Silicon Valley Programmer Indicted for Manslaughter - 1

Program Error Caused Death of Robot Operator

Special to the SILICON VALLEY SENTINEL-OBSERVER, Silicon Valley, USA

Jane McMurdock, Prosecuting Attorney for the City of Silicon Valley, announced today the indictment of Randy Samuels on charges of manslaughter. Samuels was formerly employed as a programmer at Silicon Techtronics, Inc., one of Silicon Valley's newest entries into the high technology arena. The charge involves the death of Bart Matthews, who was killed last May by an assembly line robot.

Matthews, who worked as a robot operator at Cybernetics, Inc., in Silicon Heights, was crushed to death when the robot he was operating malfunctioned and started to wave its "arm" violently. The robot arm struck Matthews, throwing him against a wall and crushing his skull. Matthews died almost instantly in a case which shocked and angered many in Silicon Valley. According to the indictment, Samuels wrote the particular piece of computer program which was responsible for the robot malfunction.

"There's a smoking gun!," McMurdock announced triumphantly at a press conference held in the Hall of Justice. "We have the hand-written formula, provided by the project physicist, which Samuels was supposed to program. But, he negligently misinterpreted the formula, leading to this gruesome death. Society must protect itself against programmers who make careless mistakes or else no one will be safe, least of all our families and our children," she said.

The Sentinel-Observer has been able to obtain a copy of the hand-written formula in question. Actually, there are three similar formulas, scrawled on a piece of yellow legal pad paper. Each formula describes the motion of the robot arm in one direction: east-west, north-south and up-down.

The Sentinel-Observer showed the formulas to Bill Park, a Professor of Physics at Silicon Valley University. He confirmed that these equations could be used to describe the motion of a robot arm.

The Sentinel-Observer then showed Professor Park the program code, written by the accused in the C programming language. We asked Professor Park, who is fluent in C and several other languages, whether the program code was correct for the given robot arm formulas.

Professor Park's response was immediate. "By Jove! It looks like he misinterpreted the y-dots in the formulas as y-bars and he made the same mistake for the x's and the z's. He was supposed to use the derivatives, but he took the averages instead! He's guilty as hell, if you ask me."

The Sentinel-Observer was unable to contact Samuels for comment. "He is deeply depressed about all this," his girlfriend told us over the phone. "But, Randy believes he will be acquitted when he gets a chance to tell his side of the story."

Developers of 'Killer Robot' Worked Under Enormous Stress - 2

Special to the SILICON VALLEY SENTINEL-OBSERVER, Silicon Valley, USA

by Mabel Muckraker

The Sentinel-Observer learned today that Randy Samuels and others who worked on the 'killer robot' project at Silicon Techtronics were under tremendous pressure to finish the robot software by January 1 of this year. According to an informed source, top level management warned killer robot project staffers that "heads would roll" if the January 1st deadline was not met. Randy Samuels, a Silicon Techtronics programmer, was indicted last week on charges of manslaughter in the now famous 'killer robot case'. Samuels wrote the flawed software which caused a Silicon Techtronics Robbie CX30 industrial robot to crush and fatally injure its operator, Bart Matthews. Matthews was a robot operator at Cybernetics, Inc. According to Silicon Valley Prosecuting Attorney Jane McMurdock, Samuels misinterpreted a mathematical formula, "turning harmless Robbie into a savage killer."

Our informed source, who wishes to remain anonymous and whom we shall call 'Martha' for the rest of this article, has intimate knowledge of all aspects of the Robbie CX30 project. Martha told the Sentinel-Observer that there was an enormous amount of friction between Robotics Division Chief Ray Johnson and the Robbie CX30 Project Manager, Sam Reynolds. "They hated each others' guts," Martha told the Sentinel-Observer in an exclusive interview.

"By June of last year the robot project had fallen six months behind schedule and Johnson went through the roof. There were rumors that the entire Robotics Division, which he headed, would be terminated if Robbie [the CX30 robot] didn't prove a commercial success. He [Johnson] called Sam [Reynolds] into his office and he really chewed Sam out. I mean you could hear the yelling all the way down the hall. Johnson told Sam to finish Robbie by the first of January or 'heads would roll'."

"I'm not saying that Johnson was ordering Sam to cut corners," Martha added. "I think the idea of cutting corners was implicit. The message was, 'cut corners if you want to keep your job'." According to documents which Martha provided the Sentinel- Observer, twenty new programmers were added to the Robbie CX30 project on June 12th of last year. This was just several days after the stormy meeting between Johnson and Reynolds which Martha recounted.

According to Martha, the new hirees were a disaster. "Johnson unilaterally arranged for these new hires, presumably by shifting resources from other aspects of the Robbie [CX30] project. Reynolds was vehemently opposed to this. Johnson only knew about manufacturing hardware. That was his background. He couldn't understand the difficulties that we were having with the robotics software. You can't speed up a software project by adding more people. It's not like an assembly line."

According to Martha and other sources inside the project, the hiring of twenty new programmers led to a staff meeting attended by Johnson, Reynolds and all members of the Robbie CX30 software project. "This time it was Sam [Reynolds] who went through the roof. He complained that the project didn't need more

people. He argued that the main problem was that Johnson and other management people did not understand that Robbie CX30 was fundamentally different from earlier versions of the robot."

These sources tell the Sentinel-Observer that the new hirees were not fully integrated into the project, even six months later, when ten Robbie CX30 robots, including the robot which killed Bart Matthews, were shipped out. According to Martha, "Sam just wanted to keep things as simple as possible. He didn't want the new people to complicate matters. They spent six months reading manuals. Most of the new hirees didn't know diddly about robots and Sam wasn't about to waste his time trying to teach them."

According to Martha, the June 12th meeting has become famous in Silicon Techtronics corporate lore because it was at that meeting that Ray Johnson announced his "Ivory Snow Theory" of software design and development. According to Martha, "Ray [Johnson] gave us a big multi-media presentation, with slides and everything. The gist of his 'Ivory Snow Theory' is simply that Ivory Snow is 99 and 44/100 per cent pure and there was no reason why robotics software had to be any purer than that. He stated repeatedly that 'Perfect software is an oxymoron.'"

Martha, and the other insiders who came forward with information, consistently portrayed Johnson as a manager in desperate need of a successful project. Earlier versions of Robbie, the CX10 and the CX20, were experimental in nature and no one expected them to be commercial successes. In fact, the Robotics Division of Silicon Techtronics was operating heavily in the red since its inception six years ago. Either CX30 would succeed or Silicon Techtronics would be out of the industrial robotics business altogether.

"The earlier Robbie robots got a lot of press, especially here in Silicon Valley," said another source, who also wishes to remain anonymous. "Robbie CX30 was going to capitalize on the good publicity generated by the earlier projects. The only thing was that Robbie CX30 was more revolutionary than Johnson wanted to admit. CX30 represented a gigantic step forward in terms of sophistication. There were a lot of questions about the industrial settings that the CX30 would be working in. Much of what we had to do was entirely new, but Johnson couldn't bring himself to understand that. He just saw us as unyielding perfectionists. One of his favorite quotes was 'Perfection is the enemy of the good'."

'Killer Robot' Programmer Was Prima Donna, Co-Workers Claim - 3

Special to the SILICON VALLEY SENTINEL-OBSERVER, Silicon Valley, USA

by Mabel Muckraker

Randy Samuels, the former Silicon Techtronics programmer who was indicted for writing the software that was responsible for the gruesome 'killer robot' incident last May, was apparently a 'prima donna' who found it very difficult to accept criticism, several of his co-workers claimed today.

In a free-wheeling interview with several of Samuels' co-workers on the 'killer robot' project, the Sentinel-Observer was able to gain important insights into the psyche of the man who may have been criminally responsible for the death of Bart Matthews, robot operator and father of three small children.

With the permission of those interviewed, the Sentinel-Observer allowed Professor Sharon Skinner of the Department of Software Psychology at Silicon Valley University to listen to a recording of the interview. Professor Skinner studies the psychology of programmers and other psychological factors which impact upon the software development process.

"I would agree with the woman who called him a 'prima donna'," Professor Skinner explained. "This is a term used to refer to a programmer who just cannot accept criticism, or more accurately, cannot accept his or her own fallibility."

"Randy Samuels has what we software psychologists call a task- oriented personality, bordering on self-oriented. He likes to get things done, but his ego is heavily involved in his work. In the programming world this is considered a 'no-no,'" Professor Skinner added in her book-lined office.

Professor Skinner went on to explain some additional facts about programming teams and programmer personalities. "Basically, we have found that a good programming team requires a mixture of personality types, including a person who is interaction-oriented, who derives a lot of satisfaction from working with other people, someone who can help keep the peace and keep things moving in a positive direction. Most programmers are task-oriented, and this can be a problem if one has a team in which everyone is task-oriented."

Samuels' co-workers were very reluctant to lay the blame for the robot disaster at his feet, but when pressed to comment on Samuels' personality and work habits, several important facts emerged. Samuels worked on a team consisting of about a dozen analysts, programmers and software testers. (This does not include twenty programmers who were later hired and who never became actively involved in the development of the robotics software.) Although individual team members had definite specialties, almost all were involved in the entire software process from beginning to end.

