via pressing buttons. She has two additional content screens (page 9c only) that explain in more detail gray and white matter in words and pictures.

Together, the screens created by each student and the jointly created screens (table of contents, title page, etc.) make up Team 4's completed project. Note that all software screen pages were created in color but are shown here in black and white. More important to point out is the problem of representing active, animated representations via static graphics on paper—only part of the simulation can be shown and thus does not display the real complexity of what the designers created. Despite this difficulty, the screen shots in Figure 1 should give the reader some idea of how teams of students were able to represent their science understandings as both descriptive content and systemic phenomena in the Microworlds™ authoring environment.

As students create these software products, they engage in another design project activity called "usability testing." Twice during the project, at Week 5 and 8, a team of 14 third graders is asked to come and test out the instructional software designs for review. But the evaluation visits by the third graders are not the only time points when students demonstrate and discuss their instructional science software designs. Throughout the project students are free to move within the classroom and visit other teams' workstations. Furthermore, the teacher schedules class demos on a rotating schedule in which each team is given 5 min at the beginning or end of class to showcase their work in progress. One of these demonstrations runs under the heading "What have you accomplished to answer your research question?" and other students as well as the teacher provide suggestions. At the end of the project, the whole class meets to discuss criteria for software evaluations and generates a list of questions for final evaluations by the other design teams. The next day, each team visits three other software designs and tests the software.

Students' participation in *all* of these activities provided the context for their integrative learning of software design, collaboration, project management, and science. It is important to note that although the outline of activities might suggest a lockstep sequencing, most activities were scheduled based on a perceived need by the teacher or the researchers to discuss particular topics (e.g., the current state of software development and science integration).

Data Collection

The main data come from the three project planning sessions (Week 1, Week 5, and Week 9) in which all student teams participated**.** During the planning sessions teams were given approximately 10 to 25 min exclusively devoted to jointly creating and negotiating their plans for their developing software design software. All team planning time for each team was videotaped and transcribed completely.

Transcripts were then coded for the extent to which students engaged in science talk during planning about their designs. The data corpus for this article was 21 transcripts, 3 for each team. In addition, we included the software designs created by each team in our analyses.

Identifying Science Talk

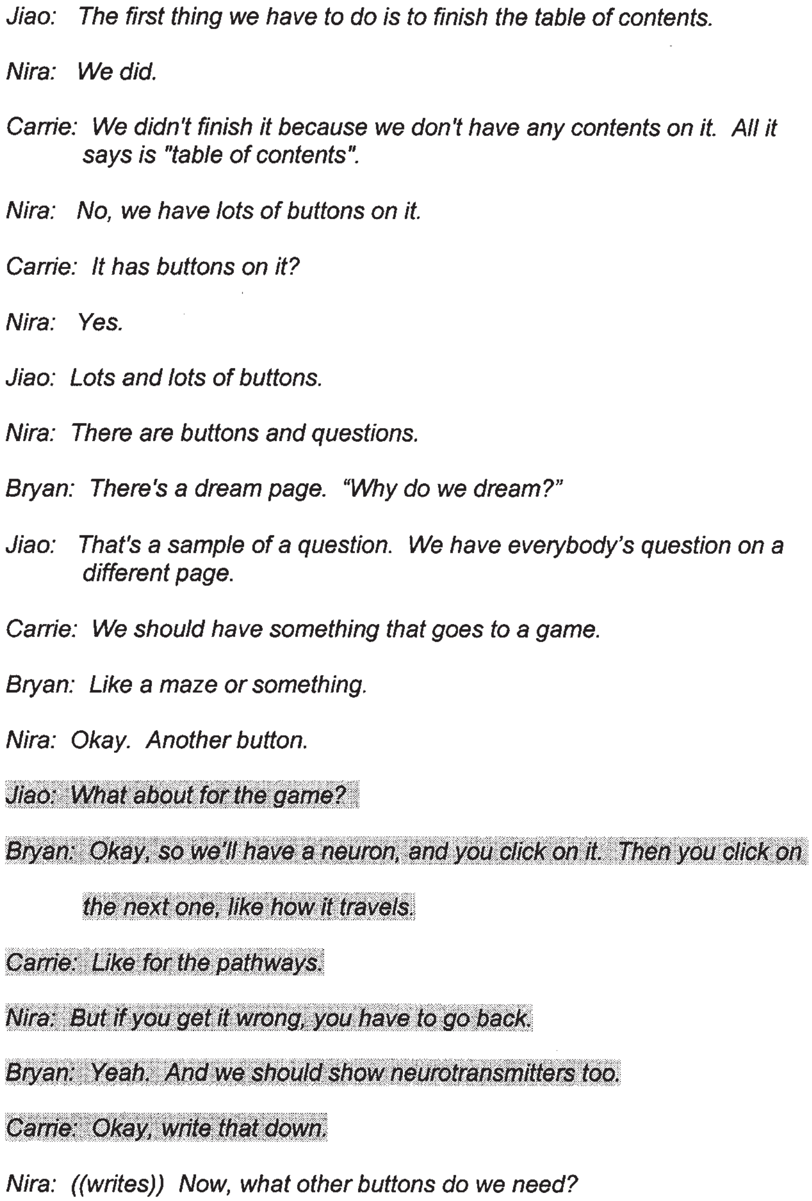
The first step was to determine which parts of the transcripts from students' planning sessions would be coded. We identified our unit of analysis, which was any segment of discourse in the transcripts containing references to and discussions of science concepts. Although this identification is based on a very open definition, it is in line with definitions used by Lemke (1990) and Gallas (1995). This phase of the method was performed in line with Erickson's first three stages of ethnographic analysis of interaction (Erickson, 1992). Our process involved viewing the entire data corpus for all three time points and creating an index of what activities and interactions were occurring, followed by decisions about what segments of discourse to include in the analysis and what to leave out.

Segments were defined as however many conversational turns it took for a complete idea to be introduced, treated, and then dropped or closed. Thus segments could be of varying lengths. Science talk segments were typically contexted within talk about other design issues. An example of a short science segment contexted within other design talk is shown in Figure 2. This segment is taken from Team 3's first planning session. Talk during this first session for Team 3 consisted largely of brainstorming many ideas with little extended treatment of any one science topic. Here Team 3 is talking about what links should appear on their table of contents page, when they veer off briefly into a discussion of a possible game they might include in their software. The brief portion we identified as "science talk" out of this longer discussion is shaded.

In this section Bryan brings up the idea of having a maze game that has something to do with neuron impulses. The shaded segment in Figure 2 shows other members of Team 3 helping to construct the game idea. The goal here is obviously not to elaborate either the game logistics or the science content, however, but rather to simply state the idea and record it (as Carrie says, "write it down"). Consequently, the segment is fairly short. Many segments were longer than this, however, and they comprised more conversational turns, but they still dealt with a single complete idea and were thus selected for segments the same as the short one earlier. Several examples of longer science talk segments can be found later in the article.

Timing Science Talk

We wanted to determine not only if science talk was present in these sessions, but also how much of the planning sessions students actually spent engaged in science



Jiao: The first thing we have to do is to finish the table of contents.

Nira: We did.

Carrie: We didn't finish it because we don't have any contents on it. All it says is "table of contents".

Nira: No, we have lots of buttons on it.

Carrie: It has buttons on it?

Nira: Yes.

Jiao: Lots and lots of buttons.

Nira: There are buttons and questions.

Bryan: There's a dream page. "Why do we dream?"

Jiao: That's a sample of a question. We have everybody's question on a different page.

Carrie: We should have something that goes to a game.

Bryan: Like a maze or something.

Nira: Okay. Another button.

Jiao: What about for the game?

Bryan: Okay, so we'll have a neuron, and you click on it. Then you click on the next one, like how it travels.

Carrie: Like for the pathways.

Nira: But if you get it wrong, you have to go back.

Bryan: Yeah. And we should show neurotransmitters too.

Carrie: Okay, write that down.

Nira: ((writes)) Now, what other buttons do we need?

FIGURE 2 Short science talk unit contexted within other design talk.

talk. To investigate this question we had to time students' discussions. After all planning sessions had been transcribed and units of science talk were identified, session videotapes were watched to time occurrences of science talk and length of the overall sessions. Because planning sessions varied in length across teams, we timed the total length of each planning session for each team and used that amount as a baseline. Session timing started when students began discussing their ideas. Session timing stopped when the researcher decided that the students had come to some sufficient decisions to plan their work for the following weeks and thus stopped the conversation to hand out planning session evaluations. Length of individual sessions ranged from 12 to 34 min (some teams constructed plans more quickly than others). Individual science segments of discourse were also timed, using stop and start times according to the conversational turns that initiated and dropped the particular topic. Individual segment lengths were then added together for each session, and percentages of science talk for each team and each session were calculated based on the ratio of total science talk time to total session length.

Coding Scheme

The fourth phase of Erickson's ethnographic microanalysis involves identifying patterns in the data that are representative of the types of interaction and also illuminate significant points of contrast. To create these categories, at each time point we reviewed the selected segments of science talk while taking copious notes on the nature of science being discussed and the way science talk seemed to be situated in relation to other design project work. Then we created a comprehensive list from our annotations of the variety of specific kinds of talk we observed. There were different types of design topics—talk about how particular screens would look and function, talk about navigation or software organization, talk about users, and talk about distributing tasks among team members—as well as systemic and descriptive science talk. (Each of these categories will be explained in more detail in the following sections.)

The next step was more conceptual in nature: at each time point we took the list we had made and derived from it a framework wherein various kinds of science talk were classified. Then we re-read all the science talk segments again to test the framework and ensure its comprehensiveness, adjusted the framework if necessary, and selected seminal examples from the data of particular framework categories—this process is in line with Erickson's fifth and final stage of analysis: testing the representativeness of the categories. After we identified all the segments of science talk in each of the 21 transcripts, we double-coded all segments for the following: (a) the qualities of science talk in which students were engaged and (b) the conversational context, which triggered the introduction of science, discourse.

