Kafai, Y. B., & Harel, I. (1991b). Learning through design and teaching: Exploring social and collaborative aspects of Constructionism. In I. Harel & S. Papert (Eds.), *Constructionism* (pp. 111–140). Norwood, NJ: Ablex.

Learning through Design amd Teaching:

Exploring Social and Collaborative Aspects of Constructionism

Yasmin Kafai and Idit Harel

1. Introduction

Students in our Instructional Software Design Projects (ISDP, e.g., see Harel & Papert, this volume) learn about fractions by designing a piece of software that will teach fractions to younger students. They are involved in issues of representation, communication, instructional design, programming, and teaching of fractions. Later, when these same students become consultants to other younger software designers, they refine their knowledge of fractions, programming, design, and teaching. They consider their own reasoning once again, and, in this playful mode, they actively construct a broader and deeper understanding of fractions—while strengthening other concepts and skills.

Our aim is to explore various aspects of the design/teaching/consulting method of learning, while emphasizing the **social nature** of the ISDP environment. By exploring social and collaborative aspects in ISDP, we attempt to define in more detail the interactionist and affective characteristics of constructionist learning environments (e.g., Harel & Papert, 1990) and to relate our models to the ongoing research on collaborative learning and social construction of knowledge.

The thesis that cognitive change is as much a social as an individual process prompted the educational and psychological research communities to focus on the role of social contexts in learning and development (Newman, Griffith, & Cole, 1990). Researchers are now looking at individual and social worlds as intertwined, and while they continue to consider ways to capture an individual's constructivist development (e.g., in the spirit of Piaget), they simultaneously pay careful attention to social processes within which the individual participates (e.g., in the spirit of Vygotsky). The contribution of social interaction to the individual's learning has been conceptualized in several ways: as 'cognitive-apprenticeship learning' (Collins, Brown, & Newman, 1990; Palinscar & Brown, 1984), or as 'peer-collaboration learning' (e.g.,Daiute & Dalton, 1989; Johnson, 1986; Slavin, 1983)—to name but a few. One general assumption of these approaches is that working and learning in close interactions with adults, peers, or people in general, is a richer experience than learning by oneself. Another assumption is that during social interactions, participants communicate their ideas and make their thoughts explicit—to examine their own as well as someone else's views and problems and react accordingly. Finally, complex shared activity with another is often viewed as a context for building arguments substantially different from those one would build without group interaction. We agree with the above assumptions, but intend to add a twist. The purpose of this chapter is to reveal our 'twist,' express some of its aspects explicitly, and relate it to the approaches mentioned above.

During the 1989-1990 school year, three different teachers and their classes participated in our ISDP reimplementation studies—our second look at ISDP (which also occured in Project Headlight at the Hennigan School, an inner-city elementary school in the Boston area). The ***learning by teaching*** aspect of the original ISDP provided the students with a new kind of audience—the younger students, instead of the audience they usually have—their teacher. This aspect contributed greatly to the original ISDP students' understanding and learning (e.g., Harel, 1988). We chose to expand it in the following way: fifth graders designed software for fourth graders, established relationships with them, and then became consultants for the fourth graders, as they designed software for third graders.

Figure 1: Research procedure in ISDP-II

In the following sections we will explore various aspects of ISDP's social nature. We will describe our conceptualization of collaboration in ISDP and, for demonstration purposes, focus on three case studies of young software designers as they navigate through a particularly sensitive phase—what we will call the 'Incubation Phase.' Our other paper which follows (see Kafai & Harel, this volume) deals with the extension of the learning-by-teaching principle and explore how young software designers make sense of their "consulting activity." In the discussion, we summarize the findings from our case studies and define the particular social features of the ISDP constructionist learning culture and how they contribute to the individual's learning.

2. Re-defining Collaboration within ISDP

The ISDP learning environment leads to better understanding of fractions and programming through enhancing opportunities for social contacts and mutual exchange. Multiple and flexible interactions are a primary feature of ISDP. Students in ISDP interact with each other in requesting help and in giving advice, both as teachers to younger students, and as software designers and content presenters. In educational research, an interactional style of learning is usually described as “collaboration” or “cooperation.” These words are being used to describe two or more students collaborating, cooperating, expressing themselves, and working together—towards **one** product. In other words, the collaborators have one common goal—finishing their project; and another common goal—working on it together, and finding ways of incorporating each collaborator's ideas in a process of mutual work and negotiation towards one product. Many studies show the advantage of working collaboratively in this style (e.g., Slavin, 1983; Johnson, Johnson & Joulbec, 1986; Daiute & Dalton, 1989); we refer to it here as the conventional conceptualization of collaboration.