"Sam Reynolds has a background in data processing. He's managed several software projects of that nature," one of the team members said, referring to the manager of the Robbie CX30 project. "But, his

role in the project was mostly managerial. He attended all important meetings and he kept Ray [Ray Johnson, the Robotics Division Chief] off our backs as much as possible."

Sam Reynolds, as was reported in yesterday's Sentinel- Observer, was under severe pressure to deliver a working Robbie CX30 robot by January 1 of this year. Sam Reynolds could not be reached for comment either about his role in the incident or about Samuels and his work habits.

"We were a democratic team, except for the managerial guidance provided by Sam [Reynolds]," another team member observed. In the world of software development, a democratic team is a team in which all team members have an equal say in the decision-making process. "Unfortunately, we were a team of very ambitious, very talented -- if I must say so myself -- and very opinionated individualists. Randy [Samuels] was just the worst of the lot. I mean we have two guys and one gal with masters degrees from CMU who weren't as arrogant as Randy."

CMU refers to Carnegie Mellon University, a national leader in software engineering education.

One co-worker told of an incident in which Samuels stormed out of a quality assurance meeting. This meeting involved Samuels and three 'readers' of a software module which he had designed and implemented. Such a meeting is called a code review. One of the readers mentioned that Samuels had used a very inefficient algorithm (program) for achieving a certain result and Samuelson "turned beet red." He yelled a stream of obscenities and then left the meeting. He never returned.

"We sent him a memo about the faster algorithm and he eventually did use the more efficient algorithm in his module," the co-worker added.

The software module in the quality assurance incident was the very one which was found to be at fault in the robot operator "murder." However, this co-worker was quick to point out that the efficiency of the algorithm was not an issue in the malfunctioning of the robot.

"It's just that Randy made if very difficult for people to communicate their concerns to him. He took everything very personally. He graduated tops in his class at college and later graduated with honors in software engineering from Purdue. He's definitely very bright."

"Randy had this big computer-generated banner on his wall," this co-worker continued. "It said, 'YOU GIVE ME THE SPECIFICATION AND I'LL GIVE YOU THE COMPUTATION'. That's the kind of arrogance he had and it also shows that he had little patience for developing and checking the specifications. He loved the problem-solving aspect, the programming itself."

"It doesn't seem that Randy Samuels caught on to the spirit of 'egoless programming,'" Professor Skinner observed upon hearing this part of the interview with Samuels' co-workers. "The idea of egoless programming is that a software product belongs to the team and not to the individual programmers. The idea is to be open to criticism and to be less attached to one's work. Code reviews are certainly consistent with this overall philosophy."

A female co-worker spoke of another aspect of Samuelson's personality -- his helpfulness. "Randy hated meetings, but he was pretty good one on one. He was always eager to help. I remember one time when I

ran into a serious roadblock and instead of just pointing me in the right direction, he took over the problem and solved it himself. He spent nearly five entire days on my problem."

"Of course, in retrospect, it might have been better for poor Mr. Matthews and his family if Randy had stuck to his own business," she added after a long pause.

'Killer Robot' Project Mired in Controversy Right from Start - 4

Warring Factions Fought Over How Project Should Proceed

Special to the SILICON VALLEY SENTINEL-OBSERVER, Silicon Valley, USA

by Mabel Muckraker

Two groups, committed to different software development philosophies, nearly came to blows during the initial planning meetings for Robbie CX30, the Silicon Techtronics robot which killed an assembly line worker last May. At issue was whether the Robbie CX30 project should proceed according to the "waterfall model" or the "prototyping model."

The waterfall model and the prototyping model are two common methods for organizing a software project. In the waterfall model, a software project goes through definite stages of development. The first stage is requirements analysis and specification, during which an attempt is made to arrive at an agreement concerning the detailed functionality of the system. As the project passes from one stage to the next, there are limited opportunities for going back and changing earlier decisions. One drawback of this approach is that potential users do not get a chance to interact with the system until very late in the system's life cycle.

In the prototyping model, great emphasis is placed on producing a working model or prototype early during the life cycle of a system. The prototype is built for the purpose of arriving at a final specification of the functionality of the proposed system. Potential users interact with the prototype early and often until the requirements are agreed upon. This approach affords potential users the opportunity to interact with a prototype system early during the development cycle and long before the final system is designed and coded.

In a memo dated December 11th of the year before last, Jan Anderson, a member of the original Robbie CX30 project team, bitterly attacked the decision of the project manager, Sam Reynolds, to employ the waterfall model. The Sentinel-Observer has obtained a copy of Anderson's memo, which is addressed to Reynolds, and Anderson verified the authenticity of the memo for this reporter.

Reynolds fired Anderson on December 24th, just two weeks after she wrote the memo.

The Anderson memo refers to an earlier meeting at which an angry exchange occurred relating to software development philosophy. Anderson underlined the following passage in her memo:

"I did not intend to impugn your competence at our meeting yesterday, but I must protest most vehemently against the idea that we complete the Robbie CX30 software following the waterfall model which you have used in previous projects. I need not remind you that those were data processing projects involving the processing of business transactions. The Robbie CX30 project will involve a high degree of

interaction, both between robot components and between the robot and the operator. Since operator interaction with the robot is so important, the interface cannot be designed as an afterthought."

Randy Samuels, who has been charged with manslaughter in the death of robot operator Bart Matthews, father of three, was in attendance at the December 11th meeting.

In a conversation with this reporter, Anderson said that Samuels did not have much to say about the waterfall-prototyping controversy, but she did state that she would give her "eye teeth" to have Samuels exonerated.

"The project was doomed long before Samuels misinterpreted those formulas," Anderson stated emphatically, in the living room of her suburban townhouse.

In her conversation with this reporter, Anderson did her best to explain the waterfall-prototyping controversy in lay terms. "The main issue was really whether we could agree on the system requirements without allowing actual robot operators to get a feel for what we had in mind. Reynolds has been in the data processing business for three decades and he's good at that, but he never should have been made manager of this project."

According to records obtained by the Sentinel-Observer, Silicon Techtronics moved Sam Reynolds from the Data Processing Division, which took care of inventory and payroll, to the Robotics Division just three weeks before the December 11th meeting alluded to in Anderson's memo.

Reynolds was moved to the Robotics Division by Silicon Techtronics president Michael Waterson. Reynolds was replacing John Cramer, who managed the earlier Robbie projects, CX10 and CX20. Cramer was placed in charge of CX30, but he died unexpectedly in a skydiving accident. In placing Reynolds in charge of the CX30 project, our sources tell us that Waterson was going against the advice of Ray Johnson, Robotics Division Chief.

According to these sources Johnson strongly opposed Reynolds' choice as head of the Robbie CX30 project. These sources tell the Sentinel-Observer that Waterson's choice of Reynolds was purely a cost-saving decision. It was cheaper to move Reynolds to the Robotics Division than to hire a new project leader from outside the corporation.

The anonymous source that the Sentinel-Observer calls 'Martha' described the situation in this way: "Waterson thought it would be cheaper to move Reynolds to robotics rather than try to find a new manager for the Robbie project from outside. Also, Waterson tended to be suspicious of people from the outside. He often sends down memos about how long it takes people to master 'the Silicon Techtronics way of doing things'. In Waterson's view, Reynolds was a manager and he was moved to his new position in Robotics as a manager and not as a technical expert. Clearly, Reynolds saw himself as both a manager and as a technical expert. Reynolds was not aware of his own technical limitations."

According to Martha, Reynolds was very reluctant to manage a project which would not use the waterfall model which had served him so well in data processing. He attacked prototyping as a "fad" at the meeting on December 11th and after a few verbal exchanges back and forth things got pretty personal.

"Anderson was especially vocal," Martha recalled. "She had lots of experience with user interfaces and from her perspective, the operator-robot interface was critical to the success of CX30 since operator intervention would be frequent and at times critical."

In her interview with the Sentinel-Observer, Jan Anderson commented on this aspect of the December 11th meeting: "Reynolds was vehemently opposed to 'wasting time' -- to use his words -- on any kind of formal analysis of the user interface and its human factors properties. To him, user interfaces were a peripheral issue."

"Anything new was a 'fad' to him [Reynolds]," Anderson added. "Computer interfaces were a fad, object-oriented design was a fad, formal specification and verification techniques were a fad, and most of all, prototyping was a fad."

Exactly one week after the December 11th meeting, the Robbie group received a memo from Sam Reynolds concerning the project plan for the Robbie CX30 project.

"It was the waterfall model, right out of a textbook," Anderson told this reporter as she reviewed a copy of the project plan memo. "Requirements analysis and specification, then architectural design and detailed design, coding, testing, delivery and maintenance. In Reynolds' view of things, there was no need to have any user interaction with the system until very, very late in the process."

The Sentinel-Observer has learned that the very first operator to actually use the Robbie CX30 robot in an industrial setting was Bart Matthews, the man who was killed in the killer robot tragedy. This initial use of Robbie CX30 in an industrial setting was covered by the media, including this newspaper. In a great irony, the Silicon Techtronics Annual Report for Shareholders, published last March, has a picture of a smiling Bart Matthews on its glossy front cover. Matthews is shown operating the very same Robbie CX30 robot which crushed him to death barely two months after the photograph was taken.

