Kolodner, J. L., Crismond, D., Fasse, B. B., Gray, J. T., Holbrook, J., Ryan, M., & Puntambekar, S. (2003). Problem-based learning meets case-based reasoning in the middle-school science classroom: Putting a Learning by Design curriculum into practice. *Journal of the Learning Sciences, 12*(4), 495–547.
Kolodner, J. L., Crismond, D., Gray, J., Holbrook, J., & Puntambekar, S. (1998). Learning by Design from theory to practice. In *Proceedings of ICLS 98* (pp. 16–22). Charlottesville, VA: AACE.
Kolodner, J. L., Gray, J. T., & Fasse, B. B. (2003). Promoting transfer through case-based reasoning: Rituals and practices in Learning by Design™ classrooms. *Cognitive Science Quarterly, 3*(2), 119–170.
Kolodner, J. L., & Nagel, K. (1999). The design discussion area: A collaborative learning tool in support of learning from problem-solving and design activities. In *Proceedings of CSCL '99. Palo Alto, CA* (pp. 300–307). Mahwah, NJ: Lawrence Erlbaum Associates.
Nagel, K., & Kolodner, J. L. (1999). SMILE: Supportive Multi-user Interactive Learning Environment [Poster summary]. Available http://www.cc.gatech.edu/projects/lbd/htmlpubs/SMILE. html
Palincsar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension monitoring activities. *Cognition and Instruction, 1*, 117–175.
Puntambekar, S., & Kolodner, J. (1998). The Design Diary: A tool to support students in learning science by design. In *Proceedings of the International Conference of the Learning Sciences (ICLS 98)* (pp. 35–41). Charlottesville, VA: AACE.
Puntambekar, S. & Kolodner, J. L. (2003). *Distributed scaffolding: Helping students learning science from design.* Manuscript submitted for publication.
Scardamalia, M., Bereiter, C., & Steinbach, R. (1984). Teachability of reflective processes in written composition. *Cognitive Science, 8*, 173–190.
Schank, R. C. (1982). *Dynamic memory.* New York: Cambridge University Press.
Schank, R. C. (1999). *Dynamic memory revisited.* New York: Cambridge University Press.
SRI International, Center for Technology in Learning. (1999). *Performance Assessment Links in Science website* [On-line]. Available: http://www.ctl.sri.com/pals

*Chapter* 5

# Children as Instructional Designers: Apprenticeship and Evaluation in the Learning Science by Design Project

Yasmin B. Kafai
*UCLA Graduate School of Education & Information Studies*

Cynthia Carter Ching
*University of Illinois, Urbana–Champaign*

Many would argue that the purpose of instructional design is to develop a body of knowledge that prescribes instructional actions to optimize instructional outcomes for the learner. Few would consider an approach in which learners themselves are engaged as instructional designers for the purpose of developing their content knowledge and skills. This unusual approach, however, is exactly what we advocate and describe in this chapter. Our model, called *learning science by design*, is situated within a curricular tradition of project-based learning approaches. Common features of these curricular approaches are that they provide students with "long-term, problem-focused, integrative and meaningful units of instructions" (Blumenfeld et al., 1991, p. 370). Design projects, such as our Learning Science by Design project, emphasize the construction of meaningful and complex artifacts (physical or virtual), which serve as driving vehicles for students' and teachers' classroom activities. Design-project activities have been developed and examined for a variety of subject matters, such as mathematics (e.g., Harel, 1991; Kafai, 1995), sciences (Brown, 1992; Kafai, Ching, & Marshall, 1998; Penner, Lehrer, & Schauble, 1998), engineering (Hmelo, Holton, & Kolodner, 2000; Roth, 1998), and social studies (Carver, Lehrer, Connell, & Erickson, 1992). In this chapter we examine various aspects of students' participation within the Learning Science by Design project, including students' apprenticeship interactions with one another and their development of evaluation criteria for themselves and others.