In our double-coding system, the first code each segment received was one that captured the conversational context in which science talk took place. Because the main goal of the planning sessions was to negotiate designs and work plans for each team's software design, science talk in this context almost always arose out of talk about something else. We identified four topics in students' discussions, which gave rise to talk about neuroscience:

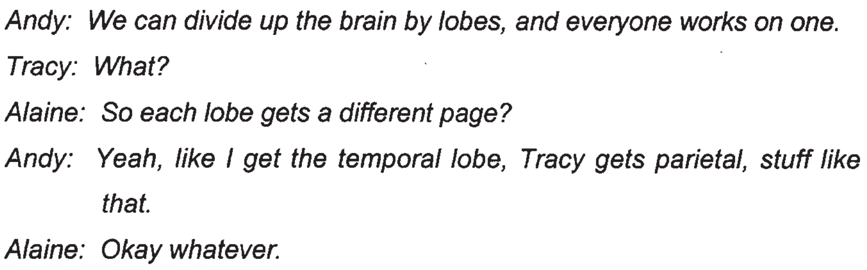
1. Software design screens. In this context students discuss the appearance, content, or functioning of an individual screen within their software.
2. Software functioning. Students talk about the overall layout of their software, navigation between pages, or other topics concerning more than one screen at a time, or screens that do not explicitly represent science content.
3. User consideration. In this context students consider how to make the software appealing or understandable to their younger users.
4. Work distribution. Students make plans for who in their teams will be responsible for various tasks in the implementation of their plans.

We used the combination of context and the level of science talk to investigate how students used the design planning session as a forum for reformulating what they were learning about neuroscience. Because all segments received a double code, examples of each of these categories are combined with examples of science talk quality, as described in the next section.

The second code each segment received identified the quality of science talk contained in the discourse. In terms of science talk quality, we were interested in capturing the diversity of science conversations observed in students' planning sessions. For this reason we chose to distinguish between *descriptive* talk, in which students incorporated the vocabulary of neuroscience into their discussions, and *systemic* talk, in which students articulated their ideas and understandings about neuroscience phenomena within the context of planning their designs. This distinction between descriptive and systemic science discourse is drawn from recent studies of students' collaborative scientific reasoning (Bianchini, 1997; Hogan et al., 2000). In these studies, researchers noted that student teams often did not move beyond procedural and observational matters in their science coverage (Bianchini). We have chosen to call such references to science descriptive. Systemic talk, on the other hand, refers to more productive content in students' science discourse, as has been identified by previous research on interactions in student teams when students explore and expand on each other's ideas (Hogan et al.). We have provided two examples of different combinations of design contexts and science talk quality to illustrate our double-coding system.

Descriptive/Work Distribution.

The segment in Figure 3, in which Team 1 discusses who will be responsible for programming which lobe of the brain, is a good example of descriptive talk within in a work distribution context.

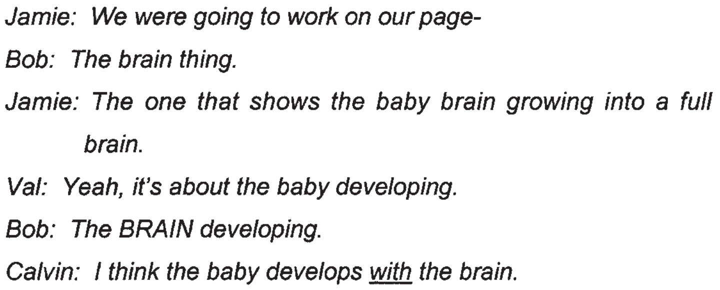


Andy: We can divide up the brain by lobes, and everyone works on one.
Tracy: What?
Alaine: So each lobe gets a different page?
Andy: Yeah, like I get the temporal lobe, Tracy gets parietal, stuff like that.
Alaine: Okay whatever.

FIGURE 3 Descriptive talk in work distribution context.

This segment is taken from Team 1's first planning session. The team is engaged in a preliminary discussion about what content will be covered in their software and who will be responsible for programming which parts. Prior to the first planning meeting students completed an activity in class, which provided a visual aid for identifying parts of the brain and basic functions of each one. Andy, an old-timer, suggested that the team incorporate this activity into their software. The discussion transcribed in Figure 3 focuses on how team members will divide the work involved in making the "lobes" part of their software. This segment is classified as descriptive because the students use correct terminology for describing the lobes (parietal, temporal, etc.), but they do not talk about brain functionality or what each lobe does.

Systemic/Screens. Compare the discussion by Andy's team (Figure 3) with another example from a different team, in which the students talk in a conceptual manner (Figure 4). This segment comes from Team 4's second planning session. Toward the end of the session the students make plans for their immediate work priorities. In response to the researcher's question, "what are you going to work on tomorrow?" the team describes one screen of their partially completed software design, which shows several phases of fetal brain development. In this segment, the team not does not use as many neuroscience terms as in the earlier example from Andy's team; however, the students are engaging with the idea of fetal brain development and are explaining to one another their views of what should appear on the screen. More important, in the last several turns among Val, Bob, and Calvin, they are attempting to decide what will appear on the screen based on their understandings of the phenomena of fetal brain development—whether they should show the brain developing independent of the fetus itself. Interestingly, although they have already begun work on this part of the software design, it is obvious that team members disagree on the nature of the science content being represented. The format of the planning session allows this disagreement to come to light, when it might have been overlooked in day-to-day programming activity.



Jamie: We were going to work on our page-

Bob: The brain thing.

Jamie: The one that shows the baby brain growing into a full brain.

Val: Yeah, it's about the baby developing.

Bob: The BRAIN developing.

Calvin: I think the baby develops with the brain.

FIGURE 4 Systemic science talk in screens context.

As our last step of analysis, we selected only the systemic talk segments for a secondary analysis of the roles old-timers and newcomers played in jointly constructing systemic discourse. Each systemic segment received one code that indicated who initiated the particular science segment, an old-timer or a newcomer. Initiation could be a question posed to the team that prompted other members to start talking in scientific terms (e.g.,"Okay, how should we draw the brain?"), or it could be an entirely new topic brought up by someone launching into a scientific description on their own (e.g., "Hey, this is my idea for my simulation"). Then each systemic segment was examined to determine if a team member extended the discussion, and if so, by an old-timer or a newcomer. Extension was defined as any conversational turn that served to draw the discussion into a deeper treatment of science. For example, a student may have challenged the science assertion put forth by another team member, or he or she may have asked a question that required a more specific and scientific answer (e.g., "How do you know that babies dream more?"). Not all science discussions were extended, however; some were fairly brief or unchallenged. We used these initiation and extension codes to examine the ways old-timers and newcomers might participate differently in systemic discourse.

Two independent raters coded all of the transcribed planning sessions. Fifty percent of the transcripts were coded for reliability purposes, and inter-rater reliability was established for both science ($\alpha = .84$) and context ($\alpha = .82$) coding. Only exact agreement was considered acceptable for reliability purposes. Rater disputes on particular items were then resolved through discussion to obtain final codes for analysis.

RESULTS

In this section we report the findings of our analysis using the coding scheme as described in the Method section. The scheme itself, though, represents only one part of our results, in that it was an emergent scheme and was constructed to describe the nature of science talk within design. We had initially hypothesized that we would, in fact, find science talk in these planning sessions; however, the quality of that talk, how it arose, and how it was situated within various design planning contexts were open questions. In describing our results here, we focus on three positive findings of our analysis:

1. The effectiveness of software design planning as an activity for promoting science talk.
2. The importance of focusing on the science content of software design screens in discussions.
3. The influential roles of old-timers in shaping science talk.

In the fourth section, we present a counterpoint to these findings by describing a problematic team that did not engage in productive science talk and offer some explanations for this result.

Effectiveness of Design for Promoting Science Talk

We found that science talk does happen within the context of design. This assertion is supported by findings from two related strands of analysis: an examination of the amount of time student teams spent talking about science during their design planning sessions overall and a differentiated view of time spent on the quality of science discussion in each of the three sessions.

On average, student teams spent about 20% of their time in the planning sessions engaged in some kind of science discussion (*M* = 20.3%, *SD* = 6.17). For the purposes of this strand of analysis, systemic and descriptive science talk were combined. Considering that science discussions were not the explicit focus of these sessions, and that students had many other logistical concerns to contend with during this time, this finding provides some support for the argument that science does not get left out of the picture.

We also noticed a trend in the percentage of science discourse over all three sessions (see Table 1). On average, student teams spent an equal amount of time in the first session (*M* = 24.3%, *SD* = 24.08) and second session (*M* = 23.4%, *SD* = 14.29) engaged in science talk. By the third session, however, the average amount of science talk for teams had decreased almost by half (*M* = 12.8%, *SD* = 10.15). Notice, however, that there was a great deal of variation in each session, due to the fact that some teams focused more or less on science content at each of the three time points, and two of the teams' discussions involved little science at all. These team differences are discussed later.

