Defensive Climate in the Computer Science Classroom

Lecia Jane Barker, Kathy Garvin-Doxas, and Michele Jackson
University of Colorado
Boulder, CO 80309
{Lecia.Barker, Michele.Jackson, Ronda.Garvin-Doxas}@Colorado.edu

Paper submitted to the 33rd ACM Technical Symposium on Computer Science Education (SIGCSE 2002)

This material is based upon work supported by the National Science Foundation under Grant No. 0090026.

Abstract

As part of an NSF-funded IT Workforce grant, the authors conducted ethnographic research to provide deep understanding of the learning environment of computer science classrooms. Categories emerging from data analysis included impersonal environment and guarded behavior, and the creation and maintenance of informal hierarchy resulting in competitive behaviors. These communication patterns lead to a defensive climate, characterized by competitiveness rather cooperation, judgments about others, superiority, and neutrality rather than empathy. The authors identify particular and recognizable types of discourse, which, when prevalent in a classroom, can preclude the development of a collaborative and supportive learning environment.

1 Introduction

The NSF Information Technology (IT) Workforce program funds research to discover ways to attract and retain under-represented professionals in IT fields. In line with these goals, we studied two programs, one which actually appeals to women in contrast with one which traditionally graduates much fewer: computer science. This paper focuses on the learning environment of the computer science major at one university. The findings, while relevant to retention of women, are also relevant to men. That is, after a year of immersion in computer science courses, we have identified communication patterns which can be characterized as engendering a defensive climate, or one in which people "perceive or anticipate threat" and in which communicative interaction is more likely to be focused on self-defense than on understanding [5]. Below we discuss learning environment and social climate, present the research methods and results of data collection and analysis, then discuss the implications of the communication patterns we observed in computer science classrooms.

2 Learning Environment

Learning environment comprises "all of the physical surroundings, psychosocial or emotional conditions, and social or cultural influences" present in a learning situation [7]. Both the physical and the social aspects of a learning environment influence student participation and satisfaction [4]. Learning environments are generally discussed and studied as affecting the learning of people who function within them and the effects can be positive or negative [9]. In fact, learning environments have effects beyond learning to include socialization, particularly when certain patterns of interaction occur across many courses in a curriculum, such as a major.

The social aspect of the learning environment, often called social climate, is influenced by traditional and emergent beliefs about appropriate class activities, relationships and roles, authority, trust, the personalities and behaviors of individuals, and typical patterns of communication. Communication patterns are extremely important, since it is by creating shared understanding that teaching and learning occur. Yet the communication process is more than the transmission of information; indeed, it is the means by which social order, culture, and meaning are created and maintained. Communication patterns that become typical provide not just information, but represent order, implying and prescribing the way things are done, and who can do them, "around here."

In sorting through our data, we began to notice patterns of communication that seemed to fit with what Gibb termed defensive communication [5]. Six overlapping categories of speech are characterized as creating defensiveness in people. First, speech that is interpreted as evaluative or judgmental (especially when negative) often creates defensive reactions; it implies that the person being evaluated is not up to par. Second, speech seen as intending to control can lead to resistance; the implication is that the speaker privately believes the listener makes inadequate judgments. Third, people resist and become defensive when they perceive that others are using strategies on them rather than acting openly. Fourth, individuals can become defensive when they perceive a lack of concern or empathy; it may be seen as a rejection of the self as a valuable person. Fifth, when people communicate either implicitly or explicitly a perception of themselves as superior, feelings of inadequacy are aroused. Finally, when people communicate certainty in a dogmatic fashion, they also tend to communication a low tolerance for disagreement. When speech is characterized in any of these ways, people may become defensive which can discourage understanding of each other, lead to distrust of others, and result in competition rather than cooperation.

While our research is part of an NSF-funded project examining what attracts and repels women in IT programs of study, the scope of this presentation is the computer science classroom. Our research questions are:

RQ1: What are the characteristics of student-student and student-teacher interactions within the learning environments in the CS program?

RQ2: What, if any, patterns are evident across courses in the program?