The collaborative style of ISDP shares the spirit of this conventional conceptualization of collaboration but implements it in a different way and with some additional features. For example, ISDP includes a collaborative process among two or more students where each of the students is working towards his/her **own** project. In ISDP, all the students have a common "umbrella" goal—to use LogoWriter to design a piece of software that teaches about fractions—but each of them also expresses his/her own ideas, and produces his/her own projects. Everyone faces similar "problems," and finds occasions for sharing ideas, asking for help, or discussing technical problems. They may choose to discuss problems with a partner, or even to jointly design and implement certain aspects of their software.

In summary, we found ourselves conceptualizing collaboration in a different way within the framework of ISDP. The main features of the ISDP collaborative style are: 1) **Optional Collaboration:** the students can work alone on their own piece of software, but they can also work with others on the same piece if they wish; and 2) **Flexible Partnerships:** students can decide with whom they want to work, when, for what purposes, etc. In other words, in the ISDP framework, students can move between both collaborative styles according to their own needs and desires.

The spatial arrangement of the working areas at Headlight allows students to be flexible in their seating arrangements. This facilitates their movement, communication, and ongoing interactions. As a result, a particular kind of interactional process caught our attention. We would like to characterize it as **"Collaboration-Through-the-Air."** This refers to a process described by two sociologists, Berger and Luckmann (1966), in their book *The Social Construction of Reality*. They state that constructing reality involves not only "face-to-face" interaction, but also interaction with the world of the "knowledge surrounding us." This last aspect is very similar to the learning situation in ISDP.

In our projects, the students are sitting at their computers, in two large circles. They are "surrounded by" other students' computer screens, ideas, knowledge, Logo code, and fractions designs. A number of different ideas are floating around in this "software design studio," waiting to be "picked up"—as Harel (1988) described in Debbie's Case in her dissertation on the original ISD Project, and as Gerald's Case (see below) illustrates in our analysis of ISDP-II. These ideas are programmed or represented in various ways on different computer screens. Most children honestly believe that they originated the ideas on their own, although in most cases the ideas probably originated with one person and then spread throughout the group. The daily interaction among students—the ability to walk around and see other students' projects, to try them out, and so on—facilitated this collaboration through the air.

Berger and Luckmann (1966) also stress the idea that there are "different timings" for the appearance of knowledge within specific individuals. Transferred to ISDP, this means that the concept of equality in fractions (when realized in the context of fractions of areas) can be very important to one person at a certain time, but not important to others at that same time. Only if and when the other students are "ready" for it, will they appropriate it. We believe that the basis for appropriation has to do not only with social aspects of learning, but also with a child's readiness. The following fieldnote by Kafai explains what we mean by “readiness” for knowledge appropriation.

> On Day 13, we introduced the QUIZ procedure to the students, which allows the student to ask a question on the screen, to read in the user's answer, and to give a reply corresponding to that answer. This procedure introduced many new and difficult programming concepts to the students, such as conditionals, variables, and branching. Many students adopted the procedure for their program immediately; they copied it and then tried it out.
> In the following days I observed that some students integrated this procedure in most of their screens, while other students deleted it completely—as did one student, Alicia.
>
> After some days we had a Focus Session in the classroom, where I asked the students if they had problems. Alicia told me she really wanted to know about READLIST (which is a component of QUIZ) because she didn't understand it. But she also told me that she didn't want to know it right then. Rather, she would tell me when.
>
> After three weeks, Alicia suddenly told me: “Now I am ready to learn about READLIST. Can you come and explain it to me?” I went to see her and discovered she had prepared everything in her program. She had put all the questions already in the Logo code, but did not include the part where you read in the answer and give the reply. After I explained this to her, and we worked together on integrating it into one of her procedures, she then proceeded to integrate it into the rest of program—on her own.

This simple example tells us that appropriating an idea is easy when you are ready for it. The same is true for understanding new ideas or concepts. Harel (1988) gave several examples of this readiness in Debbie's Case in relation to her learning new Logo techniques (e.g., Random, GetPage) and new ideas about fractions (e.g., “All these shapes show one half,” or “Fractions are everywhere!”).

To summarize, we see “collaboration through the air” as working in two ways: it provides individuals (or society) with many examples of what one person can do and how they can do it; and it allows space and time for individuals (or society) to pick up (or re-discover) such ideas when they are ready.

