4. Collaborative Learning Strategies: Storm's Weekend with Rachael

4.1 Storm's Weekend with Rachael

One Friday afternoon in April of 1996, I accepted the MOOSE Crossing application of a new member, a twelve-year-old girl who chose the character name Storm, and then I left town for the weekend. I usually log on periodically over weekends, but this particular weekend I went to Maine and was offline. When I returned on Sunday, I was surprised to learn that Storm now knew the basics of how to program. She had had limited previous experience—she had once tried Logo in school. Over the weekend, she made Jasper (a frog you can hug), Callie (a cat who purrs when you tickle her and responds wryly when you address her as "cat" rather than by name), and a catnip mouse for Callie. She also built and described three homes, and collaborated with Rachael (girl, age 13) on an extension to Rachael's castle, complete with a dead princess on the floor who had died for love. Storm and Rachael spent most of the weekend together, talking and helping one another with their projects.

I've told this story many times, and people often shake their heads with disbelief, grinning. But what is there about the story that is so remarkable? It's certainly not how we stereotypically imagine two children spending a holiday weekend. It's hard for many people to imagine any "real people" having meaningful control over computational media, much less girls. What was it that interested them so completely that they would spend most of the weekend absorbed in it? What were they doing and how did they learn to do it?

Another set of questions concerns the nature of the collaboration. What prompted Rachael to spend an entire weekend helping a stranger? This is quite different from a group of students being assigned in school to do a group project. Rachael chose to help Storm for an extended period of time simply because she enjoyed doing so. What began at the start of the weekend as an expert assisting a novice quickly became a joint effort in which what is achieved is greater than what either participant could have accomplished individually. Their collaboration was the beginning of a friendship.

In this chapter, I describe in detail what took place over that weekend. Everything typed on MOOSE Crossing is recorded, with written informed consent from both parents and children. Over that weekend alone, 3.7 Mb of data was recorded of their experiences. This data includes everything each girl typed and saw on the screen—everything they did, and everything they said to one another online. By examining what took place in extended detail, I am attempting to present what Clifford Geertz calls a "thick description" (Geertz 1973). Methodologically, studying this online medium has interesting advantages: we have a complete record of the interaction between these two

girls. They could communicate only through the computer, and every keystroke was recorded.

Seven months after this weekend took place, I sent Storm and Rachael copies of an earlier version of this chapter to read and comment on. Social science has in recent years increasingly rethought the issue of ethnographic authority. Henry Jenkins writes that "The newer ethnography offers accounts in which participation is as important as observation, the boundary between ethnographer and community dissolves, and community members may actively challenge the account offered of their experience" (Jenkins 1992). I chose to give the girls an opportunity to respond to my account of their experiences. After they had each read the chapter, I invited them to come to the Media Lab to discuss it with me, and to meet one another for the first time. It was fortunate that Rachael lives in the greater Boston area, and Storm lives an approximately ninety-minute drive away. (I'm grateful to Storm's parents for taking the time to make the trip.) I interviewed them both individually and together about their experiences that weekend and on MOOSE Crossing in general, and recorded those interviews on audio tape. Over all, they enthusiastically concurred with my account. As I had hoped, the interviews helped me to clarify what they each were thinking and feeling over the course of the weekend. As an added bonus, it also gave me an opportunity to ask the girls in what ways meeting face to face was different from meeting online.

There are four primary goals to this analysis:

- to give the reader a more concrete feel for what participating in MOOSE Crossing is like,
- to explore what children find compelling about this environment,
- to explore the learning strategies children typically use on MOOSE Crossing, and
- to explore a style of collaboration that often occurs there.

In my first outline for this thesis, there was a chapter called "Learning Strategies" followed by a chapter entitled "Collaboration and Learning from Peers." As soon as I sat down actually to write, it became clear that these were one chapter. My error was telling—we tend to think of learning first, and collaborative learning as a special sub-topic. In fact, all learning takes place in a social context. As a result, almost all learning has collaborative aspects. Even when people undertake learning experiences alone, they are still in some way responding to a broader social context—to the expectations of others, and the way we define our sense of self through our accomplishments and in relationship to others. MOOSE Crossing was designed to support the sort of self-directed, peer-supported, collaborative learning that took place for Storm and Rachael over this weekend.

4.2 Friday: Diving In

Rachael, who at this time had just turned 13 the previous week, is one of MOOSE Crossing's most dedicated and accomplished regulars. She was the first home-schooled student to join. Being home-schooled gives her greater time to devote to MOOSE Crossing, and also greater need for social contact.[1] An hour and a half after I accepted Storm's application, Rachael checks the list of all members, notices that there is a new member who hadn't logged on yet, and sends her this mail:

```
Message 1 on Storm:
From:     Rachael
To:       Storm
Subject:  hi

Dear Storm,
Hi! My name is Rachael. Who are you? I am thirteen years old and I am
female. I have been on moose crossing scince january and whould love
to be your friend.  The best times to go on moose crossing are on
mondays and fridays afterschool.

Rachael
```

At a little before 5 PM, Storm connects for the first time. Her connection message tells her she had mail, so she checks "help mail" and figures out how to read her message from Rachael. She told me in a later interview that she learned how to use the online help system and a few other basic commands from the introductory message mailed to all new members with their passwords. She learned about the mail system because she wanted to read her mail from Rachael—interactions with other children were an integral part of her explorations with the system from the very first command typed.

A moment later, Rachael notices that Storm has connected, and pages her "May I join you?" ("Paging" is a way of communicating with someone who is not in the same room in the virtual world.) Rachael waits impatiently for Storm to respond, repeatedly checking 'who' (a command that tells you who is logged on, how long they've been connected, and how long they've been idle). She also repeatedly checks Storm's last commands. It's possible to see all the commands someone has typed recently. We added this feature to MOOSE Crossing to make it easier to help others figure out what they're doing wrong when things aren't working right. When you look at someone's last commands, the system tells the person that you have looked. Thirty seconds later, Storm pages Rachael "Yes" and Rachael joins her, moving to be in the same room as Storm in the virtual world:

---

[1]Over time, MOOSE Crossing has become particularly popular with home-schoolers. The open-ended, self-directed educational philosophy behind MOOSE Crossing is consistent with the educational philosophy of most of the home-schooling movement. Additionally, it provides much-needed social contact with peers.

```
Rally says, 'Greetings Clover'
Rally arrives, following Rachael.[2]
Rachael says, 'hi'
Storm says, 'hello, all'
Rachael smiles.
Rachael says, 'Rally and Clover are my pets.'
Storm smile
Rachael says, 'how old are you?'
```