To analyze the quality of students' science talk, we then converted the percentages of descriptive and systemic talk to cumulative percentages. In other words, out of the total amount of science talk time teams had, we measured what percentage of that time was descriptive talk and what percentage was systemic talk (see Table 2). The results yielded do not sum to 100% in all cases, however, because one team had no instances of science talk at all in Sessions 2 and 3. (This problematic team is described in more detail in the last section of Results.) In terms of the quality of students' science talk, we found that students' discussions were more often characterized by descriptive talk (M = 63.9%, SD = 12.9) than by systemic talk (M = 26.5%, SD = 8.55). Over the three sessions, this pattern holds true as well; teams tended to talk more in descriptive than systemic terms (see Table 1). By the third session, however, the pattern changes. Although the average amount of descriptive talk decreased, systemic talk increased

TABLE 1
Percentage of Science Talk per Session per Team

Team	*Session 1*	*Session 2*	*Session 3*	*Average*
1	39.7	6.2	16.9	20.9
2	18.8	0.0	0.0	6.3
3	17.3	26.5	5.9	16.6
4	73.0	38.2	20.0	41.0
5	8.6	25.4	30.0	21.3
6	11.2	29.7	6.9	15.9
7	4.9	34.0	9.9	16.3
Average	24.9	23.1	12.8	

Note. Percentages are out of total planning time.

TABLE 2
Percentage of Descriptive or Systemic Talk per Session

	Session		
Talk	*1*	*2*	*3*
Descriptive			
M (%)	75.3	66.6	49.9
SD	27.2	36.98	32.5
Systemic			
M (%)	24.7	19.1	35.9
SD	27.2	24.0	28.7

Note. Percentages are out of total science talk; Sessions 2 and 3 do not sum to 100%.

in the third session, such that by Session 3, students' science talk was fairly evenly distributed between systemic and descriptive talk.

Focus on Science Content of Software Design Screens

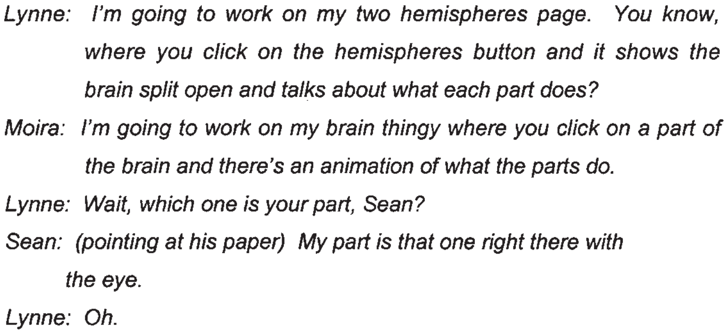
For our analysis of how different design contexts and scaffolds afforded a focus on science in the planning sessions, we moved away from the question of how much time students spent in a given kind of talk. We approached this issue from two sides: we examined the specific conversational contexts themselves that gave rise to particular kinds of talk (and not how long the segments were in relation to the rest of the planning session), and we studied conversational prompts and continuations of team members. In the first analysis, we thus decided to make tallies of the total number of instances of each kind of design context (as outlined in the Method section and Table 3), and sort them according to the quality of science talk each one yielded per session.

We found that the most total instances of science talk occurred in the context of discussing software screens. Although talk incorporating neuroscience vocabulary (descriptive talk) was well distributed over all contexts, more systemic conceptual discussions happened almost exclusively within the context of science screens. Thus it appears that although talk about software screens does not automatically lead to systemic science discussions, the affordances are stronger. Talk within other contexts, however, apparently does not often yield deeper treatment of science concepts.

We found that although many discussions situated in other contexts had the potential to evolve into fruitful science explorations, students in these teams rarely picked up on the science part of the conversational thread; more often they focused on some other aspect. Take the example of a "work distribution" context from Moira's team (Figure 5) as an example. Here team members are engaged in a calendar-making activity during the first session. After each team member has been given a day on the computer to work, Lynne writes down what all team members are going to do during the following week on their assigned days. In this segment both Moira and Lynne make reference to brain functional-

TABLE 3
Number of Instances of Science Talk in Design Contexts

	Descriptive			*Systemic*		
Design Context	*T1*	*T2*	*T3*	*T1*	*T2*	*T3*
Screens	14	15	3	4	10	14
Software functioning	5	12	2	3	2	4
User consideration	2	13	2	1	0	1
Work distribution	3	17	8	0	2	2



Lynne: I'm going to work on my two hemispheres page. You know, where you click on the hemispheres button and it shows the brain split open and talks about what each part does?

Moira: I'm going to work on my brain thingy where you click on a part of the brain and there's an animation of what the parts do.

Lynne: Wait, which one is your part, Sean?

Sean: (pointing at his paper) My part is that one right there with the eye.

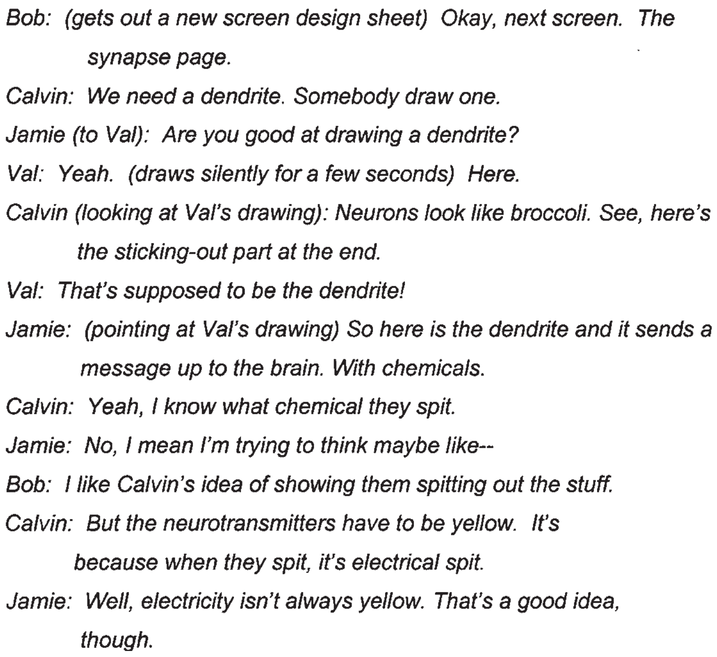
Lynne: Oh.

FIGURE 5 Work distribution context.

ity, but they don't pursue this topic any further. The team could potentially use this brief reference to open up the discussion further and explore what they each mean by "what each part does," but they do not. Once it has been established what each person is working on currently, the conversation moves on to an argument about who has more designated computer days than the others do.

When student teams focused more on the details of particular software design screens, specifically on science content and how it would be represented on particular screens in the software, we found that this context often led to much more fruitful science discussions. In an example from Jamie's team, the students start out discussing how they are going to draw a neuron on one screen, and they end up exploring ideas of neurotransmitters and electricity (Figure 6). This segment is taken from Team 4's first planning session. Early on in this session, the students in Team 4 begin describing their individual screen ideas for the software. Nothing of substance has actually been programmed yet; so talk about the students' ideas are focused on screen idea sheets. Students draw how they want a particular screen to look on these sheets and make notes about functionality below the drawing. Figure 6 shows the segment of discourse in which the team discusses Jamie's drawing of her idea for a screen about how neurons send messages.

Here an initial description of how this team's neuron screen should include something about chemical messages evolved into a more in-depth treatment of what those chemicals are, how they are transmitted, and how they should be represented. In this example we see that students not only incorporate neuroscience terms like "dendrite" and "synapse" into their talk, but they are also engaged in a discussion of how neurons send messages and the electric potential of neural signals. Note, however, that the students don't actually say "electric potential of neural signals." Rather, their systemic talk is couched in everyday, informal language—talking about "spitting out the stuff" and the necessity of using yellow to symbolize electricity.



Bob: (gets out a new screen design sheet) Okay, next screen. The synapse page.

Calvin: We need a dendrite. Somebody draw one.

Jamie (to Val): Are you good at drawing a dendrite?

Val: Yeah. (draws silently for a few seconds) Here.

Calvin (looking at Val's drawing): Neurons look like broccoli. See, here's the sticking-out part at the end.

Val: That's supposed to be the dendrite!

Jamie: (pointing at Val's drawing) So here is the dendrite and it sends a message up to the brain. With chemicals.

Calvin: Yeah, I know what chemical they spit.

Jamie: No, I mean I'm trying to think maybe like--

Bob: I like Calvin's idea of showing them spitting out the stuff.

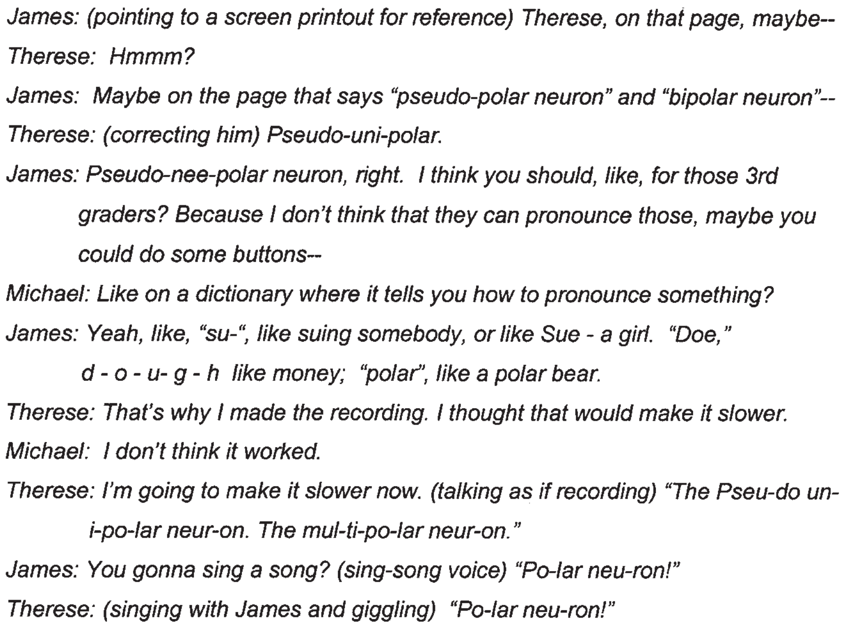
Calvin: But the neurotransmitters have to be yellow. It's because when they spit, it's electrical spit.