2.1. Case Studies

As the project begins, the only assignment given to the students is to "design a piece of software to teach younger students about fractions." No further well-defined specifications are given about the task. It is up to the participants to make their own decisions as they slowly slip into the role of educational software designers. In this beginning phase of the project—which we call the 'Incubation Phase'—the new designers are at once confronted with many hard questions: What should be the main theme of their project? What are fractions? How do you represent fractions? How do you teach fractions? Or on a more practical level: What should be programmed on the computer screen? Which colors should be used? What text might be included? Where should it be placed on the screen? and so forth. At first, the selection of themes and representations, as well as the design problems involved in handling this project, seem to be endless and quite overwhelming. Our experience shows that one of the challenges for the young software designers is to decide where and how to begin.[1]

---

[1] In the literature on problem solving and programming, the 'Incubation Phase' is usually called planning. The importance of this planning phase, for experts as well as novices, has been stressed repeatedly (e.g. Soloway, 1988). In the planning phase, the problem-solver is engaged in building up the "problem space" while decomposing the problem into smaller, more manageable parts, as well as laying out initial "sub-plans" for further steps to follow. This approach seems to be particularly appropriate when dealing with well-defined problems of which the end state is known.

In addition, many studies show that planning is not common in novice behavior (e.g. Dalbey, Tourniaire, & Linn, 1986) and more prototypical of experts (e.g. Kurland et al., 1984). Observations of expert programmers, for example, reveal that a major portion of their time is devoted to planning. In these studies, however, novices are rarely given the chance to work on problems large enough to engage them in substantial planning activities. We argue that the provision of time, complexity, and the open-ended structure of the problem in ISDP give the student designers a chance to explore further activities and therefore to engage in collaborative planning activities of various kinds.

In an ill-defined design-based problem-solving process, the student designers have to define the problem before they can identify potential obstacles and desired end-states. In short, the particularities of

The following case studies exemplify what we call "flexible and optional partnerships" and "collaboration-through-the-air." We focus here on the 'Incubation Phase'—the first 3 weeks of ISDP-II in the fifth grade. These cases demonstrate that some students actually look for and strongly need interactions with peers, while others do not. Some children, who work collaboratively at first in a 'face-to-face' style eventually do make a decision to continue their work independently. Other children prefer to work individually throughout the project. However, it would be wrong to perceive their processes as individualistic *per se*, because of collaboration-through-the-air.

We found that the initial phase has a rather "messy" character: students are trying out different designs, some get started, then stop, only to start all over again; others seem to do nothing at all for 5-10 days. From the outside, it often looks like a period of non-productive activity. There appears to be no visible progress. However, because this is the phase when learners "mess around" (Duckworth, 1987) with their ideas, intuitions, and "dirty thoughts" about fractions and software design, we consider this initial starting phase to be particularly important for learning. Eventually, this messing about—both cognitively and socially—might become finalized into a good design or an idea that is followed through for a whole week, an entire month, or until the end of the project. In fact, ISDP is advancing in individual cycles and rhythms. Individual and collaborative planning and evaluation of achievements occur quite often during the project: as students complete a representation, an entire instructional screen, a procedure, a super procedure, a Logo Page, a topic, and so on.

Following is a description of the Incubation Phase as seen through the work of several software designers. In the case of Gerald, for example, we shall see that he is not ready

---

ISDP are never defined entirely in advance. As the project moves along and becames more complex, further issues and problems are identified by the participants (students, teachers, researchers) with great sensitivity to time and context involving social interactions and collaborative efforts.

for collaboration at the beginning of the project, and needs some time on his own to identify the goals he wishes to pursue. On the other hand, Stacey and Amy worked together at the beginning, but later decided to work alone. Jeannine's case is an example of a girl who preferred to work on her own most of the time. Still, there were students who met for short periods of time on a day-to-day basis to discuss each other's projects, and then returned to their own projects.

2.1.1. Case Study 1: Gerald

Consider Gerald, a 10-year-old boy. He is one of only four boys in the class. Most of the other students do not like to work with him. In the pre-interviews they described him as “bossy,” and “not doing his work,”—remarks which were similar to what his teacher said about him. Gerald, however, does not see himself this way. He thinks that when he “is right,” things should be always done his way. (“YK” stands for the researcher):

> YK: And can you tell me on what project you were working with Eugene?
> G: I was working with Eugene on the Science Project.
> YK: What were you doing?
> G: ...making the street lights. I was working with him on making them go on and off.
> YK: Did you have discussions, arguments?
> G: ...A little.
> YK: What did you argue about?
> G: Well, if you want to make a stop you had to put the tape on it and I was telling him ‘We have to put on the tape' and he was telling me ‘Don't put the tape there.' But then I got it to be my way and I put it there, and so it worked.

Several days after the interview, when we introduced the project, the idea of a Nintendo-styled fraction-game was in everybody's mind. Gerald's first screen design in his Designer's Notebook showed a man holding something in his hand facing two towers (in "Mario Brothers" terminology these would be the Warp Zones).