[Here Rachael looks at Storm; her description is still blank.]

```
Storm says, '12'
Rachael nods.
Rachael says, 'Are you at the media lab or somewhere else?'
Storm says, 'this is the first time I've been here!:)'
Rachael says, 'I mean in real life where are you? I'm at my house.'
Storm says, 'same here'
Storm says, ' anyone here like star trek?'
Rachael says, 'Many people go to the media lab, in MIT, to do moose
   crossing. I was just wondering if you where. Yes, I do.'
Storm says, 'i'm a trekker, heart and soul!:)'
Rachael giggles.
Rachael says, 'austina/kristina[3] likes it too.'
Storm says, 'whoohoo!'
```

The expression "whoohoo!" has become popular over the last few years, and is taken from the television show *The Simpsons*. Homer Simpson says it when he's happy. Popular culture, particularly science fiction television, immediately gives these two girls something to talk about. Their mutual interest also identifies them to each other as being part of an unusual subgroup of teenagers: girls who like science fiction. While many cultural critics scorn popular culture as a negative influence on children (Postman 1985), in the right environment it can be a rich source of raw materials for personal expression (Jenkins 1992).

```
Rachael says, 'which generation do you like best?'
Rachael says, 'I like the TNG'
Storm says, 'trekkers will take over the world! I like the next
   generation
Rachael smiles. "Mabye so!"
Rachael says, 'same here, the next generation. Isn't a shame that it
   stopped?'
Storm says, 'oh, man yes! bummer!\'
Rachael says, 'Whould you like any help?'
Rachael says, 'I hear that the next TNG (next generation) movie is
   about the borg.'
```

---

[2]For technical reasons, events on MOOSE Crossing sometimes happen in an unusual order—Rachael's pet Rally says hello before he is announced as arriving in the room.

[3]The characters Austina and Kristina are played by the same person, MIT undergraduate Austina Vainius, an undergraduate research assistant working on the MOOSE Crossing project.

Rachael guesses that Storm doesn't understand that the abbreviation "TNG" stands for "The Next Generation," and here finds a way to let her know without making a big deal out of it.

```
Storm says, 'oops, didn't mean the \. no, i'm figuring this out!
   thanks!'
Storm says, 'well, the movie is about the two generations meeting.
   You should see it!'
Rachael says, 'that's fine. Just askin'. But if you want any don't be
   afraid to ask.'
```

Storm hasn't yet had time to explore or get confused about anything yet—Rachael contacted her within moments of her arrival. Rachael's immediate offer of help is a bit premature. While Storm does not immediately avail herself of that help, she now knows that help is available when she needs it.

Rachael and Storm's conversations about help with MOOSE Crossing and Star Trek become interwoven. This is typical of MUD exchanges. In face-to-face discourse, turn-taking conventions help to keep a conversation focused. In MUDs, while one person is responding to a previous point, another may introduce a new one. Typing in parallel often leads to many-threaded conversations (Cherny 1995).

```
Rachael says, 'Oh, I saw it. I meant the one they are making now.'
Storm says, 'anyone like monty python?'
Rachael says, 'Yes, but I never get a chance to see it.'
Rachael says, 'Have you heard of Babylon Five?'
```

Rachael explained earlier that Rally and Clover are her pets; however, from her use of "anyone," it's not clear if Storm yet understands that it's only really Rachael there. At this point, Storm looks at Rachael. She sees:

```
A girl with brown hair and green eyes. On her head is a sliver
headband with silver strands. At the end of each strand is a silver
ball. Around her neck is a silver chain.
She is awake and looks alert.
Carrying:
Franky                                        Rachael's Bean
Rachael is wearing a tye die shirt and overalls.
Rachael smiles.
```

Rachael had written a special script (called a "look_self" script) which is run every time someone looks at her. She is now notified "Storm just looked at you." The program also causes her automatically to smile.

Rachael next looks at herself, to see what Storm has seen. Storm tries to look at Franky and Rachael's bean, not realizing she can't see objects that someone else is holding. Rachael audits Storm (a command which shows you all the objects a person owns), and sees that she still owns nothing.

```
Rachael says, 'Do you like the way I look?'
Storm says, ' you look bea-u-ti-ful!'
Rachael says, 'thanks.'
Rachael says, 'Well, have you heard of babylon five?'
Rachael says, 'It is another science fiction TV show.'
Rachael hugs Rally.
Rally squeals happily.
```

Rachael here gets a bit fidgety, typing “who” a couple times and hugging her pig Rally. Storm is busy describing herself. Rachael guesses as much, and looks at Storm immediately after the description is finished. She sees “you see a tall, black haired, white - skinned girl, wearing all black. she is wearing lots of silver jewelry.”

```
Storm says, 'never seen it'
Rachael says, 'Most people haven't. You look nice!'
```
[Storm here looks at Rally, and sees "A small pink piggy."]
```
Storm says, 'why thank you!'
```

Storm gets immediate positive feedback. The feedback has value on several levels. First, it's a compliment to her technical ability—she was able to figure out how to describe herself. Second, it's a compliment to her writing ability. Third, being told you look nice is a generally a mark of social acceptance. Finally, given the predominance of issues of appearance in the culture of teenage girls, being told that your virtual self looks nice potentially has deep resonances.

```
Storm says, 'monty python is very funny'
Rachael says, 'You can changer your gender by typing "gender me as
   <gender>"'
Rachael says, 'the <>s are where you fill in the blank.'
```

The MOOSE Crossing documentation is written in this style, with angle brackets indicating that you should fill in the blanks. A classic slapstick routine has a bailiff trying to swear in a witness by saying “Repeat after me. I, state your name.” The witness replies literally, “I, state your name.” Giving someone directions including some things to duplicate literally and others to interpret is awkward, and is a common source of confusion for new users on MOOSE Crossing. Rachael here is anticipating the problem, giving Storm useful information not just about one command but about how to read the documentation in general.

```
Rachael says, 'Yes, monty python is.'
Storm says, ' i know. I'll love doing that! what planet are you
   from?'
Rachael says, 'Depends.'
Storm says, 'i'm from zork'
Rachael says, 'Of what universe you are referring to-this one, star
   trek...my own..'
Rachael says, 'the game?'
Storm says, 'i'm referring to a computer game.'
```

```
Storm says, 'yes. it's fun'
Rachael says, 'I thought so. I've played it.'
Rachael says, 'But I never passed it.'
Rachael says, 'Have you played MYST?'
Storm says, 'what one?'
Rachael says, 'Zapies...I made it up myself when I was in 1st grade.'
Storm says, 'no. my mom beat myst.'
Rachael says, 'I haven't yet. It's kinda hard.'
Rachael says, 'Do you live in the Boston area?'
```

Storm here attempted a bit of fantasy role playing. Rachael asked for a clarification—what is the reference frame of the conversation? They moved back to talking about real life.

```
Storm says, 'could you show me around?'
Rachael says, 'certainly!@'
Rachael says, 'anywhere in particular?'
Rachael says, 'or just everywhere?'
Storm says, 'everywhere! please'
Rachael says, 'well, let's go up!'
```

When Rachael first offered help, Storm wasn't yet ready to take advantage of it. Now that Storm has tried out a few basic commands and gotten to know Rachael a little bit, she's comfortable asking for help. Rachael leads Storm up to the clouds, where Rachael demonstrates how you can jump and roll around on one of the clouds. While Rachael is jumping around, Storm sets her gender to royal[4], making use of Rachael's earlier offered information.

