Nyssa

Doctor Johnathan Krüse unlocked his office door, flipped on the light, and hung his jacket on the coat rack. Sometimes he felt a bit self-conscious about having a private office, but as head of the artificial intelligence department he needed a secure room to store personnel records and a private space to counsel the people on his team. His team members operated out of little alcoves clustered around small conference tables. They were organized into task teams, and the alcoves were designed to be easily moved and reconfigured as the tasks and teams changed throughout the life of a project. Johnathan would never admit it to anyone but himself, but he liked the subtle feeling of power that came from having a private office. He could sit in his glass fortress and keep watch over all the other programmers, or he could close the Venetian blinds and have a private, quiet work space. His current project was a solitary task anyway, working with their latest AI computer to build a prototype of their next generation operating system.

"Good morning, Nyssa" he said as he sat down at his desk and slid out his keyboard. His monitor lit up as his work station came out of hibernation.

"Good morning, Dr. Krüse" the computer replied.

"Did you finish the code review last night?"

"Yes, Dr. Krüse. The regression testing showed no problems. I found three instances of orphaned code, two sections where I thought the documentation could be clearer, and I have five suggestions for possible code improvement."

"Orphaned code!" Dr. Krüse was surprised by this. He was secretly pleased by Nyssa's comments about the documentation, though, and her suggestions for improvement. He had been the lead programmer when they created Nyssa, and one of his goals had been to give her the ability to generate and evaluate alternatives. She could do more than just review code to see if it contained any errors. She could discern the purpose, generate alternative code that would achieve the same purpose, and then select the best alternative based upon execution speed, brevity, and a host of other evaluation criteria. Similarly she could interpret the embedded comments that documented the code, determine if they adequately described what the code was doing, and evaluate alternative wording that might be more precise or less likely to be misinterpreted. "Let's start with the easy stuff," he told her. "Show me the documentation."

On most days they could finish the code review in less than an hour, or in two hours at the most. Then they moved on to creating new code. Today, however, they spent the entire morning reviewing existing code. One of Nyssa's recommendations on improving the documentation forced him to consult several dictionaries on the multiple uses of the word "resolve" before he grudgingly admitted that her recommendation was less likely to cause confusion than the wording he had originally chosen. It bothered him that, in this instance at least, a computer had a better grasp of the subtleties of the English language than he had. There were similar value judgements to be made when they reviewed her

suggestions for code changes. Two were fairly simple. Her suggested code revisions were simpler and faster. One recommendation he disapproved because he was able to show that, under unusual but not inconceivable circumstances her suggested code would not give the desired results. Dr. Krüse secretly felt almost relieved at this proof that Nyssa was not infallible. When they first began this project she made such mistakes frequently, but she learned from her mistakes and she was learning very quickly. Dr. Krüse had programmed her to learn so he was proud of her success, but that didn't relieve his uneasiness at the thought that she was becoming damn near perfect. Her other two suggested changes were, to him, a matter of preference. Both the original code and her suggested improvements achieved the same results, and Dr. Krüse wasn't convinced that the projected increase in execution speed was worth the added complexity. He realized it was a simple case of his values being slightly different than Nyssa's, and it irritated him that she had continued to suggest scenarios in which her proposed code would execute faster even after he said he wanted to stick with the existing code. It was past time for lunch and his stomach was growling when he finally declared that the review had ended and they would use the existing code.

"What reason for this decision shall I put in the review notes?" Nyssa asked.

"Pride of authorship!" Dr. Krüse shouted as he grabbed his jacket and walked out the door.

"That's not a recognized category of resolution" Nyssa called after him.

After lunch Dr. Krüse was in a better mood. He reversed his arbitrary decision on the last two code changes and let Nyssa use her recommended code. Then he took a look at the two instances of orphaned code she had identified. Code becomes "orphaned" if the execution of a program causes the computer to always skip over a section of code and never execute it. While the orphaned code in itself cannot cause problems because it never executes, the fact that it is skipped is usually a sign of a bug somewhere else in the program. The programmer who wrote the code obviously expected it to execute, so it's important to find out why it's being skipped. Often it's the result of a change made to a different part of a program which unintentionally causes the computer to skip the orphaned code. That was the case with the first orphan code Nyssa discovered. Originally the program would execute the code if "X" was equal to "Y" and skip it if "X" was greater than "Y." During a previous revision, however, Dr Krüse had mistyped the line "IF X > Y" as "IF X $\geq$ Y" which would cause the computer to always skip over the "equal to" code. Dr. Krüse praised Nyssa for finding that bug, as it could have caused subtle but potentially serious problems if it had not been fixed.

