Dependable systems

Master of Science in Embedded Computing Systems

Prof. Cinzia Bernardeschi
Department of Information Engineering
University of Pisa
cinzia.bernardeschi@ing.unipi.it
2017-2018

Dependable computer-based systems

System dependability is the ability of the system to deliver the expected functionality during its operational life

Dependability is important in safety-critical systems, systems whose failure or malfunction may result in death or serious injury to people, loss or serious damage of equipment, or environmental harm.

Future safety-critical systems will be more automated and more dependent on computers than today's systems

Computers are increasingly used in safety-critical systems:

- transport (automotive, railways, aerospace, …)
- medicine
- process control
- ….

Transport systems
Railway Interlocking system (safe movements of trains)

mechanical (route settings by levers) -> electrical (electro-mechanical or relay-based) -> electronic/computer-based

Computer-based interlocking:

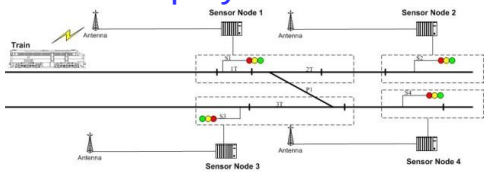
- wired networks of relays replaced by software logic running on special-purpose control hardware
- logic is implemented by software rather than hard-wired circuitry
- facilitates modifications by reprogramming rather than rewiring

https://en.wikipedia.org/wiki/Interlocking

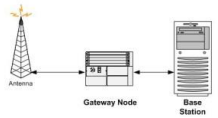
Short signal blocks on a subway system (Toronto) .
A train has just passed the most distant, leftmost signal, and the two most distant signals are red (*stop and stay* aspect). The next closest signal is yellow (*proceed with caution*), and the nearest signal shows green (*proceed*).

https://en.wikipedia.org/wiki/Interlocking

Supervisory control and data acquisition (SCADA) systems to view the location of trains and the display of signals.