Rachael is clearly delighted to meet Storm. She uses this command:

```
--> tell Rally to emote hugs Storm.
```

This causes her pig to hug Storm. Storm hugs the pig back. The pig allows Rachael to express affection indirectly.

Rachael next leads Storm to Home in the Clouds, a room that lets you build your own home by simply typing "build." Storm builds a home there.

Many kids on MOOSE Crossing make use of a special character class that lets them switch between multiple identities. One of Rachael's alter-egos is called QueenAnne. Rachael morphs into QueenAnne and invites Storm to see her castle. Storm types "queenanne" to get to her room without being prompted, and Rachael follows. Storm bows to QueenAnne. QueenAnne changes back into Rachael, and leads Storm to Palladia, part of her mythical kingdom where you can build your own home. Rachael next explains how you can build a home in certain places (not realizing Storm has already done so at Home in the Clouds). Storm builds another home at Palladia. Rachael offers

[4]Changing your gender changes your pronouns. Having the gender "royal" means you are by default referred to by "the royal we."

to show Storm her “normal” home, but Storm replies she'd like to do the tutorial. Rachael goes home. Storm types “tutorial” and then starts looking at the dog tutorials, a three-part sequence that teaches the basics of MOOSE programming by helping you to make your own pet dog. Rachael pages good bye:

```
Rachael pages you.
She pages, 'I'm disconnecting now. I hope you come again. Mondays are
   the best days.'
```

Storm doesn’t reply. Rachael disconnects for about a minute, and then logs back on. Storm wanders into Rachael's room and looks at her pet penguin and cat. Rachael is there and says hi. Storm says hi back and goes out a different way than she came in. Rachael follows her. Catching up to her, she says “I can stay somewhat longer.” Storm leaves again, and Rachael hesitates and then followed her again, this time to the recycling center. Rachael uses the 'announce' command to print to the room “THE BLUE BIN EATS THE TRASH” followed by “CRUNCH” several times. (Printing unattributed messages like this to everyone in the room is technically against the MOOSE Crossing code of conduct. Members generally remind one another of the code of conduct when it gets out of hand, but tolerate it in moderation if the content stays friendly.) Storm says “I'm getting out of here.” She told me during a later interview that she didn’t realize Rachael had generated those messages. They walk to town together. At MOOSE Crossing, Storm says “hi! where to.” Rachael replies “You choose, I'll follow.” They arrive on Main Street:

```
Main Street
You're on the edge of Our Town. Looks like there's space to build
   some shops here!
Obvious exits: ..west.........MOOSE Crossing
               ..north........North Main Street
               ..east.........Town Hall
Storm is here.
Rachael has arrived.
Rally says, 'Hello Clover'
Rally arrives, following Rachael.
Clover arrives, following Rachael.
Rachael says, 'I suggest n'
```

[Rachael is suggesting they continue their explorations by going north. “n” is usually allowed as an abbreviation for north.]

```
Storm says, 'how do you make animals?'
Rachael says, 'Well, it depends if you want a new type of animal or
   one that already exists.'
Storm says, 'new type'
```

Here Rachael checks the command “parents Rally” and gets this response output:

```
Rally(#381)   generic_greeting_creature(#402)   Generic Teachable
   Object (#225)   Generic Puppet(#223)   Generic Following
```

```
   Object(#342)   Generic Gendered Object(#77)   generic thing(#5)
   Root Class(#1)

Storm says, 'i'd like an animal to follow me around'
Rachael says, 'type "create #223 named <name>".'
```

Storm does, and chooses the name Jasper. Rachael looks at Storm and sees she's now holding a creature. She tries to look at it, but Storm is holding it. She asks Storm to drop it. Storm describes it as "a frog" and drops it for Rachael to see.

```
Rachael says, 'neat idea! I wish I thought of a frog!'
Rachael says, 'to make it follow you type "set Jasper's following to
   me'
```

Rachael responds immediately with positive reinforcement, and additional technical advice. Storm improves Jasper's description to "frog with orange skin, a black stripe and a sense of humor."

```
Storm says, 'Jasper likes Star Trek too'
Rachael smiles.
Storm says, 'how do you make him say things'
Rachael says, 'okay..now go somewhere, and it will follow you.'
Rachael says, 'it depends.'
Storm says, 'ok...'
Rachael says, 'There are two ways. One, you could make a script, so
   that if you type something it will do something in return, like
   huging Rally..'
Rachael hugs Rally.
Rally squeals happily.
Storm says, 'how'
Rachael says, 'if you go to the pencil, it will make a thing appear.
   Fill in the blanks.'
```

[The pencil icon on MacMOOSE lets you edit an object, script, or property. See Figures 3.4-3.6]

```
Rachael says, 'then, when you have a script ready to fill out, you
   type "on <script's name> this.'
Rachael says, 'let me give you an example..'
```

Rachael lists out a script of hers, and says it out loud:

```
Rachael says, 'here:on hug this    tell context "You hug " + my name
   + "."     announce_all_but context context's name + " hugs " + my
   name + "."     emote "squeals happily."   end'
Rachael says, 'where the big blanks are, is where the returns are.'
```

Rachael waits a minute, repeatedly checking to see if the script is there yet.

```
Storm says, 'what do you put on the pencil blanks'
Rachael says, 'here, fill in the object as Rally, then fill in the
   script as "hug".'
Rachael says, 'then you'll see what a normal script might look like.'
```

Storm first opens an editor on Rally:hug, and then on Jasper:hug. One of the most common learning strategies for kids on MOOSE Crossing is using other kids' programs as examples. In this respect, MOOSE Crossing shares a powerful learning feature with the World Wide Web. On the web, you can view the HTML code for any object. You don't need to go to a special library of examples—everything is an example. In the course of your normal use of the web you encounter objects that you can later go back to and use as models. Similarly, every object on MOOSE Crossing is an example. While the MOO language allows some programs to be unreadable to others, I deliberately chose not to support this feature. All programs can be viewed by everyone and learned from. Children often start with very simple variations on others' programs, and gradually progress to more original creations. This powerful learning strategy is often prohibited in schools, where it is declared to be "cheating."