The context for our study of students as instructional designers is as follows. A class of 31 students composed of seven teams of fourth and fifth graders participated with their teacher in a science project for 3 months. They designed and implemented instructional software about neuroscience to teach younger students in their school. In creating the teams and examining their interactions, we used a software design model that emulated professional perspectives of software users, designers, and consultants (Kafai & Harel, 1991). We distinguished between *users* (third-grade students who used and evaluated the software), *newcomers* (fourth-grade students who designed software for the third graders), and *oldtimers* (fifth-grade students who had previously been newcomers and then apprenticed newcomers by coparticipating with them in software design practices). This model spans three classroom grade levels and introduces all participants into forms of legitimate software design practice, albeit with different levels of access. Other studies have attempted to replicate some aspects of this development by having the same students either participate in repeated design projects (e.g., Brown & Campione, 1994; Erickson & Lehrer, 1998) or in different design activities (e.g., Barron et al., 1998).

Apprenticeship interactions within teams were one focus of our investigation, because apprenticeship has been supported as one of the best ways for learners to become expert in a given domain. Proponents of this approach to learning argue that part of the reason for educational failure stems from school's lack of resemblance to other apprenticeship-style environments where learning takes place naturally (Lave & Wenger, 1991). There has been no small difficulty, however, in applying an apprenticeship model to formal schooling. Earlier attempts focused on apprenticing students into the cognitive practices of thinking in academic domains (e.g., Collins, Brown, & Newman, 1989; Scardamalia, Bereiter, & Lamon, 1994); however, they neglected to address the physical environment, activities, and tool use that make up the culture of learning in a particular subject (cf. Roth, 1995). Furthermore, these cognitive approaches focused largely on a dialogic relationship between learners and the teacher; other students were not included in the equation. The goal of this study was to address both of these issues within the context of learning science through design (Kafai, 1996). What distinguishes our approach from other design studies is the presence of both experienced and inexperienced designers in the same classroom, who coparticipate in creating science software together.

Unlike formal schooling, in documented studies of apprenticeships explicit instruction almost never happens (Rogoff, 1990). Rather than engaging in "how to," oldtimers and newcomers (as termed by Lave & Wenger, 1991) jointly participate in a common task. The way in which labor is divided in an apprenticeship may vary based on the participants' skill levels (Hutchins, 1993); however, they work together toward the same goal. Studies examining the role of relevant artifacts in learning environments with or without an apprenticeship structure have found that tools and artifacts can constrain the tasks at hand by providing structure for newcomers (Rogoff, 1993), or they can create additional possibilities for activity depending on the users' design flexibility (Roth, 1996). The computer in particular, as an artifact being jointly used by students with varying skill levels, can enable learning and mutual respect on the one hand (Harel, 1991) and can alternately be used by more capable students to restrict others' participation. Based on the literature just cited, we wanted to investigate the nature of the apprenticeship relationship among more and less experienced student designers, how this relationship might change as the project progressed, the role of computer and nonelectronic design artifacts in that apprenticeship relationship, and how roles of particular artifacts might change over time.



We wanted to complement these investigations of apprenticeship interactions with an analysis of students' perceptions of their design skills and their evaluations of the finished products. A critical aspect in design projects is a continuous evaluation of the progressively evolving artifact. This evaluation demands that the student designers have developed some criteria to judge the success. Erickson and Lehrer (1998) called these criteria "critical evaluation standards," whereas Roth (1998) spoke about "final product knowledge." In the context of human–computer interaction work, interface designers call these principles user-centered design (e.g., Norman & Draper, 1986). Educational software as such represents a particular challenge because it requires the designer to take into account the knowledge background, interests, and motivations of learners who are often distinct from professional users (Soloway, Guzdial, & Hay, 1994). In terms of evaluation, we were thus interested not only in students' perceived software design competencies, but also in the kinds of software evaluation standards they employed. In the Learning Science by Design project, students were introduced to software design practices, which involved mastering a programming tool. For that reason, it is important to understand in which ways the young designers perceived themselves as competent tool users and what their programming competencies were. Furthermore, we chose students' own evaluation standards because these have been recognized as one essential component in design projects (Erickson & Lehrer, 1998).

Taken together, these avenues of research—apprenticeship processes and students' evaluations of their skills and final software designs—represent crucial phases in the process of instructional design for learning. In this chapter we examine how students enter a community of instructional design as both learners and designers as well as how they collaborate with others in that community, and how they reflect on and evaluate their design skills and final instructional design products.

