

(1989)

Writing in the Computer Science Curriculum

William J. Taffe

Why Writing Is Necessary

College students in career-oriented majors such as computer science have two curricular thrusts, professional studies and general education. Writing is an important component of each.

Led by the ubiquitous freshman composition course, writing has long been prominent in American general education. Following this introduction, the term papers usually assigned in literature and history courses build additional skill in a particular type of writing - the research paper. Recently, however, many academics have recognized that writing education cannot be isolated to a few courses and have advocated a much broader approach called "Writing Across the Curriculum." Ideas advocated by Writing Across the Curriculum proponents are beginning to influence Computer Science curricula.(1)

Unfortunately, in some science and engineering curricula, the importance of writing is not yet recognized. Perhaps it is because some faculty in technical curricula feel that writing should be taught in general education or that they are not competent to teach

writing. Perhaps it is because some technical students have more difficulty with writing than do humanities or social science students, suggesting that more, not less, writing instruction is needed. Perhaps it is because technical students sometimes tend to concentrate more on symbolic expression and less on the development of natural language. In addition, programming language skill development or communication with machines may crowd out the development of human-to-human communication and possibly lead computer science students to shortchange this facet of their education. Indeed, the stereotypical “nerd” is often portrayed as impoverished in written and oral human language skills.(2)

However, our computer science students need strengthened communication skills, not only for personal enrichment, but also for professional activity. Computer scientists must communicate with each other as clearly as with their machines. And, an extremely important Computer Science subdiscipline, the “man-machine interface,” clearly rests on a thorough understanding of human communication.

Modes of Writing in Computer Science Courses

There seem to be three categories which adequately describe most of the writing used in Computer Science courses: writing to develop facility with the specialized language of the discipline, writing to explain results of a study, and writing as a process for clarification of fuzzy ideas. The first two categories are the most common, but the distinction between them is frequently lost. This is unfortunate because they require different skills. A student’s lack of facility with the technical language can be mistaken by the instructor for an inability to organize thought; likewise, disorganized thinking may be passed off as merely a lack of writing skill.

The last category, clarification of thinking, although possibly new to computer science faculty, has important pedagogical potential. Teachers of writing recognize that the process of writing about a topic helps clarify the writer's thinking.(3) The simplistic model:

1. Collect all thoughts
2. Write them down

has yielded to the recognition of a feedback loop in the thinking-writing process. Attempting to express an idea often sharpens and clarifies the concept, frequently exposes lacunae in the thought chain, and possibly creates new questions. Thus, writing to clarify thinking may be an emerging tool for the Computer Science educator.

Developing Professional Language Facility

In a specialized discipline there are many new terms, phrasings, and modes of expression which have evolved to allow specialists to communicate more efficiently. When abused they create jargon, but their proper use is necessary if students are to fully join the professional community. This cannot be done passively; students must practice professional writing on a regular basis.

Since learning two things simultaneously is difficult, learning to use this new professional language and concurrently learning to organize professional material is often too large a first step for many students. For them, the first writing assignments in the discipline should be straightforward. An assignment I have used successfully is the writing of summaries.(4) Students are asked to choose an article from a recent issue of a technical journal and summarize it. Although students must be able to abstract essential points, the original article generally provides the organization for the summary and examples of using the professional language. Students learn to express themselves professionally by mimicking

professionals in their use of specialized terminology. Plagiarism is clearly a concern and needs to be discussed with the students straightforwardly. Happily, this assignment also has several beneficial side-effects which are described in the reference cited.

Learning to Organize and Present Professional Results

A second level of technical language skill is organization and presentation of the results of a study. In the natural sciences, this is often a laboratory report. In Computer Science, it can assume several formats and is currently used in a variety of courses.(5) In the laboratory portion of my Computer Architecture course, I generally give an experiment which is somewhat open-ended and ask for a standard laboratory report. In addition to specialized language and organizational skills, students learn to combine text and graphics to explain their results.

This term in Computer Graphics, I am attempting a similar approach through an open-ended programming project. The students are asked to develop three “typeface characters” in two different fonts and examine various problems associated with rendering these fonts on a computer screen. They are asked to explain what they did, what problems resulted, and how the problems were (or weren’t) overcome.

Systems analysis courses are “naturals” for writing, and several authors have described the written assignments given to their students.(6) The discipline demands written materials of varying types, such as user questionnaires and diaries, formal specifications, project correspondence, system documentation, requests for proposals or quotations, the final report, and standards for the analysis process itself. Technical writing is crucial for the systems analyst and “Systems Analysis and Design” courses often require the preparation of technical material.

Because oral communication skill is equal in importance to its written counterpart, Computer Science students also need opportunities to enhance their presentation skills.(7) In our curriculum several courses present the occasion for brief oral reports, but the best opportunity to practice presentation skills is in the required senior-project course, “Directed Study in Computer Applications.” In a multi-presentation colloquium at the semester’s end, each student presents a 30 minute project report to an audience consisting of the department faculty and fellow seniors. Faculty coach students in the preparation of their talks, showing them how to develop effective presentations.

Development of Thinking

Earlier I described the feedback loop that exists in the thinking-writing system. The writing process forces the writer to clarify thinking by exposing the holes in a progression of ideas and frequently raising new questions. Presenting an idea improves the idea, a concept familiarly expressed through the teachers’ adage “the best way to learn a subject is to teach it.” Computer Science faculty are beginning to use writing to help their students understand Computer Science better.