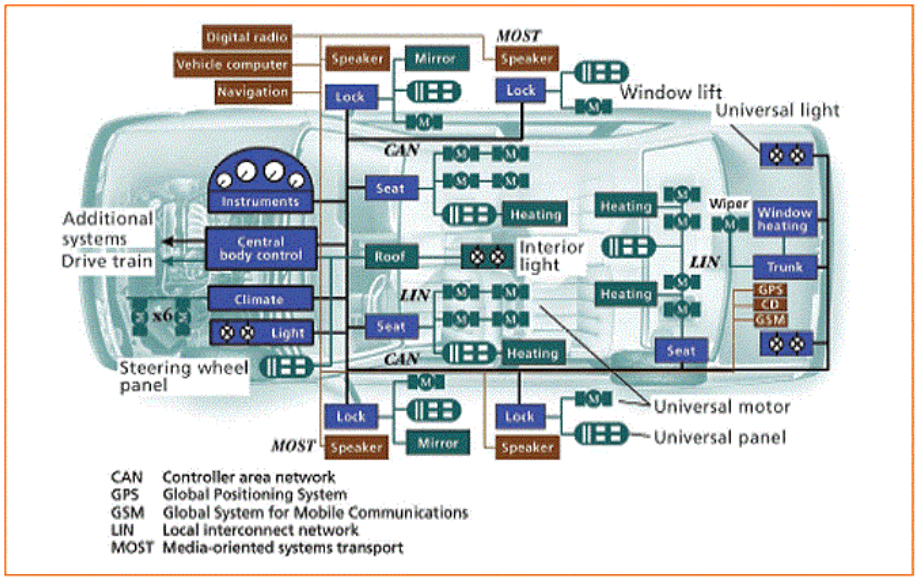


Railway Signaling using WSNs

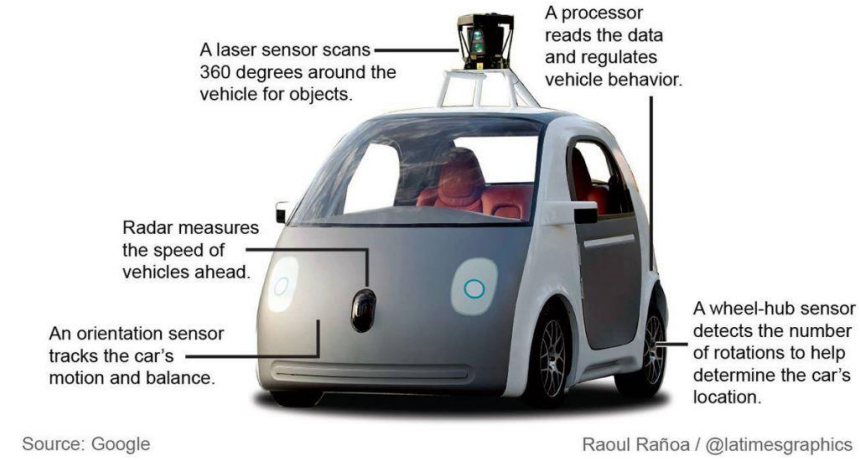


Transport systems
Automotive

Sensing and Computing in Cars



Autonomous Vehicles (capable of sensing its environment and navigating without human input)



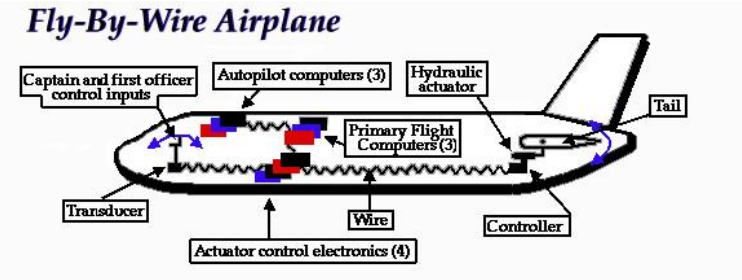
Over 80 different embedded processors, interconnected with each other.

Key ECUs (Electronic Control Unit):

- Engine Control Modul (ECM)
- Electonic Brake Control Module (EBCM)
- Transmission Control Module (TCM)
- Vehicle Vision System (VVS)
- Navigation Control Module (NCM)
- …

Array of sensors needed to provide the autonomous system with situational awareness about the physical world. Embedded processors use this information to make appropriate decisions about what actions the autonomous system should perform.

Aerospace



http://www.aviationcoaching.com/wp-content/uploads/2015/08/fly-by-wire-system.jpg

Earliest aircraft: controlled by the pilot using the steel cables, pulleys and hydraulic actuators

Fly-by-wire (FBW) system: all commands and signals are transmitted electrically along wires.

These signals are sent to **flight-control computers (FCS)** that reconvert the electrical impulses into instructions for control surfaces like wing flaps or the tail.

Potentiometers, or pots, in the control surfaces measure their position and transmit that data back to the flight computer. This technology accounts as one of the greatest breakthrough in the aviation.

Flight computers can be programmed to carry out adjustments to control surfaces automatically.

Air traffic Control

http://www.adp-i.com/en/our-solutions/airport-expert-appraisals/air-navigation

Air Traffic Control (ATC) is a service provided by ground-based controllers who are responsible for maintaining a safe and efficient air traffic flow.

ATC is transitioning to use of the Global Positioning System for Navigation and precision approaches

Future generation of ATC:
Airborne Self-Separation

an operating environment where pilots are allowed to select their flight paths in real-time.

Main challenge:
coordination between aircrafts within a dynamic environment, where the set of surrounding aircraft is constantly changing

Medical devices

PCA devices

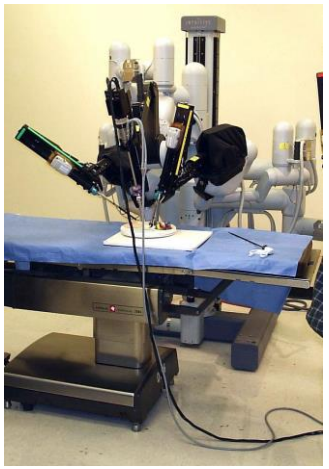
A patient-controlled analgesia (PCA) infusion pump, configured for intravenous administration of morphine for postoperative analgesia

Implantable Cardiac Pacemakers

The bulk of the device contains its battery and electronic control systems. The leads detect the heart's electrical activity, transmit that information to the artificial pacemaker's electronics for analysis and, if the natural activity is deemed irregular, deliver an electrical charge from the artificial pacemaker's batteries that causes the cardiac muscle to contract, pacing the pumping of the heart.