## THE LEARNING SCIENCE BY DESIGN PROJECT

One classroom with 31 fourth- and fifth-grade students (14 boys and 17 girls) and their classroom teacher from a school in metropolitan Los Angeles participated in the study. Eleven of the 31 students had participated in a similar research project the previous year. The group of experienced students, or oldtimers, consisted of 10 fifth-grade girls and 1 fifth-grade boy. These experienced students were distributed throughout the seven teams of designers; groups were also matched for grade level and gender. The classroom consisted of different areas: a "rug" area in front of a whiteboard and seven table clusters, each of which had a 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, the other served as a scanner station. All the computers contained Microworlds™ Logo software, which students used to program their science software.

The science intervention and related design-project activities lasted 10 weeks (Galas, 1997–1998). In general, the class spent 75 minutes, 4 days per week, on science- and software-design-related activities. The introduction to the science unit started with an all-class discussion in which students generated all the questions they had about the science topic of neuroscience. 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?" "How does memorizing things and memory work?" A central information source for the neuroscience simulation project was a website called Neuroscience for Kids,[1] which contains a variety of resources such as background information in text and graphics, online memory and vision experiments, and game activities. In addition, students visited the neuroscience laboratory on the university campus, conducted a dissection of a sheep eye, and a brain surgeon (the father of one of the students) came to talk about his work and brought a human brain and a cow brain for comparison purposes. Parallel to the generation of research questions, all teams participated in a 3-hour introduction to basic graphic and software design functionalities of their programming environment. During this introduction, the teacher asked the oldtimers to help newcomers learn the basic programming functions while creating their first animation.

Several kinds of evaluation activities took place throughout the project. Twice, at Weeks 5 and 8, a group of 14 third graders came to visit and use the developing instructional simulations. The student instructional designers conducted these "usability sessions" themselves. Before the usability visit, design teams met and discussed the questions they wanted to ask the

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



third graders in regard to their liking and understanding of the software. Informal evaluations and demonstrations also occurred throughout the project, as students visited different teams' workstations to view others' computer screens. At the end of the project, the whole class met to discuss criteria for final software evaluations and generated a list of evaluation questions. The next day, each team visited three other software simulations and tested the software. (Examples of students' instructional screen designs can be found in Fig. 5.1.) Each team received a summary report of other teams' evaluations that listed the ratings received and comments provided by students. Students' participation in *all* of these activities provides 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 are scheduled based on a perceived need by the teacher or the researchers to address student questions or emerging problems.

### Research Documentation

We used a variety of methods to document student learning, classroom activities, and group work. Student teams were videotaped as they met during class time to plan, research, and implement their software. Each team was captured for at least 30 minutes on tape approximately once a week. We videotaped the class session at the end of the project in which students generated criteria for the final evaluation activity, and we collected all the final evaluations and student comments. Finally, we also interviewed all the students at the end of the project. The interview questions varied depending on whether students were newcomers or oldtimers to the project. The questions asked students about their design-project experiences by examining their collaborative and individual contributions, the development and application of project-management strategies, their generation and implementation of research questions, and their expectations for prospective design projects. All interviews were videotaped and then transcribed in preparation for coding.

We decided to use a case-study approach for the examination of instructional design apprenticeship. The case-study team was the most normative: It contained no uniquely skilled (or unskilled) students, did not have any unusually difficult problems getting along, and was not characterized by frequent or extended absences on the part of one or more members. Each of the four team members seemed typical of the different kinds of students in the classroom as a whole. Caren,[2] a female oldtimer, was a good planner and organizer as well as a competent, but not superb, programmer. Naisha,

[2]All student names have been changed.