Jamie: Well, electricity isn't always yellow. That's a good idea, though.

FIGURE 6 Systemic talk about a simulation.

Important to point out here is that not all talk about screens yielded a systemic treatment of science. There is a crucial difference between a science screen that conveys content through text, a recording, or static graphics and a systemic animation screen. Talk about systemic animation screens is more likely to yield discussions about process—how particular phenomena should be represented. Compare Team 4's discussion about their synapse screen with a conversation from Team 5, in which they talk about a science screen that does not contain a systemic animation (Figure 7). During the last session, James provides a suggestion for Therese for a modification of her screen containing static graphics and the names of different kinds of neurons.

In this segment, both James and Therese employ scientific terminology, as they refer to different kinds of neurons by their official names and attempt to pronounce them correctly. They do not, however, engage in any kind of discussion about the differences among these neurons, what their names mean, or why Therese has included this screen in the software at all. The textual nature of Therese's screen likely contributes to the fact that James' query focuses not on whether the third-grade users will *understand* the different kinds of neurons, but rather, whether or not they will be able to pronounce the names. The names of



James: (pointing to a screen printout for reference) Therese, on that page, maybe--

Therese: Hmmm?

James: Maybe on the page that says "pseudo-polar neuron" and "bipolar neuron"--

Therese: (correcting him) Pseudo-uni-polar.

James: Pseudo-nee-polar neuron, right. I think you should, like, for those 3rd graders? Because I don't think that they can pronounce those, maybe you could do some buttons--

Michael: Like on a dictionary where it tells you how to pronounce something?

James: Yeah, like, "su-", like suing somebody, or like Sue - a girl. "Doe," d - o - u- g - h like money; "polar", like a polar bear.

Therese: That's why I made the recording. I thought that would make it slower.

Michael: I don't think it worked.

Therese: I'm going to make it slower now. (talking as if recording) "The Pseu-do un-i-po-lar neur-on. The mul-ti-po-lar neur-on."

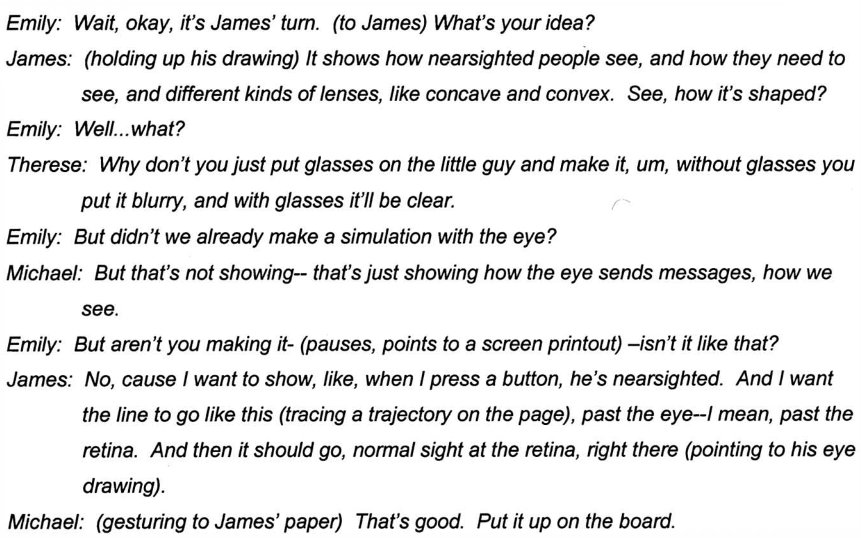
James: You gonna sing a song? (sing-song voice) "Po-lar neu-ron!"

Therese: (singing with James and giggling) "Po-lar neu-ron!"

FIGURE 7 Descriptive talk in screen context.

the neurons thus become an opportunity for word play among the designers, and Therese does not explain her screen or its contents to her team members.

Not only is talk about systemic animations (previously referred to as *simulations*) more productive than talk about other kinds of screens or other design issues, it is unique in quality and form. It became evident in examining the transcripts that science talk situated in an in-depth discussion of systemic animation screens has a very different character than either "standard science" discussions in other contexts or science discussions in the classroom. Not only do children use everyday terminology in referring to science content, but they also use software design terminology and processes to describe systemic phenomena. It appears that children are creating a fascinating hybrid of talk in these segments—talk that uses computer animation descriptions to convey systematic processes. As an example of this hybrid talk, a segment is provided in which James tries to explain his idea for a systemic animation about nearsightedness and farsightedness to his team (Figure 8). This segment takes place during Team 5's last session. James had not generated an idea for his own question screen until late in the project, and a large chunk of the last session for his team is thus devoted to helping him flesh out his plans and finish on time. As in the previous example, here too Team 5's science discussion centers on a drawing James has made of his screen idea. Talk about



Emily: Wait, okay, it's James' turn. (to James) What's your idea?

James: (holding up his drawing) It shows how nearsighted people see, and how they need to see, and different kinds of lenses, like concave and convex. See, how it's shaped?

Emily: Well...what?

Therese: Why don't you just put glasses on the little guy and make it, um, without glasses you put it blurry, and with glasses it'll be clear.

Emily: But didn't we already make a simulation with the eye?

Michael: But that's not showing-- that's just showing how the eye sends messages, how we see.

Emily: But aren't you making it- (pauses, points to a screen printout) –isn't it like that?

James: No, cause I want to show, like, when I press a button, he's nearsighted. And I want the line to go like this (tracing a trajectory on the page), past the eye--I mean, past the retina. And then it should go, normal sight at the retina, right there (pointing to his eye drawing).

Michael: (gesturing to James' paper) That's good. Put it up on the board.

FIGURE 8 Simulation/Science hybrid talk.

James' screen design continues for some time after the segment shown in Figure 8; however, the content of the remaining talk focuses exclusively on time management and programming suggestions.

James' idea is to make a systemic animation that shows how eyeglasses help reposition images to the correct area on the retinas of nearsighted or farsighted individuals. Rather than describing the biological phenomena to his team, however, James describes what will happen in the software design, with the help of a free-hand drawing and his own physical movements. He holds up his eye drawing and traces the trajectory of the lines of refraction that will be drawn by animation on the screen ("And I want the line to go like this."). He then indicates by pointing on the paper where the image should appear on the eye after vision has been corrected ("normal sight at the retina, right there"). James' talk represents a fairly sophisticated understanding of vision correction, and it appears as though he has been able to convey this understanding to his team. The question of whether his discussion is actually about the software design or about vision correction is irrelevant. It is clearly both.

Influential Role of Old-Timers

In addition to the different types of design contexts that situated science talk, another factor that contributed to students' science engagement was the influence of newcomers and old-timers in constructing systemic talk. As an example of how old-timers play an important role in structuring team discourse, look again at the exchange involving James' vision correction design (Figure 8). Emily is the old-timer in this team. Three different times during the conversation, Emily asks questions that require James and the rest of the team to elaborate beyond the initial animation description about differently shaped lenses. She forces the team to establish how James' design will be distinct from a screen they created previously that demonstrates how the eye works. Without Emily's extending questions, the team might have moved on to a different topic after James' initial description and consequently missed out on a moment of rich engagement with neuroscience concepts. We postulate that conversations that begin as descriptive treatments of science (such as James' initial description) can be drawn out and become systemic treatments through the use of extensions by other team members.

Before delving into the results of our analysis of old-timers' roles across all teams in our study, an important point should be addressed. One might argue that the old-timers in our study perform the extending function, as illustrated in Emily's example, not because of their prior experience in design, but because they are older. True, all the old-timers in our study are fifth graders, whereas most of the newcomers are fourth graders; thus, it would be easy to attribute old-timers' weighty role and positive influence in promoting science talk to age-related factors, such as a greater capacity for reflection or better verbal skills. In another study, however, this comparison of age versus experience was examined directly by creating parallel teams—half of which were composed similarly to those reported in this article, and half of which contained fourth and fifth graders who were all newcomers. Results revealed that the fifth-grade newcomers did not display the same kinds of leadership, insight into the design process, or influential roles in their teams as fifth-grade old-timers did (Ching, 2000). We can thus be fairly confident in asserting that old-timers in this study manifest their previous experience in their extensions of team science talk.

In this study, we found that Teams 1, 2, and 7 had very few instances of systemic talk, and what instances there were had been initiated by old-timers and not extended by anyone (see Table 4). Teams 3, 4, and 5, on the other hand, had more instances of systemic talk, which were initiated by both old-timers and newcomers, and these instances were more often extended. It is this extending function that marks a crucial difference between successful and unsuccessful teams. The fact that members of Teams 1, 2, and 7 were not engaged with extensive questioning in the segments we examined may help explain the fact that they spent a smaller percentage of planning time engaged in systemic discussions.