It's worth noting that on MOOSE (unlike the web), you are likely to know or have the opportunity to meet the creator of each item in the world of examples. Storm is not learning from a randomly selected example; she's learning from an example created by her new friend, Rachael. The social and intellectual relationships are mutually supportive.

```
Rachael says, 'do you understand...or should I try explaining some
   more?'
Storm says, 'wait a sec..'
Rachael says, 'okay..i'll be right back..'
```

Rachael wanders around a bit and comes back to see how Storm is doing. Storm asks how she's supposed to save her work—whether she's supposed to click "Change" or "Revert". Rachael tells her to use 'Change'. (Based on feedback from users, we've since changed "Change" to "Save".)

```
Storm says, 'what do you do after clicking change?'
Rachael says, 'nothing...it should work..'
```

Rachael here lists out Storm's script and sees this:

```
on hug this
tell context "You hug" + "my name+ + "."
announce_all_but context context's name + "hugs" + my name + my name
   + "."     emote "grins widly and jumps with joy."   end'

Storm says, ' it says I'm missing "end"'
Rachael says, 'hi..sorry...my computer crashed.'[5]
Rachael says, 'you always need to put end at the end.'
```

[5]The version of MacMOOSE Rachael was using was still somewhat unstable. The application did not become reliable until summer 1996. Rachael knows that she may not have heard everything Storm said, since Storm may have continued typing while Rachael was disconnected.

Storm examines Rally, checks 'parents Rally' and 'parents Jasper' and tries to chparent Jasper to #402, generic greeting creature, a program written by an eleven-year-old boy which serves as the parent object for Rachael's pig. She told me during a later interview that she learned this from the tutorial she read earlier. She gets the syntax slightly wrong (forgetting the word "to") and the chparent doesn't work.

```
Rachael says, 'sorry it crashed again.'
Rachael says, 'it needs to have end on the last line.'
```

Storm reads 'help chparent'.

```
Rachael says, 'like this 1:  on hug this 2:  tell context "You hug" +
   "my name+ + "." 3:  announce_all_but context context's name +
   "hugs" + my name + "."     emote "grins widely and jumps with
   joy." end
Rachael says, 'I mean..'
Rachael says, 'you need to have end on a line below the others..'
```

This is somewhat confusing advice. Rachael here checks Storm's last commands. The ability to check someone else's last commands was designed for exactly what Rachael is doing: helping someone figure out what they're doing wrong.

Storm now moves the 'end' to a separate line of her program, and also correctly chparents Jasper to generic_greeting_creature.

```
Rachael says, 'I'm sorry..but I need to go...any questions...?'
Storm says, ' nope! Thanks a lot, Rachael! Bye, Rally, Clover!'
Rachael says, 'bye...see you again soon okay! Can you come by
   monday?'
Rachael says, 'Or this weekend?'
Storm says, 'probably!'
Rachael says, 'By the way, this monday most people won't be
   here...sorry. (because of vacation)'
Rachael says, 'bye!'
Storm says, 'i'll be there, though!'
Rachael says, 'okay..see you then!'
```

Rachael now goes home, and Storm tries to hug Jasper. She gets this error:

```
OOPS!:  Can't find an object named "my name+ + "
     :  Called from Jasper (#913):hug
     :  Line: <2> tell context "You hug" + "my name+ + "."
```

She gets rid of the stray quote before the "my" in line 2, and next gets an error message from the missing return in line 3. She changes it a few times, and finally gets it to compile, but now gets a run-time error. Next she simplifies the script, eliminating the last few lines:

```
on hug this
  tell context "You hug" + my name + "."
  announce_all_but context context's name + "hugs" + my name + "."
```

This gives her a “missing end” error. She now changes it to:

```
on hug this
  tell context "You hug" + my name + "."
  announce_all_but context context's name + "hugs" + my name + "."
  emote "grins widely and jumps with joy."
end
```

This compiles, and she tries it out:

```
--> hug Jasper
You hugJasper.
Jasper grins widely and jumps with joy.
```

She edits the program to add in the missing space after the word “hug.” Next she looks at herself, adds a few adjectives to her description, and changes her gender from royal to female. She looks at Jasper and hugs him one more time, goes home, and then disconnects for the day.

4.3 Saturday: Independent Progress

On Saturday, Rachael is on a few times during the day; no one else is around.

Storm logs on in the late afternoon, sees that no one else is on, and wanders around a bit. Next she creates a new creature and calls it Callie. She sets Callie's gender to female and Jasper's to male. She describes Callie as “You see a large, furry cat, with big orange eyes and a royal air around her.” (She uses the client to edit Callie's description property this time, instead of using the “describe” command as she had previously.) She adds this script to Callie:

```
on tickle this
  emote purrs
end
```

That works, so now she adds this one:

```
on utter "cat"
  say "I am not a mere cat! I am deeply offended!"
  emote "looks indignant and raises her nose haughtily."
end

on utter "Callie"
  say "It is a beautiful name, is it not? Just like me."
  emote "purrs and sits down in a regal pose."
end
```

MOOSE's pattern-matching parser makes it easy to have multiple scripts with the same name. These new scripts both work as well. When you type "utter cat," everyone in the room sees:

```
Callie says, "I am not a mere cat! I am deeply offended!"
Callie looks indignant and raises her nose haughtily.
```

She looks back at Jasper's hug script again, and next changes Callie's utter scripts to be:

```
on utter "cat"
  tell context "Callie says, 'I am not a mere cat! I am deeply
   offended!'"
  announce_all_but context context's name
  emote "looks indignant and raises her nose haughtily."
end

on utter "Callie"
  tell context "It is a beautiful name, is it not? Just like me."
  announce_all_but context context's name
  emote "purrs and sits down in a regal pose."
end
```

In her first version, she was able to put the now familiar "say" and "emote" commands—commands she uses all the time to communicate with Rachael—into her program. Her second version makes her program more like the other program she has seen before. However, she doesn't understand what these commands really do. In this version, when you run the "utter Callie" script:

```
--> utter Callie
[The context (the person typing the command) sees:] It is a beautiful
   name, is it not? Just like me.
[Everyone else but the context sees:] Callie
[Everyone sees:] Callie purrs and sits down in a regal pose.
```

Next she changes the "utter cat" script:

```
on utter "cat"
  tell context "Callie says, 'I am not a mere cat! I am deeply
   offended!'"
  announce_all_but context context's name +  "looks indignant and
   raises her nose haughtily."
end
```

Now the output is:

```
--> utter cat
[Context sees:] Callie says, 'I am not a mere cat! I am deeply
   offended!'
[Everyone else sees:] Callielooks indignant and raises her nose
   haughtily.
```

In her first version, the words “Callie says” were generated automatically by the say command. When she switched to “tell” instead of “say,” those were missing. She now has manually added them back in. Storm tests this version and observes that she doesn’t see Callie look indignant any more, so she puts it back to an 'emote' and makes a similar change to the “utter Callie” script:

```
on utter "Callie"
  tell context "Callie says 'It is a beautiful name, is it not? Just
   like me.'"
  emote "She purrs and sits down in a regal, but dainty, pose."
end
```

This doesn't do quite what she wanted:

```
--> utter Callie
[The context sees:] Callie says 'It is a beautiful name, is it not?
   Just like me.'
[Everyone sees:] From Storm:[6] Callie She purrs and sits down in a
   regal, but dainty, pose.
```

Next she changes “tell context” in the second script to “tell Callie”. When she tries it, that line simply doesn't appear to Storm. She puts it back to “context”. In the emote, she changes the “She” to “Callie”, so now she gets a line beginning with “Callie Callie”. Next she changes the start of the line to “emote say.” This prints out “Callie say Callie”. Next she simply scrunches it into one long “tell context” command:

```
on utter "cat"
  tell context "Callie says, 'I am not a mere cat! I am deeply
   offended!' Callie looks indignant and raises her nose haughtily at
   you, with a sniff of disgust."
end

on utter "Callie"
  tell context "She says 'It is a beautiful name, is it not? Just
   like me.' She sits in a regal pose, with an air of such splendor
   that makes you feel unworthy of her presence."
end
```

Now only the person typing the command sees Callie’s response. Storm may not understand that other people won’t see it. In fact, her original version was probably closer to what she wanted. She has spent 29 minutes experimenting.