3 Method

The best way to understand student experience of the social environment in IT classes is to immerse oneself in that culture by observing from within. This data collection method is primarily associated with ethnographic inquiry, the goal of which is to capture the details and meanings of interaction from the perspective of members of a group. Through this approach, social scientists work to articulate the shared, yet often unspoken, rules, beliefs, and values which are produced communicatively and which surround and influence the everyday practices of members of a social setting.

Over the course of the 2000-2001 academic year, we observed courses in two different types of IT programs: a traditional Computer Science (CS) major and the Technology, Arts, and Media (TAM) certificate program, in which students acquire in-depth skill with high-end multimedia software packages (e.g., Director, Flash) as well as some html programming. Over the 2000-2001 academic year, we observed 10 courses for a total of 254 hours, as shown in Table 1 below. The extensive fieldnotes recorded: number of students attending, sex, and appearance; physical layout of classrooms and seating arrangements; and descriptions of interaction (student-student and student-instructor interaction) and interactants male/ female; major). Fieldnotes were typed and the resulting 648 pages of text bound into "books," by course.

The analytical method employed best fits into content analysis, in which researchers read through the data several times, labeling them and establishing a classification scheme. Patterns, themes, and categories of analysis emerged through the systematic examination of the data both during semesters and after; we then analyzed the degree to which our categories overlapped. In an iterative process, relevant data were coded according to category. These categories are discussed in the results section, below.

Table 1: Observation Details

Program	Level	Lectures	Recitations/ Labs/Other	Hours
CS	Lower	16	7	29
CS	Lower	11	4	20.75
CS	Lower	14	3	20.5
CS	Lower	9	-	9
CS	Upper	12	9	24
CS/TAM	Mixed	28	2	47
CS/TAM	Upper	20	-	25
TAM	Upper	-	15	37.5
TAM	Upper	-	9	22.5
TAM	Upper	15	-	18.75
	Total	125	49	254

4 Categories of Interaction

The results included here are limited to Computer Science courses, though understanding of the social environment of TAM courses was useful for comparison. Two categories which emerged from the data are presented below. Although these categories are analytically separable, they are part of a system (i.e., the social environment of the computer science classroom) and are mutually influential. While the actions of all members of the system contribute to the particular "flavor" of the CS major and the classrooms we observed, it is most heavily shaped by the discourse and actions of those in authority: the instructors and teaching assistants.

4.1 Impersonal Environment and Guarded Behavior

The social environment of most of the computer science courses we observed can be characterized as impersonal, an environment in which it is easy to remain relatively anonymous and socially distant. Interpersonal relationships usually begin with learning a person's name, then learning more about a person's interests through intentional self-disclosure and inferences drawn from others' speech and actions. In observations, it was rare to hear the name of a student in class or any personal information beyond that learned or guessed by appearance and other obvious facts (e.g., only computer science majors are allowed in this class, so a student would know that all students in the room are computer science majors).

In only two courses were students required to introduce themselves to the class; in only one of these did student names continue to be used by the instructor. In fact, instructors rarely used students' names, even at the end of the semester. Instructors sometimes called on students according to their clothing, such as "the woman in the red shirt." When names were used, it seemed surprising, leading the researchers to wonder if the student and professor knew each other from extra-classroom interaction. Sometimes a professor would call on a student by name, adding, "I'm calling on you only because I know your name." In other words, he didn't know others' names, though this occurred more than two months into the semester. Further, unlike field notes from TAM observations, field notes from the computer science courses are characterized by identifying students as “F1,” “M2,” “red hair,” and so on, even in courses with very small enrollment (one of these had only 12 students by the end of the semester). As observers, we try to remember names so that we can observe patterns by student, but the lack of name use made this difficult.

Self-disclosure is sharing information that others would not usually know or discover. Self-disclosure functions to cement interpersonal relationships by helping people to predict others’ behavior and deepening mutual trust [3]. It was very rare to hear a student disclose personal information that was related to anything other than work (a serious and important topic) even in their limited before-class chitchat. Indeed, one CS instructor repeatedly disclosed personal information such as his love of poetry, music, and hiking, but students typically did not respond and remained silent, violating an unspoken rule of self-disclosure in American culture: disclosure is mutual. Students rarely responded to instructors or even spoke much in class, suggesting that the social environment was very guarded. A few examples are the lack of or very subdued student chatting before class, students sitting isolated from each other, non-response to instructor queries, silence when the instructor paused or erased the board, and students not asking expected questions such as “what was the average grade on the test?”