Figure 2: Gerald's first Designer's Notebook entry of a Nintendo-like design

Immediately, all four boys went to the computers to start working on their Nintendo-like projects, and all four sat next to each other. They started by taking animation programs (from their other class) and transferring them into their fraction project. This idea remained the dominant one during the first few days until Matt decided to start working on his own project and Eugene began designing geometrical shapes in the Logo Shapes Page representing 1/2, 1/3, 1/4, etc. Eugene's program was designed so he could assign the Logo Turtle a fraction shape (by using SETSH and a number) and have it move across the screen. In this way, the user could control the movement of several fractions on the computer screen, resulting in an animated sequence of fractions moving on the screen. Antonio immediately used Eugene's idea for his project, and they both started working on the idea together.

For a while, at least, the idea of a very sophisticated Nintendo-like game had died. Gerald remained the only one pursuing this idea. For the next three days we saw complicated mazes and various race tracks in his Designer's Notebook. His reflection entries were always the same:"*I had no problems. I didn't make any changes. I don't know.*"

Gerard's insistence on working on this particular piece of software isolated him from the other three boys. He kept trying day after day to change one of the already existing programs called ‘car racing.' At one point, as Kafai tried to help him with something, she accidently deleted his whole procedure and he stopped working on his project completely. He wrote in his notebook: "*I lost everything.*"

For the next 3 days whenever Kafai asked Gerald what he wanted to do, he would say: “I am not working on the project.” After 10 days, Kafai found a design in his notebook with wild scribbles on it: "*I am not doing this project anymore*."

On Day 11 we had a 'Show and Tell' session in the class. Mira, Stacey, and Robin volunteered to present their projects to the other students. Not all of the children came to look, but Gerald did. Later that same day, he started programming some fraction representations on his computer screen for the first time. He began by working on displaying a circle divided into fourths. When Kafai asked him why he gave up on his car program, Gerald answered: “It's too difficult for now.”

After the 'Show and Tell' session, we had a meeting with all the children in the class to discuss project ideas and any problems they were having. This was the first time a competitive atmosphere clearly emerged. Most of the students expressed their feelings about how they “did not want to have other students look over their shoulders” and “take away their ideas.” Through this discussion they reached an agreement that “Logo ideas could be shared, but not fraction representations ideas.” Strangely enough, Gerald was one of the students who strongly advocated this agreement![2]

Nevertheless, it seemed to Kafai that Gerald grasped something during the 'Show and Tell' session which allowed him to re-direct his approach to his own project. Did seeing the other students' projects stimulate his thoughts about what to do? Every day, he would start by working on a new screen, a new representation of a fraction. He even invited Kafai to see what he was working on—something he had refused to do before. One day, we had a guest student from another class. Gerald eagerlytook him to his computer for his own ‘Show and Tell’ session during which he explained to the guest what he was doing.

---

[2] This session is further discussed in the following chapter, Episode 8 (see Kafai & Harel, this volume).

Figure 3: An example of Gerald's screen on fractions

A week later, Matt and Gerald decided to put both their programs together. Matt had designed a maze with fractions displayed all over it. A Pacman-like shape was running across the screen, in a maze, and passing over fractions (see Figure 4). Interestingly, he had incorporated the concept of equal fractions in the design of the maze: opposing each other were equal fractions such as 3/6 to 4/8; 2/4 to 5/10, and 1/2 to 6/12.

Figure 4: Matt's PacMan fractions maze

Matt's initial ideas displayed in his Designer Notebook and software were the "Mario Brothers" warp zones, a little man standing on them with fractions. When Kafai asked him what his game was all about, Matt presented an elaborated perspective of what his software project would be once it was finished:

> "I will make a man on the Shapes page. The man is going to have a fraction on his shirt saying 1/2 or 2/4. And then he goes to the dimension where there is a lot of mumbles [he pointed on the screen to the warp zones]. And they are going through the pipes...and when they come out of the pipes, they go over there. And the person who has 1/2, he has to jump up and down. They will come up real fast and on another warp, when he got one false, the guy grows as tall as this, he grows and he runs. That's what happens. Creatures come out, birds come down dropping things on him, when you make it to the King Tut, he throws fractions at you. You have to dodge the fractions so you don't die..If you get this, you go to another dimension."

For several days, Matt elaborated his software, called "videogame," trying out different placements of the figure shapes and warp zones. Much in contrast to the detailed designs, he wrote in his notes most of the time:"*I have no plans and ideas for tomorrow. None*."