Satisfied, she wanders around the world a bit. She chparents Callie to be a greeting creature, as she had done with Jasper. She logs off for the day.

---

[6]Storm is holding Callie. The system prefaces the output with the words “From Storm” to identify its source.

4.4 Sunday: More Mutual Reinforcement

On Sunday, Storm connects again in the morning. Her character wakes up in her home, which she notices she hasn't yet described. She describes it as:

```
You are in a black room, with silver stars all over the ceiling. In
one corner, you see a little pond with a fountain and lots of lily
pads. In another, there is a large, comfy black chair with long,
black cat fur all over
```

She modifies this a few times, settling on:

```
You are in a black room, with silver stars all over the ceiling. In
one corner, you see a little pond with a fountain and lots of lily
pads. In another, there is a large, comfy black chair with long,
black cat fur all over it. There are two open windows, and s light
breeze gently flaps the black lace curtains. Silver bells tinkle
merrily.
```

Next she reads the dog tutorial parts 1 and 2, and makes herself a child of generic dog named Toby. She adds an "on tickle this" script to Toby which just "emote laughs". She next changes it to this:

```
on tickle this
  say "That is beneath my dignity."
  emote looks at you with scorn and contempt.
end
```

She looks at the pet script on Toby's parent generic dog, and then uses that as a model to improve her tickle script:

```
on tickle this
  tell context "You tickle " + my name + "."
  announce_all_but context context's name + " tickles " + my name +
   "."
  say "That is beneath my dignity."
  emote looks at you with scorn and contempt.
end
```

The previous day, she struggled to use another object's program as a model. This time, she is successful. This program will tell the context "You tickle Toby" and tell everyone else that "<context's name> tickles Toby." The previous day's experimentation has paid off. It's worth noting also that her program is not an exact copy of what the tutorial or existing dogs do—she has customized it, making her dog react with a different personality.

She next tries a couple simple string constructions out at the command line, as suggested by one of the system tutorials (the second part of the dog tutorial):

```
--> "You pet " + my name + "."
=> "You pet Storm."

--> my name
 => "Storm"
```

She next reads help on the dig command, and digs a new room called Cat's Corner, and describes it:

```
You are in a smaller black room, and cat pictures are all over the
place!  Toys and a food dish decorate the floor. A well used
scratching post is in one corner. A big, black sofa with red cushions
is on your right. Lots of cat fur is shed all over the place. There
is a window by the sofa, where bird song is drifting in.
```

Unfortunately, she realizes that she accidentally put that description on her original room instead of the new one. With a bit of struggling, she manages to switch the descriptions. She told me during a later interview that she did this by reusing different describe commands from her input buffer to switch them.

Next she digs Jasper's Pond off of Callie's Corner. She is careful this time to make sure she is describing the right room. She describes it as:

```
Water splashes into the large pond in the center of this room!
Several frogscroak amid the weeds and stones. You can see some
minnows swim  racefully in the water. Lily pads are abundant here.
```

Storm wanders around the world a bit and then makes a new object inheriting from generic ticklish object. She told me later that she believes she saw the object number on her visit to the library. She immediately recycles the new object without trying it out. She walks back to the library, looks at the parents of one of the library's objects, goes home, and makes herself another object: catnip mouse, with parent generic thing. She describes it as " A catnip mouse that seems loved well. Made of red felt and a yarn tail."

At this point, Rachael connects and pages her 'HI!!!' They have both been online several hours since they last met, but not at the same time. Rachael audits Storm and examines some of her creations remotely. Storm is in the process of adding a 'throw' script to her catnip mouse. She looks back at the hug script on Jasper as an example. She hasn't responded to Rachael yet, who pages a second time: 'May I join you, oh great power of wind?' Storm pages back 'yes'.

```
Jasper says, 'Welcome Rachael'
Rachael beams in.
Jasper says, 'Welcome Clover'
Clover arrives, following Rachael.
Rachael says, 'hi!'
```

```
Jasper says, 'HiRally'
Rally arrives, following Rachael.
Storm says, 'Hi!'
Rachael says, 'What are you doing?'
StormhugsJasper.
Jasper grins widely and jumps with joy.
RachaelhugsJasper.
Jasper grins widely and jumps with joy.
Rachael smiles.
Storm says, 'Just experimenting'
Rachael says, 'ahh...I see you know about dig.'
Storm says, 'yup'
Rachael says, 'Well, you are learning fast!'
Storm says, 'What do you think of my rooms?'
Rachael says, 'They are very nice.'
Storm says, 'Thanks! Did you see Callie'
Storm says, 'she's my cat'
Rachael says, ' could you tell me what :-) means, or what ever that
   thing like it means? I haven't figured it out, and everyone does
   it!'
Rachael says, 'no I haven't. I'd love too.'
Storm says, 'look at it sideways! It's a smily face!'
Rachael says, 'Oh!'
Rachael grins!
```

On Rachael's arrival, Storm immediately shows off her new creation, Jasper's hug script. Rachael acknowledges it by hugging it herself and then smiling. Throughout the conversation, Rachael offers positive feedback. The exchange is nicely two-way: Rachael has a question that Storm can help her with.

Storm next invites Rachael to her other room, where she shows off her cat Callie. Storm suggests that Rachael say “cat.” Rachael does, and nothing happens. Storm tries to figure out what's wrong. Rachael looks at the code on the cat and figures out that she needs to type “utter cat”, and does. She laughs. This prints output to only her. Storm urges Rachael to try it, not realizing she already has. Rachael explains she did try it—that's why she laughed. They continue to play with Storm's cat, and then Rachael pets her own cat Clover. Storm pets Clover as well and says she's pretty:

```
Storm says, 'Clover is pretty'
Rachael says, 'I think you are worthy of being a
   Crowned_player_class'
Storm says, 'cool! like what?'
Rachael says, 'Well, you'll see.'
Rachael smiles mysteriously.
Storm says, ''wanna see my frog room'
Rachael says, 'sure!'
```

They continue talking, and showing off creations to one another. Rachael explains how her special player class gives you clothes, and suggests Storm use it. Storm explains that her creatures are named after friends of hers—her real life friend Jasper likes frogs. Rachael invites Storm to her game room,

which boots you out (sending you back to your home) if you don't type the right thing. She doesn't explain how it works to Storm, but suggests she look at the code to figure it out. After getting booted, Storm just goes idle for a while and doesn't answer Rachael's pages. Rachael goes to work on Clover's enter script, and fixes it. Storm comes back and they talk about the game room a bit. Rachael tells Storm how to use the "list" command to look at code. (Storm looks for help on "list," which unfortunately doesn't yet exist. We have added it since then.) Storm says she has to go, and they make plans to meet the next day around 3 PM. They both log off.