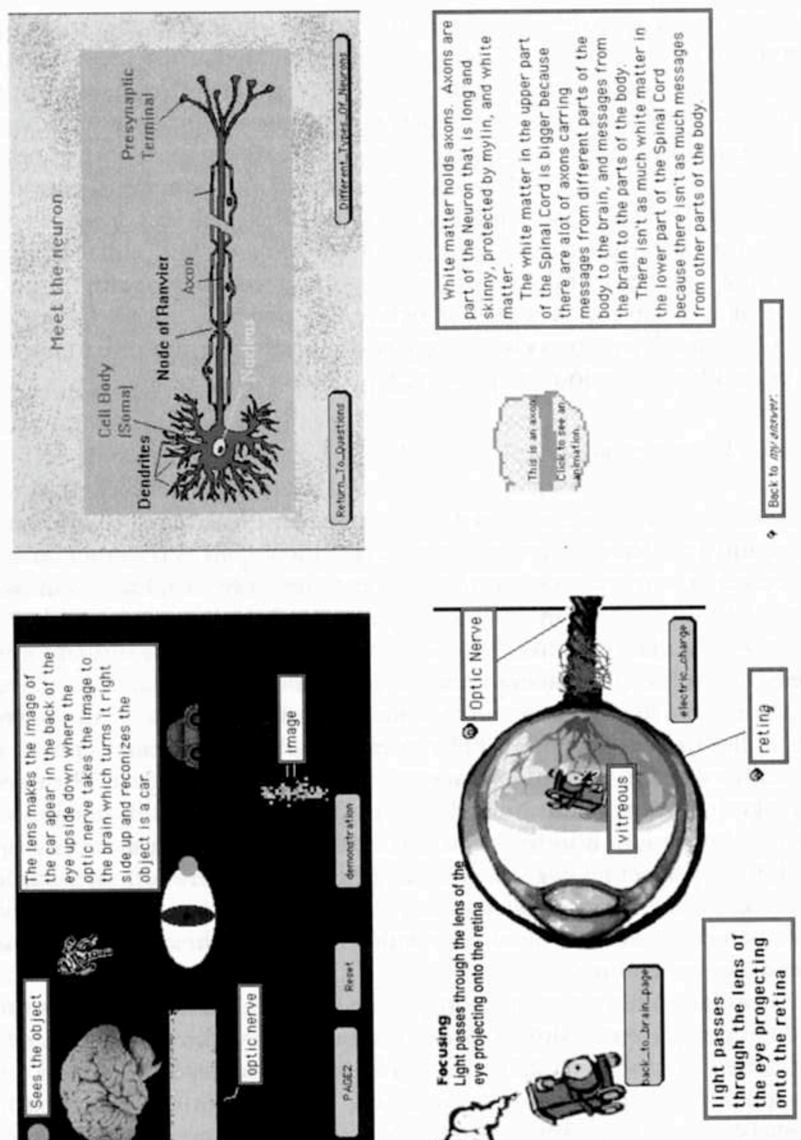


FIG. 5.1. Examples of instructional software screens.



also a female oldtimer, was less interested in organization and still needed to master some aspects of programming. Brian, a male newcomer, had computer experience and skills surpassing that of Naisha and Caren, but he had never used Logo before. Finally, Jiao, a male newcomer, was also a competent computer user but was somewhat shy and appeared to be confused by the open-endedness of the design task. The fact that both of the case-team oldtimers were girls was not atypical; most of the students who repeated the design experience in fifth grade were female.

## Results

In the following sections, we provide an overview of different complementary aspects of the Learning Science by Design project: apprenticeships among experienced and inexperienced instructional designers/learners, and evaluative criteria derived and employed by students in reflecting on their design skills and final products.

***Apprenticeship Into an Instructional Design Community.*** One of the key questions in our research deals with how students learn to become instructional designers. We were interested in not only how students learn to represent knowledge for the benefit of younger learners (as in traditional instructional design), but also how they master the use of various tools and artifacts specific to the learning-through-design process. In this project, these tools were comprised mainly of the Logo software, which students used to create simulations, and "planning boards": freestanding posters that served as storyboards for the emerging software designs, central message centers, and project calendars for each design team. In our study, students' mastery of these tools took place largely in the realm of apprenticeship interactions between newcomers and oldtimers. We were interested in how both the apprenticeship interactions and tool use would change over time as newcomers became more expert instructional designers. Overall, we found that apprenticeship relationships, working patterns, and tools that appeared to serve particular functions at the project outset were changed to suit new goals and circumstances as the project progressed and the developing software took shape.

