

LATEST DEVELOPMENTS IN THE "KILLER ROBOT" COMPUTER ETHICS SCENARIO

Richard Gary Epstein
Department of Mathematics and Computer Science
West Chester University of Pennsylvania
West Chester, PA 19383
epstein@golden.wcupa.edu

Introduction

[When this paper was submitted to SIGCSE in August, the "Case of the Killer Robot" was a 100 page computer ethics scenario. Now, in December, it is nearly 200 pages and at least four more articles are planned. The expanded version has broader coverage of software engineering, ethics, professional issues and societal impact of computers. Consequently, the new version greatly expands the scope and intent of the first. I will describe the 100 page version in some detail and the new materials in lesser detail since the original reviewers of this paper did not know of the new materials.]

The "Case of the Killer Robot" is a computer ethics scenario that explores issues in software engineering and computer ethics. It is in two parts. Part I is the scenario itself. It uses the waterfall software process as a skeletal framework for presenting factors that played a role in the death of a robot operator. Part II consists of two radio talk show transcripts. The fictitious radio programs are devoted to an analysis of the transcripts of killer robot team meetings.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association of Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.
SIGCSE '95 3/95 Nashville, TN USA
© 1995 ACM 0-89791-693-x/95/0003....\$3.50

Fortunately, the team meetings were being recorded as part of a research project on team dynamics.

The primary purposes of this paper are (1) to make faculty aware of the availability of this scenario for their classes and (2) to explain the philosophy behind the construction of the scenario so that faculty can see the potential power of fiction as a device for teaching computer science and engineering.

Philosophy

The use of fictitious scenarios are the norm for text books on computer ethics (for example, see [1,2]). However, the "Case of the Killer Robot" differs from those that are normally used in that it explores a situation in very great detail, almost as one would expect from a novel. However, unlike a novel, the scenario is presented in the form of "articles" of various types. The scenario [i.e., the 100 page version] now consists of eight ordinary newspaper articles, one scholarly journal article, one newspaper magazine interview and two radio talk show transcripts. The latter are in a section that is completely separate from the scenario which is primary concerned with unfolding events along a historical time line. These twelve articles together present the facts in the case.

The "Case of the Killer Robot" is 100 pages in length, whereas ethics scenarios in a typical computer text book is 2 - 4 pages in length. What emerges from a 100 page scenario is the awesome complexity of software development, something that cannot be communicated in a brief scenario of two to four pages.

Another significant difference between a typical computer ethics scenario and the "Case of the Killer Robot" is that the latter is intended to teach software engineering technical material as well as computer ethics material. Writing the scenario has been an exciting learning experience and one of the major lessons that I have learned is that software engineering and computer ethics are inexorably intertwined. In part this is due to the lack of standard practices in software engineering. Fortunately, I was able to get one of my characters, computer ethics Professor Harry Yoder, to express this opinion for me. And that is fun because if people disagree with Professor Yoder, they can attack him instead of me.

The killer robot scenario introduces software engineering topics such as the software life cycle, CASE tools, object-oriented analysis and design, the psychology of programmers and software teams, the nature of requirements, user interfaces, and software testing. A lot of attention is paid to the challenge of teamwork and human interactions on a team because I feel that these are essential subjects for students who are going to be practitioners.

The scenario, of course, also discusses many ethics issues, including intellectual property and software theft, software reliability, honesty and trust, conflicts of interests, ethics of human communication, codes of ethics in computing, and tests of ethical behavior.

An important part of the philosophy behind the killer robot is to show that a programmer

who makes an error works in a complex web of relationships. The scenario illuminates these relationships more and more completely, article by article.

The use of fiction in CS education

One of the original goals of the killer robot scenario (as presented at SEI CSEE in 1994 [3]) was to use fictitious characters to exemplify ethical and professional dilemmas and to express professional conflicts. The cast of characters in the killer robot is at least 30 (much more in the newer 200 page version). This is a complex scenario, but a real software project involves many people and impacts upon many more.

The writing process for me has been a matter of absorbing new knowledge about software engineering, computer ethics and social implications of computing, and giving it vivid expression (hopefully) in the form of fictitious characters. For any ethical issue in computing, there are nearly limitless possibilities for creating characters that embody that dilemma.

