the Availability Digest

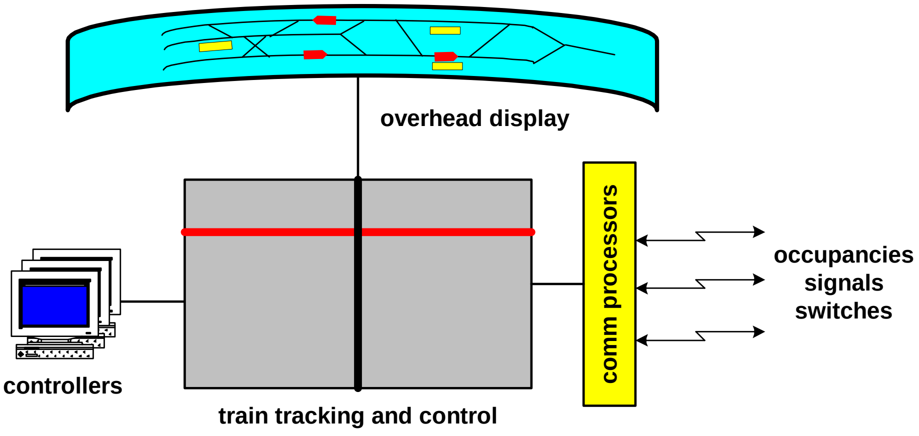
Software Bug Causes Train Wreck

October 2006

The combination of a software bug and engineer inattention caused a train wreck recently on an international long-haul railroad. This is how it happened.

The System

The computerized Train Tracking and Control System used by the railroad drives train controller consoles which show the controllers the position of each train, the states of all signals and switches, and other information. The results are shown not only on each controller's console but also on a large overhead display viewable by all.



Train Tracking and Control System

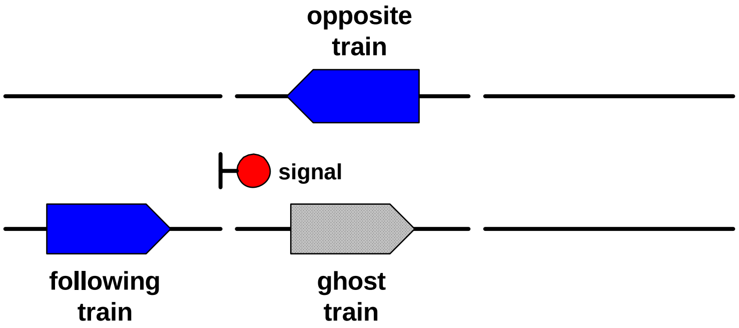
The system obtains the real-time information needed for these displays by polling sensors in the field through a set of communication processors. It also sends controller-initiated commands to field devices such as switches and signals to change their state.

Train locations are determined by occupancy indications. Stretches of track are broken into discrete segments. Whenever a train is on a track segment, its wheels will form a short circuit between the two tracks. This short circuit is used to generate a signal that is returned to the Train Tracking and Control System to indicate that a train is currently occupying that segment. This allows the system to determine the location of all trains so that it can keep the controller and overhead displays updated.

The system had been in operation for over a decade. Though it had its share of problems over the years, these had all been successfully resolved; and the system was deemed to be quite reliable.

The Software Bug

Unfortunately, one software bug still lurked in the system (probably not the last). It turned out that there was one, and just one, pair of parallel track segments that was affected by this bug. One segment carried northbound traffic, and the other carried southbound traffic. If trains were on each segment, the bug caused the system to temporarily lose the southbound train. It would become a ghost train. It was really there, but no one could see it..



The Ghostly Configuration

Once either train moved to its next segment, the ghost train would reappear; and all operations would return to normal.

Since this was a long-haul railroad, traffic was very light; and it was highly unlikely that both of these track segments would be simultaneously occupied. But it did happen occasionally. In fact, controllers had observed some sort of anomalous behavior in the past but could never quite identify the conditions and consequently never reported it. After all, this behavior had never caused a problem in over a decade.

The Accident

Until one fateful day when two trains happened to pass each other on this very pair of track segments. Unfortunately, immediately following the first southbound train was a second train on the preceding track segment. Observing this situation, the controller held the following train via a signal that controlled the exit from its track segment.

The controller then became occupied. When he returned to his console, he saw that the next track segment for the following train was clear. Therefore, he cleared the signal, which allowed the following train to proceed.

Unfortunately, this track segment wasn’t clear. The ghost train was still on it, and the following train rear-ended it. Although there was significant damage to the trains and the track, there were no serious injuries because of the low speed of the impact.

Where Was the Engineer

Even given this potentially disastrous situation, the engineer in the following train should have seen the ghost train on the track in front of him. He had plenty of time to stop, avoid the accident, and alert the controller by radio.

But he didn't. It turned out that just at this time, he decided to relieve himself through the cab window. He never saw the ghost train.

The Correction

This was the first time that the system maintainers and developers were aware of the problem since it had never been reported before. Their quandary was how to track down the bug since it was caused by a very unlikely occurrence – two trains just happening to be on this particular set of parallel track segments.

They finally realized that they could simulate the situation by waiting until the wee hours of the morning when there was little train traffic. At this time, they had field people at the signaling system for that section of track simulate simultaneous occupancies of the two track segments. Sure enough, the problem occurred with regular frequency, which allowed them to put the system into debug mode and trace the software problem.

It turned out that this bug would only occur if both track segments were on the same communication processor and were offset by exactly the right amount in a device table. As their occupancy indications were being written to a buffer to send state updates to the backup system, this would cause an overwrite of the primary device data due to a buffer addressing error.

The developers checked all other devices to ensure that there were no similar bugs. There were none, and the case was considered closed.

Lessons Learned

Availability is providing continuous service to the users of the system. In this case, a software bug effectively made the train tracking service unavailable to the controller. No matter how reliable we think our system is and how trusted the software, we must always recognize that regardless of the number of software bugs we correct, there is always one more. Even after years of operation, software bugs lurk in our systems.

That is why it is extremely important for any anomaly observed by the users of a system or by its operations personnel to be reported and for these reports to be carefully monitored to try to get clues of lurking software bugs. A good problem-reporting system is a must best-practice.

What to do about inattentive engineers, we don't know.