

Critical Systems Engineering

- Processes and techniques for developing critical systems

Recommended reading for this part of the course

Software Safety in Embedded Computer Systems. N.G. Leveson, Comm. ACM, 34 (2), February 1991. (supplied as handout)

A good introduction to safety issues - the focus is on software safety but general system safety issues are also covered.

Software Engineering, 5th edition. Ian Sommerville. Chapters 18, 19 and 21. Covers reliability and safety issues in much more detail than is discussed here

Many critical systems are REAL-TIME SYSTEMS. You should be aware of the principles of real-time systems and understand the notion of concurrent process architectures that are fundamental to real-time systems. As an ESSENTIAL preparation for this course, you should read Chapter 16, Real-time Systems Design of Software Engineering, 5th edition.

Course organisation

- Ten 2-hour sessions from 9-11 on Mondays
- Each session is structured into 3 components
 - A 'standard' lecture on the background to a course topic (9-10)
 - Short break
 - A complementary session that may be a guest lecture, a case study of a critical system, an example presentation, etc. (10-11)
- Lecture notes and supplementary papers will be provided and put onto the intranet. ALL material that is distributed is potentially examinable
- Assessment will be 80% exam, 20% coursework

Week 1 - Introduction. Systems engineering and critical systems

Insulin pump - concept of operation

Week 2 - Dependability

To be decided

Week 3 - Critical Systems Specification 1

Insulin pump safety specification

Week 4 - Critical Systems Specification 2

Insulin pump formal specification in Z

Week 5 - Critical Systems Development 1

Space shuttle computer systems

Week 6 - Critical Systems Development 2

Security architectures (???)

Week 7 - Critical Systems Validation 1

Insulin pump safety validation

Week 8 - Critical Systems Validation 2

ARIANE 5 failure study

Week 9 - Human Factors and Critical Systems

London Ambulance failure study

Week 10 - Ethics and Professional Responsibility

Airbus 320 crash

What is a system?

- A collection of elements which are assembled to fulfil some defined purpose. Elements may be hardware or software components, organisational policies and procedures and operational processes.
- Systems have properties which are emergent i.e. they only come to light when the parts are put together, they have structure and mechanisms for communication and control.

The world is full of different types of system ranging from very very simple systems to systems whose complexity is incomprehensible.

The most significant distinction between a system and some other collection of things is that a system has emergent properties. Emergent properties are properties which only become apparent when the system is put together e.g. people have the emergent property of consciousness. We discuss this in more detail in a later slide.

Things which DON'T have these emergent properties shouldn't really be considered as a system - e.g. a collection of books isn't a system as you can predict more or less everything about the collection by looking at its individual components.

Socio-technical computer-based systems

- Systems which some of the elements are software-controlled computers and which are used by people for some purpose. They typically include:
 - Computer hardware
 - Software
 - Policies and procedures
 - Operational processes

In this course, we are only interested in systems which include computers and where software plays a major part in the control of the systems.

Over the past 15 years, we have had an explosion in the numbers of computer-based systems. Indeed, this has really changed the nature of systems engineering itself as, up until recently, it was mostly concerned with complex mechanical/electrical systems such as chemical plants, power stations etc.

For this reason, many of the books you will find called “systems engineering” aren’t really useful for this course as they say very little about computer-based systems.

It is **ESSENTIAL** to understand we are not just talking about machines but also about people. People are a critical component of all computer-based systems as, ultimately, these systems are intended to support some human activity.

Examples

- A payroll system
- A navigation system
- A system for testing blood samples
- A mobile phone
- A ticket reservation system
- A chemical process control system
- A pollution monitoring system
- An air traffic control system

The rationale for these systems is that they contribute to some human activities. Human activities supported:

1. Some form of economic work
2. Moving from place to place
3. Maintaining the health of the population
4. Communicating with other people
5. Moving from place to place, leisure activities
6. Fabricating materials
7. Maintaining the environment
8. Moving from place to place

Different systems vary in the software/hardware/operational procedures that are needed

Emergent properties

- Properties which are properties of the system AS A WHOLE rather than of the collection of parts.
- Not determined solely from the properties of the system parts but also from the system's structure.
- Examples
 - The reliability of a computer depends on the reliability of the processor, memory, keyboard, monitor, disk, etc.
 - A mobile phone has the emergent property of being a communication device.

These distinguish systems from collections of parts. You can take all the parts of a bicycle and put them in a box and you don't have a system - the emergent property of being a transportation device only emerges when you put them together in the right way.

Some emergent properties are predictable and the goal of the system design is to ensure that the system has these properties (a mobile phone has to be usable as a communication device). Other emergent properties are planned but not predictable (a mobile phone will have the property of interfering with other communications equipment but you can't usually be very definite about this during the design). Other properties are unplanned - the designers didn't think about them. These properties often cause lots of problems.

Examples of these unplanned properties are (un)usability properties - e.g. it is often very difficult to use the features on a video recorder or mobile phone.