***Apprenticeship Structures.*** At the beginning of the 10 weeks of designing, all groups were informed that they would have to divide their group in half and pick working partners, since on certain days half of the group members would be doing instructional science activities with the teacher in another part of the classroom. The resulting partnerships in our case-study group were (1) Caren and Brian and (2) Naisha and Jiao. These pairings were particularly interesting in that they allowed the maximum distance in skill and assertiveness between the oldtimers and newcomers involved. For example, had Naisha been paired with Brian, she would not have been much more skilled than he was and therefore not able to help him a great deal with programming. The maximum-distance pairings of oldtimers and newcomers that occurred in the case-study group seemed to facilitate good working relationships, group harmony, and productivity during the first half of the project. A few other groups who did not use maximum-distance pairings had trouble getting along, and the newcomers learned slowly due to floundering when working without the presence of an oldtimer.

***Role of Artifacts.*** During the first 4 weeks of working on the software, the case-study group members were largely occupied with planning and researching their product; not a great deal was actually made yet. Consequently, the foam storyboard, or planning board, and its components played the important role of documenting what was going to be implemented. All group members contributed to the board, although Caren was in charge of determining how it should be organized. Initially, the relation between the computer and the planning board was directly related to the apprenticeship structure. For example, when Naisha and Jiao were paired together for computer work, Naisha would do most of the programming while Jiao checked the board to make sure they were working according to plan, as in the following segment:

Naisha: Okay, so I'm doin' the background. What color?
Jiao: I don't know. (*gets up to look at the board*) Ben has it brown on here.
Naisha: It's brown on his sheet?
Jiao: Yeah, I'm looking at it right now.
Naisha: Okay.

This division of responsibility in the group served to enforce the notion with newcomers of the need to stay focused and follow the group's plans. In a few other groups, when newcomers were left to their own devices, they played with Logo, making nonscience animations and ignoring the planning boards, behavior that was detrimental both to their own learning and to group productivity.

***Change Over Time.*** As the project progressed and the newcomers became more skilled in Logo, the case-study group found that the maximum-distance pairings were no longer necessary. Working pairs became much more flexible, and who was working on the computer together was determined more by common interests (such as Naisha and Ben's screen about



brain waves during dreaming) than by a need for assistance. The role of various design artifacts changed as well. Though the planning board was crucial in the beginning, its use tapered off during the second half of the project. As the emerging software took shape and the group accumulated a number of half-finished screens in their folder on the hard drive, the developing product itself seemed to dictate what needed to be completed, making the planning boards obsolete. Likewise, Caren's role as the group organizer also waned when her control of the planning board was no longer as relevant. In the end, group members behaved much more like equals than oldtimers and newcomers—which was appropriate since the fourth graders were no longer inexperienced in software design.

### *Instructional Designers' Self-Evaluations*

Designing software, and for that matter any kind of artifact, is a complex enterprise. We examined how students' perspectives on their programming expertise expressed themselves and what kind of criteria or evaluation standards students had developed. One central feature of the Learning Science by Design project is that students learn to program in Logo not just for the sake of learning how to program but also for the purposes of knowledge reformulation and personal expression (Harel & Papert, 1990). The goal is to introduce students to a tool that facilitates their creation of dynamic, interactive representations in science.

All students participate in the basic programming introduction, with oldtimers taking the lead in their teams in the beginning. Over the course of 10 weeks, newcomers get introduced to more sophisticated aspects of software design by either their team members or other students. Some newcomers very quickly become the programming experts in their teams. Yet when we look at interview results, 50% of all newcomers report mastery of the software design environment as one of the hardest parts in the project, whereas only 10% of the oldtimers listed this as a challenge. All oldtimers listed this as one aspect that was different from their first design-project experience. Later, when asked whether they considered themselves "junior software designers," 72% of oldtimers agreed compared to only 55% of newcomers.

The students' explanatory statements shed some light on the different ways in which oldtimers and newcomers saw themselves as junior software designers. Oldtimers saw their programming expertise relative to where they started from and to others, *and* to what they still had to learn:

Heidi: Well I'm not like the best programmer software maker and I'm not the worst programmer software maker, so I'm sort of in the middle.

Jade: Probably last year I don't think I would have thought myself but this year I think that I am.

Elisa: I know that I can do much more and I know that there are people that know much more than me, but kind of. And if I think of the next project, I will do much better.

ewcomers situated their understanding of programming only in regard to here they were in the beginning:

John: I know more than the first time.