Silicon Techtronics Promised to Deliver a Safe Robot - 5

Quality of Operator Training Questioned

Special to the SILICON VALLEY SENTINEL-OBSERVER, Silicon Valley, USA

by Mabel Muckraker

At a news conference this afternoon, a ragtag group of programmers who call themselves the "Justice for Randy Samuels Committee" distributed documents which show that Silicon Techtronics had obligated itself to deliver robots which would "cause no bodily injury to the human operator." Randy Samuels is the programmer who has been charged with manslaughter in the infamous "killer robot" case.

"We cannot understand how the prosecuting attorney could charge Randy with manslaughter when, in fact, Silicon Techtronics was legally bound to deliver a safe robot to Cybernetics," said committee spokesperson, Ruth Witherspoon.

"We believe that there is a cover-up going on and that there is some kind of collusion between SiliTech [Silicon Techtronics] management and the prosecuting attorney's office. Michael Waterson was a major contributor to Ms. McMurdock's re- election campaign last year."

Michael Waterson is President and CEO of Silicon Techtronics. Jane McMurdock is the Prosecuting Attorney for the city of Silicon Valley. The Sentinel-Observer has confirmed that Waterson made several large contributions to the McMurdock re-election campaign last fall.

"Randy is being made the scapegoat for a company which had lax quality control standards and we are not going to stand for it!" Witherspoon shouted in an emotional statement to reporters. "We believe that politics has entered this case."

The documents which were distributed by the Justice for Randy Samuels committee were portions of what is called a "requirements document." According to Ruth Witherspoon and other committee members, this document proves that Samuels was not legally responsible for the death of Bart Matthews, the unfortunate robot operator who was killed by a Silicon Techtronics robot at Cybernetics, Inc. in Silicon Heights last April.

The requirements document amounts to a contract between Silicon Techtronics and Cybernetics, Inc. The requirements document spells out in complete detail the functionality of the Robbie CX30 robot which Silicon Techtronics promised to deliver to Cybernetics.

According to Witherspoon, the Robbie CX30 robot was designed to be an "intelligent" robot which would be capable of operating in a variety of industrial settings. Separate requirements documents were required for each corporate customer since Robbie CX30 was not an "off-the-shelf" robot, but a robot that needed to be programmed differently for each application.

However, all requirements documents which were agreed upon under the auspices of the Robbie CX30 project, including the agreement between Silicon Techtronics and Cybernetics, contain the following important statements:

"The robot will be safe to operate and even under exceptional conditions (see Section 5.2) the robot will cause no bodily injury to the human operator."

"In the event of the exceptional conditions which potentially contain the risk of bodily injury (see Section 5.2.4 and all of its subsections), the human operator will be able to enter a sequence of command codes, as described in the relevant sections of the functional specification (see Section 3.5.2), which will arrest robot motion long before bodily injury can actually occur."

"Exceptional conditions" include unusual events such as bizarre data from the robot sensors, erratic or violent robot motion or operator error. It was exactly just such an exceptional condition which led to the death of Bart Matthews.

These paragraphs were extracted from the portion of the requirements document which dealt with "non-functional requirements." The non-functional requirements list in complete detail the constraints under which the robot would be operating. For example, the requirement that the robot be incapable of harming its human operator is a constraint and Silicon Techtronics, according to Ruth Witherspoon, was legally obligated to satisfy this constraint.

The functional requirements portion of the requirements document covers (again in complete detail) the behavior of the robot and its interaction with its environment and its human operator. In particular, the functional requirements specified the behavior of the robot under each and every anticipated exceptional condition.

In her statement to reporters at the news conference, Witherspoon explained that Bart Matthews was killed when exceptional condition 5.2.4.26 arose. This involved an exceptionally violent and unpredictable robot arm motion. This condition required operator intervention, namely the entering of the command codes mentioned in the document, but apparently, Bart Matthews became confused and could not enter the codes successfully.

"Although Randy Samuels' program was in error -- he did misinterpret the robot dynamics formulas, as reported in the media -- exceptional condition 5.2.4.26 was designed to protect against just this sort of contingency," Witherspoon told reporters. "The robot motion values generated by Randy's program correctly set off this exceptional condition and the robot operator received due warning that something was wrong."

Witherspoon claimed that she has a signed affidavit from another Cybernetics robot operator to the effect that the training sessions offered by Silicon Techtronics never mentioned this and many other exceptional conditions. According to Witherspoon, the robot operator has sworn that neither she nor any other robot operator was ever told that the robot arm could oscillate violently. Witherspoon quoted the affidavit at the news conference. "Neither I not Bart Matthews was ever trained to handle this sort of exceptional condition. I doubt that the Bart Matthews had any idea what he was supposed to do when the computer screen started flashing the error message on the screen."

Exceptional conditions requiring operator intervention cause an error message to be generated at the operator console. Silicon Valley Police confirm that when Bart Matthews was killed, the reference manual at his console was opened to the page of the index which contained entries for "errors."

Witherspoon then quoted sections of the requirements document which obligated Silicon Techtronics (the vendor) to adequately train robot operators:

"The vendor shall provide forty (40) hours of operator training. This training shall cover all aspects of robot operation including exhaustive coverage of the safety procedures which must be followed in the case of exceptional conditions which potentially contain the risk of bodily injury.

"The vendor shall provide and administer appropriate test instruments which shall be used to certify sufficient operator understanding of robot console operations and safety procedures. Only employees of customer who have passed this test shall be allowed to operate the Robbie CX30 robot in an actual industrial setting.

"The reference manual shall provide clear instructions for operator intervention in all exceptional situations, especially and including those which potentially contain the risk of bodily injury."

According to Witherspoon, sworn affidavits from several robot operators at Cybernetics, Inc., state that only one work day (approximately eight hours) was spent in operator training. Furthermore, almost no time was spent discussing potentially dangerous exceptional conditions.

"The written test developed by Silicon Techtronics to certify a robot operator was considered a 'joke' by Cybernetics employees," Witherspoon asserted. "Silicon Techtronics obviously did not give much thought to the training and testing procedures mandated by the requirements document according to the evidence in our possession."

The 'Killer Robot' Interface - 6

Dr. Horace Gritty
Department of Computer Science and Related Concerns
Silicon Valley University
Silicon Valley, USA

Abstract: The Robbie CX30 industrial robot was supposed to set a new standard for industrial robot intelligence. Unfortunately, one of the first Robbie CX30 robots killed an assembly line worker, leading to the indictment of one of the robot's software developers, Randy Samuels. This paper propounds the theory that it was the operator-robot interface designer who should be on trial in this case. The Robbie CX30 robot violates nearly every rule of interface design. This paper focuses on how the Robbie CX30 interface violated every one of Shneiderman's "Eight Golden Rules."

1. Introduction

On May 17, 1992 a Silicon Techtronics Robbie CX30 industrial robot killed its operator, Bart Matthews, at Cybernetics, Inc., in Silicon Heights, a suburb of Silicon Valley. An investigation into the cause of the accident led authorities to the conclusion that a software module, written and developed by Randy Samuels, a Silicon Techtronics programmer, was responsible for the erratic and violent robot behavior which in turn lead to the death by decapitation of Bart Matthews.

As an expert in the area of user interfaces (1,2,3), I was asked to help police reconstruct the accident. In order to accomplish this, Silicon Techtronics was asked to provide me with a Robbie CX30 simulator which included the complete robot operator console. This allowed me to investigate the robot's behavior without actually risking serious harm. Due to my extensive understanding of user interfaces and human factors I was able to reconstruct the accident with uncanny accuracy. On the basis of this reconstruction, I came to the conclusion that it was the interface design and not the admittedly flawed software which should be viewed as the culprit in this case.

Despite my finding, Prosecuting Attorney Jane McMurdock insisted on pursuing the case against Randy Samuels. I believe that any competent computer scientist, given an opportunity to interact with the Robbie CX30 simulator, would also conclude that the interface designer and not the programmer should be charged with negligence, if not manslaughter.

2. Shneiderman's "Eight Golden Rules"

My evaluation of the Robbie CX30 user interface is based upon Shneiderman's "eight golden rules" (4). I also used other techniques to evaluate the interface, but those will be published in separate papers. In this section, I offer a brief review of Shneiderman's eight golden rules, a subject which would be more familiar to computer interface experts such as myself as opposed to the robot hackers who read this obscure journal.

The eight golden rules are:

1. Strive for consistency. As we shall see below, it is important for a user interface to be consistent on many levels. For example, screen layouts should be consistent from one screen to another. In an environment using a graphical user interface (GUI), this also implies consistency from one application to another.

2. Enable frequent users to use shortcuts. Frequent users (or, power users) may be turned off by overly tedious procedures. Allow those users a less tedious procedure for accomplishing a given task.

3. Offer informative feedback. Users need to see the consequences of their actions. If a user enters a command but the computer does not show that it is either processing or has processed that command, this can leave the user confused and disoriented.

4. Design dialogues to yield closure. Interacting with a computer is somewhat like a dialogue or conversation. Every task should have a beginning, a middle and an end. It is important for the user to know when a task is at its end. The user needs to have the feeling that a task has reached closure.

5. Offer simple error handling. User errors should be designed into the system. Another way of stating this is that no user action should be considered an error that is beyond the ability of the system to manage. If the user makes a mistake, the user should receive useful, concise and clear information about the nature of the mistake. It should be easy for the user to undo his or her mistake.