Emergent system-wide properties

- Important emergent properties of a system are
 - Performance
 - Reliability
 - Safety
 - Security
 - Usability
 - Maintainability
- These are non-functional properties - they do not relate to any specific functionality of the system
- Some or all of these properties are usually more important than detailed system functionality

These properties are emergent properties because they depend on the relationships between components as well as the components themselves. For example, if you have a set of very reliable components yet connect them up wrongly, your final system will be unreliable.

Properties such as safety and security are even more slippery as it doesn't really make sense to talk about the safety of an individual component. This is something that only applies to the system as a whole.

These are often more important than functional system properties because the system supports a broader business process and the level of support which can be provided is not a fixed requirement. If it isn't possible to provide some support, then maybe it will be left out of the process or maybe it will be done manually (e.g. until recently, it was not possible to find out how many of our applicants came from a specific postcode area in the admissions system).

However, if the system is unreliable it may produce incorrect results or results at the wrong time or in the wrong place. There may be no manual fall-back position so this can have serious business consequences

The role of software in systems

- Software in complex systems now has a number of different roles. For example:
 - *Control and coordination* The operation of different parts of the system is coordinated by a controlling software system
 - *Information management* Large amounts of information that is required in many systems is managed and organised by software
 - *Input and output filtering* System inputs and outputs are pre and post processed by software to simplify their subsequent processing
 - *User interface* The user interface to many systems is now largely a software-based interface
 - *System monitoring* The operation of the system is monitored by software and anomalies reported

Systems engineering

- The process of specifying, designing, implementing and installing computer-based systems to solve some identified problem.
- Generally concerned with complex systems involving hardware, software and people.
- Concerned with wider 'systems issues' rather than details of a system's functionality and implementation.

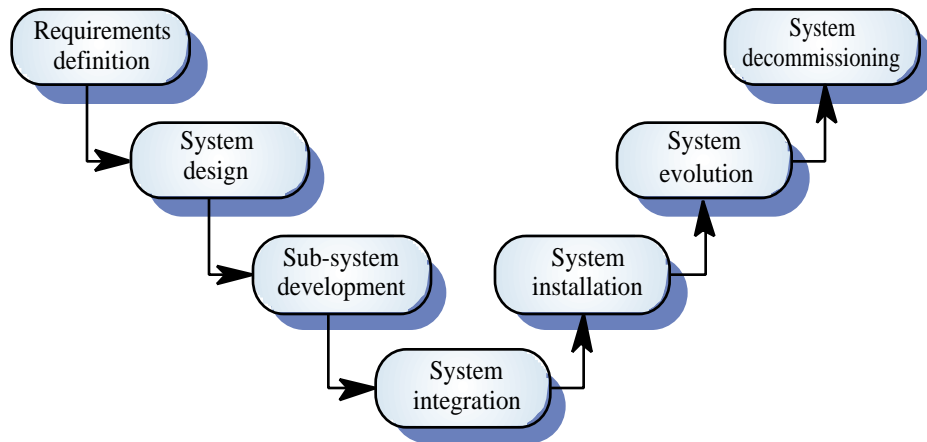
A process is a structured set of activities which is intended to achieve some outcome. Therefore, a systems engineering process is a set of engineering activities whose goal is to produce a system of some type.

Processes can be very simple (e.g. the process of buying a lottery ticket) or very complex (the process of designing and building the Channel Tunnel).

In the systems engineering processes, we are mostly concerned with those activities which lead to more detailed specifications of sub-systems (these are then implemented using some other engineering processes) and integrating these sub-systems to form a system.

Surprisingly, perhaps, the systems engineering process is rarely concerned with the details of the functionality provided by the system, only with a broad outline of what the system is supposed to do.

The systems engineering process



©Ian Sommerville 2000

CS 365. Critical Systems Engineering

Slide 10

We look at each of the activities in this process in turn.

This model suggests that the process is linear with a smooth transition from one phase of the process to another. In reality, it's never as simple as that because different sub-systems are inevitably at different stages of development, there are various problems which arise which affect some but not all parts of the system, etc.

In practice, there tends to be a fairly lengthy transition period between one phase of the process and the next and a lot of process iteration where activities in previous stages have to be redone to correct problems which have been detected.

System and software engineering

- Software is increasingly used in systems because it allows for more complex information processing AND it is malleable - changes can be made after the design is complete
- Many software system problems are a consequence of this malleability. Software changes become necessary because of problems elsewhere in the system or new requirements that emerge when the system is integrated

Critical systems

- A critical system is any system whose ‘failure’ could threaten human life, the system’s environment or the existence of the organisation which operates the system.
- ‘Failure’ in this context does NOT mean failure to conform to a specification but means any potentially threatening system behaviour.

Critical systems, in the broadest sense of the term, are not new. It has always been the case that an aircraft is a critical system. However, until relatively recently the control element of the system was primarily based on the human operator of the system and (wrongly perhaps) they were often seen as outside of the system.