Assignments which require students to express their thinking about problems and concepts may help the student sharpen their understanding of concepts.(8) Getting students to carefully pose questions about the subject may lead them to think about the answers. I have assigned as a homework problem, “Write a potential question for the next hour exam, and explain what this question measures about knowledge of the subject.” In addition to the wonderful side-effect of giving me some great exam questions, answering this question helps students focus and clarify their own knowledge.

My colleague Peggy Eaton formalized this approach last spring during her “Organization of Programming Languages” course. She broadened the traditional concept of the course notebook by requiring her students to keep a Programming Languages Journal. The journal contained lecture notes, but also notes taken while studying. More importantly, in the journal students wrote down concepts or ideas they didn’t understand. In the process of explaining what confused them, they often removed the confusion, and if not, they had a well-focused question for class discussion. Writing helped students learn Computer Science.

Summary

Writing is both an end and a means. Computer Science students need to write to communicate, and professional writing must be taught in Computer Science courses as a continuation of the more general writing instruction of general education courses. But also, through the process of writing, writers are forced to clarify their thinking. By this means students have an additional tool for learning Computer Science.

Notes

- (1) Hartman, Janet D., “Writing to Learn and Communicate in a Data Structures Course,” *SIGSCE Bulletin*, February 1989: 32-36.
- (2) For example, see the Dick Tracy comic strips for the weeks of February 25 and March 5, 1989.
- (3) For example, *New Directions for Teaching and Learning: Teaching Writing in All Disciplines*, Jossey-Bass, San Francisco, 1982.
- (4) Taffe, William J. “Teaching Computer Science Through Writing.” *SIGSCE Bulletin* June 1986: 82-83. See also Flaningam, Dona Lee, and Warriner, Sandra “Another Way to Teach Com-

puter Science Through Writing.” *SIGSCE Bulletin* September 1987: 15-16.

(5) See Quirk, James R. F. “Teaching Computer Networks and a Writing Intensive Course.” *SIGSCE Bulletin*, June 1988: 30-35, and Brown, Dale A., “Requiring CS1 Students of Write Requirements Specifications: A Rationale, Implementation Suggestions, and a Case Study.” *SIGSCE Bulletin*, February 1988: 13-16.

(6) For example, Jordan, Donald L., “Integrating Desktop Publishing into a Systems Analysis and Design Course.” *SIGSCE Bulletin*, February 1989: 74-77.

(7) Cote, Vianney. “Teaching Oral Communication in Computer Science.” *SIGSCE Bulletin*, June 1987: 58-60.

(8) Hartman, p. 34, The microthemes on “supporting a thesis” and “quandary posing.”



(1997)

Did I Really Write That? A Retrospective Introspection

It was a good experience to reread my thoughts expressed in that first issue of *The PSC Journal of Writing Across the Curriculum*. As I read, I began to ask myself “do I still believe what I said?”, and “do I still practice what I preached?” Well, yes and no. Yes, I still believe it, but I’ve “backslid” a bit.

Thinking about the uses of writing, and its importance in the learning process, makes me remember some of the successes my students have had in developing their expressive abilities and cognitive capacities through that written expression. Early in the

semester, many students in my senior-level “W” course in computer design express disbelief at the writing assignments, a disbelief that evolves into dismay when they discover that I’m serious about writing. However, as the semester progresses, little-by-little they grudgingly tend to accept the notion that as computer professionals they will have to write proposals, reports, documentation and other forms of tangible instantiations of their ideas, and they begin to put some effort into writing clearly and expressively. Some even begin to enjoy the challenge of saying something clearly, though they usually won’t admit it. As the semester progresses, I see the evidence of their efforts. Gradually, the papers begin to become coherent, then a bit polished, and by semester’s end, some are even writing like professionals. They have it in them; it just needs to be induced to come out.

But there’s a price--and I pay a large part of it. The steady pressure required to persuade students to do what they often insist is irrelevant, useless, and is perhaps even an “unnatural act” for a computer scientist, takes its toll. Consoling the student who gets a rejection slip (“This writing is not at an acceptable level for a college senior - rewrite.”) demands a lot of physic energy. Working with students who have weak organizational skills on documents where text, tables and graphics need be coordinated into a coherent package can be exhausting. I read, in this Journal, suggestions from colleagues about “how to read papers without having to read papers” but I haven’t mastered the art. I still find reading papers to be labor intensive, hard work.

So, I’ve backslid a bit. I still give writing assignments, but I give them less frequently. And sometimes I tend to refine the assignments so that the writing “fits a template,” allowing students to slide through with less thinking. And at times I accept a still weak third draft of a paper just so that I don’t have to read it once again. I hear President Wharton say, “We must raise our expectations of performance for students ...” but I also hear myself ask

“what is reasonable to expect from a generation that doesn’t read?”

So it was good to reread what I wrote eight years ago. I am pleased that after almost a decade, I really have no argument with myself on this matter. But I do have to think about my ideals and what I’m doing to reach them. I need to remember the fervor I (we) had when “writing across the curriculum” was new to PSC, and ask how to rekindle it. I should concentrate on the successes and not on the obstacles that arise as surely as potholes in the roads of spring. It was good to see my reflection in the mirror.