6 Dependable systems evolution

Jim Woodcock, University of Kent

CNN News, June 4th, 1996: ìThe Ariane 5 rocket was destroyed seconds after it took off,î a spokesman for Arianespace said today.

The European Space Agency spent 10 years and $7 billion developing the giant rocket that can put six tonnes of satellite into orbit. So what went wrong? A piece of software that had been extensively tested, and which worked well in Ariane 4, caused a simple arithmetic error in Ariane 5. This led the guidance system to make a dramatic, but unnecessary change to the rocket's course. A backup system took over control, and repeated the same mistake. The rocket was destroyed before aerodynamic forces ripped it apart.

The millions of software faults that afflict ordinary PC users every day are less dramatic, but much more familiar, and collectively more expensive even than the loss of Ariane 5. Each software fault offers an opening to a virus. Just one infection like the Code Red virus was estimated to have caused losses of $4 billion world wide. It all adds up. In May 2002, the US department of Commerce estimated that the total cost to the US economy of avoidable faults in software is $60 billion.

A computing system is *dependable* if we can justify the reliance that we place on the things that it does for us. Evidence is needed in advance to back up any promises about a computer system's future service, and this evidence must be scientifically rigorous. At the moment it's very expensive and difficult to produce such evidence. Exhaustive testing is usually out of the question: checking every test case in an aircraft control system would take thousands of computers hundreds of years to complete. A more sophisticated approach to checking correctness is based on mathematics. This was used to show, for instance, that certain smart cards don't let crooks counterfeit money. But even these techniques can be costly to use. It took over 100 man-years of effort to produce the safety case for the Sizewell B nuclear power station.

We need the scientific foundation to be able to build systems whose dependability can be justified, even in the face of the most extreme threats. We need to be able to put systems in inaccessible places, knowing that they will continue to work over decades. We need to be able to build very large scale systems with controllable costs and risks. We need the ability to evolve such systems rapidly, at costs which reflect the size of change, not the scale of the system. The scientific and technical advances that we hope will result from this Grand Challenge could be the basis and trigger of a radical change in the practice of developing computer systems. We want suppliers to sell software for its safety, security, and reliability, as well as for its functionality. Perhaps in the future:

- Commercial and industrial-scale software can be developed to be truly dependable, at lower cost and with less development risk than today.
- The vulnerabilities in existing computer systems can be discovered and corrected more effectively, improving their dependability.
- Dependable systems can be evolved dependably including, for a class of applications, just-in-time creation of required services.

The various technologies are now sufficiently advanced that this project can be planned with a reasonable prediction of success. Within fifteen years we hope to produce prototype tools and examples of their successful use that are sufficiently persuasive to encourage the industry to make these improvements.