Computer grading of programming assignments is employed to address the very real problems of scale. However, this is another impersonal aspect of CS. Though the grading program has a name and is talked about as if it is a person, it cannot interact with students like a tutor or teacher and the only feedback it gives is whether the program compiled.

4.2 Informal Student Hierarchy

Hierarchy and status are characteristics of every social situation and relationship, whether equal or unequal [1, 10]. Hierarchy may be formal or informal, or have elements of both. Formal hierarchy occurs when certain persons have the authority and duty to govern the interaction of others, such as in teacher-student relationships. Informal hierarchy is created through the acquisition and display of status by participants in a social situation and is relevant to the values shared by members. Individuals learn the values of groups in subtle ways through interaction and present themselves as members through the expression of shared values; they make a bid to be treated as having higher status when they talk in ways that suggest they excel at the kind of skills or knowledge required for functioning in that social context. In CS classrooms, status is informally accorded to those who display technical skill or provide valued information. Who belongs and where they belong in the informal hierarchy are negotiated throughout the semester in the courses we observed.

Even on the first day of class for first semester freshmen, the process of identifying who belongs and who does not begins. One professor explained to his class that the course had no prerequisites and that all levels of experience, “never programmed, a bit of programming, and rocket scientists” are all in the same course. Thus all students belong, but the status of different types of belonging is already set up, with experience being attributed to a term popularly used to mean ‘very smart’. The instructor further defines the “rocket scientists” by saying that they often compete to do more and to out-do one another. He gives an example of a student who designed a complex game for his final project and says, “Did it help his grade [to go beyond the assignment]? No, but he had fun.” Instructors reinforce views about belonging both explicitly and implicitly. In one course, the professor demonstrated a possible project for the final component of the course (an extension of the peg-board game) and asked “[who in the class] thinks it’s a cool one?” When all of the students raised their hands, he said, “Good, because if you don’t think it’s cool, you’re in the wrong class. If you don’t think it’s cool, I won’t be able to entertain you. This is about as good as it gets.”

Setting up hierarchy and status can help students to understand the goals of courses and the major. For example, a professor compared two versions of the same code, telling the class, “don’t hesitate to write code like this [indicates the beginner’s version], but eventually you’ll write code like this [indicates the more compact, efficient version].” However, problems can arise when students confuse the source of knowledge that can lead to high status: intelligence versus experience. This is especially problematic for those with less experience, a group to which most female CS students belong.

Students with programming experience are frequently referred to as “smart,” both explicitly and implicitly. For example, one instructor announced that “we have people in this class who have never programmed and some who have created game software. By the way, we have [TA] jobs for smart students like that.” In fact, one professor described a computer scientist as “a very smart person, who knows how to create software.” Clearly, students who are successful in the major come to believe that computer scientist have knowledge which is superior in nature to other types of knowledge. An undergraduate CS major who worked with a group of non-majors said, “I tried to make sure that ideas from non-technical people could be heard by not making judgments.” He implies that he deliberately chose not to express judgments. The judgment of another person is a “one-down” move, a move which shows a perception of, in this case, intellectual superiority.

Over time, students become aware of whether they belong as well as where they fit in the CS social hierarchy. That is, they are developing their identity as “computer scientists” – or not – through interaction with each other, their instructors, and TAs. Prior to the deadline for the first assignment for a junior-level course, one TA spent the entire recitation session telling students how ‘easy’ the assignment would be: "You won't have any problem with [the assignment]. It will be simple." When a student asked, "did you already do this project?" the TA replied with a wave of his hand, "No. I spent about an hour putting together [this presentation for class] and I've written five languages commercially – this is all scraps to me." Only 26 out of more than 80 students received passing grades on the assignment. Likewise, in a lower-division course, the instructor repeatedly told the class that "this [lab] test is a slam dunk; you're all going to get 100 percent." Those who made the average grade 87 percent must have wondered, like the 54 students who failed the assignment above, whether they were "smart," like the other computer scientists (i.e., experienced students).