Gerald, on the other hand, had so far designed five screens displaying different fractions. For three days, they both sat together and worked on copying all the procedures and shapes from both of their programs into one. At one point, Matt indicated to Gerald the incorrect phrasing of one of his fraction questions:

> M: You have to write 'What fraction is <u>this</u>?' and not 'What fraction is it?'
>
> G: Why 'this'?
>
> M: Because that's what you are asking your users, about this one (points to the representation Gerald created on the computer screen).

At the end of this collaborative period between Matt and Gerald, Matt had a fractions project combining both his and Gerald's program segments. Gerald also incorporated Matt's segments. However, after a week, Gerald decided to continue working only on the fractions representations he had created, and to leave Matt's part out. From then on, both students continued working individually on their fractions project, although every once in a while they did interact on various matters.

2.1.2. Case Study 2: Amy and Stacey

Amy and Stacey, decided to work together at the beginning of the project. Amy is a short, rather shy, girl. Stacey, in contrast, is much taller and has a more outgoing personality. She smiles quite often and enthusiastically jumps up and down when she gets excited. She is always ready to laugh and joke around with a visitor or a friend. Frequently she would hug her teacher or her classmates.

The entries in their Designer's Notebooks reflect this difference: Stacey wrote on the first day in a determined voice: "*I am going to give the students 20 questions, and if they get them all right they will win a TV. and if they get half of them right they will win a radio.*" Her notebook's screen design (see Figure 5) included many ideas at once: examples for different fractions, how to represent fractions with either shaded or circled parts, a question she remembered from the Fractions Interview about how to cut a cake.

The next day she had a new idea and wrote, "*I will make a maze and the pacman will get you. If you do not get to home first you must get all of the fractions right in 20 minutes and I will put 20 fractions for you to do.*"

Amy, in contrast, was less sure of her plans and she wrote: " *I am going to just make some fractions* ." Her screen design in the notebook included two fractions, a circle cut into quarters and a rectangle cut into eighths.

Figure 5: Stacey's first design of quarters and eighths in her Designer's Notebook

Once they decided to work together, the two girls had a discussion with Kafai about how they could do that. She said that it was "perfectly o.k. to share ideas, give help to each other, and to work on the same program if they wanted to." Amy then asked if they could work together on one computer. Again, the researcher said this was fine if that was what they wanted.

Figure 6: Amy's and Stacy's fractions screen representing a 1/2, and the pit

Their writings in their Designers Notebooks were nearly the same for the first 2 weeks. When Stacey wrote, "*Amy and I had problems today we couldn't figure out what we were trying to do*" or on the following day, "My problem was that my procedure didn't work and my partner and I were getting very angry. I added a whole procedure." Amy put it this way "*Stacey and I couldn't get it to work when the kids type the answer. I didn't make any changes.*" They both decided to continue working and Amy added in her notebook:"*[we need] to work on the game and fix our problems.*"

For 2 weeks, we could also see Amy and Stacey finishing their designs and writings at the same time, walking together to the computer area, and always choosing to sit next to each other. However, at one point, they decided to continue working on the same thing, but implementing it using two adjoining computers, one for each girl. While Amy and Stacey sat next to each other, they checked each other's screens continuously and waited for each other before continuing work. The following discussion at the computer shows Stacey and Amy discussing their screen design.

> Stacey: "We should it put up there [she gestures with her hand over Amy's screen; see figure 6]." A few minutes later, Stacey asks: "Ok, what do we do?" Amy answers: "Now how we are going to type this in [she talks loud while she is typing in code in the command center] if ...[unclear]?" Stacey: "What are you doing?" and she discusses different examples of how to provide feedback to the user. A few minutes later, in the same session, both talk about screen design again. Stacey [while she is talking she is pointing with her hand to the corresponding parts on the screen]: "Maybe we could do this stuff here. Make some things smaller, have a lot of them here."

We observed that Stacey, the more outgoing of the two girls, had the more active part in the team, as far as deciding what to do next, and how. One day, we introduced the “QUIZ” procedure to the students. Most of the students tried to integrate this procedure into their programs. Amy tried, and when she didn't know how to continue, she asked the researcher to help her. In the meantime, Stacey sat and waited. Afterwards, she asked Amy to leave the Flip Side on the screen (the Logo Editor) so that she could correct the procedure she had copied onto her own Logo Page. Because Amy wanted to work on her Shapes Page, she leaned over to Stacey's computer and said: “Let me do this for you fast.” During this time, it was very important to the girls that they proceed at the same pace, and create identical programs but on two different computers.

Suddenly, on Day 10 of the project, Stacey began designing a new representation: it had a rectangle divided into unequal pieces, some of which were shaded in. She also printed a question at the top of the computer screen, above the representation: “Are these fractions equal? Type yes or no” see figure 7.