Robotic Surgical Systems

Da Vinci Surgical System: Approved by the Food and Drug Administration (FDA) in 2000, it is designed to facilitate complex surgery using a minimally invasive approach, and is controlled by a surgeon from a console.



http://www.davincisurgery.com/da-vinci-surgery/da-vinci-surgical-system/

Digital Instrumentation and Control

A Digital Control System samples feedback from the system under control and issues commands to the system in an attempt to achieve some desired behaviour

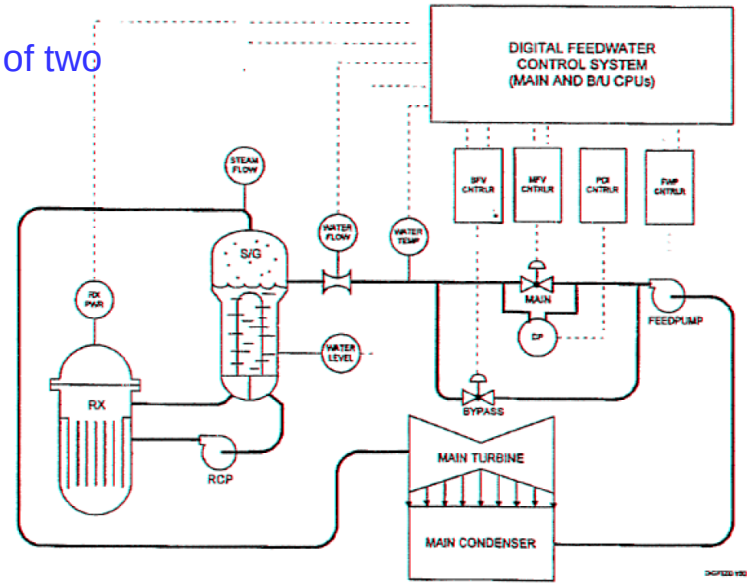
Digital I&C: analog and mechanical parts are replaced by CPUs and software

One of the reactor coolant loops with its associated Digital FeedWater Control System

Nuclear Power Plant (NPP) has two units, each consisting of two reactor coolant loops

For each reactor coolant loop:
- Reactor coolant pump (RCP)
- Steam Generator (S/G)

Main components of the FeedWater Systems (FWS)
- FWPs (FeedWater Pumps)
- MFRVs (Main FeedWater Regulating Valves)
- BPFRVs (Bypass FeedWater Regulating Valves)



''Traditional Probabilistic Risk Assessment Methods for Digital Systems'', U.S. Nuclear Regulatory Commission, NUREG/CR-6962, October 2008

Dependable Systems

For a computer based safety-critical system, the safety of the system depends strongly on its computers.

Faults are unexpected events that may compromise the system functionality

Faults in computer systems:

- hardware faults
- software faults

General questions:

how to build dependable computer-based systems ?
can we justifiably trust the dependability of such systems?

Computer-based systems

Hw and sw systems relaying on hidden components

- A system is as strong as its weakest component
- Design failures: unintended system function due to incomplete problem description
- Human failure: the system includes the operator.
- Hursh environment (wide temperature range , …)

Computer failures differ from failures of other equipment

- Subtler failures than “breaking down” or “stopping working”, ..
- The computer is used to store information: there are many ways information can be wrong, many different effects both within and outside the computer
- Small hidden faults may have large effects (digital machine)

«Computer designers and programmers would be students of reliability and so do computer system users»

D. P. Siewiorek R.S. Swarz, Reliable Computer Systems, Prentice Hall, 1992

Examples of safety-critical systems failures

Ariane 5 - Flight 501

Ariane 5 is a European heavy lift launch vehicle that is a part of the Ariane rocket family, an expendable launch system used to deliver payloads into geostationary transfer orbit or low Earth orbit .