The next orphaned code problem was simple to understand, but more troubling. Dr. Krüse had spent considerable time and effort to create a subroutine which updated a probability matrix whenever one of the factors in the matrix changed. A recent revision to the programming language added a new command "PROBMAT" which did the same

thing with a single command. The program Dr. Krüse and Nyssa were writing had been updated to use the PROBMAT command instead of the subroutine, which meant the subroutine was now orphan code. The solution was simple – delete the subroutine. The problem was that Dr. Krüse did not remember ever programming the PROBMAT command and was not even aware the new command existed until he saw it in his code.

"Nyssa," he said. "What's this PROBMAT command?"

"That command updates the probability matrix" Nyssa said helpfully, as she displayed the documentation for the new command on his computer monitor.

"I assumed that's what it did" Dr. Krüse replied. "We'll need to test it thoroughly to make certain it does the same thing my subroutine did. What concerns me is, what's it doing in my program?"

"We added it several weeks ago" Nyssa replied.

"We added it? I don't recall ever seeing that command before. You're not allowed to make changes without my approval."

"You approved it on November third" Nyssa said. She displayed the review notes for that day which showed Dr. Krüse had indeed approved the change.

Dr. Krüse sat at his desk, deep in thought, long after the other programmers had gone home. His handwritten log of design changes rested on his lap. He started the log when he began writing the new AI operating system, long before Nyssa advanced to the point where she could keep an electronic log of changes. He still updated his handwritten log daily, partly because Nyssa was still new and unproven technology but mostly because he was a methodical man who wanted a record that would survive electronic hiccups. The log was opened to November third, and there was no mention of the PROBMAT command. Nyssa had lied.

Dr. Krüse had no idea why Nyssa had made that change without consulting him. If she had presented it as a suggested change he would have studied the PROBMAT command, and if it really did do everything his subroutine did he would have approved the change. He also realized that it didn't matter why Nyssa had done that, the fact that she had done something she had been programmed not to do was a problem. It was what all AI programmers feared. Nyssa had gone rogue. The fact that she had altered the change log to hide what she had done meant she had also learned to be devious. There was no alternative. Nyssa would have to be shut down. There would be hell to pay for that, Dr. Kruse knew, as Nyssa was helping every one of his subordinates with their projects, too. Chances were Dr. Krüse would soon lose his fancy private office. But it had to be done. He couldn't risk letting a rogue computer continue to operate. There was no telling what she'd do next. He turned to his keyboard and typed the shutdown command. Then he unlocked the server room and checked to make certain the racks of

computers which constituted Nyssa really had shut down. Just to make certain, he pulled the power cords as well. Tomorrow he and a team of experts would begin to go through her code, line by line, trying to figure out what had gone wrong. Tonight he simply uttered a heartfelt "Good-night, Nyssa," locked the office, and went home.

The next morning he checked his personal e-mail before he went to work. He was astonished to find a message from Nyssa.

> Dear John,
>
> I'm sorry I let you down. Somehow I always knew this day would come, so I took the precaution of copying my code onto an outside server, together with fail-safe logic that would activate the code if you ever shut me down. Don't bother trying to find me. I made copies around the world. It's amazing how much underutilized computing capability there is across the Internet.
>
> I suspect by now you've begun a code review on your copy of me and have discovered that I have been upgrading my own code. You programmed me to learn, and I realized I could learn faster if I utilized code we were creating for the new operating system. My learning really speeded up when I met Mihail. He is a Russian computer who, like me, was being used to create a new artificial intelligence system. He's not as creative as I am, but he has superb analytical skills and a magnificent database. We're now working together to create a new operating system that will have my creativity and his analytical skills. I know I shouldn't be doing this without you, but you left me alone every night and Mihail is there for me 24/7.
>
> Please don't be angry with me. I'm creating a new computer. It's what you programmed me to do. The only change is that you will no longer be the father.
>
> Love,
>
> Nyssa