Crash and Burn – The Story of the Ariane 5 Explosion

Kristen Hieronymus

UTD SYSM6309 Advanced Requirements Analysis

Summer 2013

 On June 4, 1996, after $7 billion in development costs, France’s Ariane 5 rocket exploded 40 seconds after the initiation of the flight sequence, 33 seconds after liftoff, resulting in the loss of the rocket and its payload of 4 satellites. The short version of the story is that it was due to a conversion of a 64-bit floating point value to a 16-bit signed integer and the overflow condition that resulted. But how that came to occur is much more interesting. This paper will attempt to tell the full story of how this came about, and what could have been done differently to prevent it.

 France was trying to prove its technical competence in the Aerospace industry, and fend off international competition, which was increasing at the time.[[1]](#endnote-1) France wanted to be the European leader in this economically important industry as Europe moved toward the eventual European Union.[[2]](#endnote-2) Military expenditures were falling, and commercial applications were on the rise.[[3]](#endnote-3) So there was perhaps more pressure than in the past to streamline costs and maintain schedule.

 This was the maiden flight of the Ariane 5. A lot rode on that flight -- specifically, four satellites to be used to measure the earth’s magnetosphere.[[4]](#endnote-4) But more than that – future business with not only French but other European companies who would depend on rocket launches to get their satellites into orbit. And a failure would mean an economic impact greater than just the mere loss of the rocket and its uninsured satellites.[[5]](#endnote-5)

 According to the “Report by the Inquiry Board,”[[6]](#endnote-6) The rocket self-destructed, as designed, at H0+40 seconds (H0 being main cryogenic engine ignition), due to detection that the boosters had begun to detach. The boosters began to detach at H0+39 seconds, due to aerodynamic forces caused by an attempt to change direction by 20 degrees. The direction change was caused by full nozzle deflections of the booster and main engine. These deflections were due to commands received from the On-Board Computer software. The software made these commands due to calculations made. These calculations were made based on an incorrect value. It arrived at this incorrect value because of a misinterpretation of the data from the active Inertial Reference System (SRI, due to its French name). The SRI was actually reporting a diagnostic bit pattern rather than flight data; this bit pattern indicated that the system was shutting down, but this error condition had not been accounted for in the design of the On-Board Computer system or tested in interface tests.

 The SRI had actually failed over to the back-up SRI at H0+36.7 seconds, and the back-up had failed for the same reason 0.05 seconds later. The SRI had failed due to a software exception. The software exception was caused by trying to convert a 64-bit floating point value into a signed 16-bit integer, but the value was too large to fit, resulting in an Operand Error. That conversion was not protected by error-handling code, in spite of the original analysis back on Ariane 4 indicating that it was one of seven such conversions which should be so protected.

The larger-than-expected value was for a value called “Horizontal Bias,” which is related to the horizontal component of velocity sensed by the platform, which should be very small before lift-off. The software which uses this value is for alignment of the strap-down inertial platform, and computes meaningful data only prior to lift-off. But, on the Ariane 4, this function was left on for 47 seconds after liftoff as a workaround for a possible temporary delay before lift-off which was not otherwise accounted for in the time-tracking. The reason for allowing the count-down to continue rather than start over was due to the 45-minute reset of other hardware if the launch had to be stopped and restarted. This workaround never negatively affected the Ariane 4 launches, because the horizontal component of velocity on the Ariane 4 is only 20% of that on the Ariane 5, due to its different trajectory.

That is the technical reason for the failure. However, there were problems with the way the system tests had been conducted prior to launch. Before that, there were requirements issues which led to the problem in the first place. There were also problems in communications among the various players who contributed components to the system.

The system level tests had not included the SRIs. This is partly due to the impossibility of including them entirely without actually launching the system, due to needing completely realistic data from the sensors. But the original test plans had called for including the SRIs and providing them simulated inputs from what would otherwise be the real accelerometers and Ring Laser Gyros. However, in the end, it was decided to simply accept the SRIs as previously verified by way of successful Ariane 4 launches, and to use SRI simulators instead when testing the rest of the system. It was for this reason that the failure of the SRIs to accommodate Ariane 5 horizontal velocity values was not discovered. Also, the diagnostic bit pattern from the SRI as it shut down was never tested with the rest of the system.

The requirements themselves were at fault in several ways. First, the trajectory of the Ariane 5 was never documented in the functional requirements. Second, the diagnostic bit pattern and the need for the commanding software to handle it was not documented in the interface document. Third, there was an assumption that software never errored, and that the only failures required to be handled from software components were ones due to hardware failures – hence, the shutdown and failover of the unit. There should have been a requirement to detect possible error conditions in the software and to generate “best estimate” values. Fourth, there should have been a requirement to not have software running that served no purpose at the moment; this would have guaranteed that the function which served no value after lift-off would not have caused problems. And finally, the need to test the system with the SRI during system-level tests was not listed in the requirements. If any of these requirements had been documented, the Ariane 5 would not have crashed. Or at least, it would not have crashed for this reason, but that is another story, an allusion to which can also be found in the Board of Inquiry’s findings regarding the hydraulic pressure in the actuators for the main engine nozzle.