“ The morning of the 4th of June 1996 was partially cloudy at Kourou in Guyana as the European Space Agency (ESA) prepared for the first launch of the French-built Ariane 5 rocket. The rocket lifted off at 09:34. Just 37 seconds later, the rocket veered on its side and began to break up. The range safety mechanism identified the impending catastrophe and initiated explosive charges that blew up the rocket to prevent further damages and possible casualties. An investigation by the ESA determined that the accident was caused by a software ‘bug’. This is the story of that bug. ”

The Bug That Destroyed a Rocket, Mordechai Ben-Ari. *SIGCSE Bulletin*, n. 2, 2008

ARIANE 5 Flight 501 failure, Inquiry Board Report, 1996
(http://sunnyday.mit.edu/accidents/Ariane5accidentreport.html)

- Numbers coming from the inertial guidance system were wrong

- on board computer mistakenly though the rocket needed a course change

- the computer started to reconfigure the boosters accordingly

- this reconfiguration physically disrupted the boosters from the rocket body

- self-destruction started because boosters were ripping from the rocket on a populated area (explosion reduced the risk for population)

Ariane 5

The data the computer collected were actually a diagnostic error message

The diagnostic message was output by some other subsystem that had an overflow error. The control systems took this data.

There was a backup inertial system, but the backup system, an identical redundant unit, failed in the identical manner a few milliseconds before. It was running the same software.

Original problem:

64 bit number too big to be converted to 16 bit

In Arianne 5, they reused the same software but they replaced the sensors.

"new sensor and old software that read these sensors"

The old software was not able to deal with large numbers produced by new sensors, we had an overflow problem.

This was never tested due to budget constraints.

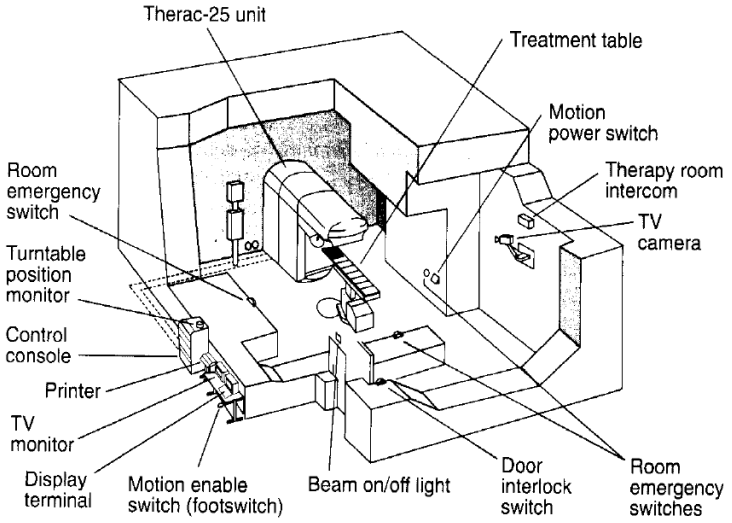
"The subsystem that had the problem was a part dedicated to align the system before lunch – it should have been turned off!"

Therac 25

Therac 25 is a machine for radiation therapy
(to treat cancer)

It was involved in at least six accidents between 1985 and 1987, in which patients were given massive overdoses of radiation

Because of concurrent programming errors, it sometimes gave its patients radiation doses that were hundreds of times greater than normal, resulting in death or serious injury.



Functional principle:

"scanning magnets" are used to spread the beam and vary the beam energy

Two operation modes:

1) high power spreaded beam or

2) low power focus beam

Accident: high power mode (a lot of energy) without spreader plate activated
-> caused by software flaws

Therac 25

These accidents highlighted the dangers of software control of safety-critical systems, and they have become a standard case study in health informatics and software engineering

https://en.wikipedia.org/wiki/Therac-25

There was a software bug in the software control and there was a problem in the coordination between plain position sensors and control software

The defect was as follows: a one-byte counter in a testing routine frequently overflowed; if an operator provided manual input to the machine at the precise moment that this counter overflowed, the interlock would fail.