Katia: Because I know what Microworlds is, how to program it, how to make animations, how to make buttons, how to do all the stuff.

Sven: I think of myself as more experienced than when we started.

All students became more competent in programming, which was also ndicated in their use of programming terminology. Newcomers and oldimers alike used terms such as "pages," "buttons," "back buttons" (when reerring to buttons linking pages back to the table of contents or previous ages), and "textboxes" and actions such as "making things move," "finding here the turtle is," or "putting them into order" (when referring to the seuencing of procedures) to describe their science software.

### *Instructional Designers' Product Evaluations*

During the design project, the software development was evaluated not nly by the third graders but also by their peers in informal reviews on a lay-to-day basis. Furthermore, in official classroom demos, designers preented their science simulation designs to the whole class. During the reiew sessions, students would not only comment on screen designs but also n information content. In the context of the debriefing interview, we osed several questions that asked students to evaluate and review their wn educational software.

The results are not surprising given the different ways oldtimers and ewcomers judged features of their software. Students used several evaluaion criteria to judge the software content and organization:

- *Liking/entertainment value:* Students referred to page organization and content presentation as "I thought it was cute," "I liked my animation," "This was interesting," or "This was fun."



- *Information value:* Students focused on the information content in relation to science or media forms. For example, students commented: "And we just had this dinky animation and it wasn't very good, and it had a lot of words. Like words pretty much the whole page"; "We had too much text, we didn't give enough information"; "The picture doesn't explain what you are supposed to do . . ."
- *Navigational structure and performance:* Students evaluated the operation of various software features, such as: "It was kind of jerky . . . like I say it didn't work"; "Maybe we could make another little table of contents and that little table of contents would lead to games about the senses"; or "I would change the buttons cause they're pretty plain."
- *Audience consideration:* Students included in their considerations of navigation and content their expected users by stating: "And if you are saying a hard word say it slowly"; "I would probably put it more not so you can push buttons, not so make it is easy to learn. Some things you have to make hard to learn"; or "We just showed what the brain did and it wasn't a simulation so the kids didn't learn anything . . ."
- *Production value:* Students evaluated the relationship between quantity and quality of information production by stating: "It is better to have more pages that are pretty good because than you have a variety of things to do," or "They should make the project smaller but the pages better."

In comparing newcomers and oldtimers, we found several differences in how they used criteria such as information and navigational organization for evaluating their science simulation (see Table 5.1). More newcomers used entertainment values in their evaluations. Both groups considered information and navigation features in their evaluations. The most striking difference was in the near absence of audience considerations of software features among the newcomers. Only two newcomers (10%) made any reference to their potential users, whereas five oldtimers (45%) included this factor in their considerations. In a later question in the interview, we asked the designers how they would change their software when designing it for

TABLE 5.1
Software Evaluation Standards by Newcomers and Oldtimers

| *Category* | *Oldtimers* | *Newcomers* |
|---|---|---|
| Entertainment/interest | 9% | 33% |
| Information | 64% | 50% |
| Navigational performance | 18% | 33% |
| Audience consideration | 45% | 10% |
| Production value | 27% | 10% |

older students. Here we found no differences between newcomers and oldtimers in considering different aspects such as different wording or more text, based on the assumption that their older audiences know how to read. This result may point out that newcomers are able to include the audience in their software design considerations when specifically asked to do so. But it appears not to be a systematic factor in considering information and navigational organization of their software.

These results reflect those found by Erickson and Lehrer (1998) in their comparisons of students' changes in hypermedia development over 2 years, with the exception of audience consideration and production value. In the Learning Science by Design project, students designed software for actual users who visited them twice, and one half of the newcomers were previous users. Consequently, it is not surprising that this evaluation standard was included in their considerations for information presentation and software organization.

These analyses present a complex picture of apprenticeships within a school classroom. From the observational analyses, we learned that collaborative interactions between oldtimers and newcomers shifted over the course of the project. The interview analyses indicated that oldtimer students evaluated their instructional software tool use and instructional design products according to different criteria. In the following section, we discuss what these findings tell us about the feasibility and benefits of curriculum designed around students as instructional software designers.