6. Permit easy reversal of actions. More generally, users must be permitted to undo what they have done, whether it is in the nature of an error or not.

7. Support internal locus of control. User satisfaction is high when the user feels that he or she is in control and user satisfaction is low when the user feels that the computer is in control. Design interfaces to reinforce the feeling that the user is the locus of control in the human-computer interaction.

8. Reduce short-term memory load. Human short-term memory is remarkably limited. Psychologists often quote Miller's law to the effect that short-term memory is limited to seven discrete pieces of information. Do everything possible to free the user's memory burden. For example, instead of asking the user to type in the name of a file which is going to be retrieved, present the user with a list of files currently available.

3. Robot console overview

The Robbie CX30 operator interface violated each and every one of Shneiderman's rules. Several of these violations were directly responsible for the accident which ended in the death of the robot operator.

The robot console was an IBM PS/2 model 55SX with a 80386 processor and an EGA color monitor with 640x480 resolution. The console had a keyboard, but no mouse. The console was embedded in a workstation which included shelves for manuals and an area for taking notes and for reading manuals. However, the reading/writing area was quite a distance from the computer screen, so that it was quite awkward and tiresome for the operator to manage any task which required looking something up in the manual and then acting quickly with respect to the console keyboard. The operator's chair was poorly designed and much too high relative to the console and the writing/reading area. This placed much strain on the operator's back and also caused excessive eyestrain.

I cannot understand why a sophisticated system such as this would not include a better device for input. One can only conclude that Silicon Techtronics did not have much experience with user interface technology. The requirements document (5) specified a menu-driven system, which was a reasonable choice.

However, in an application where speed was of the essence, especially when operator safety was at issue, the use of a keyboard for all menu selection tasks was an extremely poor choice, requiring many keystrokes to achieve the same effect which could be achieved almost instantaneously with a mouse. (See the paper by Foley et al. (6). Actually, I had most of these ideas before Foley published them, but he beat me to the punch.)

The robot operator could interact with the robot and thus impact upon its behavior by making choices in a menu system. The main menu consisted of twenty items, too many in my opinion, and each main menu item had a pull-down submenu associated with it. Some of the submenus contained as many as twenty items -- again, too many. Furthermore, there seemed to be little rhyme or reason as to why the menu items were listed in the order in which they were listed. A functional or alphabetical organization would have been better.

Some items in the pull-down submenus had up to four pop-up menus associated with them. These would appear in sequence as submenu choices were made. Occasionally, a submenu choice would cause a dialogue box to appear at the screen. A dialogue box requires some kind of interaction between the operator and the system to resolve some issue, such as the diameter of the widgets being lowered into the acid bath. The menu system presents a strict hierarchy of menu choices. The operator could backtrack up the hierarchy by pressing the escape key. The escape key could also terminate any dialogue. The use of color in the interface was very unprofessional. There were too many colors in too small a space. The contrasts were glaring and the result, for this reviewer, was severe eye strain in just fifteen minutes. There was excessive use of flashing and silly musical effects when erroneous choices or erroneous inputs were made.

One has to wonder why Silicon Techtronics did not attempt a more sophisticated approach to the interface design. After a careful study of the Robbie CX30 applications domain, I have come to the conclusion that a direct-manipulation interface, which literally displayed the robot at the operator console, would have been ideal. The very visual domain that the robot operated within would lend itself naturally to the design of appropriate screen metaphors for that environment, metaphors which the operator could easily understand. This would allow the operator to manipulate the robot by manipulating the graphical representation of the robot in its environment at the computer console. I have asked one of my doctoral students, Susan Farnsworth, to give up her personal life for the better part of a decade in order to investigate this possibility a bit further.

4. How the Robbie CX30 interface violated the eight golden rules

The Robbie CX30 user interface violated each and every golden rule in multitudinous ways. I shall only discuss a few instances of rule violation in this paper, leaving a more detailed discussion of these violations for future articles and my forthcoming book1. I will emphasize those violations which were relevant to this particular accident.

4.1 Strive for consistency

There were many violations of consistency in the Robbie CX30 user interface. Error messages could appear in almost any color and could be accompanied by almost any kind of musical effect. Error messages could appear almost anywhere at the screen.

When Bart Matthews saw the error message for the exceptional condition which occurred, an exceptional condition which required operator intervention, it was probably the first time he saw that particular message. In addition, the error message appeared in a green box, without any audio effects. This is the only error message in the entire system which appears in green and without some kind of orchestral accompaniment.

4.2 Enable frequent users to use shortcuts

This principle does not appear in any way in the entire interface design. For example, it would have been a good idea to allow frequent users to enter the first letter of a submenu or menu choice in lieu of requiring the use of the cursor keys and the enter key to effect a menu choice. The menu selection mechanism in this system must have been quite a mental strain on the operator.

Furthermore, a form of type-ahead should have been supported, which would have allowed a frequent user to enter a sequence of menu choices without having to wait for the actual menus to appear.

4.3 Offer informative feedback

In many cases, the user has no idea whether a command that was entered is being processed. This problem is exaggerated by inconsistencies in the user-interface design. In some cases the operator is given detailed feedback concerning what the robot is doing. In other cases the system is mysteriously silent. In general, the user is led to expect feedback and consequently becomes confused when no feedback is given. There is no visual representation of the robot and its environment at the screen and the operator's view of the robot is sometimes obstructed.

4.4 Design dialogues to yield closure

There are many cases in which a given sequence of keystrokes represents one holistic idea, one complete task, but the operator is left without the kind of feedback which would confirm that the task has been completed. For example, there is a fairly complicated dialogue which is necessary in order to remove a widget from the acid bath. However, upon completion of this dialogue, the user is led into a new, unrelated dialogue, without being informed that the widget removal dialogue has been completed.

4.5 Offer simple error handling

The system seems to be designed to make the user regret any erroneous input. Not only does the system allow numerous opportunities for error, but when an error actually occurs, it is something that is not likely to be repeated for some time. This is because the user interface makes recovery from an error a tedious, frustrating and at times infuriating ordeal. Some of the error messages were downright offensive and condescending.

4.6 Permit easy reversal of actions

As mentioned in the previous paragraph, the user interface makes it very difficult to recover from erroneous inputs. In general, the menu system does allow easy reversal of actions, but this philosophy is not carried through to the design of dialogue boxes and to the handling of exceptional conditions. From a practical (as opposed to theoretical) point of view, most actions are irreversible when the system is in an exceptional state, and this helped lead to the killer robot tragedy.

4.7 Support internal locus of control

Many of the deficiencies discussed in the previous paragraphs diminished the feeling of "internal locus of control." For example, not receiving feedback, not bringing interactions to closure, not allowing easy reversal of actions when exceptions arose, all of these things act to diminish the user's feeling of being in control of the robot. There were many features of this interface which make the operator feel that there is an enormous gap between the operator console and the robot itself, whereas a good interface design would have made the user interface transparent and would have given the robot operator a feeling of being in direct contact with the robot. In one case, I commanded the robot to move a widget from the acid bath to the drying chamber and it took 20 seconds before the robot seemed to respond. Thus, I did not feel like I was controlling the robot. The robot's delayed response along with the lack of informative feedback at the computer screen made me feel that the robot was an autonomous agent -- an unsettling feeling to say the least.

4.8 Reduce short-term memory load

A menu-driven system is generally good in terms of the memory burden it places upon users. However, there is a great variation among particular implementations of menu systems insofar as memory burden is concerned. The Robbie CX30 user interface had very large menus without any obvious internal organization. These place a great burden upon the operator in terms of memory and also in terms of scan time, the time it takes the operator to locate a particular menu choice.

Many dialogue boxes required the user to enter part numbers, file names, and other information from the keyboard. The system could easily have been designed to present the user with these part numbers and so forth without requiring the user to recall these things from his or her own memory. This greatly increased to memory burden upon the user.

Finally, and this is really unforgivable, incredible as it may seem, there was no on-line, context-sensitive help facility! Although I was taken through the training course offered by Silicon Techtronics, I often found myself leafing through the reference manuals in order to find the answer to even the most basic questions, such as: "What does this menu choice mean? What will happen if I make this choice?"

5. A reconstruction of the "killer robot" tragedy

Police photographs of the accident scene are not a pleasant sight. The operator console was splattered with a considerable amount of blood. However, the photographs are of exceptional quality and using blow-up techniques, I was able to ascertain the following important facts about the moment when Bart Matthews was decapitated:

1. The NUM LOCK light was on.

The IBM keyboard contains a calculator pad which can operate in two modes. When the NUM LOCK light is on, it behaves like a calculator. Otherwise, the keys can be used to move the cursor at the screen.

2. Blood was smeared on the calculator pad.

Bloody fingerprints indicate that Bart Matthews was using the calculator pad when he was struck and killed.

3. A green error message was flashing.

This tells us the error situation in force when the tragedy occurred. The error message said, "ROBOT DYNAMICS INTEGRITY ERROR -- 45 ."

4. A reference manual was open and was laid flat in the workstation reading/writing area.

One volume of the four volume reference manual was open to the index page which contained the entry 'ERRORS / MESSAGES'.

5. A message giving operator instructions was also showing on the screen.

This message was displayed in yellow at the bottom of the screen. This message read "PLEASE ENTER DYNAMICAL ERROR ROBOT ABORT COMMAND SEQUENCE PROMPTLY!!!"