Previous models had hardware interlocks in place to prevent this, but Therac-25 had removed them, depending instead on software interlocks for safety. The software interlock could fail due to a race condition

A commission concluded that the primary reason should be attributed to the bad software design and development practices, and not explicitly to several coding errors that were found. In particular, the software was designed so that it was realistically impossible to test it in a clean automated way

Medical Devices: The Therac-25, Nancy Leveson, U. of Washington. In *Safeware: System Safety and Computers*, Addison-Wesley, 1995

Patriot Missile Launcher (US Military)

During Gulf war a Scud missile broke through
the Patriot anti-missile defense barrier
and hit American forces killing 28 people
and injuring 98

https://it.wikipedia.org/wiki/MIM-104_Patriot

Patriot missile launcher:

- mobile missile launcher
- radar sweeps the sky for threats.
  If an incoming threat is found,
  the launch of a missile is guided
  by the control station
- designed for a few hours of operations

The launcher was used for Scud defense operation, never designed for it, failed to intercept a Scud missile

Use of the system in an unexpected mode of operation. The station did not move, the system was at the same position for many days

Patriot Missile Launcher (US Military)

There was an aging problem in the software.The software run too long and started getting overflow, inaccuracy, …

Target velocity and time demanded as real values, was stored as 24 bit integers

with the advent of time this conversion loses accuracy (> 100 hours)

tracking of enemy missiles becomes inaccurate

The software problem was already known, and the update was delivered the next day

Software related recalls in medical devices

Over the last few years, a series of device recalls were mandated by the FDA for dangerous or defective medical devices

Recalls have increased since 2006 as shown here, due to more sophisticated software

The Biomedical Instrumentation & Technology journal
Software-Related Recalls: An Analysis of Records
Lisa K. Simone, a *biomedical and software engineer with the Center for Devices and Radiological Health at the U.S. Food and Drug Administration*

Year	Total Recalls	Software-Related Recalls	Percent
2005	604	84	13.9%
2006	663	119	17.9%
2007	638	119	18.7%
2008	847	192	22.7%
2009	782	146	18.7%
2010	981	147	15.0%
2011	1,277	315	24.7%

Percentage of Recalls Related to Software

User interface issues were one of the main causes of these recalls. The issues were observed across multiple device manufacturers and device types, and appear to be related to poor device design and engineering.

https://chimedblog.wordpress.com/2014/03/27/video-medical-device-training-user-interfaces-design-issues-and-avoiding-medical-error/

discuss user interface issues commonly observed in commercial devices such as infusion pumps, ventilators, and patient monitors

ExoMars 2016: Trace Gas Orbiter (TGO) and Schiaparelli, the entry, descent and landing demonstrator module

http://exploration.esa.int/mars/47852-entry-descent-and-landing-demonstrator-module/

The TGO was launched with Schiaparelli in a composite configuration to Mars. Three days before reaching the atmosphere of the Red Planet, Schiaparelli will be ejected from the TGO towards the planet.

Progress has been made in investigating the ExoMars Schiaparelli anomaly of 19 October. A large volume of data recovered from the Mars lander shows that the atmospheric entry and associated braking occurred exactly as expected

The parachute deployed normally at an altitude of 12 km and a speed of 1730 km/h. The vehicle's heatshield, having served its purpose, was released at an altitude of 7.8 km.

As Schiaparelli descended under its parachute, its radar Doppler altimeter functioned correctly and the measurements were included in the guidance, navigation and control system.