Figure 7: Stacey's screen representing and quizzingabout fractions equality

Kafai asked Stacey why she thought the equality of fractions was an important thing to teach. She answered that one must know how to cut areas into equal pieces. Later, Kafai asked why she was “designing this screen alone, and why Amy doesn't have it too.” Stacey answered: “Because it was my idea.” The girls did not fight, and it seemed that their agreement to take a break from working together was mutual. In their Notebooks, both write on the same day *"I will continue to work on my game. It is getting easier for me. I changed something in my procedure."* (Stacey) and *" I had no problems today. I didn't make any changes. I will continue to work on my game. It is getting easier"* (Amy)

After that, Amy and Stacey did not work together again in this fashion. In fact, both started new partnerships with others: Stacey working most of the time with Karen, and Amy worked with Sara and Jeannine.

2.2.3. Case Study 3: Jeannine

Jeannine's teacher, Mrs. Mar, regards Jeannine as a very successful student and thinks very highly of her school work and her accomplishments with LogoWriter. In an interview we conducted at the end of ISDP-II, Mrs. Mar talked about Jeannine's software design project:

> Jeannine always puts a lot of time in anything she does. Her software program includes [an animation of] the bus picking up the people showing them [fractions]. That something, a little extra, I am not sure if anyone else has that. That's nice. Jeannine always goes above and beyond for anything she does!

We will follow Jeannine through her Incubaction Phase. For one-third of the project, Jeannine explored, on her own, different ideas of what to do in her project. She first designed a candy bar, then a baseball game, until she settled down on her final idea. Even though Jeannine worked mostly on her own, she rarely sat alone. Every day she changed partners. One could frequently see her looking at other students' projects, trying them out, or making a comment. On several occasions Jeannine was hanging around and playing with her classmates.

On her first day of the project, Jeannine wrote in her Designer's Notebook: *"I want to make a candy bar and make it into half and as I go on I want to make it harder."* She then drew the picture of a Hershey Bar in her Designer's Notebook and wrote the symbol 1/2 over it. In LogoWriter that day, she programmed a picture of a Hershey Bar. At the end of the session, she wrote again: *"I had absolutely no problem. And it's fun! I made no changes but I might change my mind on what I am going to do on the game."*

Figure 8: Jeannine's design of a Hershey chocolate bar

For some reason, Jeannine chose not to save this program. The next day she started with another idea and erased the whole Hershey bar procedure she programmed. She wrote in her Designer's Notebook: *"I am going to try to think of a new project, and while I am doing that I am going to finish what I did. My problem I had was to think of what I am going to do but I found out. I erased my Hershey Bar because I thought it was stupid. So I made a baseball game"* (see figure 9).

Figure 9: Jeannine's computer screen of the Baseball diamond

Figure 10 is the instructional screen she designed for her students:

Figure 10: Jeannine's instructions for how to use her software

For the next 3 days, Jeannine elaborated upon the idea of the baseball diamond, adding other features such as bases. She included a procedure which takes the name of the user as an input and says "Hello [name]." At the computer, where other students tried out her software, she kept saying: "It is not finished yet!" and then explained to Leslie that she was trying to fix something. However, she was still not sure whether or this was what she wanted to do. In her Designer's Notebook entry from the third day of the project, she wrote: *"I am going to finish what I had on Friday and if I get frustrated I am going to erase it."* After this session she decided to stick further with this plan: *"I had no problems and I didn't need to erase my procedure but I almost erased it."* She also corrected the spelling mistakes in the instructions screen, and changed the shape of the turtle from a diamond into a bus stop sign. Furthermore, she added a little procedure:

```
to r
```

```
game
end
```

On the next day, there were no changes in her code. During the day, some students, among them Jeannine, Stacey and Amy asked Kafai how to quiz people ("How to ask a question and to give an answer back.") Kafai started explaining it to them but had the vague feeling they did not understand. However, a day later, Jeannine changed her instructional screen, indicating that she had a new design idea for her project. Even though the programming part of quizzing seemed unclear, Jeannine grasped the idea of what it would do and integrated it accordingly into her project (see figure 11).

Figure 11: Jeannine's new instructions

After we introduced the students to the idea of a quizzing procedure that, in fact, was requested by the students, Jeannine changed her strategy once again. On Day 6 of the project, she started working on the representation of a square cut into five pieces, two of which were colored in different shades of blue. In her quiz, she asked the student to choose an answer from the given fractions: 1/2, 2/3 or 2/5.

Figure 12: Jeannine's representation of 2/5

At this point, the diamond procedure had been disconnected from the program, and on the following days, Jeannine continued to work on different fraction representations without the diamond idea. She also added new fraction representations to her software. Her program was now more modularized: she had separated the parts of the program that quizzed the user. A further feature of her program's flow of control was that each procedure called the next.