In addition to illuminating variation in the instances of systemic talk among teams, Table 4 also reveals an interesting trend with regard to how old-timers and newcomers contribute to systemic discussions. The "total" row at the bottom shows that systemic talk was initiated by both old-timers (16) and newcomers (10)

TABLE 4
Number of Old Timers' and Newcomers' Contributions to Systemic Talk

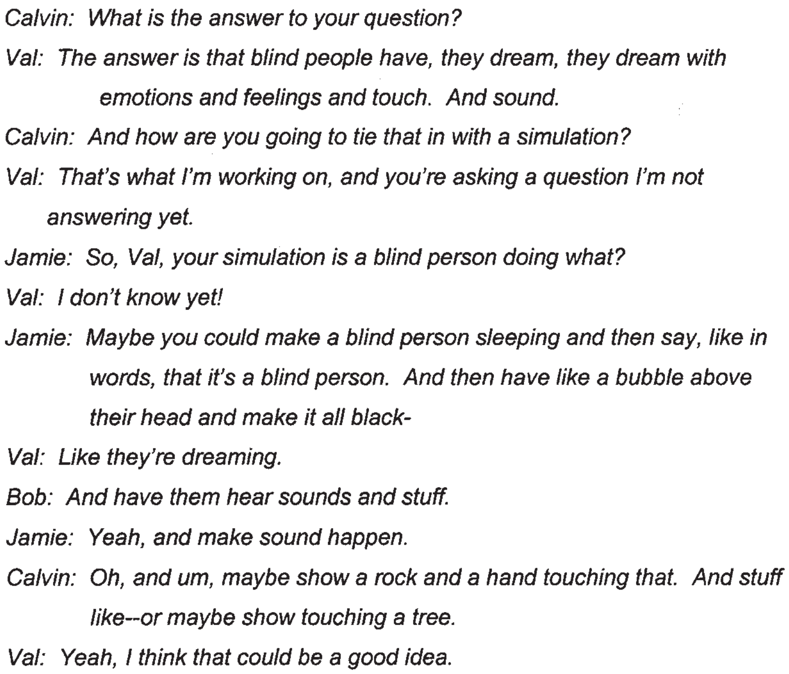
	Old Ttimers		*Newcomers*	
Team	*Initiates*	*Extends*	*Initiates*	*Extends*
1	0	1	1	0
2	0	0	0	0
3	5	2	3	0
4	6	4	1	1
5	2	3	4	1
6	1	0	1	0
7	2	0	0	0
Totals	16	10	10	2

in the sessions. Extensions, however, were contributed almost exclusively by old-timers (10), whereas newcomers rarely extended systemic science discourse (2). This type of extending talk shows that the old-timers in Teams 3, 4, and 5 are concerned about deeper level science issues than allocating tasks and ensuring content coverage. They help stretch other students' design ideas to engage more with science.

In the example in Figure 9, Jamie's team tries to help one of its member's come up with a concrete idea for a software design screen. This segment is also taken from the last planning session. As was the case with James in the previous example, Val did not generate a software design idea until near the end of the project. Prior to the segment transcribed in Figure 9, the team has been helping Val articulate her question in a concrete manner: "how do blind people dream?" When they start discussing how she will represent the answer to her question, the entire team grapples with how to translate her knowledge into a software design. Here all team members, not just old-timers, become involved in extending the discourse by asking questions and building on one another's ideas. We can also see more of the software design hybrid talk occurring here, as the team simultaneously tries to establish design ideas and grapple with the complex idea of how blind people dream. Of particular interest is how they attempt to design ways of representing nonvisual phenomena in the visual medium of a graphic animation.

Problematic Team Results

We observed a great deal of variation among teams in terms of the quality of their engagement with science across the three sessions. A listing for each individual team provides an explanation for this variation (see Table 5). Although we were able to see a general trend in science talk over two planning sessions with a decrease in the third, this does not apply to each team. It becomes appar-



Calvin: What is the answer to your question?

Val: The answer is that blind people have, they dream, they dream with emotions and feelings and touch. And sound.

Calvin: And how are you going to tie that in with a simulation?

Val: That's what I'm working on, and you're asking a question I'm not answering yet.

Jamie: So, Val, your simulation is a blind person doing what?

Val: I don't know yet!

Jamie: Maybe you could make a blind person sleeping and then say, like in words, that it's a blind person. And then have like a bubble above their head and make it all black-

Val: Like they're dreaming.

Bob: And have them hear sounds and stuff.

Jamie: Yeah, and make sound happen.

Calvin: Oh, and um, maybe show a rock and a hand touching that. And stuff like--or maybe show touching a tree.

Val: Yeah, I think that could be a good idea.

FIGURE 9 Multi-party extensions and simulation talk.

TABLE 5
Percentages of Descriptive (D) and Systemic (S) Science Talk per Team per Planning Session

	Session 1		*Session 2*		*Session 3*	
Team	*D*	*S*	*D*	*S*	*D*	*S*
1	100	0	100	0	76	24
2	100	0	0	0	0	0
3	66	34	79	21	15	85
4	81	19	31	69	61	39
5	40	60	95	5	40	60
6	40	60	76	24	86	14
7	100	0	85	15	71	29
Average	75.3	24.7	66.6	19.1	49.9	35.9

Note. Percentages are cumulative, but some results do not sum to 100, due to Group 2's having no science talk.

ent that there are at least two other trends: some teams have more than average science talk in the third planning session, and two teams at least show very little engagement in science talk at all.

We see the counter-trend observation of increased science talk as further evidence for team-specific developments. Although classroom activities such as the visits of third graders or the public inquiry provide “milestones” for all teams to compare their progress with that of other teams, it is difficult in such a complex project for all members and teams to be in lockstep development. As a closer examination of the transcripts revealed, all those teams were still discussing specific details of software screens covering research questions—hence the greater incidence of science talk—while at the same time covering assignments for finalizing software. It should also be noted that not all students decided on a research question for their software design in the first week; some students changed their questions in the middle of the project and hence needed more time to complete their software designs.

Here we turn to a more problematic development, that of some teams’ sparse engagement with any science talk over the course of the three planning sessions. If we look at the percentages of systemic talk that each team engages in over the three sessions, we see some interesting groupings. Teams 1, 2, and 7 have very little systemic talk at all, whereas Teams 3, 4, and 5 engaged in more systemic talk overall and more consistently over the three sessions (see Table 3). Although Teams 1, 2, and 7 may have productively constructed plans for work distribution, programming procedures, and other issues in their sessions, they were not very successful at engaging with the science content in their software designs.

A comparison of two teams, one considered successful (Team 4) and another considered less successful (Team 2), will help to elucidate differences. Team 2 in particular had no systemic science talk in all three sessions and very little descriptive science talk in the first session. In reviewing the transcripts of the three sessions it becomes apparent that Team 2 was discussing matters related to the software design but framed their conversations in terms of software organization and navigation. A resource allocation principle suggested by one of the old-timers in the first session, “I think everybody should have equal time [on the computer] with their page,” becomes a dominating theme to be picked up in the following planning session again. There is nothing wrong with deciding on the equality of resource allocation for software designs. Yet such a planning neglects to take into account the decision that not all students need equal amounts of time for conducting and implementing their research. Some of them are old-timers, and others are newcomers and hence less familiar with the programming environment. We have found similar issues with the “equality” principle for work organization in other studies of children’s collaborative planning of complex problems (e.g., Marshall & Kafai, 1998).

It also appears from the transcripts that the members of Team 2 chose to discuss their software designs in terms of navigational structure and connections rather than in the discussion of individual software screens; for example, Anna claims,

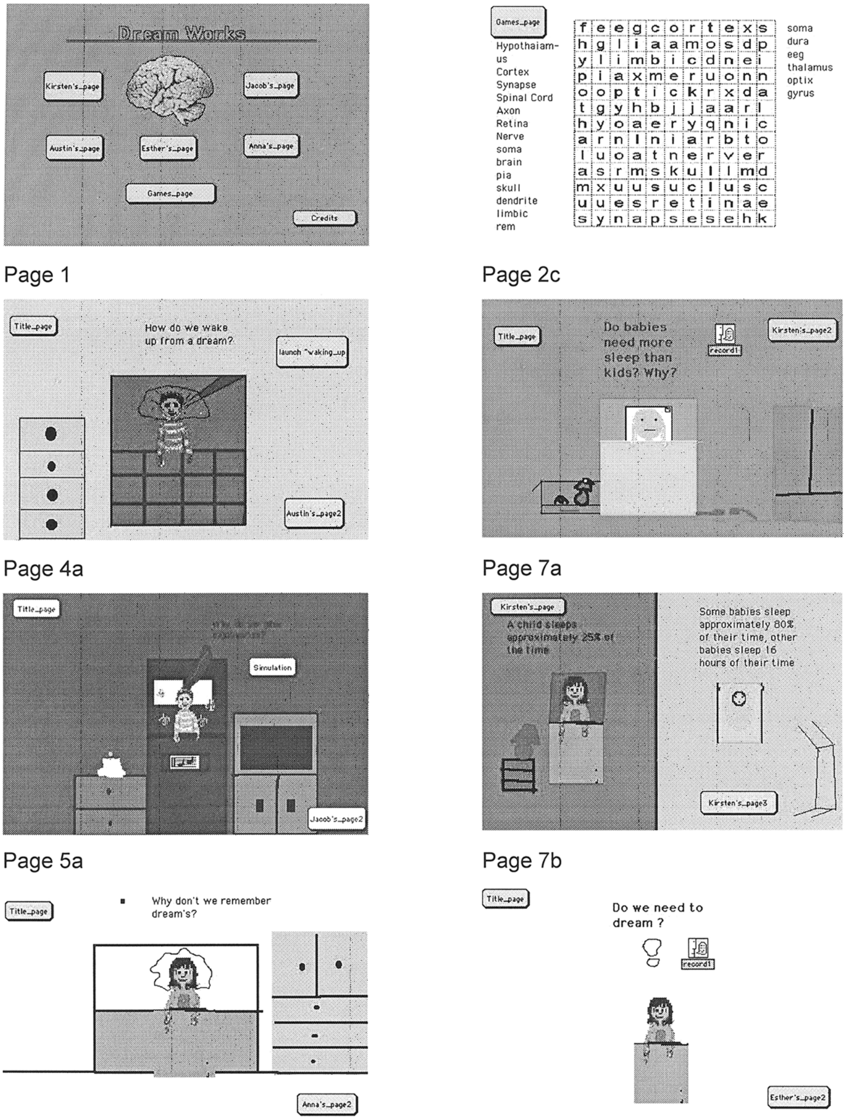
> I know, that's why we should all erase all of them by page one so you can go from page eight to one and then to nine. That way we don't waste all of our saving space on buttons that all it does to move us to another page that goes to a different page.

