Why Computer Science Students Need Language

Theresa Beaubouef
Department of Computer Science
Southeastern Louisiana University
Hammond, Louisiana 70402 USA
tbeaubouef@selu.edu

Abstract

Many students enter the field of computer science with misconceptions about the importance of communication skills. They often choose this field, thinking they will end up with jobs working alone or with other "techies" developing computer games, and not having to deal with people. These students often do not realize the significance of reading, writing, and speaking skills in computer science. This paper discusses several relevant areas of computer science, and explains why computer science students need skills covered in English, speech, technical writing, and even foreign language courses in order to achieve success as a computing professional.

1. Introduction

Computer Science is an interesting field. It integrates knowledge and skills from several areas of study in addition to the fundamental computer science courses. Because the field has so many diverse specializations, coursework in the areas of biological, chemical, or physical sciences may be appropriate for those individuals who will eventually work with scientists, engineers, or medical specialists. Advanced study in accounting, finance, and economics will prove useful for those working with business applications. No matter what the area of specialization or application, a good understanding of mathematics is required, and a previous work [1] discusses the importance of math for computer science students and the reluctance of many students to take these courses seriously. It is also imperative that students develop strong communication skills, which are vital to almost all professions in computer science. Therefore, courses such as English, technical writing, speech, and a foreign language are typically required for computer science majors. These courses are not simply for providing humanities courses for rounding out the curriculum. They offer skills that are critical for success in any profession, especially computer science.

Communication skills will determine a graduate's likelihood of landing a position in the first place. Job candidates with spelling and grammatical errors on their resumes and cover letters are often perceived as careless at best, undereducated at worst. Neither is a desirable trait for a candidate. A computer science graduate may obtain a position with a company based on his or her degree and computer skills. However, it is usually that person's ability to communicate effectively that will determine career advancement and potential for success.

This paper, therefore, discusses the issue of communication as it relates to typical positions and activities associated with the computer science profession. By pointing out the importance of language skills in all areas of computer science, I hope to better prepare students who are considering computer science as a major, and to emphasize that mastery of verbal and written communication is critical to success in computer science.

2. Requirements Specification

Requirements specification is usually considered the most important step in the system development process. Incomplete, incorrect, or ambiguous requirements lead to many problems and subsequent re-work. A good requirements specification is therefore critical to the development of a large system. However, just what goes into developing these requirements? What skills are necessary? Basically, it involves two things: an understanding of technology and competence in communication. By far, the more important of these is communication.

First of all, the system analyst must communicate effectively with end users, management, programmers, and non-technical personnel in order to fully understand the current system and the changes necessary for the new system [4, 7]. Verbal skills are used in individual interviews, focus groups, presentations, and joint application development (JAD) sessions. Written skills are used in developing surveys, which serve as an information gathering tool, and for general correspondence such as business letters and memos. The analyst must be able to listen well, to speak well, and to write well.

The world is getting smaller, and many systems are developed for companies with international interests and/or locations. Fluency in a foreign language will prove useful for communicating with users in such a system, and in identifying requirements for the finished product that relate to languages used in locations that differ from the local language. (For the purpose of illustration in this paper, assume that the native language of the student is English.) Again, the ability to communicate effectively in English, as well as a foreign language is a skill to be desired.

Finally, a requirements specification is a formal written document. It serves as a contract between those paying for the system and those designing and developing it. It must be clearly written so that all parties involved understand exactly what needs to be accomplished and when the work should be completed. Without a clearly written specification defining the scope of the project, features tend to be added, unspecified or ambiguous details lead to confusion, and tasks within the project become delayed. Excellent technical writing skills are a must for developing a working requirements specification for any significant project.

3. Problem Analysis and Programming

Humans are constrained to a certain extent by language; any thinking or problem solving done by a person is mapped onto constructs of his native language, which he uses in understanding a problem and describing a solution to others (or to a computer). A person who is fluent in more than one language can actually "think" in more than one language. When a person studies other languages, therefore, he becomes more aware of this fact, leading to a higher level of thinking.

There are many additional reasons that computer science students should study a foreign language. Often when a student studies a foreign language, he also becomes better at his native language. Learning language concepts such as verb tense in Spanish, for example, reinforces what the student knows about tenses in English.