Rachael comes back a few hours later, and examines some of Storm's things. She makes a small change to one of her own programs, and logs off. Two hours later, she logs on briefly again. Finding no one on, she disconnects. She connects for half an hour again in the evening, and investigates what some basic system objects do.

4.5 Monday: Camaraderie Combats Frustration

Monday (a school holiday), Rachael logs on and off quickly in the morning. Storm connects a little after 1 PM. She starts to work on the dog tutorial, part 3. Rachael notices she's on and page 'Hi!' Storm pages back 'I'm working on a tutorial. When I'm done, I'll page you again. Thanks!' Rachael pages back 'Okay. See you then!!' Storm works through the tutorial, and then recycles the dog she's made. She goes back to working on the throw script on her catnip mouse. She experiments with variations on using say and emote and leaving them out entirely, with little luck.

Rachael works on improving some of her programs, and then snoops into various system functions. She checks Amy's last commands (which relate to the registration room) and then tries to figure out what the registration room does. Finally, she gets curious about what Storm is doing and checks Storm's last commands. Storm has been tossing her mouse over and over. Rachael asks if she can come over, and Storm says 'sure'.

```
Rachael says, 'hi'
Storm says, 'Hi!'
Rachael says, 'Whatca doin'?'
Storm says, 'Being unsucessful with the dratted mouse!'
Rachael says, 'AIIIE! What seems to be the problem?'
[Rachael here checks 'scripts mouse' and tosses it.]
Rachael says, 'Hmm..what do you want it to do?
Storm says, 'well, the mouse should be thrown across the room but it
   just won't GO!!!!'
Rachael says, 'okay this is the script:1:  on throw this   2:
   drop   3:    " thrown across room. "    4:  end'
Rachael says, 'well, first, " thrown across room. " dosen't do
   anything. You need to announce it.'
Storm says, ' what the heck is that supposed to mean?'
Rachael says, 'Pardon?'
```

```
Rachael says, 'You need to make it say 'announce " is thrown across
   the room."' '
Rachael says, 'I mean, 'announce my name + " is thrown across the
   room."' '
```

Storm changes the script to this:

```
on toss this
  drop
  announce myname + 'is tossed across room.'
end
```

The old throw script is unchanged, which Rachael points out. Storm tells her to try the toss script.

```
Rachael says, 'You made a tiny error. You said 'announce myname + 'is
   tossed across room.'' when it should say 'announce my name + " is
   tossed across the room."' '
[Storm adds in the space.]
Rachael says, 'When your done could you please drop your dog? I want
   to test something with Clover.'
```

Rachael tries 'toss catnip' and 'catnip mouse is tossed across the room' is printed to both of them.

```
Rachael says, 'okay, now it announces it.'
Rachael says, 'But you want it to go into the room right?'
Storm says, 'yeah'
Rachael says, 'well, I think that if you said 'this:moveto my owner's
   location' it might work..'
Rachael says, 'instead of drop...let's try it.'
Storm says, 'well, let's lay off the mouse before i strangle
   it!!!!!!!!!'
```

Storm tries to take Rachael's advice, but adds the new code to the first line instead of putting it on its own line. Now you can't toss it at all. In my face-to-face discussion with Storm and Rachael together, we were talking about the benefits of working with someone else, and Storm noted particularly appreciating Rachael's support at this moment:

> It was sort of handy when she came in. Like that catnip mouse, which I finally threw away, cause I was just sick of it. And it wasn't working, so.... It was nice to have somebody who I could just sort of rant and rave all my woes about this mouse to. [Laughing.] We just started saying "it's ok, it's ok, calm down, it's just a little object, it won't kill you."

Storm drops her smart dog as Rachael requested, and it reacts to Rachael's cat as Rachael had hoped. Storm compliments Rachael on her cat. Storm suggests they go to the game room, and Rachael agrees. This time Rachael explains how it works right away. After jumping around in the game room

for a while, Storm suggests they explore. They go to Paradise Island together, a room built by an eleven-year-old girl where you can swim, climb trees, and build a summer home. Storm builds another home there. Rachael teaches Storm about the 'audit' command. Rachael suggests Storm connect her houses in different places together to make a loop. Storm asks how. Rachael reads “help dig” and gives Storm a concrete example of a command. Storm works on making the circle of exits, with help (and positive reinforcement) from Rachael. Storm runs out of quota, and Rachael advises her to recycle stuff or ask Amy for more quota. (Each member of MOOSE Crossing is given a quota for the total number of objects he or she can build. This is necessary to prevent the database from getting too big for the machine. Children who ask for more quota are almost always granted more.)

Rachael has to leave. They plan to meet the next day at 10:30 AM. Rachael logs off and Storm logs off. Rachael logs on again a few minutes later. I am back from Maine, and talk to her a while later:

```
Rachael laughs. "Have you seen any of Storm's things
Amy says, 'she's made stuff already?'
Amy says, 'wow!'
Rachael says, 'Yes! Will almost full credit!'
Amy says, 'she's been busy!'
Rachael says, 'Yep! I've showen her somethings, but she's done some
   on her own.'
Rachael says, 'It is. She is really nice.'
Rachael says, 'We spent almost 2 hours together today.'
Amy says, 'great!'
```

Rachael is logged on for several more hours that day. She programs Marj, a robot that acts as a second self for her online.

4.6 Tuesday: Collaboration

Storm logs on early in the morning. She continues to work on the dog tutorial, part 3. As part of her work on the tutorial, she has made a pet dog named Tao and another pet called Melvin the Moose. She changes the parent of her objects from generic dog to generic cliché-spouting object, a program I wrote that inserts clichés into the conversation at random intervals, and to which you can teach new clichés. She disconnects.

Rachael and Storm both connect promptly at 10:30, the time at which they had agreed to meet. Rachael sends her new puppet, Marj, to talk to Storm. Storm teaches Melvin a cliché. Rachael drops Clover. She's made some coding improvements on Clover since they last talked. Each of them is immediately showing off the new work they've done.