 Recommendations of the board for future prevention of this type of error included fixing the flaws in the requirements as indicated above. It also included a recommendation to review all flight software and embedded software for implicit assumptions made by the code, and a recommendation that, wherever possible, exceptions should be confined to the originating function rather than propagating them to the calling or interfacing software to handle. The final recommendation was for better communication between participants in the system project, to include reviews of specifications, code, and what they call “justification documents.” These justification documents contain the rationale for implementation decisions, and the recommendation was to give them the same attention and maintenance as the code itself.

 I would add a few additional recommendations, though. The first recommendation would be to perform an Analytical Hierarchy Process analysis of the requirements when choosing between requirements which seem to not be able to be met simultaneously. The AHP is based on pairwise comparisons of the relative importance of requirements. Had such a process been implemented across all the system requirements, it would have been a herculean task. However, it could at least be used for requirements which are identified as perhaps being in conflict. The reason that the Horizontal Bias value was not protected with code which could have detected the out of range value and made a “best estimate” adjustment for it was because of a competing requirement which set the maximum target workload at 80% of the CPU’s capacity. There had been seven variables which had been identified as needing range protection, but only four were given such protection due to trying to not exceed this 80% threshold. It would have been preferable to protect all seven of the variables. Better to meet all the functional requirements and slightly exceed a desired performance metric, than to meet the desired performance metric but fail the functional requirement. After all, the reason the threshold was set at 80% rather than 100% was to provide margin, and the impact of slightly exceeding 80% would have been far less catastrophic than an Operand Error causing the SRI to shut down.

 A second recommendation I would add is to treat the reuse of modules more carefully. When reusing a system component originally designed for one system, on a different system, special care needs to be given to reexamine all the assumptions on which that the code was based. Reuse of modules – particularly when they can be reused in total without modification – is absolutely a way to save money and time, but it is not without its risks. Integration testing of the reused component, within the context of its new system, is very important, and care must be given to exercising all of its interface values – the error cases as well as the “happy path” cases. In this case, thorough integration testing would have highlighted the lack of code to handle the diagnostic bit pattern.

 A final recommendation I would add is to more thoroughly document assumptions via comments in the code when it is originally written. No one understands the assumptions of a piece of code better than the author does when writing it. While it is advisable to document these assumptions in comments in the code at the point of use, it is also advisable to include critical assumptions in method prologues, since this is information that the caller would need to know when deciding whether and how to call a method. In this particular case, according to the Inquiry Board, there were no comments at all in the code which explained why there was no protection logic for the three variables or otherwise addressed the assumptions regarding their possible range of values.

 Interestingly, the spin story on the Ariane 5 disaster continues to this day. The Ariane 5 rocket is still used today to launch satellites. In fact, it has been used for sixty-nine launches – the vast majority being successful, and the most recent being July 23, 2013. However, the Wikipedia page for the Ariane 5 lists this June 4, 1996 launch as a “test launch.”[[7]](#endnote-7) However, test launches do not carry expensive payloads.

1. ECORYS Nederland BV. “FWC Sector Competitiveness Studies – Competitiveness of the EU Aerospace Industry with a focus on Aeronautics Industry.” (2009, Dec 15). Retrieved from <http://ec.europa.eu/enterprise/sectors/aerospace/files/aerospace_studies/aerospace_study_en.pdf> [↑](#endnote-ref-1)
2. Cook, C. “The Aerospace Industry: Its history and how it affects the U.S. economy.” (2013). Retrieved from: <http://www.yale.edu/ynhti/curriculum/units/1990/7/90.07.06.x.html> [↑](#endnote-ref-2)
3. Carrinacazeaux, C. and Frigant, V. “The internationalization of the French Aerospace industry in the 1990s: a break with the past? (In French).” (2006). Retrieved from: <http://cahiersdugres.u-bordeaux4.fr/2006/2006-15.pdf> [↑](#endnote-ref-3)
4. Wikipedia. “Cluster II (spacecraft).” (Last modified 2013, Jul 23). Retrieved from: <http://en.wikipedia.org/wiki/Cluster_mission> [↑](#endnote-ref-4)
5. Gleick, J. “A bug and a crash: sometimes a bug is more than a nuisance.” (1996, Dec). Retrieved from: <http://www.around.com/ariane.html> [↑](#endnote-ref-5)
6. Lions, J.L. “Ariane 5 Flight 501 Failure: Report by the Inquiry Board.” (1996). Retrieved from: <http://www.ima.umn.edu/~arnold/disasters/ariane5rep.html> [↑](#endnote-ref-6)
7. Wikipedia. “Ariane 5.” (Last modified 2013, Aug 1). Retrieved from: <http://en.wikipedia.org/wiki/Ariane_5> [↑](#endnote-ref-7)