I do not have proof at this time, but I believe that students will get much more out of a killer robot sort of presentation of computer ethics and professionalism than a completely dry, formal discourse. Ethical dilemmas, when they occur in real life, are very vivid. They are the stuff of life and they always involve other people. I should mention, however, that the killer robot should be used with other reading materials that expand upon the technical and/or ethical issues that are raised.

Fiction allows one to juxtaposition characters and world views that are important in reality but that are rarely juxtaposed in Computer Science education. For example, computer ethics Professor Harry Yoder is a blend of an ordained minister and a Computer Scientist. Because he comes in from a somewhat different sphere, he can bring in new

perceptions that might be missing if everyone in the scenario were a card-carrying member of the ACM. (Actually, Harry Yoder is a member of the ACM, but I hope you get my point.)

In the new materials that I have recently generated, the power and potential of fiction in Computer Science education really impressed itself upon me. One question I am constantly asking is, "What do the students need to know when they go out into the real world? What can I offer to them in the way of helpful information?" Part of my agenda is perhaps that the university education is not doing all it should be doing to prepare people for real life, which is mostly about human relationships and doing the right thing. I am also trying to illuminate the creative process itself and suggest to students that new forms of creativity are always available in our field.

Fiction is a great device to help one envision the future and to imagine new concepts and even applications. One of the new articles [in the 200 page version] is about software maintenance. It takes the form of a human interest newspaper story in which the reporter is visiting the lab of Professor Cleareye, a Native American whose shaman father taught him how to "see" one starry night on Elk Mountain. He applied his ability to "see" to proving theorems about concrete objects in seven dimensional space, but when that ability faded away, he became a Computer Scientist.

Professor Cleareye developed a process visualization lab and his graduate students are developing remarkable new systems that allow one to see the properties of computer systems. As we tour the lab we see visual representations of the concepts of coupling, cohesion and information hiding for function-oriented systems, and also coupling for object-oriented systems. We learn, of course, that the killer robot software was tightly coupled and lacked cohesion and that the vision processing part of the system,

which uses an object-oriented language, is nearly impossible to maintain. Reading this, students might anticipate a day when a Computer Scientist will be able to look at the visual representation of a system and on that basis decide whether it is well-designed or not. The new materials are influenced by many articles read over the past six months. For example, some of the inspiration for Professor Cleareye's lab came from the article "Maintaining Object-Oriented Systems" by Wilde et al. [4]. The main borrowing from the Wilde et al. paper is that visual representations of object-oriented systems are really needed to improve the maintenance process.

Outline of the Case of the Killer Robot

Here is a brief accounting of the ten articles that comprise the scenario part (part I) of the "Case of the Killer Robot" and *some* of the issues that they raise. This is an outline of the 100 page version and not the more recent 200 page version. The second part (consisting of the radio talk show transcripts) is not described, due to limitations of space. Each article is listed with its type, a summary of its "story line", a list of technical issues it touches upon and a similar list of ethical issues. New articles that have been added since the SEI Conference in January 1994 are marked with asterisk (***) .

Silicon Valley Programmer Indicted for Manslaughter (newspaper article)

Story line: Randy Samuels, programmer for Silicon Techtronics, is indicted for manslaughter in the death of Bart Matthews, robot operator. *Technical issues:* scenario draws upon two historical incidents that were reported in SIGSOFT notes as a starting point for this fictitious scenario. *Ethics issues:* Should a programmer be held legally responsible if his or her code causes bodily injury or death to a human being?

Developers of "Killer Robot" Worked Under Enormous Stress (newspaper article)

Story line: The killer robot project was already far behind schedule when management hired twenty new programmers to speed things up. The project leader, Ray Johnson, expounds his "Ivory Snow Theory" of software reliability: 99 44/100% pure is pure enough. *Technical issues:* human relationships, conflicting software philosophies, mythical man-month, just how reliable does software have to be? *Ethical issues:* Just how reliable does software have to be? Competence.

"Killer Robot" Programmer was Prima Donna, Co-Workers Claim (newspaper article)

Story line: A software psychologist analyzes Randy Samuels' personality and discovers that he is task-oriented, like everyone else on his team. Various team members are interviewed to get a more complete picture of his personality and of the team dynamics on the team. *Technical issues:* Programmer psychology, personality types, egoless programming, team organizations, properties of successful versus unsuccessful teams. *Ethical issues:* Implication of abusive verbal behavior.