Then, for the first time a new design idea emerged: she created a shape in the form of a man that was stamped at the bottom of the computer screen below the fraction. For the following days, she disconnected part of the program from the control flow and worked separately on each fractions screen procedure to include the little man. When Kafai asked her what she was planning to do, she replied: "with every correct answer, the student will see an additional little man at the bottom of the screen." At the end of her game, "a bus will come and pick them all up" (see figure 13).

Figure 13: Jeannine's screen of a fraction representation with the "little man"

In the following weeks, Jeannine expanded her idea, and by the end of ISDP, her personal software included 10 instructional screens asking questions about fractions. In the course of ISDP, she changed the shape of the little man twice and worked on completing her quiz procedures. She managed to implement the particular animation with the little men. After each time a correct answer was given by the software user, a little man was placed on the bottom of the screen. After 10 correct answers, ten little men were waiting in line. Then, at the end, after all the fraction quizzes are completed by the user, there is an animation of a bus which comes to pick the men up—motivating feedback (or reward) for Jeannine's software users.

2.2. Summary of Collaborational Features in ISDP

In this chapter, we examined the interplay between social and individual processes in ISDP. In describing the collaborative dynamics in ISDP, we identified some prominent features that are partly different from those found in existing literature. We also specified two particular kinds of interactions that influenced this learning process: one of collaborative nature, where students actually worked together on specific designs and problems, the other a collaboration-through-the-air, in which students interacted with free-flowing ideas and concepts. We strongly believe that the success of ISDP is drawn upon the *integration* of these interaction processes. The case studies of three young software designers provided examples of the flexibility of ISDP, and the different ways students can use the building of software to explore their individual ideas and meet their individual learning styles, needs, and desires—by working alone or with a partner.

The 'Incubation Phase' of the project was chosen to explore these issues of collaboration. It is a challenging stage within which students tease out their intuitions and preliminary design ideas and is a useful period for illustrating the optional and flexible collaborative features of the project. In the process of constructing an understanding of this complex enterprise, students chose what seemed best-suited to their styles and personal knowledge of programming and fractions. Gerald, for example, was not able to immediately build the cornerstone for his project. Initially, he lost interest and was somewhat frustrated. However, his interest in the project was sparked again through his walking around, chatting with friends, and watching other students' software projects. He eventually decided to collaborate with one of them. This collaboration turned out to be a fruitful act which resulted in Gerald's creating very interesting designs. In contrast, Stacey and Amy needed the collaboration in order to get started. Jeannine, on the other hand, prefered to work on her own, although other students used her procedures and were inspired by her designs. This illustrated the ways in which students took advantage of the ideas and knowledge floating around the computer circles and how they shared their impressions with each other. However, they chose different ways of building their own ideas of what **their** project would be.

In many ways, this is comparable with the environment of professional software designers, or scientists in general, as described by Latour (1987) or Schoenfeld (1988). Schoenfeld, for example, describes the influence of “ideas in the air” on his own research agenda within the culture of his scientific community:

> I have claimed that our work is, in large part, the product of our environment, and that in other locales the work could not have evolved the way it did here... It shows how two of the most important features of our analysis [of data from videotapes, for example] have their roots in ideas current in our local intellectual community—how we are conditioned to see new or different things in our data, as a result of living in a community whose ‘common sense’ supported our seeing things that way (Schoenfeld, 1988, p. 6).

Two features of Schoenfeld's work are of a similar nature to the atmosphere and activities of ISDP: His casual conversations with other researchers in his community, which were seemingly unrelated to the ongoing work; and the culture that he worked in, which supported the growth of these conversations.

In addition, the culture created during the Incubation Phase can be described as providing a "home base" for the young software designers. It supported conversations and interactions that allowed students to explore various ideas, to experiment with different approaches, to continue working on some ideas, and to dismiss notions that were not working. Our vision of the "home base" is that of a place where students can return to start new ideas while retaining the original task of designing a piece of software. In the same sense, it is also a "comfortable zone" where students can explore more familiar ideas—individually and collaborativelly—until they are able to "break away" with their new concepts and make discoveries. The home base is established in different places: in the Designer's Notebook, where the students can invent, describe, and scribble their designs without actually having to implement them; in their programs, where the students can write segments of code and disconnect them later from the flow of the program; and in trying out different screens before settling on one design idea to implement throught their software. Harel also described in Debbie's case (e.g., 1988) how Debbie was frustrated at first, and did not know how to get started. She also generated many ideas in her Designer's Notebook, but did not implement all of them in her software project. She used the Designer's Notebook as well as her software project as a home base for exploration. In one particular example, Debbie first expressed her personal feelings by writing poems before she could actively begin work on her software project. In this case, writing poems was a comfortable zone for Debbie to start thinking about the fractions project.