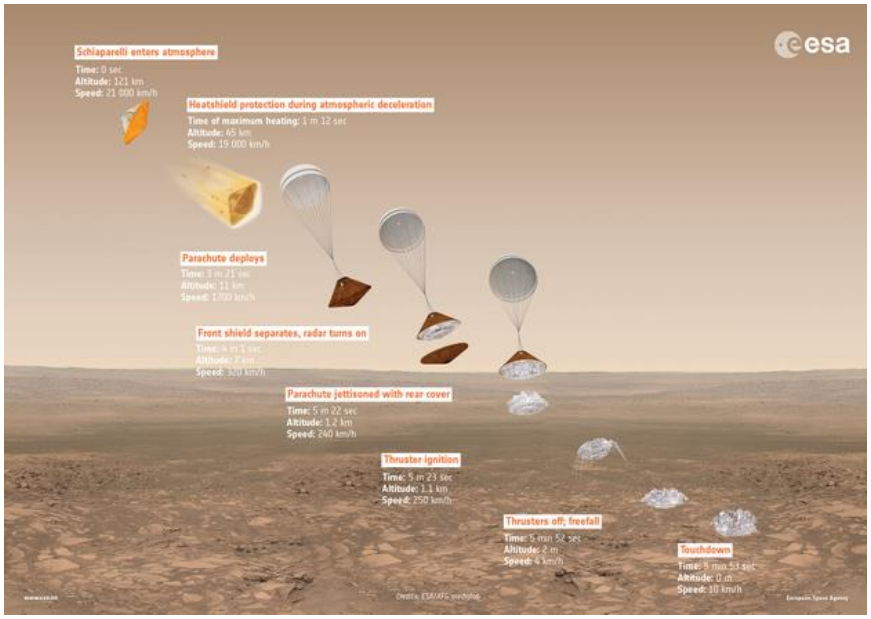
ExoMars 2016: Trace Gas Orbiter (TGO) and Schiaparelli, the entry, descent and landing demonstrator module

http://exploration.esa.int/mars/47852-entry-descent-and-landing-demonstrator-module/

However, saturation – maximum measurement – of the Inertial Measurement Unit (IMU) had occurred shortly after the parachute deployment.
The IMU measures the rotation rates of the vehicle. Its output was generally as predicted except for this event, which persisted for about one second – longer than would be expected.

When merged into the navigation system, the erroneous information generated an estimated altitude that was negative – that is, below ground level.

This in turn successively triggered a premature release of the parachute and the backshell, a brief firing of the braking thrusters and finally activation of the on-ground systems as if Schiaparelli had already landed. In reality, the vehicle was still at an altitude of around 3.7 km.



The computer made a mistake!

Hw fault / sw fault / specification fault/ design fault/ ….. ?

ExoMars 2016: Trace Gas Orbiter (TGO) and Schiaparelli, the entry, descent and landing demonstrator module

http://exploration.esa.int/mars/47852-entry-descent-and-landing-demonstrator-module/

This behaviour has been clearly reproduced in computer simulations of the control system's response to the erroneous information.

"This is still a very preliminary conclusion of our technical investigations," says David Parker, ESA's Director of Human Spaceflight and Robotic Exploration.

"The full picture will be provided in early 2017 by the future report of an external independent inquiry board, which is now being set up, as requested by ESA's Director General, under the chairmanship of ESA's Inspector General.

Toyota Unintended Acceleration

Sudden acceleration is one of the most deadly automotive defects in history.

https://www.viva64.com/en/b/0439/

The issue became public in August 2009 after an accident in San Diego, Calif., killed a family of four. The mat entrapped the gas pedal, accelerating the car at full throttle. The incident occurred after the National Highway Traffic Safety Administration (NHTSA) had opened a defect investigation into the ES350 over that issue in 2007 and identified other Lexus models that might be similarly defective.

http://www.cbsnews.com/news/toyota-unintended-acceleration-has-killed-89/
http://mashable.com/2014/03/19/toyota-lied-aceleration recall/#KHnZ2PBWjsq3

- Car's electronics cause the throttle to go wide open, making it impossible for the driver to return the car to idle if it remains in gear

- severely limits the ability of the brakes to bring the vehicle under control -- leaving the unsuspecting driver at the mercy of a runaway car.

- Thousands of people, including drivers, passengers, and innocent bystanders, have been killed or seriously injured in sudden acceleration accidents.

See the set of slides of Prof. Phil Koopman (Carnegie Mellon University)

https://users.ece.cmu.edu/~koopman/pubs/koopman14_toyota_ua_slides.pdf

Dependable computer-based systems

In real world, dependability problems are really subtle.

- There is a root cause that evolves. It propagates into the system, something happens in a subsystem, something else happins in another subsystem, …., and then we have a failure.

This is very complicated to predict; in this case fault tolerance can be usefull

System are designed to endure within a given operational conditions. It is very hard to anticipate the operational conditions correctly

In safety critical systems community, people are forced to document and publish the problems (accidents)

You have to publish problems, to document them, to analyse them to be sure that nobody else has the same problem again

In this way, data for dependability research can be collected