Storm must be a bit confused, because she pages Rachael asking if she would like to join her. Rachael has Marj explain that she is already there. As Rachael is talking, Melvin the Moose is generating errors.

```
Marj looks distasefully at Melvin. It is giving Rachael lots of
   errors because she is talking through a puppet.
Marj says, 'I am not a MEAR puppet. I am a non - orgainic lifeform.'
Storm says, 'I am sorry about my moose. She is very...strange.'
```
[What Storm is saying isn't actually getting across to Rachael, because Rachael's puppet isn't working correctly yet.]
```
Marj says, 'We non - orgainic lifeforms have feelings too!'
(from Rachael's room) Rachael grins broadly.
Storm says, 'And I know, o mighty one, that you are not a MERE
   puppet!'   [Rachael again doesn't hear this line.]
```

The conversation continues, with Rachael not hearing Storm's side. Rachael wonders why Storm is being so quiet. Storm responds that she's not; Rachael of course doesn't hear that either. Storm asks a question, and gets no answer. She repeats it with increased urgency (i.e. six question marks). Finally, Rachael checks Storm's last commands and realizes something is wrong. In this instance, Storm has helped Rachael to detect a bug in Rachael's program. The exchange of technical help has already become very much two way.

Storm suggests they make a medieval room. Rachael likes the idea. Storm asks where they should make it. Rachael suggests off of her castle. Rachael makes the room. Storm asks if she can be the one to describe it, and Rachael says yes. Joint ownership of objects is not currently supported. It would be a desirable feature. In this case, Storm ends up building an additional room so that she can be the one to describe it. Rachael sets up the exits to connect the new room to her castle. Storm writes a first description, and Rachael offers suggestions for additions. Storm starts with:

```
You are in a red room. Armor and wepons are all over, but they are
old and dull
```

She expands it to:

```
You are in a red room. Armor and weapons are all over, but they are
old and dull, so you can't use them. But the armor and weapons are
strikingly well made.
```

This conversation ensues:
```
Storm says, 'hi. Look here and tell me what you think...'
```
[Rachael isn't in the room at this moment, and doesn't hear this line.]
[Rachael arrives.]
```
Rachael says, 'hi'
Rachael says, 'hmm...'
Rachael says, 'Great!'
Rachael says, 'The only problem is the exits...look it says
   passage...armory'
Rachael says, 'We can have amy help us fix that.'
```

```
Storm says, 'any other ideas for describing?'
Rachael says, 'wooden benches?'
Rachael says, 'blood stains?'
Rachael says, 'rusty weapons?'
Storm says, 'i like it! give me a second...'
Rachael says, 'a fire'
Rachael says, 'ok'
Rachael says, 'it is dim'
Rachael says, 'the only light is from the window and fire'
Rachael says, 'the fire is roaring'
Rachael says, 'in the center of the room.
Rachael says, 'We could rename the room to "hero's hall!"'
Rachael says, 'I mean Hero's Hall'
Storm says, 'try the description now...'
```

Rachael's brainstorming inspires Storm to improve the description:

```
You are in a red room. Armor and weapons are all over, but they are
old and dull, so you can't use them. But the armor and weapons are
strikingly well made. Tapastries of unicorns and knights decorate the
walls. There is a lone window here, with red velvet curtains. You see
a blood stain, old and brown, streaked across the wall. The only
light comes from the fire, for in this room it is eternally night.
The fire is in the center of the room, and it is crackling wickedly.
You see a white flutter in a shadowed corner, and your heart stands
still. But it is only a piece of cloth...or so you think...

Rachael says, 'wow! It's great!'
Storm says, 'Thanks.'
Storm says, 'I love writing, you see...'
Rachael says, 'It's like a murder mysteroy'
```

We discussed the construction of the room in our face-to-face interview:

> Amy: Which parts of that were whose idea? How did it start, and whose idea was which part?
>
> Rachael: I think it was her idea originally to build the room
>
> Storm: Yeah, I wanted to build a room that we could both do stuff in. And she suggested a castle. And castles and dungeons and dragons are just stuff I've always liked. When I was like 2 I had a dream about dragons that would take me places. So I always liked them and I thought, that would be pretty cool if we did all this stuff. I wanted to describe it cause I like to write. My original description was like one sentence, which was boring.
>
> Amy: And then what happened?
>
> Storm: Then she [turning to Rachael] then you said all the suggestions, you know, like blood stains and stuff like that. And I thought "Ah yeah!" And I thought of a catalog that we get, it's called Design

Toscano. It's got gargoyles and tapestries, swords and that kind of stuff. So I just added a whole bunch of stuff.

Rachael: And I think after I saw her description I thought we should add some objects or something in the room or something to follow up the description.

Storm: Like blood stains. And I thought "Ah ha! Yes, There we go! Now I have some ideas"

Amy: Do you think either of you could have done as good a job on your own?

Storm: No.

Rachael: No, I don't think so.

Amy: Is it fun working with someone else?

Rachael: It's much more fun than working by yourself. Cause if there's nobody to see it, why even bother doing it?

This is a nice example of collaboration. The girls concur that the result is better than if either girl had made it on her own. They are learning from each other and inspiring each other to do better work. The result is a positive experience with creative writing: constructive feedback, a feeling of success, increased confidence, and increased interest in pursuing creative writing further.

Rachael and Storm decide the room needs a dead body and note. Is it a murder or a suicide? Who is dead and why? Their shared vision of what has happened here evolves as they work together. This time Rachael writes, and Storm offers detailed feedback. Rachael says she has to go. They agree to meet Thursday at 5.

Rachael comes back half an hour later. She asks Storm if she's going to go the Media Lab. Storm doesn't understand; Rachael explains that some times people go to the Media Lab to do MOOSE Crossing. Storm says no, she's not going. Storm shows Rachael her new Gargoyle. It has a couple of errors: it's an adaptation of the smart dog tutorial, but it's missing some required properties. Rachael explains that to Storm, and Storm fixes it.

Storm has made a ghost object. Its description reads:

```
The ghost of the murdered girl. She says "I will not harm you. I am
left here to mourn my loved one forever. Heaven would not take me
```

```
because of my suicide, but Hell could not take me because of my
goodness. So I must stay here, and mourn Patrick until the stars grow
cold and the Earth is no more."
```

Rachael suggests Storm might make the ghost say that whenever anyone enters the room. She explains how to write an 'enter' script.

The previous day, I met and talked with Storm online. She showed me her creations, and I complimented her on them and on her fast progress in general. Storm mentions this encounter to Rachael. Despite the child-centered nature of the environment, feedback from adults is still significant:

```
Storm says, 'oh, amy loved my work and gave me some more quota to
   make stuff..'
Rachael says, 'That's great!'
Rachael says, 'she did the same to me, byron, zoro, and miranda.'
Storm says, 'I'm going to try to make that script... Why don't you
   create a sword or something? Make it out of Generic Jewelry, if
   there is such a thing...'
```

Rachael and Storm hang out together in the same room. Rachael works on getting Marj to be able to page people; Storm writes the enter script Rachael suggested. The girls' online friendship continues to the time of this writing, much in the same fashion.