In short, this time spent wrestling with different perspectives and ideas—individually and collaborativelly—during the Incubation Phase proved to be very important in Harel's as well as in the present study. It resulted in students' conversing and starting to build their conceptual models of what their instructional piece of software would be, and it helped them conceptualize what they wanted to communicate about fractions. Later, we were able to expand this by implementing “consulting activities” (see Kafai & Harel, the following chapter in this volume) which allowed students to revisit their knowledge of fractions and Logo programming through play and social interaction, without feeling at risk.

Acknowledgements

We wish to thank the fifth-grade teacher Marquita Minot and her students, the fourth-grade teacher Gwen Gibson and her students, as well as the third-grade teacher Fran Streeter and her students for their collaboration with us and their great contribution to this work. Without them, this research would not have been possible. The research reported here was conducted at Project Headlight's Model School of the Future and was supported by the IBM Corporation (Grant # OSP95952), the National Science Foundation (Grant # 851031-0195), the McArthur Foundation (Grant # 874304), the LEGO Company, Fukatake, and the Apple Computer Inc. The preparation of this paper was supported by the National Science Foundation (Grant # MDR 8751190), and Nintendo Inc., Japan. The ideas expressed here do not necessarily reflect the positions of the supporting agencies. We thank Seymour Papert, Ricki Goldman Segall, Uri Wilensky, Mitchel Resnick, Aaron Falbel, and Colette Daiute for their insightful comments on our work and on previous drafts of this chapter.

References

Berger, P., & Luckmann, T. (1966). *The social construction of reality*. New York: Irvington.

Collins, A. S., Brown, J. S., & Newman, S. (1990). Cognitve Apprenticeship: Teaching the craft of reading, writing, and mathematics. In L. B. Resnick (Ed.) *Cognition and instruction: Issues and agendas*. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Dalbey, J., Tourniaire, F., & Linn, M.C. (1986). Making programming instruction cognitively demanding: An intervention study. *Journal of Research in Science Teaching*, 23.

Daiute, C., & Dalton, B. (1989). *Collaboration between children learning to write: Can novices be masters?* Paper presented at the American Educational Research Association, April 1989, San Fransisco, CA.

Duckworth, E. (1987). *The Having of Wonderful Ideas and Other Essays on Teaching and Learning*. New York, NY: Teachers College Press.

Harel, I. (1988). *Software design for learning: Children's construction of meaning for fractions and Logo programming*. Unpublished Ph.D. Thesis Cambridge, MA: Media Laboratory, MIT. (Available through Ablex, Spring 1991)

Harel, I, & Papert, S. (in this volume). Software Design as a Learning Environment. pp...

Johnson, D., Johnson, R., & Jolubec, E. (1986). *Circles of learning: Cooperation in the classroom*. Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Kafai, Y. & Harel, I. (1990a). Replicating the instructional software design project: A preliminary research report. In I. Harel (Ed.) *Constructionist Learning: A 5th Anniversary Collection*. p. 150-170. Cambridge: MIT Media Lab.

Kafai, Y. & Harel, I. (this volume). Children Learning through Consulting: When Mathematical Ideas, Programming and Design Knowledge, and Playful Discourse are Intertwined. In this volume...pp....

Kurland, D. M., Pea, R. D., Clement, C., & Mawby, R. (1989). Development of Programming Ability. In E. Soloway & J. C. Spohrer (Eds.). *Studying the Novice Programmer*. Hillsdale, NJ: Lawrence Erlbaum, p.83-112.

Latour, B. (1987). *Science in action: How to follow scientists and engineers through society*. Cambridge, MA: Harvard University Press.

Newman, D., Griffith, P., & Cole, M. (1989) *The construction zone:* Working for cognitive change in school. Cambridge: Cambridge University Press.

Palinscar, A. S., & Brown, A. L. (1984). Reciprocal teaching. *Cognition & Instruction*, Vol. 1, p. 117-175.

Schoenfeld, A. H. (1988). Ideas in the Air. Speculations on Small-Group Learning, Environmental and Cultural Influences on Cognition and Epistemology. Institute for Research on Learning, Report No. IRL88-0011, Palo Alto, California.

Slavin, R. (1983). Cooperative learning. New York: Longman.

Soloway,E. (1988). It's 2020: Do you know what your children are learning in programming class? In R. S. Nickerson, & P. P. Zodhiates (Eds.) *Technology in Education: Looking toward 2020*. Hillsdale, N. J.: Lawrence Erlbaum.