Although such a discussion is important for the design of the overall structure, it does not afford as great an opportunity to address science issues unless a navigational principle based on science concepts is used (e.g., using a representation of the different brain parts as the navigational introduction page to find out information about individual brain parts as found in the software of Team 4; see Method section).

In contrast, one of the most successful teams in terms of consistent science talk, Team 4, not only addressed individual screen designs in their conversations throughout the planning session but engaged in "progressively deepening" topic discussions, which led to more systemic conversations. Referring back to Figure 6, for example, we see that Calvin, a newcomer, introduces the idea of drawing a dendrite ("Are you good at drawing a dendrite?"), but then all team members engage in drawing and commenting on dendrites. Then the conversation evolves to describe what a dendrite is doing ("So this is the dendrite and it sends a message up to the brain. With chemicals."). This team's interactions overall were largely characterized by this type of discussion. Science concepts were not only introduced but also discussed in more detail, as the high number of systemic science talk indicates. This is not to say that this team did not address issues of work organization or software navigation; they did so. What differentiates them from Team 2 is their approach, which uses individual screens as the starting point rather than the general software structure or distribution of tasks.

This comparison points out some striking differences between the two teams in terms of engagement in science talk. There is a question of whether these differences had farther reaching impact, such as on the science content represented in the final software products created by the teams. We have already described in a previous section the software product of Team 4, which was considered one of the most successful teams in terms of science talk integration. In contrast, Team 2 was one the least successful. Their team consisted of five members Jacob (newcomer), Esther (old-timer), Anna (old-timer), Kirsten (newcomer), and Austin (newcomer).

Team 2's software project titled "Brainworks" (p. 1) focuses on dream explorations (see Figure 10). The title page contains a number of different buttons that lead to credits to a word recognition game and a vision experiment. The remainder



Page 1

Page 2c

Hypothaiam-
us
Cortex
Synapse
Spinal Cord
Axon
Retina
Nerve
soma
brain
pia
skull
dendrite
limbic
rem

f	e	e	g	c	o	r	t	e	x	s
h	g	l	i	a	a	m	o	s	d	p
y	l	i	m	b	i	c	d	n	e	i
p	i	a	x	m	e	r	u	o	n	n
o	o	p	t	i	c	k	r	x	d	a
t	g	y	h	b	j	j	a	a	r	l
h	y	o	a	e	r	y	q	n	i	c
a	r	n	l	n	i	a	r	b	t	o
l	u	o	a	t	n	e	r	v	e	r
a	s	r	m	s	k	u	l	l	m	d
m	x	u	u	s	u	c	l	u	s	c
u	u	e	s	r	e	t	i	n	a	e
s	y	n	a	p	s	e	s	e	h	k

soma
dura
eeg
thalamus
optix
gyrus

Page 4a

How do we wake
up from a dream?

Page 7a

Do babies
need more
sleep than
kids? Why?

Page 5a

Page 7b

A child sleeps
approximately 25% of
the time

Some babies sleep
approximately 80%
of their time, other
babies sleep 16
hours of their time

Page 6a

Why don't we remember
dream's?

Page 8

Do we need to
dream ?

FIGURE 10 Final software product of Team 2.

of the buttons link to the individual research questions and answers. Jacob's page provides an illustration and verbal description of what we dream (page 5a). Anna's page addresses why we can't remember dream and shows an explanation (page 6a). Austin's pages ask, "How do we wake up from dreams?" and show an animation of light entering the room (page 4a). Kirsten's project focused on "Do babies need more sleep than kids? And why?" and includes two follow-up pages in which she provides the amount of sleep needed for babies versus kids and a baby dream page explaining the need for making connections in the brain (pages 7a, b, and c). Finally, Esther's page questions "Do we need to dream?" and her answer showcases in triptych screen what happens when we don't dream and provides a verbal answer (page 8a).

Interesting to note about this team is that while many teams' table of contents screens direct users to individual question pages based on the science topics themselves (e.g., buttons for "how do we dream?" "how do glasses help you see?" etc.), Team 2 takes a different approach. Instead, their table of contents makes explicit that each section is owned and created by a particular team member (e.g., separate buttons for "Esther's page" and "Austin's page"). Only when the user gets to the individual pages does the user find out what the science topic is. This focus on individual ownership and who created what is emblematic of the topics other than science that dominated Team 2's planning sessions: work distribution, individual responsibility and initiative, and computer resource allocation to individual designers.

A comparative analysis of software pages based on a classification scheme we developed earlier (Kafai, Ching, & Marshall, 1998) reveals some interesting differences (see Table 6). We classified the software pages into five categories. Content pages represent some piece of knowledge about the field of science and could take the form of text, pasted pictures, drawings, or any combination of those three design elements. Animation pages contain animations that exemplified dynamic aspects of science concepts. Quiz/Feedback pages usually asked questions about the content displayed elsewhere in the product but occasionally introduced new material. Most quiz screens contained one or two multiple-choice questions with buttons linking the user to feedback on his or her response. Feedback screens con-

TABLE 6
Distribution of Software Page Types in Final Projects

	Team 2[a]	*Team 4*[b]
Content	0	36
Animation	35	31
Quiz/feedback	47	23
Information/navigation	6	10
Games	6	0

[a] $n = 17$. [b] $n = 22$.

sisted primarily of simple pages exclaiming “right!” or “wrong!” in a very large font. Information/Navigational screens provided information about the designers themselves or displayed the title of the software and subtitles of topic areas. Other screens contained buttons or turtles, which linked to different topic areas and provided information to the user on how to navigate the software, such as a table of contents. Finally, games pages contained activities such as a maze or a word recognition puzzle.

Although the number of developed pages is not substantially different between both teams, the most striking difference lies in the types of pages generated by team members. Both teams have about a third of their pages dedicated to animations and software designs, a small number of navigation pages. Team 4 has more multistep animations. The most substantial difference is the number of content pages created in Team 4 compared to Team 2. A further difference lies in the quiz and feedback pages, which constitute about half of the pages, created for Team 2 and significantly less for Team 4. This comparison seems to indicate that science content generation in software design also determines the amount of science talk generated in a team planning session. This is not to say that Team 2 did not contain any science content; it did. But it was situated within a different context that of explaining dreams. The team’s software actually reveals a strong thematic cohesiveness as all questions deal with dreams but from different perspectives. One can debate whether the topic of dreams as such does not afford as great an opportunity to engage with the science content or whether other factors can account for the lack of content focus.

DISCUSSION

The goal of our analyses was to show how science can naturally be integrated into design activities without dividing activities into “science” and “not science” parts (which may or may not compliment one another). We selected the software planning sessions for analysis to show that science talk in design activities occurs not only in designated “public” science discussions guided by teachers, but also in more “private” conversations among students when they are focused primarily on the designs. In our analysis of how students contextualized science within their planning sessions, we found evidence that science is not left at the door, and that there were numerous opportunities for making science part of the planning discourse. We found that a focus on software screens and the presence of more experienced students were particularly beneficial in helping students to “reformulate” their science designs. We also found that considerable differences existed among teams and that not all teams were as successful at integrating science within their planning discussions. Our analyses of productive and problematic contexts in which science talk occurs revealed a complex picture of interactions between software artifacts and student contributions.

In the following sections, we discuss in more detail the nature of science talk in student-directed discussions, the quality of particular context affordances, and the roles of scaffolds in design-based activities.

Nature of Science Talk in Student-Directed Discussions

One of the issues that is often mentioned in conjunction with design projects (and also project-based activities in general) is whether too many learning activities are vying for students' attention. What we found was that the design activity provided many different opportunities for students to bring science into the conversation.

Students had to think about science "in context"; that is, in the context of design. They were concerned not only with understanding science topics, but also with how to represent them. For example, in Team 4's discussion about a screen showing the synapse (Figure 6), they talk about what part should be shown (the dendrite) and what color should be used to represent the electrochemicals (yellow). Students also considered science in the contexts of how to create a game about neural pathways (Figure 2), how to make complex terminology palpable for third graders (Figure 7), and how to make parallels between physiological divisions of the brain and divisions of labor in terms of who would program each one (Figure 3). We also saw that students' science talk is informal and does not resemble "standard science" talk. The "electrical spit" conversation is one example of this phenomenon; in this same segment a student compares the appearance of neurons to broccoli (Figure 6). Although these informal discussions are somewhat humorous, we should not trivialize them. Student designers are doing essential work through this talk by putting what they are learning in everyday speech and connecting neuroscience to things they are familiar with. We have to be careful to not discourage students' discussions about "electrical spit" and the uncanny resemblance between neurons and broccoli because they don't sound like what we think science should sound like.

Our results also reveal the emergence of a type of scientific discourse found in other design-based classroom projects (e.g., Hmelo et al., 2000; Penner, Schauble, & Lehrer, 1996). The discussions around software screen designs by Teams 4 and 5 (see Figures 5 and 6) are characterized by talk that can be seen as having the vocabulary of science, as students utilize neuroscience terms and concepts, but the grammar of software designs. Within this "grammatical" structure, students talk about how biological processes will be represented on the screen via an animation and/or still graphics. It may be difficult for fourth and fifth graders to grapple with the how's and why's of scientific phenomena, but their "software design science talk," as exemplified by James' vision correction animation and Team 4's blind dreaming screens, reveals a sophisticated engagement with the complex field of neuroscience that belies their young age. What is apparent from these examples is that students are engaging in science in a very real sense within these conversations. Students used their software screen designs as an opportunity to talk in simplified ways about process in the same way the design of an elbow (Penner et al., 1998) or an artificial lung (Hmelo et al., 2000) facilitates students' understanding of the structure, behavior, and function of the science system under investigation.

