CRITICAL THINKING AND COMPUTER SCIENCE: IMPLICIT AND EXPLICIT CONNECTIONS

Barry Fagin, Jim Harper, Leemon Baird, Steve Hadfield and Ricky Sward
US Air Force Academy Dept of Computer Science
2354 Fairchild Drive USAFA, CO 80840
719-333-3590
{firstname.lastname}@usafa.af.mil

ABSTRACT
Critical thinking is an essential skill for an educated society. Our experience as computer science educators in an environment with an explicit emphasis on critical thinking has led us to investigate the connection between the two. We describe examples of how critical thinking skills can be developed throughout the computer science curriculum, and suggest future avenues where the connection between critical thinking and computer science could be fruitfully explored.

1. INTRODUCTION
Critical thinking abilities are considered a cornerstone of academic maturity and a trademark of a well-educated person. They are particularly important at our university, a military institution where critical thinking skills are an explicitly desired educational outcome.

Our experience as computer scientists charged with developing critical thinking skills in our students has led us to spend some time examining the connection between computer science and critical thinking. This paper discusses how progress through the discipline of computer science can develop critical thinking, and how to assist that development through carefully chosen exercises. We believe the connection between CS and critical thinking is stronger than the literature would suggest.

2. CRITICAL THINKING
Paul [7] describes critical thinking as an intellectual discipline for examining information and determining validity. It is based on universal intellectual values that transcend divisions of subject matter. Three key parts of critical thinking are clarity, accuracy, and relevance.

Clarity is crucial to understanding the information received. Questions such as "can you elaborate on your comment" can determine if the information is clear. *Accuracy* probes the gaps between the information and factual reality. Questions like "how can we find out if that is really true" can help determine accuracy. *Relevance* helps ensure that the information received is pertinent. A skilled critical thinker can quickly distinguish relevant facts from red herrings.

Critical thinkers are also skeptical [5]. Every commander knows the poisonous effect rumor can have on an organization. Skepticism and critical thinking are the best antidotes to rumor, particularly in the internet age [2]. This makes the development of critically thinking officers vitally important to the nation.

3. CRITICAL THINKING IN MILITARY EDUCATION

It is a common perception outside the armed forces that the military does not value critical thinking[1]. In fact, the opposite is true.

One of us [Harper] taught critical thinking in the Air Force ROTC Program. AFROTC course materials on critical thinking emphasize the inculcation of habits of thought and action: The desire for clarity, accuracy, and relevance, the ability to detect logical fallacies, and the examination of implicit assumptions.

These skills are vital to officers because of the unique nature of the profession of arms. As part of normal day-to-day activities, members of the military may have in their possession weapons of considerable lethality. Even officers without direct access to weapons systems may wield considerable power and influence, particularly as computer scientists where they may be network administrators or information warfare specialists.

The importance of a functioning chain of command in wartime, combined with the societal need for confidence that military force will be used only when authorized under lawful civilian control, create two opposing tendencies that officers must carefully balance. Lawful orders must be obeyed, but unlawful orders must not. Concern with this tension often emerges in popular culture[2].

Critical thinking skills enable officers to walk the middle path, and make them more likely to accomplish the mission. Officers who think critically are less likely to act on rumor, are more likely to give their superiors tough messages when they need to, and are more likely to make the right decision at the right time. They are also more likely to spend public money correctly [2], to generate and respond to criticism effectively, and to lead with wisdom and judgment. Officers are obligated to apply critical thinking to all tasks in the profession of arms.

4. CRITICAL THINKING IN THE CS CURRICULUM

Critical thinking skills can be cultivated in many other ways besides programming exercises, and at many other points in the curriculum besides the introductory courses. We discuss a few examples here.

Figure 2 shows a simplified view of the CS major at the Air Force Academy:

[1] Holsti, in [5], reported almost 80% of civilian leaders without military experience described military culture as "rigid". Only about 15% considered the term "creative" appropriate.

[2] See for example Columbia Pictures' "A Few Good Men" (1992) or Hollywood Pictures' "Crimson Tide" (1995).