On the basis of this physical evidence, plus other evidence contained in the system log, and based upon the nature of the error which occurred (robot dynamics integrity error -- 45, the error which was caused by Randy Samuels' program), I have concluded that the following sequence of events occurred on the fateful morning of the killer robot tragedy:

10:22.30. "ROBOT DYNAMICS INTEGRITY ERROR -- 45" appears on the screen. Bart Matthews does not notice this because there is no beep or audio effect such as occurs with every other error situation. Also, the error message appears in green, which in all other contexts means that some process is proceeding normally.

10:24.00. Robot enters state violent enough for Bart Matthews to notice.

10:24.05. Bart Matthews notices error message, does not know what it means. Does not know what to do. He tries "emergency abort" submenu, a general purpose submenu for turning off the robot. This involves SIX separate menu choices, but Mr. Matthews does not notice that the NUM LOCK light is lit. Thus, the menu choices aren't registering because the cursor keys are operating as calculator keys.

10:24.45. Robot turns from acid bath and begins sweep towards operator console, its jagged robot arms flailing wildly. No one anticipated that the operator might have to flee a runaway robot, so Bart Matthews is cornered in his work area by the advancing robot. At about this time, Bart Matthews retrieves the

reference manual and starts looking for a reference to ROBOT DYNAMICS INTEGRITY ERROR -- 45 in the index. He successfully locates a reference to error messages in the index.

10:25.00. Robot enters the operator area. Bart Matthews gives up on trying to find the operator procedure for the robot dynamics integrity error. Instead, he tries once again to enter the "emergency abort" sequence from the calculator keypad, when he is struck.

6. Summary and conclusions

While the software module written by Randy Samuels did cause the Robbie CX30 robot to oscillate out of control and attack its human operator, a good interface design would have allowed the operator to terminate the erratic robot behavior. Based upon an analysis of the robot user interface using Shneiderman's eight golden rules, this interface design expert has come to the conclusion that the interface designer and not the programmer was the more guilty party in this unfortunate fiasco.

7. References

1. Gritty, Horace (1990). The Only User Interface Book You'll Ever Need. Vanity Press, Oshkosh, WI, 212 pp.

2. Gritty, Horace (1992). "What We Can Learn from the Killer Robot," invited talk given at the Silicon Valley University International Symposium on Robot Safety and User Interfaces, March 1991. Also to appear in Silicon Valley University Alumni Notes.

3. Gritty, Horace (expected 1993). CODEPENDENCY2: How Computer Users Enable Poor User Interfaces, Angst Press, New York.

4. Shneiderman, Ben (1987). Designing the User Interface, Addison-Wesley, Reading MA, 448 pp.

5. Robbie CX30 INTELLIGENT INDUSTRIAL ROBOT REQUIREMENTS DOCUMENT : Cybernetics Inc. Version, Technical Document Number 91-0023XA, Silicon Techntronics Corporation, Silicon Valley, USA, 1245 pp.

6. Foley, J. P., Wallace, V. L., and Chan, P. (1984). "The Human Factors of Computer Graphics Interaction Techniques." IEEE COMPUTER GRAPHICS AND APPLICATIONS, 4(11), pp. 13- 48.

---

1 The media were misled to believe that Bart Matthews was crushed by the robot, but the photographic evidence given to this author shows otherwise. Perhaps authorities were attempting to protect public sensibilities.

2 CODEPENDENCY: How Computer Users Enable Poor User Interfaces, Angst Press, New York. This book presents a radically new theory concerning the relationship between people and their machines. Essentially, some people need a poor interface in order to avoid some unresolved psychological problems in their lives.

Foley and Shneiderman are actual authors. The other references are fictitious.

Software Engineer Challenges Authenticity of 'Killer Robot' Software Tests - 7

SVU Professor's Inquiry Raises Serious Legal and Ethical Issues

Special to the SILICON VALLEY SENTINEL-OBSERVER, Silicon Valley, USA

by Mabel Muckraker

The "killer robot" case took a significant turn yesterday when a Silicon Valley University professor issued a report questioning the authenticity of software tests that were purportedly performed on the "killer robot" software by Silicon Techtronics. Professor Wesley Silber, Professor of Software Engineering, told a packed news conference held at the university that the test results reflected in Silicon Techtronics internal documents were not consistent with test results obtained when he and his associates tested the actual robot software. Silicon Valley is still reacting to Professor Silber's announcement, which could play an important role in the trial of Randy Samuels, the Silicon Techtronics programmer who has been changed with manslaughter in the now infamous "killer robot" incident.

Pressed for her reaction to Professor Silber's report, Prosecuting Attorney Jane McMurdock reiterated her confidence that a jury will find Randy Samuels guilty. McMurdock shocked reporters, however, when she added, "But, this does raise the possibility of new indictments."

Ruth Witherspoon, spokesperson for the "Justice for Randy Samuels Committee," was almost exultant when she spoke to this reporter. "McMurdock cannot have it both ways. Either the programmer is responsible for this tragedy or management must be held responsible. We believe that the Silber report exonerates our friend and colleague Randy Samuels." Silicon Techtronics CEO Michael Waterson issued a terse statement concerning the Silber report:

"Soon after the indictment of Randy Samuels was announced, I personally asked the esteemed software engineer, Dr. Wesley Silber, to conduct an impartial inquiry into quality assurance procedures at Silicon Techtronics. As the chief executive of this corporation, I have always insisted on quality first, despite what you might have read in the press.

"I asked Professor Silber to conduct an impartial inquiry into all aspects of quality assurance at Silicon Techtronics. I promised Professor Silber that he would have access to all information relevant to this unfortunate situation. I told him in a face-to-face meeting in my office that he should pursue his investigation wherever it might lead, regardless of the implications.

"It never occurred to me, based upon the information that I was getting from my managers, that there might be a problem in which software quality assurance procedures were either lax or deliberately circumvented. I want the public to be reassured that the person or persons who were responsible for the failure of software quality assurance within the Robotics Division of Silicon Techtronics will be asked to find employment elsewhere." Roberta Matthews, widow of Bart Matthews, the robot operator who was killed in the incident, spoke to the Sentinel-Observer by telephone from her home. "I still want to see Mr.

Samuels punished for what he did to my husband. I don't understand what all the commotion is about. The man who murdered my husband should have tested his own software!"

The Sentinel-Observer interviewed Professor Silber in his office shortly after his news conference. On his office wall were numerous awards he has received because of his work in the field of software engineering and software quality assurance. We began the interview by asking Professor Silber to explain why it is that software is sometimes unreliable. He answered our question by citing the enormous complexity of software. "Large computer programs are arguably the most complex artifacts ever fashioned by the human mind," Professor Silber explained, seated in front of a large computer monitor. "At any point in time, a computer program is in one of an extremely large number of possible states, and it is a practical impossibility to assure that the program will behave properly in each of those states. We do not have enough time to do that kind of exhaustive testing. Thus, we use testing strategies or heuristics that are very likely to find bugs, if they exist."

Professor Silber has published numerous papers on software engineering. He made headlines last year when he published his list of "Airlines to Avoid as if Your Life Depended Upon It." That list named domestic airlines that he deemed irresponsible because of their purchase of airplanes that are almost completely controlled by computer software.

Soon after Randy Samuels was indicted in the "killer robot" case, the CEO of Silicon Techronics, Michael Waterson, asked Professor Silber to conduct an impartial review of quality assurance procedures at Silicon Techtronics. Waterson was acting in order to counter the bad publicity generated for his company after the Samuels indictment.

"Quality assurance" refers to those methods a software developer uses to assure that the software is reliable: correct and robust. These methods are applied throughout the development lifecycle of the software product. At each stage, appropriate quality assurance methods are applied. For example, when a programmer writes code, one quality assurance measure is to test the code by actually running it against test data. Another would be to run special programs, called static analyzers, against the new code. A static analyzer is a program that looks for suspicious patterns in programs, patterns that might indicate an error or bug.

These two forms of quality assurance are called dynamic testing and static testing, respectively.

Software consists of discrete components or units that are eventually combined to create larger systems. The units themselves must be tested, and this process of testing individual units is called unit testing. When the units are combined, the integrated subsystems must be tested and this process of testing the integrated subsystems is called integration testing. Professor Silber told the Sentinel-Observer about his work at Silicon Techtronics: "Mike [Waterson] told me to go in there [into the company] and conduct an impartial review of his software testing procedures and to make my findings public. Mike seemed confident, perhaps because of what his managers had told him, that I would find nothing wrong with quality assurance at Silicon Techtronics."

Soon after arriving at Silicon Techtronics, Professor Silber focused his attention on procedures for dynamically testing software at the high tech company.

Assisted by a cadre of graduate students, Professor Silber discovered a discrepancy between the actual behavior of the section of program code (written by Randy Samuels) that caused the Robbie CX30 robot to kill its operator and the behavior as recorded in test documentation at Silicon Techtronics. This discovery was actually made by Sandra Henderson, a graduate student in software engineering who is completing her doctorate under Professor Silber. We interviewed Ms. Henderson in one of the graduate computer laboratories at Silicon Valley University.