Furthermore, we take the results as another example of how discussions directed by students differ from those directed by teachers. As Hogan et al. (2000) pointed out in their comparison of student- versus teacher-directed discussions, it was not the quality of arguments but the quantity of explorations that differed mostly between the two teams. Student-directed discussions showed a significantly higher number of idea generations. When students engage in science discussions, they focus and elaborate on concepts but they also build a connection to their common vocabulary, which is essential for their understanding. Lemke (1990) made similar observations. He also argued that such informal examples of science talk might indeed be better cases for students' science discourse than the "official" science discourse advocated in the National Science Standards (National Research Council, 1995) and numerous state science frameworks.

Affordances of Software Design Screens

Our analysis showed that focusing on planning for individual screens within the software, rather than more abstract issues of product organization or implementation concerns like work distribution, was the most productive context for yielding systemic science discourse. These results were supported by a comparative analysis of final software products. It seems that encouraging students to focus on particular screens not only allows them to establish the most specific level of planning for their designs, but it also affords more reformulation of the academic content at higher levels. This finding is complementary to results from other design-based learning approaches, which work more with an "engineering" model of design rather than the "architectural" model employed in this pedagogical approach. For example, Hmelo et al. (2000) asked students to design physical artifacts such as an artificial lung that provides direct feedback about its functionality as well as the opportunity to improve its design in iterative steps. Here particular behaviors of the design artifact itself (i.e., the lung) offer feedback to the designer. In the software design project, the specifics of science screen designs, but not the whole software itself, can invite students to discuss the appropriateness of represented processes.

But we do not think that individual screens are the only context in which systemic science talk can and will occur with greater frequency. Integration in terms of software organization is another potential candidate. In our students' software projects, we found several examples, such as a navigational structure created by Team 1, in which clicking on different lobes of the brain will take the user to a page about the functionality of the different lobes of the brain. Another example is thematic integration, as found in the software product of Team 2, in which all contributions focused on different perspectives of dreams. These software design frameworks are potentially productive contexts in which science software design talk can be situated. Furthermore, thematic integration can be both context specific, (i.e., tied to the task of designing a piece of instructional software) and content general (i.e., allowing for the integration of different science topics). These results confirmed research we conducted previously in the field of mathematics education (Kafai, Franke, Ching, & Shih, 1998). One conclusion to be drawn from these analyses is that instructional software designs with a focus on screen content or software organization can help students to situate their conversations about science in more systemic ways.

Supportive Role of Experienced Students

Further analysis indicated that experienced students facilitated more systemic science talk by initiating and expanding on team members' contributions. Eleven students (old-timers) had experience from participating in a previous design project (at that time they were the newcomers) on a different science topic—ocean environments. At several times in the current planning sessions, they made either direct or indirect reference to their previous design experience. We know from companion research conducted on software planning sessions by Marshall (2000) that these explicit and implicit references are unique to the old-timers. What we observe here is not only how students choose to reformulate science within their design projects but, equally important, what learning insights and practices they carried over into a new design experience. In those teams that managed to successfully integrate science into their planning sessions, the old-timers were able to use their previous knowledge of how content representation is valued in the software design project to help their team extend their science discussions. Previous research has also demonstrated the value of critical questioning by particular team members in making progress toward team problem solving (Fischer & Granott, 1995).

These results need to be taken with caution, however, as it is difficult to determine what factors impacted how old-timers did or did not extend their teams' science discourse. It is possible that old-timers were acting in their own best interests—wanting their teams to have impressive and scientifically accurate end products. More likely, however, is that some old-timers were assuming pedagogical roles in their teams and were thus explicitly teaching or helping newcomers. It could also well be that old-timer students had a better understanding of what appropriate science explanations are. A recent study that compared teaching interactions in a software design class composed of fourth-grade newcomers and fifth-grade old-timers with a class composed of fourth- and fifth-grade newcomers provides support for this argument (Ching, 2000). The comparative analysis showed that teams with old-timers provided better learning opportunities for the newcomers, thus excluding age difference between old-timers (fifth graders) and newcomers (fourth graders) as a potential factor. In any case, whether old-timers were acting pragmatically or pedagogically, their extensive questioning functioned as a way for their teams to bootstrap up to the next level of engagement.

These findings also contribute to current developments in training students in collaborative practices (see Cohen, 1994). Previous efforts have shown mixed results: although students were able to learn particular collaborative practices, they were less successful in employing them in their own team conversations (Bianchini, 1997). One interpretation of this result in light of our findings is that students in Bianchini's study simply did not have enough practice time; in other words, all team members were "newcomers." It is conceivable that old-timers' previous experience as newcomers in instructional software design provided them with a model of what it means to be an old-timer. In other words, they not only learned as newcomers the new collaborative practices in context, but they also saw the old-timers model for them conversational interactions in context. In addition, newcomers learn not just for the moment of the classroom activity; they know that in the following year it will be their turn to take on the role as old-timers in their teams. The combination of these factors—experience, modeling, and purpose—suggests some ways in which the learning of collaborative practices can be improved.

Problematic Developments

Within this context, we also need to address the problematic developments. The presence of old-timers is obviously no guarantee that systemic science talk will happen, but the affordances are there. The same can be said about contextual affordances for structuring science conversations. Team-specific choices emphasized different aspects of the design process (e.g., work distribution, navigation, science coverage) and for this reason team-specific progressions in science talk turned out to be more difficult to analyze. This is not only because of the placement of the sessions themselves at different time points in the project, but also because within each of those time points, the teams were at different places in terms of their engagement with the design process. Some teams were still working on getting their questions solidified as late as the last session, whereas others were focused exclusively on assigning time for debugging and polishing at that point. Even within teams, individuals were at different points at different times. There are two examples in the article (see Figures 8 and 9) of students who are still struggling with how to represent their knowledge in the systemic animation format in the last session, yet some others in their teams are completely finished and claim that they have nothing to do at other times in the transcript. So making any sweeping statements about trends or "progressions" turns out to be more complicated given the complex nature of the software design project work. Students and teams work at there own pace within a long-term deadline—this means not just programming work, but also the work of grappling with science.

Although these team-specific developments were not surprising given the findings of previous research, the differences between teams in science talk integration were more problematic. The near absence of systemic science talk in several sessions and in particular teams raises legitimate concerns in which ways these students were engaged in science inquiry. If we are to take these planning sessions as an indicator of how science can be integrated within design, then those teams were missing out, at least within the context of the planning session. Yet one should note that planning sessions took place within the context of a larger science inquiry environment. Students in our project engaged in various science-related activities such as dissections, three-dimensional modeling, field trips, and other investigations. Indeed, students had many whole-class and smaller team discussions that were devoted entirely to the treatment of a neuroscience concept. Under these circumstances, the absence of science talk in the planning sessions might be less of a problem. Of further importance should be that these teams did not engage in off-task activities; they were discussing plans for their software designs but not with a focus on science. We have discussed in the previous sections responsible aspects such as the screen focus and old-timer intervention. In future implementations of the instructional software design project, these aspects can provide helpful pointers for preparing and directing student discussions.

Implications for Design-based Activities

The discussion so far has focused on issues particular to instructional design projects, one model of project-based design activities for learning. The relevance of these findings to other project-based approaches (e.g., Baumgartner & Reiser, 1998; Brown & Campione, 1994; Linn & Hsi, 2000; Penner et al., 1998; Puntambakar et al., 1998; Roth, 1998; Scardamalia & Bereiter, 1994) is the focus of this section. Although each approach has developed a framework for introducing, engaging, and guiding students in design activities, the common theme is that students engage in creating artifacts, computer-based or not. Where projects differ most is in the degree to which constraints have been placed on students' design process, either in the choice of design problem or in the tools that they can use to accomplish the task. Projects have also implemented various constraints in the form of scaffolds that structure students' conversations and reflections on learning. Our software design project is more minimalist in respect to all those features: students can choose their own research question as long as it is within the confines of the science subject, and students decide what software screens they design and how to conduct their research. We structure team composition by distributing experienced students into all teams. Planning meetings, like the ones used in the analysis of science talk, are scheduled, but the format and content is left to students.

We also found that the presence of more experienced students can function as a successful scaffold. In most other projects the teacher or high-ability students often assume this role. There is an important and nontrivial difference between high-ability and design-experienced (i.e., old-timer) students. Although a "high-ability" designation is often confounded with status and other factors, design experience draws its position from having participated in a previous project, which applies to all old-timers independent of their science or programming background. It would be worthwhile to investigate how such "human" scaffolds can be build into other design activities. This would not only provide support in-group collaboration but also facilitate the introduction and continuation of new classroom practices as found in many design-based approaches. For example, "ramping-up" activities in which students practice smaller scaled versions of design problems or particular components could provide such experiences (see Hmelo et al., 2000).

Another concrete suggestion involves perhaps separating out the students' formal meetings and having some planning sessions devoted exclusively to productive contexts, like how the science topics will be covered and what screens will look like. Other concerns such as allocating computer time, team member loafing, and programming and navigation specifics could occupy a different meeting. Supportive artifacts for each distinct meeting, such as screen design sheets for science planning and calendars for the logistical planning meetings, would also be divided accordingly. But here we enter a catch-22 of sorts, in that while such compartmentalization would probably foster more science talk in the science-specific meetings, it begins to look less like the complex and multilayered process of design. Whether one wants to sacrifice authenticity for content coverage is a tradeoff each practitioner must decide for him or herself.