Increasingly, critical systems are becoming computer-based with many control decisions made by the computer rather than an operator. Because computers work faster than people, this allows much more complex approaches to control to be used which reduces system costs and improves their capabilities

It also means that we have a poorer understanding of how to design systems to avoid failure.

Critical system classes

- **Safety-critical systems**
 - A system whose failure may result in the loss of human life, injury or major environmental damage
- **Mission-critical systems**
 - A system whose failure may result in the consequent failure of a goal-directed activity
- **Business-critical systems**
 - A system whose failure may result in the failure of the business that is using that system

Examples of critical systems

- Communication systems such as telephone switching systems, aircraft radio systems, etc.
- Embedded control systems for process plants, medical devices.
- Command and control systems such as air-traffic control systems, disaster management systems.
- Financial systems such as foreign exchange transaction systems, account management systems.

Failure of a communication system can be critical for both the ‘owners’ of the system (e.g. the telephone company) and the ‘users’ of the system. For example, if a telephone system fails, it may be impossible to report an accident to the emergency services

Embedded systems are ‘primary’ critical systems i.e. failure can directly cause loss or damage. Systems such as account management systems are ‘secondary’ critical systems in that the consequences of system failure (lack of information, incorrect information) may mean that the organisation as a whole cannot function properly

Critical systems usage

- Most critical systems are now computer-based systems
- Critical systems are becoming more widespread as society becomes more complex and more complex activities are automated
- People and operational processes are very important elements of critical systems - they cannot simply be considered in terms of hardware and software

Computer-based control and data management allows for an order of magnitude increase in the complexity of critical systems

However, the procedures and practices which have evolved to integrate critical systems into society are based on much less complex systems. In many cases, we do not really understand what the overall impact of these critical systems will be (e.g. automated share trading systems).

Critical systems are usually *socio-technical* systems - they are part of a broader social and organisational framework and the people working with these systems are influenced by social and organisational factors. Changes to the social and organisational environment in which the system is installed, can mean that the systems have to change.

Critical systems failure

- The cost of failure in a critical system is likely to exceed the cost of the system itself
- As well as direct failure costs, there are indirect costs from a critical systems failure. These may be significantly greater than the direct costs
- Society's views of critical systems are not static - they are modified by each high-profile system failure

Costs of failure may include:

- Direct costs of repairing the system - can be high if expensive hardware is physically damaged
- Costs of investigating the cause of the problem - again this, can be very high if there has been an accident with associated loss of life
- Loss of revenue while the system is out of service
- Compensation costs for people/things damaged by the failure
- Legal costs associated with compensation claims
- Re-design and change costs for other systems which may be vulnerable to the same type of fail

People do not necessarily react logically to critical systems failure - they are much more concerned about train and plane crashes than car accidents although cars kill many more people annually

Critical emergent properties

- **Reliability**
 - Concerned with failure to perform to specification
- **Availability**
 - Concerned with failure to deliver required services
- **Safety**
 - Concerned with behaviour which directly or indirectly threatens human life
- **Security**
 - Concerned with the ability of the system to protect itself from external attack

Each of these attributes are covered in the following slides.

Not all of the attributes necessarily apply to all types of critical systems. In particular, the safety attribute only applies to a sub-set of critical systems called *safety-critical systems*. Other attributes may be more or less important but for all systems, reliability and security are normally important attributes.

The criticality attributes are not independent. For example, availability and maintainability are closely related. If a system has to be taken out of service to be maintained, it is clearly not available. If a system is insecure, data corruption can cause the system to become unreliable.

Critical systems development

- Critical systems attributes are NOT independent - the systems development process must be organised so that all of them are satisfied at least to some minimum level
- More rigorous (and expensive) development techniques have to be used for critical systems development because of the potential cost of failure

There is a trade-off between the different critical systems attributes and there may be conflicts between them. For example, security requirements may limit the number of people who have privileged access to a system but maintainability requirements may require that all maintenance staff have such access. Systems may often be made safe by making them unavailable (e.g. when a robot is switched off, it cannot cause any damage).

Critical systems usually cost more and take longer to develop than non-critical system. Different techniques of specification, design, implementation and testing may be used. The system may be independently inspected and some classes of critical system (e.g. aircraft, nuclear systems) must be certified before they may be used.

Development techniques

- Use of formal methods for system specification
- Use of formal verification to demonstrate that a program is consistent with its specification
- Separate teams for implementation and testing
- Incorporation of redundant code and self-checking in programs
- Redundant hardware units
- Measurement of test coverage

The most extensive use of formal methods has been in the development of critical systems particularly safety-critical and security-critical systems.

Key points

- Computer-based systems are socio-technical systems which include hardware, software, operational processes and procedures and people.
- An increasing number of socio-technical systems are critical systems
- Systems have emergent properties i.e. properties which are only apparent when all sub-systems are integrated.
- Critical system attributes are reliability, availability, safety and security