"We found a problem with the unit testing," Ms. Henderson explained. "Here are the test results, given to us by Mr. Waterson at Silicon Techtronics, which are purported to be for the C [programming language] code which Randy Samuels wrote and which caused the killer robot incident. As you can see, everything is clearly documented and organized. There are two test suites: one based upon white-box testing and another based upon black-box testing. Based upon our own standards for testing software, these test suites are well-designed, complete and rigorous."

Black-box testing involves viewing the software unit (or component) as a black box which has expected input and output behaviors. If the component demonstrates the expected behaviors for all inputs in the test suite, then it passes the test. Test suites are designed to cover all "interesting" behaviors that the unit might exhibit but without any knowledge of the structure or nature of the actual code.

White-box testing involves covering all possible paths through the unit. Thus, white-box testing is done with thorough knowledge of the unit's structure. In white-box testing, the test suite must cause each program statement to execute at least once so that no program statement escapes execution. Sandra Henderson went on to explain the significance of software testing: "Neither black-box nor white-box testing 'proves' that a program is correct. However, software testers, such as those employed at Silicon Techtronics, can become quite skillful at designing test cases so as to discover new bugs in the software. The proper attitude is that a test succeeds when a bug is found."

"Basically, the tester is given a set of specifications and does his or her best to show that the code being tested does not satisfy its specifications," Ms. Henderson explained.

Ms. Henderson then showed this reporter the test results that she actually obtained when she ran the critical "killer robot" code using the company's test suites for white-box and black-box testing. In many cases, the outputs recorded in the company's test documents were not the same as those generated by the actual killer robot code.

During his interview with the Sentinel-Observer yesterday, Professor Silber discussed the discrepancy "You see, the software that was actually delivered with the Robbie CX30 robot was not the same as the software that was supposedly tested -- at least according to these documents! We have been able to determine that the actual "killer code," as we call it, was written after the software tests were supposedly conducted. This suggests several possibilities: First, the software testing process, at least for this critical part of the software, was deliberately faked. We all know that there was enormous pressure to get this robot 'out the door' by a date certain. Another possibility is that there was some kind of version management difficulty at Silicon Techtronics, so that correct code was written, successfully tested, but the wrong code was inserted into the delivered product."

We asked Professor Silber to explain what he meant by "version management." "In a given project, a given software component might have several versions: version 1, version 2 and so forth. These reflect the evolution of that component as the project progresses. Some kind of mechanism needs to be in place to keep track of versions of software components in a project as complex as this one. Perhaps the software testers tested a correct version of the robot dynamics code, but an incorrect version was actually delivered. However, this raises the question as to what happened to the correct code."

Professor Silber sat back in his chair and sighed. "This really is a great tragedy. If the 'killer code' had gone through the testing process, in an honest manner, the robot would never have killed Bart Matthews. So, the question becomes, what was going on at Silicon Techtronics that prevented the honest testing of the critical code?"

The Silicon-Observer asked Professor Silber whether he agreed with the notion that the user interface was the ultimate culprit in this case. "I don't buy the argument, being put forth by my colleague, Professor Gritty, that all of the culpability in this case belongs to the user-interface designer or designers. I agree with some of what he says, but not all of it. I have to ask myself whether Silicon Techtronics was placing too much emphasis on the user interface as a last line of defense against disaster. That is, they knew there was a problem, but they felt that the user interface could allow the operator to handle that problem."

The Silicon-Observer then asked Professor Silber about the charge made against him that he should never have accepted Waterson's appointment to conduct an impartial investigation into the accident. Critics point out that Silicon Valley University and Professor Silber, in particular, had many business ties with Silicon Techtronics, and thus he could not be counted upon to conduct an impartial investigation.

"I think my report speaks for itself," Professor Silber replied, visibly annoyed by our question. "I have told you reporters over and over again that this was not a government investigation but a corporate investigation, so I believe that Silicon Techtronics had the right to choose whomever they desired. I believe I was known to them as a person of integrity."

Late yesterday, Sam Reynolds, the Robbie CX30 Project Manager, hired an attorney, Valerie Thomas. Ms. Thomas issued this statement on behalf of her client:

"My client is shocked that someone at Silicon Techtronics has misled Professor Silber concerning the software tests for the Robbie CX30 robot. Mr. Reynolds asserts that the software was tested and that he and others were well aware of the fact that there was something wrong with the robot dynamics software. However, Mr. Ray Johnson, my client's immediate superior at Silicon Techtronics, decided that the robot could be delivered to Cybernetics, Inc., based upon Mr. Johnson's 'Ivory Snow Theory'. According to that theory, the software was nearly bug free and thus could be released. According to Mr. Johnson, the risk of failure was very small and the cost of further delaying delivery of the robot was very great.

"According to my client, Mr. Johnson felt that the environmental conditions that could trigger erratic and violent robot behavior were extremely unlikely to occur. Furthermore, Mr. Johnson felt that the robot operator would not be in danger because the user interface was designed so as to permit the operator to stop the robot dead in its tracks in the case of any life-threatening robot motion."

Mr. Johnson, Robotics Division Chief at Silicon Techtronics, could not be reached for comment.

Randy Samuels will be placed on trial next month at the Silicon Valley Courthouse. When contacted by phone, Samuels referred all questions to his attorney, Alex Allendale. Allendale had this to say concerning Professor Silber's findings: "My client submitted the software in question in the usual way and with the usual documentation and with the usual expectation that his code would be thoroughly tested. He was not aware until Professor Silber's report came out that the code involved in this terrible tragedy had not been tested properly or that the test results might have been faked.

"Mr. Samuels wants to again express the great sorrow he feels about this accident. He, more than anyone else, wants to see justice done in this case. Mr. Samuels once again extends his heartfelt condolences to the Mrs. Matthews and her children."

Silicon Techtronics Employee Admits Faking Software Tests - 8

Electronic Mail Messages Reveal New Details in 'Killer Robot' Case

Association of Computer Scientists Launches Investigation Into Ethics Code Violations

Special to the SILICON VALLEY SENTINEL-OBSERVER, Silicon Valley, USA

by Mabel Muckraker

Cindy Yardley, a software tester at Silicon Techtronics, admitted today that she was the person who created the fraudulent "killer robot" software tests. The fraudulent tests were revealed earlier this week by Silicon Valley University professor Wesley Silber in what has come to be known as the "Silber Report."

At issue are quality assurance procedures that were performed on the program code written by Randy Samuels, the programmer charged with manslaughter in the killer robot incident. The Silber Report asserted that the test results reflected in internal Silicon Techtronics documents were inconsistent with the test results obtained when the actual killer-robot code was tested.

In a startling development at noontime yesterday, Max Worthington, Chief Security Officer for Silicon Techtronics, announced his resignation at a packed news conference that was broadcast live by CNN and other news organizations.

Worthington stunned the assembled reporters when he began his news conference with the announcement, "I am Martha."

Worthington described his responsibilities at Silicon Techtronics in this way: "Basically, my job was to protect Silicon Techtronics from all enemies -- domestic and foreign. By foreign I mean adversaries from outside the corporation. My role was mostly managerial. Those working under me had many responsibilities, including protecting the physical plant, watching out for industrial spying and even sabotage. I was also responsible for keeping an eye out for employees who might be abusing drugs or who might be disloyal in some way to Silicon Techtronics."

Worthington then pointed to a stack of bound volumes which were on a table to his left. "These volumes represent just some of the electronic surveillance of employees that I conducted over the years for my superior, Mr. Waterson. These are printouts of electronic mail messages that Silicon Techtronics employees sent to one another and to persons at other sites. I can say with great certainly that no employee was ever told that this kind of electronic surveillance was being conducted. However, I think the evidence shows that some employees suspected that this might be going on."

Several reporters shouted questions asking who at Silicon Techtronics knew about the electronic surveillance.

Worthington replied, "No one knew about this except Mr. Waterson, myself, and one of my assistants, who was responsible for conducting the actual monitoring. My assistant produced a special report, summarizing e-mail [electronic mail] activity once a week, and that report was for Waterson's eyes and my eyes, only. Upon request, my assistant could produce a more detailed accounting of electronic communications."

Worthington explained that he was making the electronic mail transcripts available to the press because he wanted the whole truth to come out concerning Silicon Techtronics and the killer robot incident.

The electronic mail messages between employees at Silicon Techtronics indeed revealed new facets of the case. A message from Cindy Yardley to Robotics Division Chief Ray Johnson indicates that she faked the test results at his request. Here is the text of that message:

To: ray.johnson
From: cindy.yardlay

Re: samuels software

I have finished creating the software test results for that troublesome robot software, as per your idea of using a simulation rather than the actual software. Attached you will find the modified test document, showing the successful simulation.

Should we tell Randy about this?

-- Cindy

Johnson's response to Yardley's message suggests that he suspected that electronic mail might not be secure:

In-reply-to: cindy.yardley
From: ray.johnson

Re: samuel's software

I knew I could count on you! I am sure that your devotion to Silicon Techtronics will be repaid in full.

Please use a more secure form of communication in the future when discussing this matter. I assure you that the way we handled this was completely above board, but I have my enemies here at good ol' SiliTech.