## Discussion

Students' apprenticeship interactions and differences between oldtimers and newcomers were focal points of our analysis of children as instructional designers. Our particular approach within the Learning Science by Design project provided an apprenticeship model within a school context. The accounts of successful apprenticeships in traditional societies describe the social networks involved as complex, changing structures in which participants move fluidly from one stage to the next (Lave & Wenger, 1991). This investigation found that student apprenticeships are dynamic systems as well. Oldtimers and newcomers inside the classroom, within the context of learning through design, do not stay constant in their respective roles or their relationship with various tools and artifacts in the environment. Existing studies have documented the fact that in design projects, the student designs eventually become artifacts in the environment (Roth, 1996). Here, however, the emerging design was not only added to the artifact arsenal, but it actually affected the utility value of other artifacts. As the gradual shift in the importance of the storyboard took place, a shift in the group control



structure also occurred. As the collaborative product emerged, student roles shifted to be more equal.

The interviews conducted with students at the project's end required students to do "talking about design" rather than "talking within design," as they had practiced with their team members during the 10 weeks of the project. Schön (1986) pointed out one of the inherent problems in design: the difficulties of getting designers to articulate their thinking and doing—not just a step-by-step description of design activities but an articulation of principles that carry over from project to project. All the students were articulate about the many aspects of software evaluation and design. Students referenced their statements with multiple examples describing the design and operations of particular software screens or interactions with other students. Their statements were grounded in experience of the particular design-project case or cases (for the oldtimers). The issue at hand is to what extent the students' own descriptions are representative of actual mindful practice within a project. What we heard students describe resonates with descriptions and analyses of mindful practice in action such as provided by Roth (1998) and Hall and Stevens (1995). Furthermore, within the particular case of project management, we had actually documented and analyzed students' performances during the project (Marshall, 2000).

The comparison between newcomers and oldtimers gave us a first view of what it might mean to become an instructional software designer for learning. The observed differences between newcomers and oldtimers should not come as a surprise as we would expect oldtimers with time and experience to become more sophisticated. The differences between newcomers and oldtimers were not of a conceptual nature (as it is often pointed out in the research literature focusing on expert–novice differences). The difference was that the repertoire of oldtimers was more expanded. The ability to consider multiple standards for software evaluation and to understand one own's programming competence were features of the oldtimers' expanded repertoire. We should note that there was also a substantial degree of shared understanding between newcomers and oldtimers of which standards to apply for evaluating software products.

To conclude, we see a particular benefit in our version of the apprenticeship model in that it sustains learning over several classroom years and thus provides the essential experiential component. It sustains a classroom environment not only with teachers' careful and knowledgeable guidance and facilitation but also with students who know and understand classroom activities and apprentice others into it. This model begins with the third-grade students who get introduced to software design practice through the perspective of being a user. It continues with the newcomer students who move from users to codesigners under the guidance of the oldtimers. The oldtimers are the students who apprentice the newcomers to the various aspects of instructional software design practice. This apprenticeship model not only gradually moves responsibility from teacher to student (as reciprocal teaching does), it also moves responsibility from student to student over years. Both teachers and students have to be cognizant of learning practices to make a classroom intervention work and sustain over time.

Future directions for this work include an increasing focus on the teacher's role and on individual student differences. The role the teacher plays in such collaborative environments—how he or she interacts with teams and individual students, in which ways he or she provides assistance—is definitely a complex one and deserves further investigation. It is also clear that our description and investigation of newcomer and oldtimer students provides us with only a beginning understanding of the cognitive and social differences between them and of how these differences impact team interactions (Ching, 2000). Existing collaborative research provides us with little understanding of the multiple factors that come into play in such a long-term learning environment. Our analyses presented in this chapter are a first step in coming to understand the complexities of implementing such collaborative instructional design models in school classrooms.

## ACKNOWLEDGMENTS

An Early CAREER Award from the National Science Foundation to the first author (REC-9632695) supported this research. The ideas expressed in this chapter do not necessarily reflect the positions of the funding agency. This chapter is based on previous presentations by both authors at the annual meetings of the American Educational Research Association in 1998, 1999, and 2000. We thank the teacher, Cathleen Galas, and her science students at UCLA's Seeds University Elementary School for their participation. Sue Marshall and numerous UCLA undergraduates assisted in collecting, transcribing, coding, and analyzing the data.