Additionally, when one learns a second programming language, certain constructs may be similar those in the first language learned. For example, if a student has learned the semantics of an **if** statement in C, he may easily relate to the **if** statement in Java when seen for the first time. Everything the student learns in Java, then, reinforces understanding of the constructs learned in C, the first language learned. Finally, when a senior-level programming languages course is taken, these language constructs are abstracted, and the student can think in terms of selection structures, and focus more on semantics rather than syntax.

The study of programming languages [3, 5] also involves natural language. Formally, programming languages are expressed mathematically, using some notation such as Backus-Naur form. Definitions are precise, and abstraction is necessary for describing the syntax and semantics of a language. A student who is well versed in formal English will understand how words, clauses, phrases, and sentences fit into the correct grammar of the English language. He will be familiar with parts of speech, subject/verb agreement, verb tenses, etc., and be better able to understand that a programming language also has "parts of speech" and rules for connecting primitives in the language correctly. Moreover, the writing of compiler software also requires this knowledge since compilers are programs written to take code written in some programming language and convert it into machine language. Knowledge of the grammar and rules of a foreign language reinforces understanding of all these concepts.

Having gained insight into the expressive power and limitations of different natural languages gives a person a broader understanding of the need for different means of expression. This supports the idea that it is beneficial to be fluent in several programming languages as well. If a programmer can "think" in more than one programming language, he can better choose the one that more accurately models the data and programming needs of the application.

In order to solve problems on the computer, programmers must understand user requirements, develop solutions, and then convert those solutions into code using a programming language. Communication is necessary for understanding and clarifying requirements with others, in working with other systems analysts and programmers individually and in groups to develop code and test solutions, and in explaining intermediate and final results to both technical and non-technical personnel through code documentation, training sessions, and printed user's guides.

4. Documentation

Documentation is necessary for describing the purpose of the code, along with recording administrative and technical details such as programmer names, dates of code modification, assumptions, conditions, and system requirements. Large systems, in particular, require well written and complete code documentation. Documentation requires excellent writing skills. It is performed at almost every stage of code design and development, and continues long after the initial completion of the code to serve as reference, and as a record for code amendments. Code documentation should clearly state everything a programmer needs to know about code should there be a need to understand and modify it. Non-programmers should also easily understand it. Often management personnel or code librarians must work with code printouts or electronic copies. These people do not need to know much about code. They only need to know what it is, who worked on it, and most importantly, what it does.

5. Training and Maintenance

Once a system has been put in effect (and usually also before) users must be made familiar with its operation. Many types of people including high-level management, sophisticated users and programmers, as well as naïve end users must be informed about the part of the system that concerns them.

Communication with the customer or upper management often involves formal presentations highlighting the non-technical aspects of the system and discussing issues related to budget and decision making. The presenter must have good speaking skills in order to maintain the attention and respect of the audience. He must also be aware of the level of expertise of those in the audience, so as not to bore experienced personnel with simple details, nor confuse inexperienced personnel with technical jargon and unnecessary low-level system details.

Public speaking is easier for some people than for others, but everyone can and should improve his or her speaking skills. A good speech course can help students learn to prepare various types of speeches and presentations, and provide much needed experience at speaking before a live audience.

It is also important for professional presentations, that written material appearing on slides or handouts be well-written and free from grammatical and other types of errors. Good proofreading skills and the ability to recognize and correct these mistakes before the presentation takes place can save the speaker from embarrassment. Students can develop these skills in English and technical writing courses. They can also strive to improve both written composition and speaking skills by embracing these types of assignments in all courses, rather than simply complaining about them, and doing the minimum required.

6. Artificial Intelligence

Artificial Intelligence (AI) encompasses many areas of study, including such things as computer vision, image understanding, and planning. It also includes research in the areas of knowledge representation, voice recognition, natural language processing, and computational linguistics. These areas are often associated with human intelligence, and according to Luger and Stubblefield:

> "One of the long-standing goals of artificial intelligence is the creation of programs that are capable of understanding human language. Not only does the ability to understand natural language seem to be one of the most fundamental aspects of human intelligence, but also its successful automation would have an incredible impact on the usability and effectiveness of computers themselves." [6]

Artificial intelligence experts also explain that computer scientists can develop computer languages, but that they "do not have the luxury of defining natural languages," but must study them in form in which they naturally occur [2].