What our results have to offer to other projects speaks to students' competencies to operate and learn within a less constrained environment. Our take in presenting scaffolds is that rather than prescribing instructional dialogue for students, we see these design directives as conceptual design tools that can support and direct students' science negotiations. Our results indicate that minimal constraints, such as suggesting to conduct planning sessions with a focus on screen discussions or content-specific navigation designs, can leave room for students' individual research focus and interests, thus also rendering the projects more personally meaningful. This also has implications for what kind of science areas to choose for instructional design activities. Currently, we have implemented software design activities with other science topics such as ocean life (Ching, 2000; Kafai, 1998) and astronomy (Kafai, Ching, & Marshall, 1997) and found that all the topics lend themselves to be used as science contexts for learning design activities. We found that the combination of whole-classroom science activities with more individualized research questions allowed both for content coverage as well as activities that were intellectually and socially meaningful to individuals.

ACKNOWLEDGMENTS

An Early CAREER Award from the National Science Foundation to Yasmin B. Kafai (REC–9632695) supported this research.

We would like to thank the teacher, Cathleen Galas, and her science students at UCLA’s Seeds University Elementary School for their participation. Sue Marshall, a former collaborator at UCLA, has been instrumental to our understanding of the complexity of students’ planning in learning through design; she originally developed the planning session format. Kate Muir, Stephanie Lau, Audrey Farrales, Deborah Escalante, and Jennifer Park assisted in transcribing, timing, and coding the planning sessions. Janet Kolodner and three anonymous reviewers provided insightful comments on earlier drafts of this article.

This article is based on previous presentations of preliminary results at the Third International Conference for the Learning Sciences in 1998 in Atlanta, Georgia, and the annual meeting of the American Educational Research Association in 1999 in Montreal, Canada.

REFERENCES

Abrams, E. (1997). Talking and doing science: Important elements in a teaching-for-understanding approach. In J. J. Mintze, J. H. Wandersee, & J. D. Novak (Eds.), *Teaching science for understanding: A constructivist approach* (pp. 307–323). New York: Academic.

Anderson, C. W., Holland, J. D., & Palincsar, A. S. (1997). Canonical and sociocultural approaches to research and reform in science education: The story of Juan and his group. *The Elementary School Journal, 94*, 359–383.

Artzt, A. F., & Armour-Thomas, E. (1992). Development of cognitive-metacognitive framework for a protocol analysis of mathematical problem solving in small groups. *Cognition & Instruction, 9*, 137–175.

Baumgartner, E., & Reiser, B. (1998). *Strategies for supporting student inquiry in design tasks.* Paper presented at the annual meeting of the American Educational Research Association, San Diego, CA.

Benett, N., & Dunne, E. (1991). The nature and quality of talk in cooperative classroom groups. *Learning and Instruction, 1*, 103–118.

Bianchini, J. A. (1997). When knowledge construction, equity and context intersect: Student learning of science in small groups. *Journal of Research in Science Teaching, 34*, 1039–1065.

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

Brown, A., & Campione, J. (1994). Guided discovery in a community of learners. In K. McGilly (Ed.), *Classroom lessons: Integrating cognitive theory and classroom practice* (pp. 229–272). Cambridge, MA: MIT Press.

Carlsen, W. S. (1992). Closing down the conversation: Discouraging student talk on unfamiliar science content. *Journal of Classroom Interaction, 27*(2), 15–21.

Ching, C. C. (2000). *Apprenticeship, learning and technology: Children as oldtimers and newcomers in the culture of learning through design.* Unpublished doctoral dissertation, University of California, Los Angles.

Collins, A., Brown, J. S., & Newman, S. (1989). Cognitive apprenticeship: Teaching the crafts of reading, writing, and mathematics. In L. Resnick (Ed.), *Knowing, learning, and instruction: Essays in honor of Robert Glaser* (pp. 453–494). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Cuthbert, A., & Hoadley, C. (1998). *Designing desert houses in the Knowledge Integration Environment (KIE)*. Paper presented at the annual meeting of the American Educational Research Association, San Diego, CA.

Daiute, C., & Dalton, B. (1993). Collaborating between children learning to write: Can novices be masters? *Cognition & Instruction 10*, 281–333.

Fischer, K., & Granott, N. (1995). Beyond one-dimensional change: Parallel, concurrent, socially distributed processes in learning and development. *Human Development, 38*, 302–314.

Fusch, L. S., Fuschs, D., Kazdan, S., Karns, K., Calhoon, M. B., Hamlett, C. L., & Hewlett, S. (2000). Effects of workgroup structure and size on student productivity during collaborative work on complex tasks. *The Elementary School Journal, 100*(3), 184–212.

Galas, C. (1997–1998). From presentation to programming: Designing something different, not the same thing differently. *Learning and Leading With Technology, 25(*4), 18–21.

Gallas, K. (1995). *Talking their way into science: Hearing children's questions and theories responding with curricula.* Columbia University, NY: Teachers College Press.

Gordin, D., Polman, J., & Pea, R. (1994). The climate visualizer: Sense-making through scientific visualization. *Journal of Science Education and Technology, 3,* 203–226.

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

Harel, I., & Papert, S. (1990). Software design as a learning environment. *Interactive Learning Environments, 1*(1), 1–30.

Hmelo, C. E., Holton, D. L., & Kolodner, J. L. (2000). Designing to learn about complex systems. *The Journal of the Learning Sciences, 9*(3), 247–298.

Hogan, K. (1999). Thinking aloud together: A test of an intervention to foster students' collaborative scientific reasoning. *Journal of Research in Science Teaching, 36*, 1085–1109.

Hogan, K., Nastasi, B. K., & Pressley, M. (2000). Discourse patterns and collaborative scientific reasoning in peer and teacher-guided discussions. *Cognition & Instruction, 17,* 379–432.

Hunt, E., & Minstrell, J. (1994). A cognitive approach to the teaching of physics. In K. McGilly (Ed.), *Classroom lessons: Integrating cognitive theory and classroom practice* (pp. 51–74). Cambridge, MA: MIT Press.

Jackson, S., Krajcik, J., & Soloway, E. (2000). Model-It: A design retrospective. In M. Jacobson & R. Kozma (Eds.), *Advanced designs for the technologies of learning: Innovations in science and mathematics education*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Jones, M. G., Brader-Araje, L., Carboni, L. W., Carter, G., Rua, M., Banilower, E., & Hatch, H. (2000). Tool time: Gender and students' use of tools, control and authority. *Journal of Research in Science Teaching, 37,* 760–783.

Kafai, Y. B. (1995). *Minds in play: Computer game design as context for children's learning*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Kafai, Y. B. (1998). Children as software users, designers, and evaluators. In A. Druin (Ed.), *The design of children's interactive technologies* (pp. 123–145). San Francisco: Kaufman.

Kafai, Y. B., Ching, C. C., & Marshall, S. (1997). Children as designers of educational multimedia software. *Computers and Education*, *29*, 117–126.

Kafai, Y. B., Franke, M., Ching, C. C., & Shih, J. C. (1998). Games as an interactive learning environment for fostering students' and teachers' mathematical inquiry. *International Journal of Computers for Mathematics Education, 3,* 149–184.

Kolodner, J. L., Crismond, D., Holbrook, J., & Puntambekar, S. (1998). Learning by design from theory to practice. In A. S. Bruckman, M. Guzdial, J. L. Kolodner, & A. Ram (Eds.), *Proceedings of the Third International Conference on the Learning Sciences* (pp. 16–22). Charlottsville, VA: AACE.

Kolodner, J. L., Hmelo, C., & Narayanan, N. (1996). Problem-based learning meets case-based reasoning. In D. Edelson & E. Domoshek (Eds.), *Proceedings of the Second International Conference on the Learning Sciences* (pp. 188–195). Charlottesville, VA: AACE.

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

Linn, M., & Hsi, S. (2000). *Computers, teachers, peers: Science learning partners*. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Lumpe, A. T., & Staver, J. R. (1992). Peer collaboration and concept development: Learning about photosynthesis. *Journal of Research in Science Teaching*, *32*(1), 71–98.

Marshall, S. (2000). *Planning in context*. Unpublished doctoral dissertation, University of California, Los Angeles.

Marshall, S., & Kafai, Y. B. (1998). Children's development of planning tools for managing complex software design projects. In A. S. Bruckman, M. Guzdial, J. L. Kolodner, & A. Ram (Eds.), *Proceedings of the Third International Conference on the Learning Sciences* (pp. 202–208). Charlottsville, VA: AACE.

National Research Council. (1995). *National Science Education Standards*. Washington, DC: National Academy Press.

Penner, D. E., Lehrer, R., & Schauble, L. (1998). From physical models to biomechanics: A design-based modeling approach. *The Journal of the Learning Sciences, 7*, 429–502.

Puntambekar, S., Nagel, K., et al. (1997). Intra-group and intergroup: An exploration of learning with complementary collaboration tools. In R. Hall, N. Miyake, & N. Enyedy (Eds.), *Proceedings of CSCL'97* (207–214). Toronto: AACE.

Richmond, G., & Striley, J. (1996). Making meaning in classrooms: Social processes in small group discourse and scientific reasoning knowledge building. *Journal of Research in Science Teaching, 33*, 839–858.

Roth, W. M. (1998). *Designing communities*. Nordrecht, The Netherlands: Kluwer.

Roth, W. M., & McGinn, M. (1998). Inscriptions: Toward a theory of representing as a social practice. *Review of Educational Research*, *68*, 35–59.

Scardemalia, M., Bereiter, C., & Lamon, M. (1994). The CSILE project: Trying to bring the classroom into World 3. In K. McGilly (Ed.), *Classroom lessons: Integrating cognitive theory and classroom practice* (pp. 201–228). Cambridge, MA: MIT Press.

Songer, N. (1996). Exploring learning opportunities in coordinated network-enhanced classrooms. *Journal of the Learning Sciences*, *5*, 297–327.

Webb, N. (1983). Predicting learning from student interaction: Defining the interaction variables. *Educational Psychologist*, *18*, 33–41.