4.7 Seven Months Later: Meeting Face to Face

The girls finally met face to face for the first time on the day that they both came to the Media Lab to talk with me about this chapter. They hadn't met previously because they live more than a 90-minute drive apart, and because the MOOSE Crossing code of conduct strongly discourages meeting other members face to face for safety reasons. Storm said Rachael was much like she had imagined. Rachael confessed that she hadn't really given much thought to what Storm was like in real life. Storm arrived at the Media Lab first, and I interviewed her separately. I had interviewed Rachael on several previous occasions. Next Rachael arrived, and I interviewed the two of them together. I then gave them both an hour to talk without me present, and went to chat with their parents. The girls immediately decided to log on to MOOSE Crossing. Afterwards, I asked them to compare connecting together versus connecting alone from home:

> Amy: How is doing this sitting next to each other at two terminals different from doing it from two towns a hundred miles away?
>
> Storm: It can be a lot more funny, cause we can talk to each other.
>
> Rachael: Yeah, like when they're at the Media Lab, at least in the beginning especially, they'd be talking and they'd say like "that's not

funny." And I'm like, what are they talking about? And then when I went to the Media Lab,[7] I realized they're joking around at the same time. So we were just doing that to Zeus. I told him eventually what was going on.

Amy: So it's more fun with someone in the same room?

Storm: It's a lot of fun that way.

Amy: Do you think it would be just as much fun if everyone was always in the same room?

Rachael: Well, no.

Storm: That won't work if everyone was just in the same room.

Amy: Well, it *could* be. We could just have the people at the Media Lab and that's it.

Rachael: It's fun to have people you've never seen before. And it's fun, the aspect of it being global.

Rachael and Storm hit it off face to face. After leaving the Media Lab late that Sunday afternoon, their families went out to dinner together. A few days later, I asked them each via MOOmail (email internal to MOOSE Crossing) how she thought meeting face to face would affect their online friendship. Rachael responded "I'm not sure. I think that it might strengthen our friendship...but I honestly don't know." Storm said "I don't think it really changed our friendship much at all." They hope to meet up face to face again, but the driving distance between them may be an obstacle.

Rachael's experience meeting Storm stands in contrast to her experience meeting Miranda. Rachael describes herself as "unpopular" and says "I'm a very intellectual person. I'm not very physical." I would characterize Storm as earthy. On meeting face to face, they appeared immediately comfortable with one another. On the other hand, I would characterize Miranda as a popular child. Miranda is smart, pretty, and adept and relaxed in her social relations. Rachael and Miranda had developed a warm friendship online. However, on meeting face to face, they seemed immediately uncomfortable with one another. Online, they hadn't been aware that one of them was a popular sort of kid, and the other was unpopular. Rachael says that after meeting Miranda face to face, she was initially much less comfortable with her; however, over the months that followed, she slowly became more

[7]Rachael is referring to the children who come to the regular MOOSE Crossing after school program at the Media Lab. She attended that program occasionally—three times in 1996.

comfortable again. Different factors are significant in determining social success and social compatibility online versus face to face.

4.8 Conclusion: Integrating Technological and Social Contexts

Storm's experiences that weekend are quite typical of the general patterns of interaction on MOOSE Crossing. However, she was unusually lucky in one key respect: she immediately met someone with whom she has significant shared interests. The population of MOOSE Crossing is still smaller than is desirable (primarily because not enough kids have Macintosh computers on the Internet, and also as a result of the cumbersome nature of the registration process). This makes it more difficult for individuals to meet people they particularly like, because there are fewer people to chose from. Even within a larger community, it may take time before a new member happens to meet someone with shared interests. Storm was fortunate to meet Rachael right away.

Over that weekend, Storm used a variety of learning strategies and sources of support. In this transcript she makes use of:

- the introductory message sent to all new members with their passwords,
- the online help system,
- online tutorials,
- unsolicited support from peers,
- solicited support from peers,
- trial and error, and
- using others' projects as models.

Of particular interest is the inseparability of the social and intellectual activity going on. It matters not just that a learner have access to adequate support. One must also ask, support from whom or what? What is the relationship between the learner and the person or program offering assistance? That relationship is rarely if ever neutral. If a computer program attempts to affect a disembodied, non-anthropomorphized tone, working with that program will still evoke the learner's basic feelings about computers and technology in general. For many people, those feelings are not positive. For this reason, the tutorials on MOOSE Crossing are written in a first-person, chatty style. The dog tutorial begins, "Hi there! This is Amy and I thought I'd show you how I made my dog, Pumpernickel." (Storm noticed the tone of the tutorials, and commented that it was just right—"It's not totally perky, but it's not like [speaking in a monotone] 'this is how you do it.'") The informal, warm tone is intended to put kids at ease.

While some attention to the "personality" of a computer program can help make a more supportive learning environment, there is no substitute for

human contact. A human can, for example, tailor assistance to the particular situation. Rachael was responding to Storm in a way computer programs can not, at least yet. Newman *et al* (Newman, Griffin et al. 1989).comment that "current computer systems are actually quite far from being able to perform the feats of sensitive interpretation performed routinely by human teachers"(Newman, Griffin et al. 1989). Research in artificial intelligence and education aims to help tailor support to the user, often by attempting to model what the user knows and anticipating typical mistakes. Unfortunately, such systems are still largely unsuccessful, and it's unclear how much progress in this area is even possible. Unless the general project of creating an artificial intelligence is successful, interacting with intelligent tutoring systems will remain unsatisfying. A teacher's relationship to a student is fundamentally social in nature, and computers can not yet fill that role. It's not likely they will ever be able to do so.

Storm received help not from just anyone, but from a girl her own age who also loves *Star Trek* and *Monty Python*. Similarly, she used as examples not some official library of sample programs written by a disembodied authority figure, but programs designed and actively used by her new friend, Rachael. She also looked at additional projects that Rachael recommended, written by Rachael's other online friends. Her choice of examples to embrace is a significant act that takes place in the context of a network of social relations. A child who mimics another's way of dressing or speaking is paying that child a high compliment, and taking on that child as a role model. Similarly, using someone's project as an example to learn from is to some extent accepting the project's author as a role model. Making use of a sample program can be as much a social as an intellectual act.

Social context is of central importance to any learning experience. One of the strengths of networked learning environments is their ability to help integrate a supportive social context with the computational context. The Logo language traditionally arrives in a shrink-wrapped box, with little social support for its use. Teacher workshops try to build networks of support for the use of Logo as a learning tool; however, the workshops are usually too short, and too few teachers can attend. Once a workshop is over, typically so is the support. Additionally, the model most often used is one to many: a workshop leader supports a group of teachers; each teacher supports many kids. In a many-to-many model, kids support one another. MOOSE Crossing attempts to deliver a rich computational artifact to kids, with a network of many-to-many social support built in.