Through self-presentation as "smart," students also contribute to the construction of hierarchy in the classroom. For example, a male freshman made a point of telling one of us about his web design business and extensive programming experience. He explained his presence in the introductory class by saying that he did not want to place out of it so that he could have a more 'relaxed' semester. He said all of this before being reminded that the researcher wasn't another student in the class and was later over-heard sharing the same information with other students in the class.

In many cases, students ask questions that do not appear to be seeking information, but to be displays of their own knowledge. Students tend to use question forms such as, "You can do _____, right?" and "Isn't it true that ____ " "But doesn't it work [this way] in Java?" These are voiced not as questions, but as statements. The strategy seems to persuade at least some students in their classes. In a recitation, a slightly older female student said to a male student, "why are you taking this class? From what you said the other day, it sounds like you already know this stuff." One day later in the semester, the same male student brought up another way of programming an operator. The instructor agreed that there is more than one way to do it and called on him to explain to the rest of the class. The student could not articulate an explanation, but instead said "that's what I get for opening my mouth." Interestingly, when we tested our belief that students were posturing in discussions with three CS professors, all agreed that students often do that. To people who consider themselves to be outsiders to the CS classroom, however, such posturing seems unusual.

While students rarely speak in class relative to other majors in our experience, some 'experienced' programmers seem to be compelled to show off their knowledge by pointing out mistakes in syntax that they see on professor's slides and/or on any work written on the blackboard. Students frequently interrupt lecture with comments like, "Shouldn't 'expression' and 'term' be the other way around?" and "you've made a mistake in ______ [on that slide]." These comments frustrate the professors. As several told us in informal interviews, they feel that students who publicly challenge their knowledge and who point out minor mistakes on slides are interfering with lectures simply to make the point that they are experts. These professors felt that such attention-drawing strategies took away from the real point of the lectures. One described it as focusing too much on details and not on the concepts.

However, professor reactions to such challenges and displays of knowledge as they take place in the classroom suggest that they feel forced to respond to such comments. They consistently either offer 'good reasons' for having made a mistake (e.g., "that comes from doing things at the last minute") and/or attempt to make light of the mistakes in a way that indicates acute embarrassment (e.g., "obviously, I believe in text readers"). Accounts such as those offered by the instructors acknowledge and attempt to mitigate some behavior seen as dispreferred in a particular community. After a series of such corrections, one professor began discovering his own mistakes as he lectured and hurried to correct them live as he joked about making the corrections "so [that] people who have me next time will benefit from these debugged slides." Over time and in conjunction with other similar messages, such as a grading system that penalizes students a full 80 percent of their grade if the program doesn't compile (regardless of the correct planning and steps indicated in their programs), these sorts of reactions indicate that it is not *really* okay to make trivial mistakes – especially not in public – and that judging others' knowledge and abilities and defending your own is an expected part of interaction.

5 The Creation of a Defensive Social Climate

The interactions described above all work together to create, maintain, and reinforce a learning environment characterized by a defensive social climate, as discussed by Gibb [5]. No single person and no one course create a defensive climate nor did we see all the types of defensive communication Gibb identified. Rather, a series of interactions over extended periods of time lead to such a climate. In the case of the CS courses examined for this study, it is important to note that CS faculty members, as is the case in most Science, Mathematics, Engineering, and Technology (SMET) education, are actively seeking to find ways to encourage interaction in their classes and to re-enfranchise those students who have traditionally become disenfranchised from the CS major. Be that as it may, the types of interaction we saw in these courses, impersonal communication, guardedness, and jockeying for superior status, lead to and lend their support to the creation and maintenance of a defensive communication climate which works against these goals.

The impersonal environment and guarded behavior we describe is characterized by neutral, as opposed to empathetic, communication. The depersonalization of students through a failure to use their names and an environment that discourages self-disclosure of anything other than work-related information can be perceived as a lack of concern for others, which violates the normal human desire to be perceived as important, someone who matters, someone for whom others are concerned. This is a particularly salient desire for women in that they tend to define themselves in relation to others, as opposed to men, who often define themselves in terms of their occupational contributions to society [6].

