The Role of Software in Spacecraft Accidents

Nancy G. Leveson∗
Aeronautics and Astronautics Department
Massachusetts Institute of Technology

Abstract: The first and most important step in solving any problem is understanding the problem well enough to create effective solutions. To this end, several software-related spacecraft accidents were studied to determine common systemic factors. Although the details in each accident were different, very similar factors related to flaws in the safety culture, the management and organization, and technical deficiencies were identified. These factors include complacency and discounting of software risk, diffusion of responsibility and authority, limited communication channels and poor information flow, inadequate system and software engineering (poor or missing specifications, unnecessary complexity and software functionality, software reuse without appropriate safety analysis, violation of basic safety engineering practices in the digital components), inadequate review activities, ineffective system safety engineering, flawed test and simulation environments, and inadequate human factors engineering. Each of these factors is discussed along with some recommendations on how to eliminate them in future projects.

1 Introduction

Software is playing an increasingly important role in aerospace systems. Is it also playing an increasing role in accidents and, if so, what type of role? In the process of a research project to evaluate accident models, I looked in detail at a variety of aerospace accidents that in some way involved software.1,2 Many of the factors were in common across several of the accidents. To prevent accidents in the future, we need to attack these problems.

The spacecraft accidents investigated were the explosion of the Ariane 5 launcher on its maiden flight in 1996; the loss of the Mars Climate Orbiter in 1999; the destruction of the Mars Polar Lander sometime during the entry, deployment, and landing phase in the following year; the placing of a Milstar satellite in an incorrect and unusable orbit by the Titan IV B-32/Centaur launch in 1999; and the loss of contact with the SOHO (SOlar Heliospheric Observatory) spacecraft in 1998.

On the surface, the events and conditions involved in the accidents appear to be very different. A more careful, detailed analysis of the systemic factors, however, reveals striking similarities. Systemic factors are those that go beyond the specific technical causes, such as a flawed O-ring design in the Space Shuttle Challenger accident, and include the reasons why those failures or design errors were made. For Challenger, the latter include flawed decision making, poor problem reporting, lack of trend analysis, a “silent” or ineffective safety program,
communication problems, etc. Systemic factors are those related to the overall system within which the technical device is developed and operated.

A difficulty was encountered in that several of the accident reports implicated the software but then, for some unknown reason, never investigated the software development process in any depth to determine why the error was made. In some cases, it was possible to find information about the software development problems from sources outside the official accident investigation report. One conclusion from this observation might be that accident investigation boards must include more software experts and must more thoroughly investigate the reasons for the introduction of the errors and their lack of detection once introduced if we are to learn from our mistakes and improve our processes.

The accidents are first briefly described for those unfamiliar with them, and then the common factors are identified and discussed. These factors are divided into three groups: (1) flaws in the safety culture, (2) management and organizational problems, and (3) technical deficiencies.

2 The Accidents

Ariane 501
On June 4, 1996, the maiden flight of the Ariane 5 launcher ended in failure. About 40 s after initiation of the flight sequence, at an altitude of 2700 m, the launcher veered off its flight path, broke up, and exploded. The accident report describes what they called the “primary cause” as the complete loss of guidance and attitude information 37 s after start of the main engine ignition sequence (30 seconds after liftoff). The loss of information was due to specification and design errors in the software of the inertial reference system. The software was reused from the Ariane 4 and included functions that were not needed for Ariane 5 but were left in for “commonality.” In fact, these functions were useful but not required for the Ariane 4 either.

Mars Climate Orbiter (MCO)
The Mars Climate Orbiter (MCO) was launched December 11, 1998 atop a Delta II launch vehicle. Nine and a half months after launch, in September 1999, the spacecraft was to fire its main engine to achieve an elliptical orbit around Mars and to skim through the Mars upper atmosphere for several weeks, in a technique called aerobraking, to move into a low circular orbit. On September 23, 1999, the MCO was lost when it entered the Martian atmosphere in a lower than expected trajectory. The investigation board identified what it called the “root” cause of the accident as the failure to use metric units in the coding of a ground software file used in the trajectory models. Thruster performance data were instead in English units.

Mars Polar Lander (MPL)
Like MCO, Mars Polar Lander (MPL) was part of the Mars Surveyor program. It was launched January 3, 1999, using the same type of Delta II launch vehicle as MCO. Although the cause of the MPL loss is unknown, the most likely scenario is that the problem occurred during the entry, deployment, and landing (EDL) sequence when the three landing legs were to be deployed from their stowed condition to the landed position. Each leg was fitted with a Hall Effect magnetic sensor that generates a voltage when its leg contacts the surface of Mars. The descent engines were to be shut down by a command initiated by the flight software when touchdown
was detected. The engine thrust must be terminated within 50 milliseconds after touchdown to avoid overturning the lander. The flight software was also required to protect against a premature touchdown signal or a failed sensor in any of the landing legs.

The touchdown sensors characteristically generate a false momentary signal at leg deployment. This behavior was understood and the flight software should have ignored it. The software requirements did not specifically describe these events, however, and consequently the software designers did not account for them. It is believed that the software interpreted the spurious signals generated at leg deployment as valid touchdown events. When the sensor data was enabled at an altitude of 40 meters, the software shut down the engines and the lander free fell to the surface, impacting at a velocity of 22 meters per second and was destroyed.

**Titan/Centaur/Milstar**

On April 30, 1999, a Titan IV B-32/Centaur TC-14/Milstar-3 was launched from Cape Canaveral. The mission was to place the Milstar satellite in geosynchronous orbit. An incorrect roll rate filter constant zeroed the roll rate data, resulting in the loss of roll axis control and then yaw and pitch control. The loss of attitude control caused excessive firings of the reaction control system and subsequent hydrazine depletion. This erratic vehicle flight during the Centaur main engine burns in turn led to an orbit apogee and perigee much lower than desired, placing the Milstar satellite in an incorrect and unusable low elliptical final orbit instead of the intended geosynchronous orbit.

The accident investigation board concluded that failure of the Titan IV B-32 mission was due to an inadequate software development, testing, and quality assurance process for the Centaur upper stage. That process did not detect the incorrect entry by a flight software engineer of a roll rate filter constant into the Inertial Navigation Unit software file.

The roll rate filter itself was included early in the design phase of the first Milstar spacecraft, but the spacecraft manufacturer later determined that filtering was not required at that frequency. A decision was made to leave the filter in place for the first and later Milstar flights for “consistency.”

**SOHO (SOlar Heliospheric Observatory)**

SOHO was a joint effort between NASA and ESA to perform helioseismology and to monitor the solar atmosphere, corona, and wind. The spacecraft completed a successful two-year primary mission in May 1998 and then entered into its extended mission phase. After roughly two months of nominal activity, contact with SOHO was lost June 25, 1998. The loss was preceded by a routine calibration of the spacecraft’s three roll gyroscopes and by a momentum management maneuver.

The flight operations team had modified the ground operations procedures as part of a ground systems reengineering effort to reduce operations costs and streamline operations, to minimize science downtime, and to conserve gyro life. Though some of the modifications were made at the request of the SOHO science team, they were not necessarily driven by any specific requirements changes. A series of errors in making the software changes along with errors in performing the calibration and momentum management maneuver and in recovering from an emergency safing mode led to the loss of telemetry. Communication with the spacecraft was eventually restored after a gap of four months.