"Killer Robot" Project Mired in Controversy Right from Start (newspaper article)

Story line: A dispute arose on project concerning the use of waterfall model versus prototyping. *Technical issues:* software process models, waterfall model, prototyping model, role of institutional inertia in decision making. *Ethical issues:* Awareness of one's own technical limitations, is it ethical to ignore the end-user in product development?

***** Fallen Project Director Accused of Conflict of Interest in Killer Robot Case** (newspaper human interest story)

Story line: This article exposes the important role of the original project director, who was killed in a sky-diving accident. In a blatant conflict of interest he

succeeds in getting Silicon Techtronics to use the Sheol CASE tool, a product of a company he partly owns. His insistence on moving over to o-o technology is a disaster and his death left the project in a shambles. *Technical issues:* OOA/D, CASE tools, use of untested, new technologies, transitions to new technologies, need for project librarian. *Ethical issues:* Conflicts of interest, alcoholism and irresponsible behavior in the workplace

The "Killer Robot" Interface (paper in academic journal, although not a good one)

Story line: Professor Gritty argues that the design of the user interface was ultimately responsible for the disaster. He explains Shneiderman's Eight Golden Rules [5] in some detail and shows how the killer robot interface violated each and every rule. *Technical issues:* user interfaces and software quality, Shneiderman's Eight Golden rules, computers in workplace and ergonomics.

Ethical issues: Computers in the workplace

Silicon Techtronics Promised to Deliver a Safe Robot (newspaper article)

Story line: A group of programmers come to Randy Samuels' defense by making public the requirements document that served as a legal contract between Silicon Techtronics and Cybernetics, Inc. (the customer). In so doing they explain the nature of requirements and the manner in which a requirements document might handle exceptional conditions.

Technical issues: Nature of a requirements document, functional and non-functional requirements, user and operator training.

Ethical issues: Duties and obligations with respect to a document of this nature.

***** Software Engineer Challenges Authenticity of "Killer Robot" Software Tests** (newspaper article)

Story line: Wesley Silber, Professor of Software Engineering, discovers a discrepancy between documented software

tests and the robot's actual behavior. In so doing, he explains the nature of software testing. *Technical issues:* Software testing, white box vs. black box testing, static vs. dynamic testing, unit vs. integration testing, version management. *Ethical issues:* Just how reliable does software have to be?

***** Silicon Techtronics Employee Admits Faking Software Tests** (newspaper article)
Story line: Cindy Yardley is exposed as the software tester that faked the software tests, acting at the request of her boss, Ray Johnson. These and other goings on at Silicon Techtronics are exposed by Max Worthington, corporate security officer, who had the job of monitoring employee e-mail.
Technical issues: Just how reliable does software have to be? *Ethical issues:* whistle blowing, just how reliable does software have to be, data privacy, software theft, faking software tests, codes of ethics.

***** A Conversation with Dr. Harry Yoder** (Sunday newspaper magazine article / interview)
Story line: Dr. Yoder presents his thesis that a corporation is a social organism, and as such, the leaders of that organism bear a special responsibility for what has transpired.
Technical issues: Corporation as organism.
Ethical issues: Techniques for making ethical decisions, legal vs. ethical, accountability and responsibility in software failures.

Summary

The killer robot scenario is now being used to teach courses on software engineering and computer ethics. It is also being used at several non-academic sites to teach software professionals about professionalism. It is also being used in a course on ethics (in a Department of Philosophy) and a course on global technology (in a Department of Psychology). The purpose of this paper was to introduce the scenario to a wider audience and to encourage people to consider the

power of fiction in teaching ethics and values within a technical curriculum.

References

1. Forrester, T. and Morrison, P., *Computer Ethics: Cautionary Tales and Ethical Dilemmas in Computing*, MIT Press, Cambridge, MA 1990.
2. Kallman and Grillo, *Ethical Decision Making and Information Technology*, McGraw-Hill, New York, NY, 1993.
3. Epstein, R., "The Use of Computer Ethics Scenarios in Software Engineering Education: The Case of the Killer Robot", Seventh SEI CSEE Conference, San Antonio, TX, Lecture Notes in Computer Science, Number 750, Springer-Verlag, New York, 1994.
4. Wilde, N., Matthews, P., and Huitt, R., "Maintaining Object-Oriented Software", *IEEE Software*, January 1993, pp. 75 - 80.
5. Shneiderman, B. *Designing the User Interface*, Addison-Wesley, Reading, MA, 1986.