-- Ray

These communications were exchanged just a few days before the Robbie CX30 robot was shipped out to Cybernetics, Inc. This fact is important because the fake software tests were not part of a cover-up of the killer robot incident. These facts seem to indicate that the purpose of the fake software tests was to make sure that the Robbie CX30 robot was delivered to Cybernetics by a deadline that was negotiated between Silicon Techtronics and Cybernetics.

The electronic mail transcripts reveal repeated messages from Ray Johnson to various people to the effect that the Robotics Division would definitely be closed down if the Robbie CX30 project was not completed on time. In one message, he lectures project leader, Sam Reynolds, on his "Ivory Snow Theory":

To: sam.reynolds
From: ray.johnson

Re: don't be a perfectionist!

Sam:

You and I have had our differences, but I must tell you that I like you personally. Please understand that everything I am doing is for the purpose of SAVING YOUR JOB AND THE JOB OF EVERYONE IN THIS DIVISION. I view you and all of the people who work with me in the Robotics Division as my family.

Waterson has made it clear: he wants the robot project completed on time. That's the bottom line. Thus, we have no recourse but "Ivory Snow." You know what I mean by that. It doesn't have to be perfect. The user interface is our fall back if this version of the robot software has some flaws. The robot operator will be safe because the operator will be able to abort any robot motion at any time.

I agree with you that the non-functional requirements are too vague in places. Ideally, if this weren't crunch time, it would be good to quantify the amount of time it would take the operator to stop the robot in case of an accident. However, we cannot renegotiate those now. Nor, do we have time to design new tests for new, more precise non- functional requirements.

I cannot emphasize enough that this is crunch time. It's no sweat off Waterson's back if he lops off the entire Robotics Division. His Wall Street friends will just say, "Congratulations!" You see, to Waterson, we are not a family, we are just corporate fat.

-- Ray

In this message, Ray Johnson seems to be less concered with the security of communicating by electronic mail.

The Silicon-Observer interviewed Cindy Yardley at her home yesterday evening. Neither Ray Johnson nor Sam Reynolds could be reached for comment.

Ms. Yardley was obviously upset that her private electronic mail messages had been released to the press. "I am relieved in some ways. I felt tremendous guilt when that guy was killed by a robot that I helped to produce. Tremendous guilt."

The Silicon-Observer asked Ms. Yardley whether she felt that she had made an ethical choice in agreeing to fake the software test results. She responded with great emotion: "Nothing, nothing in my experience or background prepared me for something like this. I studied computer science at a major university and they taught me about software testing, but they never told me that someone with power over me might ask me to produce a fake software test!"

"When Johnson asked me to do this, he called me to his office, as if to show me the trappings of power, you see, someday I would like to be in a managerial position. I sat down in his office and he came right out and said, 'I want you to fake the test results on that Samuels software. I don't want Reynolds to know anything about this.'"

Yardley fought back tears. "He assured me that no one would probably ever see the test results because the robot was perfectly safe. It was just an internal matter, a matter of cleanliness, in case anyone at Cybernetics or higher up in the corporation got curious about our test results. I asked him whether he was sure about the robot being safe and all that and he said, 'It's safe! The user interface is our line of defense. In about six months we can issue a second version of the robotics software and by then this Samuels problem will be solved.'"

Yardley leaned forward in her chair as if her next remark needed special emphasis. "He then told me that if I did not fake the software tests, then everyone in the Robotics Division would lose their job. On that basis I decided to fake the test results -- I was trying to protect my job and the job of my co-workers."

Ms. Yardley is currently pursuing an MBA degree at night at Silicon Valley University.

The Sentinel-Observer then asked Ms. Yardley whether she still felt that she had made an ethical decision, in view of the death of Bart Matthews. "I think I was misled by Ray Johnson. He told me that the robot was safe."

Another revelation, contained in the released electronic mail transcripts, was the fact that Randy Samuels stole some of the software that he used in the killer robot project. This fact was revealed in a message Samuels sent to Yardley when she first tested his software and it gave erroneous results:

In-reply-to: cindy.yardley
From: randy.samuels

Re: damned if I know

I cannot for the life of me figure out what is wrong with this function, swing_arm(). I've checked the robot dynamics formula over and over again, and it seems to be implemented correctly. As you know, swing_arm() calls 14 different functions. I lifted five of those from the PACKSTAT 1-2-3 statistical package verbatim. Please don't tell a soul! Those couldn't be the problem, could they?

-- Randy

Experts tell the Silicon-Observer that lifting software from a commercial software package like PACKSTAT 1-2-3 is a violation of the law. Software such as the immensely popular PACKSTAT 1-2-3 is protected by the same kind of copyright that protects printed materials.

Mike Waterson, CEO of Silicon Techtronics issued an angry statement concerning Max Worthington's release of "confidential" electronic mail transcripts. Waterson's statement said, in part, "I have asked our attorneys to look into this matter. We consider those transcripts the exclusive property of Silicon Techtronics. Our intent is to pursue either civil or criminal charges against Mr. Worthington."

In reaction to yesterday's developments in the killer robot case, the ACM or Association for Computing Machinery announced its intention to investigate whether any ACM members at Silicon Techtronics have violated the ACM Code of Ethics. The ACM is an international association of computer scientists with 85,000 members.

Dr. Turina Babbage, ACM President, issued a statement from the ACM's Computer Science Conference, which is held every winter and which is being held this winter in Duluth, Minnesota.

An excerpt from Dr. Babbage's statement follows:

All members of the ACM are bound by the ACM Code of Ethics and Professional Conduct1 . This code states, in part, that ACM members have the general moral imperative to contribute to society and human well-being, to avoid harm to others, to be honest and trustworthy, to give proper credit for intellectual property, to access computing and communication resources only when authorized to do so, to respect the privacy of others and to honor confidentiality.

Beyond that, there are professional responsibilities, such as the obligation to honor contracts, agreements, and assigned responsibilities, and to give comprehensive and thorough evaluations of computing systems and their impacts, with special emphasis on possible risks.

Several of the people involved in the killer robot case are ACM members and there is cause to believe that they have acted in violation of our association's code of ethics. Therefore, I am asking the ACM Board to appoint a Task Force to investigate ACM members who might be in gross violation of the code.

We do not take this step lightly. This sanction has been applied only rarely, but the killer robot incident has not only cost a human life, but it has done much to damage the reputation of the computing profession.

---

1 A draft of this code was reported in Communications of the ACM, May 1992.

Please note that the statement by the fictitious Dr. Babbage contains verbatim quotes from the actual ACM code.

A Conversation With Dr. Harry Yoder - 9

The Sunday Sentinel-Observer Magazine

by Robert Franklin

Harry Yoder is a well-known figure on the Silicon Valley University campus. The Samuel Southerland Professor of Computer Technology and Ethics, he has written numerous articles and texts on ethics and the social impact of computers. His courses are very popular, and most of his courses are closed long before the end of the registration period. Dr. Yoder received his Ph. D. in electrical engineering from the Georgia Institute of Technology in 1958. In 1976 he received a Master of Divinity degree from the Harvard Divinity School. In 1983 he received an MS in Computer Science from the University of Washington. He joined the faculty at Silicon Valley University in 1988.

I interviewed Dr. Yoder in his office on campus. My purpose was to get his reaction to the case of the killer robot and to "pick his brain" about the ethical issues involved in this case.

Sentinel-Observer: Going from electrical engineering to the study of religion seems like quite a jump.

Yoder: I was an electrical engineer by profession, but all human beings have an inner life. Don't you?

Sentinel-Observer: Yes.

Yoder: What is your inner life about?

Sentinel-Observer: It's about doing the right thing. Also, it's about achieving excellence in what I do. Is that what sent you to Harvard Divinity School? You wanted to clarify your inner life?

Yoder: There was a lot going on at the Divinity School, and much of it was very compelling. However, most of all I wanted to understand the difference between what was right and what was wrong.

Sentinel-Observer: What about God?

Yoder: Yes, I studied my own Christian religion and most of the major world religions, and they all had interesting things to say about God. However, when I discuss ethics in a forum such as this, which is secular, or when I discuss ethics in my computer ethics courses, I do not place that discussion in a religious context. I think religious faith can help a person to become ethical, but on the other hand, we all know that certain notorious people who have claimed to be religious have been highly unethical. Thus, when I discuss computer ethics, the starting point is not religion, but rather a common agreement between myself and my students that we want to be ethical people, that striving for ethical excellence is a worthwhile human endeavor. At the very least, we do not want to hurt other people, we do not want to lie, cheat, steal, maim, murder and so forth.

Sentinel-Observer: Who is responsible for the death of Bart Matthews?

Yoder: Please forgive me for taking us back to the Harvard Divinity School, but I think one of my professors there had the correct answer to your question. He was an elderly man, perhaps seventy, from Eastern Europe, a rabbi. This rabbi said that according to the Talmud, an ancient tradition of Jewish law, if innocent blood is shed in a town, then the leaders of that town must go to the edge of the town and perform an act of penance. This was in addition to any justice that would be meted out to the person or persons who committed the murder.

Sentinel-Observer: That's an interesting concept.

Yoder: And a truthful one! A town, a city, a corporation -- these are systems in which the part is related to the whole and the whole to the part.

Sentinel-Observer: You are implying that the leaders at Silicon Techtronics, such as Mike Waterson and Ray Johnson, should have assumed responsibility for this incident right from the start. In addition, perhaps other individuals, such as Randy Samuels and Cindy Yardley, bear special burdens of responsibility.