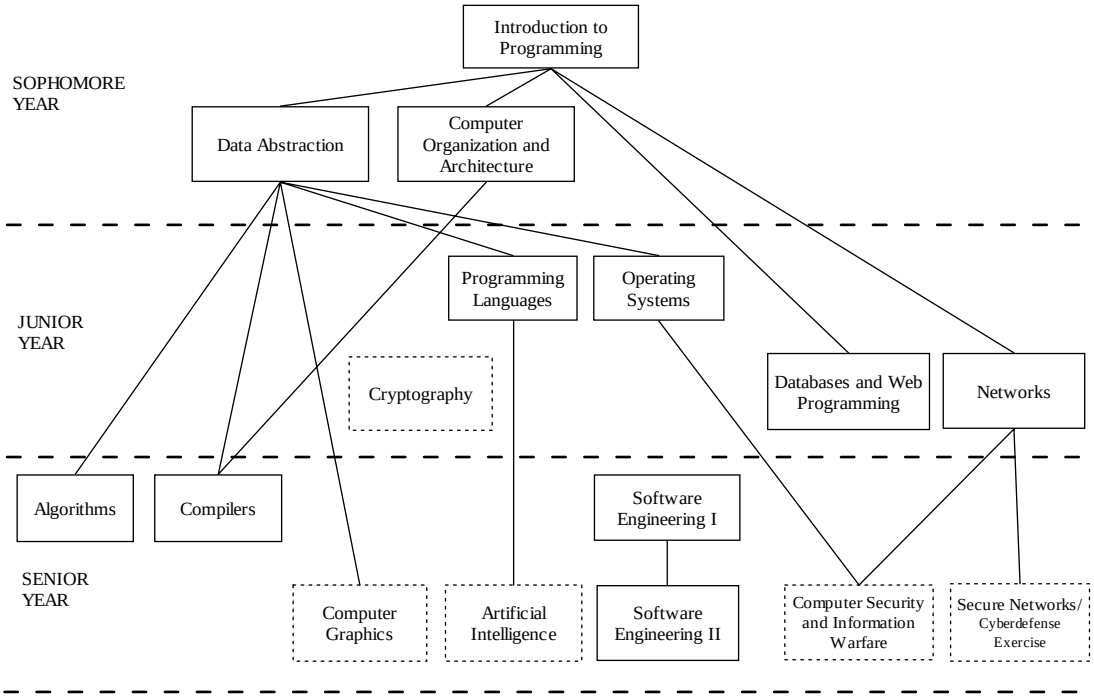

Figure 1: The USAFA computer science major

Lines indicate a prerequisite relationship from the course above to the course below[3]; electives are outlined in dashes. The program is CSAB/ABET accredited; most readers at 4-year undergraduate institutions should recognize where their corresponding courses fit in. In the sections below we offer a few examples of how we try to inculcate critical thinking using topics drawn from the course syllabus. We hope these exercises will show how critical thinking skills can be encouraged naturally throughout the CS curriculum, and encourage further work in this area.

5.1 Programming languages

Students in our program use Ada for their first and principal programming language [11]. Their exposure to other languages is normally as juniors in the programming languages course.

For their first C program, students write a simplified Ballistic Missile Defense discrete event simulator and then run it against scenarios using different engagement strategies. They are then asked to analyze the results and draw conclusions. In addition to the software development goals, our intent is to give them the opportunity to practice their analytical and critical thinking skills within a military context.

When the course turns to object oriented languages and toolkits, we discuss both the Java and .NET frameworks and ask probing questions regarding similarities and differences. The goal is to lead them down certain paths to see if they are asking the

[3] Software Engineering does not have formal prerequisites in the CS major because it is open to students in other disciplines. The only formal prerequisite is senior standing.

right questions to get the full perspective. Many times this discussion will lead to a critical review of open source versus strictly proprietary capabilities.

5.2 Operating systems

A typical OS course includes lessons on process scheduling, the standard process state machine model (e.g. blocked, waiting, running), and scheduling algorithms. This material can be difficult to teach, because students consider it uninteresting, and tend to miss the subtleties of how processes are handled.

Last fall, we tried a new approach using the game "foxes and hounds". Our version is played on a wraparound grid of arbitrary size, with one student playing the fox, and the rest playing hounds:

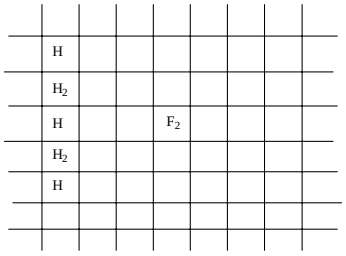

Figure 2: Foxes and Hounds

The hounds act as a team and try to catch the fox by moving into its square. The fox tries to evade capture for as long as possible. The fox and two of the hounds can move two spaces on each turn. We add a twist to the standard game by having the instructor roll a die. If the result is a 1, then the player that just moved must freeze. On a 6, all frozen players become active again.

This game maps well to the process management state diagram. Students decide the rules for when pieces are to move. Coming up with rules for a fair game is equivalent to coming up with a good scheduling algorithm. In the process, they discover how to think critically about process management issues.