In areas of AI dealing with knowledge representation and natural language, there are various classes of content words such as noun, verb, adjective, and adverb. There are also function words such as preposition, determiners (the, that), connectives, and quantifiers (all, many, some, none). It is important to understand the structure of phrases and clauses. It is also necessary to understand how various types of words, phrases, and clauses are put together based on the rules and structure of the grammar of the language to produce a meaningful construct, something that is more than an arbitrary collection of words. Competence in English is mandatory for this type of artificial intelligence work.

In this modern world, people often use software packages without being aware of the artificial intelligence involved in their development. These include programs that translate from one language to another, database information retrieval systems, spelling/grammar checkers, human/computer interaction interfaces, automatic dictation taking software, and spoken language control systems such as automated telephone service. None of these systems could have been developed without a mastery of language and communication. It is important to note moreover, that many of these systems actually do involve some type of computer system communicating with real people.

7. Research

Most of this paper has dealt with the essentials of communication skills for students graduating in computer science and entering the professional world. These jobs relate to developing products for customers, and it has been shown that communication skills are essential qualities in a successful computer science professional. What about those students who choose to continue education, those students who plan to perform research in computer science?

Research in any field requires good communication skills. First of all the researcher must be adept at reading and understanding written documents and papers. He must be able to review critically these papers and to envision and develop new ideas and designs. Finally, the culmination of all research is to be able to advance the field. In order to do this, the results must be published and/or presented to other researchers. In summary, for research to be effective, it has to be communicated to others. This requires that results be presented at some technical conference or symposium, or published in a journal or proceedings as a scientific paper. The successful researcher must be competent in both composition and speaking skills.

8. Conclusion

The computer is far from intelligent. It can do only simple things. It can store a value, it can perform basic arithmetic operations such as addition, and it can compare two values. At a low level, everything about a computer is related to numbers, arithmetic, and logic.

Real world applications, however, involve people, products, and systems for science, business, and industry that deal with real world problems. System users, managers, and scientists know how to state a problem and outline a solution in English, sometimes enhanced with complicated mathematical formulas. They do not typically have the knowledge necessary to develop hardware or software to solve the problem. That is the job of the computer scientist. This person acts as a go-between for the customer and the computer. To solve correctly a problem, he must first understand it well enough to put it in computer form. Then he must be able to explain the computer solution back to the customer. Communication is the key.

Is it any wonder then that computer science, the study of how to use a machine based on simple binary numbers to solve real world problems, requires a good understanding of language?

References

[1] Beaubouef, T., “Why Computer Science Students Need Math,” *SIGCSE Bulletin (inroads)*, vol. No. , pp. , December 2002.

[2] Dean, T., Allen, J., and Aloimonos, Y., *Artificial Intelligence: Theory and Practice, Addison-Wesley Publishing Co*., Menlo Park, CA, 1995.

[3] Hopcroft, J. and Ullman, J. *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley Publishing Co., Reading, MS, 1979, p1.

[4] Kendall, K. and Kendall, J. *Systems Analysis and Design*, Prentice Hall, New Jersey, 1999.

[5] Linz, P., *An Introduction to Formal Languages and Automata*, Jones and Bartlett Publishers, Boston, 2001.

[6] Luger, G. and Stubblefield, W. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, Benjamin/Cummings Publishing Company, Redwood City, California, 1993.

[7] Whitten, J. and Bentley, L. *Systems Analysis and Design Methods*, Irwin McGraw-Hill, Boston, 1998.

Call for Manuscripts

Computer Science Education Journal

A scholarly journal devoted to the
improvement of teaching and learning effectiveness
for students of computer science
at the college and university level

Seeks to publish manuscripts documenting
computer science educational research and practice

Editors

Sally Fincher
University of Kent
Canterbury, United Kingdom
<S.A.Fincher@ukc.ac.uk>

Renee McCauley
College of Charleston
Charleston, South Carolina, USA
<mccauley@cs.cofc.edu>

Details at

<http://www.swets.nl/sps/journals/cse1.html>