Yoder: Yes, responsibility, not guilt. Guilt is a legal concept and the guilt or innocence of the parties involved, whether criminal or civil, will be decided in the courts. I guess a person bears responsibility for the death of Bart Matthews if his or her actions helped to cause the incident -- it's a matter of causality, independent of ethical and legal judgments. Questions of responsibility might be of interest to software engineers and managers, who might want to analyze what went wrong, so as to avoid similar problems in the future.

A lot of what has emerged in the media concerning this case indicates that Silicon Techtronics was a sick organization. That sickness created the accident. Who created that sickness? Management created that sickness, but also, employees who did not make the right ethical decisions contributed to the sickness.

Randy Samuels and Cindy Yardley were both right out of school. They received degrees in computer science and their first experience in the working world was at Silicon Techtronics. One has to wonder whether they received any instruction in ethics. Related to this is the question as to whether either of them had much prior experience with group work. Did they, at the time that they were involved in the development of the killer robot, did they see the need to become ethical persons? Did they see that success as a professional requires ethical behavior? There is much more to being a computer scientist or a software engineer than technical knowledge and skills.

Sentinel-Observer: I know for a fact that neither Samuels nor Yardley ever took a course in ethics or computer ethics.

Yoder: I suspected as much. Let's look at Randy Samuels. Based upon what I've read in your newspaper and elsewhere, he was basically a hacker type. He loved computers and programming. He started programming in junior high school and continued right through college. The important point is that Samuels was still a hacker when he got to Silicon Techtronics and they allowed him to remain a hacker.

I am using the term "hacker" here in a somewhat pejorative sense and perhaps that is not fair. The point that I am trying to make is that Samuels never matured beyond his narrow focus on hacking. At Silicon

Techtronics, Samuels still had the same attitude toward his programming as he had in junior high school. His perception of his life and of his responsibilities did not grow. He did not mature. There is no evidence that he was not trying to develop as a professional and as an ethical person.

Sentinel-Observer: One difficulty, insofar as teaching ethics is concerned, is that students generally do not like being told "this is right and that is wrong."

Yoder: Students need to understand that dealing with ethical issues is a part of being a professional computer scientist or software engineer.

One thing that has fascinated me about the Silicon Techtronics situation is that it is sometimes difficult to see the boundaries between legal, technical and ethical issues. Technical issues include computer science and the management issues. I have come to the conclusion that this blurring of boundaries results from the fact that the software industry is still in its infancy. The ethical issues loom large in part because of the absence of legal and technical guidelines.

In particular, there are no standard practices for the development and testing of software. There are standards, but these are not true standards. A common joke among computer scientists is that the good thing about standards is that there are so many to choose from.

In the absence of universally accepted standard practices for software engineering, there are many value judgments, probably more than in other forms of production. For example, in the case of the killer robot there was a controversy concerning the use of the waterfall model versus prototyping. Because there was no standard software development process, this became a controversy, and ethical issues are raised by the manner in which the controversy was resolved. You might recall that the waterfall model was chosen not because of its merits but because of the background of the project manager.

Sentinel-Observer: Did Cindy Yardley act ethically?

Yoder: At first, her argument seems compelling: she lied, in effect, to save the jobs of her coworkers and, of course, her own job. But, is it ever correct to lie, to create a falsehood, in a professional setting?

One book I have used in my computer ethics course is Ethical Decision Making and Information Technology by Kallman and Grillo.1 This book gives some of the principles and theories behind ethical decision making. I use this and other books to help develop the students' appreciation for the nature of ethical dilemmas and ethical decision making.

Kallman and Grillo present a method for ethical decision making and part of their method involves the use of five tests: the mom test, would you tell your mother what you did; the TV test, would you tell a national TV audience what you did; the smell test, does what you did have a bad smell to it; the other person's shoes test, would you like what you did to be done to you, and the market test, would your action be a good sales pitch?

What Yardley did fails all of these tests -- I think nearly everyone would agree. For example, can you imagine Silicon Techtronics using an ad campaign that runs something like this:

"At Silicon Techtronics, the software you get from us is bug free, because even if there is a bug, we will distort the test results to hide it, and you will never know about it. Ignorance is bliss!"

This shows that apparent altruism is not a sufficient indicator of ethical behavior. One might wonder what other unstated motives Ms. Yardley had. Could it be that personal ambition led her to accept Ray Johnson's explanation and his assurance that the robot was safe?

Sentinel-Observer: Are there any sources of ethical guidance for people who are confronted with an ethical dilemma?

Yoder: Some companies provide ethical guidelines, in the form of corporate policies, and there is such a document at Silicon Techtronics, or so I am told. I haven't seen it. An employee could also refer to ethical guidelines provided by professional societies, such as the ACM. Beyond that, he or she could read up on the subject to get a better feel for ethical decision making. Of course, one must always consult with one's conscience and innermost convictions.

Sentinel-Observer: Did Randy Samuels act ethically?

Yoder: Stealing software the way that he did was both unethical and illegal.

I think the most important issue with Randy Samuels has never been discussed in the press. I truly doubt that Samuels had the requisite knowledge that his job required. This kind of knowledge is called domain knowledge. Samuels had a knowledge of computers and programming, but not a very strong background in physics, especially classical mechanics. His lack of knowledge in the application domain was a direct cause of the horrible accident. If someone knowledgeable in mathematics, statistics and physics had been programming the robot instead of Samuels, Bart Matthews would probably be alive today. I have no doubt about that. Samuels misinterpreted the physics formula because he didn't understand its meaning and import in the robot application. It may be that management is partly responsible for the situation. Samuels might have told them his limitations and management might have said, "What the hell!"

Samuels had difficulty with group work, peer reviews and egoless programming. It is possible that he was trying to hide his lack of expertise in the application domain?

Sentinel-Observer: Did Ray Johnson act ethically?

Yoder: This 'Ivory Snow' business! The trouble with the Ivory Snow theory is that it was just a theory. If it were more than a theory and an actual methodology for keeping the likelihood of failure within statistically determined limits, like what is called "clean room software engineering," then there would be less culpability here.

Based upon the information that I have, the Ivory Snow theory was just a rationalization for getting flawed software out the door to customers on time. The Ivory Snow theory is only valid, ethically and professionally, if the customer is told of known bugs, or impurities, if we can use the soap jargon. In the case of Silicon Techtronics the Ivory Snow theory worked like this: we know it's not pure, but the customer thinks it is!

Of course, coercing Cindy Yardley the way Ray Johnson did was also not ethical. Did he believe what he told Ms. Yardley, namely that the robot was safe, or was that an out and out lie? If he believed that the robot was safe, why cover up with the false tests? If the user interface were so important as a last line of defense, why avoid more rigorous tests of the user interface?

Sentinel-Observer: What is your view of Mike Waterson in all this?

Yoder: If Johnson is the father of the Ivory Snow theory, Waterson is the grandfather. His demand that the robot be completed by a certain date or "heads would roll" might have caused Johnson to formulate the Ivory Snow theory. You see, it is apparent that Johnson thought that the delivery of Robbie to Cybernetics by the specified date was impossible unless the robot software had bugs.

In many regards I feel that Waterson acted unethically and irresponsibly. He placed Sam Reynolds in charge of the robot project, yet he, Reynolds, lacked experience with robots and modern user interfaces, Reynolds rejected the idea of developing a prototype, which might have allowed for the development of a better user interface.

Waterson created an oppressive atmosphere for his employees, which is unethical in itself. Not only did he threaten to fire everyone in the Robotics Division if the robot was not completed on time, he "eavesdropped" on private electronic mail communications throughout the corporation, a controversial right that some companies do claim.

My personal belief is that this kind of eavesdropping is unethical. The nature of e-mail is somewhat of a hybrid of normal mail and a telephone conversation. Monitoring or spying on someone else's mail is considered unethical, as is tapping a telephone. Indeed, these activities are also illegal under almost most circumstances. So, I believe it is an abuse of power to monitor employees the way that Waterson did.

Sentinel-Observer: Does the prosecutor have a case here?

Yoder: Against Randy Samuels?

Sentinel-Observer: Yes.

Yoder: I doubt it, unless she has information that has not been made public thus far. Manslaughter, to my understanding, implies a kind of reckless and irresponsible act, causing death of another. Does this description apply to Samuels? I think the prosecutor's best bet is to stress his lack of knowledge in the application domain if it can be shown that he did engage in a deliberate deception.

I read last week that 79% of the people favor acquittal. People are inclined to blame the corporation and its managers. Last night, one of the network news anchors said, "Samuels isn't a murderer, he's a product of his environment."

Sentinel-Observer: Could you restate your position on the matter of ultimate responsibility in the case of the killer robot?

Yoder: In my mind, the issue of individual versus corporate responsibility is very important. The corporation created an environment in which this kind of accident could occur. Yet, individuals, within that system, acted unethically and irresponsibly, and actually caused the accident. A company can create an environment which brings out the worst in its employees, but individual employees can also contribute to the worsening of the corporate environment. This is a feedback loop, a system in the classical sense. Thus, there is some corporate responsibility and some individual responsibility in the case of the killer robot.

Sentinel-Observer: Thank you, Professor Yoder.

---

1 This is an actual textbook from McGraw-Hill.