5.3 Artificial intelligence

Krishna Rao [5] discusses the promotion of critical thinking in an AI course through the presentation of ill-defined problems and the use of experimental testing. We concur that this is important, and use programming assignments that do this. But AI also provides an excellent opportunity to promote critical thinking through the use of essay writing. We assign four papers to read, two “pro-AI” and two “anti-AI”, and require students to write essays on them. The papers are:

1) *Computing Machinery and Intelligence,* by Alan Turing [12]

2) *Minds, Brains and Programs,* by John Searle [10]
3) *The Godelian Argument,* by J.R. Lucas [6]
4) *Robots, Re-evolving Mind,* by Hans Moravec [8]

Typical questions from assignments include:

"What are potential arguments against Turing's thesis and how does he respond?"
"Describe the 'Chinese room' example."
"Are Searle's arguments sufficient to refute the possibility of thinking machines?"

Students approach these assignments with a common framework of questions. What is the author's claim? Are there unstated assumptions? Are there implicit value judgments? Where is the author not thinking clearly? Where are counterarguments not considered?

Feedback is extremely positive; students consistently rate AI as the most enjoyable class we teach.

5.4 Software engineering

Our CS "capstone" experience is a two-semester course in software engineering. It exposes students to a large problem they cannot solve on their own given existing time constraints.

The intent of this exercise is to force students to question their assumptions of software development. Class teams struggle with the organization and division of labor. They question, for example, what a program manager does day-to-day and why they even need one. Fortunately, since some of them may become program managers, they eventually come around.

Students also struggle when it's time for software integration, and learn valuable lessons about unit testing, integration testing, and delivery. They question the processes that they have been using and learn new approaches for the development of software. These are all lessons learned through the development and refinement of critical thinking skills.

5.5 Other issues in upper level courses

As the curriculum becomes more advanced and programming projects become more challenging, implicit assumptions that lead to errors in programs become both harder to identify and more important to discover. Typically, these assumptions concern complex components of a computer system (such as I/O devices, networks, and OS internals) to which students have only recently been exposed. In these cases, students are likely to implicitly adopt simplistic assumptions about component and user behavior that are not accurate descriptions of true performance. Student critical thinking skills are further developed during their upperclass years by requiring them to create correct software that tests these assumptions directly.

6. CONCLUSIONS AND FUTURE WORK

Our experience as computer science faculty at a military institution, where critical thinking is an explicitly desired outcome, has led us to carefully explore the relationship between CS and critical thinking skills. We have learned that these skills can be developed both implicitly, through the practice of questioning assumptions by debugging software, and explicitly, through the creation of assignments specifically designed to test critical thinking skills.

The rich connections between CS and critical thinking are worth exploring further. How might we assess the effect of an increased emphasis on critical thinking skills? Can educators develop a resource for CS exercises that promote critical thinking? Do the ideas suggested here generalize to high school CS education? The evidence suggestions the connection between CS and critical thinking warrants further attention from the computer science education community. We look forward to further work in this area.

7. ACKNOWLEDGEMENTS

The authors are grateful to Mike Collins of the National Security Agency for his assistance in the preparation of this paper.

8. REFERENCES

[1] Fagin, B. Technology and the values of a liberal education. *Academic Questions* (Fall 1999).

[2] Fagin, B., Baird, L. and Schweitzer, D. Skepticism and cryptography. Work in progress.

[3] Graham, L. and Metaxas, P.T. Of course it's true; I saw it on the internet! Communications of the ACM, 46,5 (May 2003) 71-75.

[4] Holsti, O. Of chasms and convergences: Attitudes and beliefs of civilians and military elites at the start of a new millennium. In *Soldiers and Civilians: The Civil-Military Gap and American National Security*, Feaver and Kohn eds, MIT Press, 2001, pp 15-101 . (ISBN 0262062232)

[5] Krishna Rao, M.R.K. Infusing critical thinking skills into content of an AI course. In *Proceedings of the 5th International Conference on Technology in Computer Science Education* (ITiCSE'05) (Monte de Capaica, Portugal, June 27-29 2005), pp 173-177.

[6] Lucas, J. The Godelian Argument. *Truth Journal*, 14 July 2002.

[7] Moore, B., and Parker, P. *Critical Thinking.* McGraw-Hill, 2003. (ISBN: 0072818816).

[8] Moravec, H. Robots, re-evolving mind. *Cerebrum*, 3,2 (Spring 2001) pp. 34-49 (invited by Walter Donway).

[9] Paul, R. *Critical Thinking: What every person needs to survive in a rapidly changing world.* Center for Critical Thinking and Moral Critique, Sonoma CA, 1990. (ISBN: 0944583040)

[10] Searle, J. Minds, brains, and programs. *The Behavioral and Brain Sciences*, vol. 3., 1980, Cambridge University Press.

[11] Sward, R., Carlisle, M. et al. The Case for Ada at the USAF Academy, In *Proceedings of the 2003 ACM SIGAda Conference (SIGADA-03) (San Diego CA, USA Dec 7-11 2003)*, pp 68-70.

[12] Turing, A.M., Computing machinery and intelligence. *Mind*, 59, pp 433-460, 1