This depersonalization is only reinforced and enhanced by computerized grading practices. The neutral and impersonal nature of many of the practices in the CS courses observed can unintentionally and indirectly communicate rejection rather than acceptance and therefore support of the value of the individual. This neutral communication style is further supported when teachers repeatedly tell students that what they're learning is "easy." In doing so, they implicitly deny the legitimacy of these students' concerns and fears about mastering what they perceive to be new and difficult material. Again, women are particularly sensitive to this sort of rejection and denial of their concerns [8].

Much of the system of informal hierarchy described here stems from communication that emphasizes superiority rather than equality. Equating "smart" students to experience in programming, implying that one must think that extending programs, like the peg board game, is "cool," and describing students who compete to do more and to out-do one another as "rocket scientists" (i.e., smart), all function to express and establish a social order defining the value of different types of people. This in turn leads to a competitive environment where students feel it necessary to demonstrate their superiority, not just in the execution of their assignments, but in the way they present themselves and challenge their teachers publicly. In the interrelated fashion of social construction, this behavior from students contributes to a communication environment in which teachers feel that they are being judged and evaluated. In iterative fashion, this leads them to communicative defensively, modeling behaviors that suggest that mistakes are bad and must be justified. Yet a learning environment should be a place where making a mistake is an acceptable action. Once again, these communicative practices can be discouraging to women, who tend to be less competitive and more cooperative in their approach to social interactions.

With SMET education's trend toward collaborative learning; ABET 2000's focus on collaboration; industry's emphasis on team work; and the push for diversity in SMET disciplines, institutions of higher education across the nation are looking for ways to incorporate collaboration and peer learning as well as to increase participation in their classes. As one of the processes underlying the competitive, rather than collaborative, nature of computer science [2], a defensive communication climate works against these goals.

As Gibb [5] points out, defensive communication is endemic to most traditional learning environments. To meet the needs of today's students, it will be necessary to understand the dimensions of the learning environment and to be cognizant of the ways in which our communication styles contribute to the creation and maintenance of the climate associated with our learning environments. In this article, we have presented some ways students and instructors speak and behave to unconsciously create and maintain defensive climates. Reflecting on typical communication practices found in the classroom is the first step to changing the culture and appeal of computer science.

References

[1] Burke, K. *Language as Symbolic Action*. Los Angeles, CA: University of California Press, 1966.

[2] Button, G. and Sharrock, W. (1996) Project Work: The Organization of Collaborative Design and Development in Software Engineering. *Computer Supported Cooperative Work: The Journal of Collaborative Computin*g, Vol. 5, (1996), pp 369-386.

[3] Derlega, V. J. *Self-Disclosure*. Newbury Park, CA: Sage, 1993.

[4] Fulton, R. D. A Conceptual Model for Understanding the Physical Attributes of Learning Environments. In R. Hiemstra, (Ed.) *Creative Environments for Effective Adult Learning*, San Franscisco, CA: Jossey-Bass, 1991, pp. 13-22.

[5] Gibb, J. R. Defensive Communication. *Journal of Communication*, Vol. 11-12, 1961, pp. 141-148.

[6] Gilligan, C. *In a Different Voice: Psychological Theory and Women's Development*. Harvard University Press, 1982.

[7] Hiemstra, R. (Ed.) Aspects of Effective Learning Environments. *Creative Environments for Effective Adult Learning*, San Francisco: Jossey-Bass, 1991, pp. 5-12.

[8] Tannen, D. The Display of (Gendered) Identities in Talk at Work. *Reinventing Identities: The Gendered Self in Discourse*, ed. by Mary Bucholtz, A. C. Liang, and Laurel A. Sutton, pp. 221- 240. New York: Oxford University Press, 1999.

[9] Vahala, M. E. and Winston, R. B. College Classroom Environments: Disciplinary and Institutional-Type Differences and Effects on Academic Achievement in Introductory Courses. *Innovative Higher Education*, Vol. 19, No. 2, 1994, pp. 99-121.

[10] Watzlawick, P., Beavin, J., and Jackson, D. *Pragmatics of Human Communication*, New York: W.W. Norton, 1967.