## REFERENCES

Barron, B. J. S., Schwartz, D. L., Vye, N. J., Moore, A., Petrosino, A., Zech, L., Bransford, J. D., & the Cognition and Technology Group at Vanderbilt. (1998). Doing with understanding: Lessons from research on problem- and project-based learning. *Journal of the Learning Sciences, 7*(3/4), 271–311.

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

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.

Carver, S., Lehrer, R., Connell, T., & Erickson, J. (1992). Learning by hypermedia design: Issues of assessments and implementation. *Educational Psychologist, 27*, 385–404.

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 Angeles.

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.

Erickson, J., & Lehrer, R. (1998). The evolution of critical standards as students design hypermedia documents. *Journal of the Learning Sciences, 7*(3 & 4), 351–386.

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.

Hall, R., & Stevens, R. (1995). Making space: A comparison of mathematical work in school and professional design practices. In S. L. Star (Ed.), *The cultures of computing* (pp. 118–145). London: Blackwell.

Harel, I. (1991). *Children designers.* Norwood, NJ: Ablex.

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. *Journal of the Learning Sciences, 9*(3), 247–298.

Hutchins, E. (1993). Learning to navigate. In S. Chaiklin & J. Lave (Eds.), *Understanding practice: Perspectives on activity and context* (pp. 35–63). New York: Cambridge University Press.

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

Kafai, Y. B. (1996). *Learning science by design: Developing information-rich learning environments in science for young software designers* (National Science Foundation No. REC-9632695). Los Angeles: University of California, Los Angeles.

Kafai, Y. B., Ching, C. C., & Marshall, S. (1998). Children as designers of educational multimedia software. *Computers and Education, 29*, 117–126.

Kafai, Y. B., & Harel, I. (1991). Children's learning through consulting: When mathematical ideas, programming knowledge, instructional design, and playful discourse are intertwined. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 85–110). Norwood, NJ: Ablex.

Lave, J., & Wenger, E. (1991). *Situated learning: Legitimate peripheral participation.* New York: Cambridge University Press.

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

Norman, D. A., & Draper, S. W. (Eds.). (1986). *User-centered system design.* Hillsdale, NJ: Lawrence Erlbaum Associates.

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

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

Rogoff, B. (1993). Observing sociocultural activity on three planes: Participatory appropriation, guided participation, and apprenticeship. In J. Wertsch, P. del Rio, & A. Alvarez (Eds.), *Sociocultural studies of mind* (pp. 139–164). Boston: Cambridge University Press.

Roth, W. M. (1995). Inventors, copycats, and everyone else: The emergence of shared resources and practices as defining aspects of classroom communities. *Science Education, 79,* 475–502.
Roth, W. M. (1996). Art and artifact of children's designing: A situated cognition perspective. *Journal of the Learning Sciences, 5,* 129–166.
Roth, W. M. (1998). *Designing communities.* Dordrecht, Netherlands: Kluwer.
Scardamalia, 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.
Schön, D. (1986). *The reflective practitioner: How professionals think in action.* New York: Basic Books.
Soloway, E., Guzdial, M., & Hay, K. (1994). Learner-centered design. *Interactions, 1*(2), 36–48.

Part II

# CURRICULUM DEVELOPMENT, INSTRUCTIONAL DESIGN, AND INFORMATION TECHNOLOGY

Norbert M. Seel
*Albert-Ludwigs-University of Freiburg, Germany*

The contents of the chapters of Part II are concerned with the relationships between curriculum development and instructional design (ID). Accordingly, processes of curriculum development and their impact on ID are discussed. Before the chapters are briefly outlined, a framework to integrate the contents is proposed.

## WHAT IS THE ESSENCE OF CURRICULUM DEVELOPMENT?

To answer this question it is necessary to define the term *curriculum.* A useful definition is given by Glatthorn (1987): "The curriculum is the plans made for guiding learning in schools, usually represented in retrievable documents of several levels of generality, and the implementation of those plans in the classroom; those experiences take place in a learning environment that also influences what is learned" (p. 6). Some years earlier, Beauchamp (1968) defined a curriculum as a document describing contents of subject matter domains, such as history, science, math; aims; and learning situations. This definition is still common, especially in the various German approaches of didactics. Accordingly, several