Systems Engineering

- Designing, implementing, deploying and operating systems which include the interaction amongst infrastructure, control and people (users)

Objectives

- To explain why project development is affected by broader system engineering issues
- To introduce the concept of emergent system properties such as reliability and sustainability
- To explain why the systems environment must be considered in the system design process

What is a system?

- A purposeful collection of inter-related components working together towards some common objective.
- A system may include infrastructures, mechanical, electrical and electronic hardware and be operated and managed by people.
- System components are dependent on other system components
- The properties and behaviour of system components are inextricably inter-mingled

Problems of systems engineering

- Large systems are usually designed to address and solve ‘challenging' problems
- Systems engineering requires a great deal of co-ordination across disciplines
  - Almost infinite possibilities for design trade-offs across components
  - Mutual distrust and lack of understanding across engineering disciplines
- Systems must be designed to last many years (sustainability) in a changing environment

Emergent properties

- Properties of the system as a whole rather than properties that can be derived from the properties of components of a system
- Emergent properties are a consequence of the relationships between system components
- They can therefore only be assessed and measured once the components have been integrated into a one-large well-designed well-integrated system

Examples of emergent properties

- *The overall weight of the system*
  - This is an example of an emergent property that can be computed from individual component properties.
- *The reliability of the system*
  - This depends on the reliability of system components and the relationships between the components.
- *The usability of a system*
  - This is a complex property which is not simply dependent on the system components but also depends on the system operators and the environment where it is used.

System reliability engineering

- Because of component inter-dependencies, faults can be propagated through the system
- System failures often occur because of unforeseen inter-relationships between components
- It is probably impossible to anticipate all possible component relationships
- Component reliability measures may give a false picture of the system reliability

Influences on reliability

- *Infrastructure components reliability*
  - What is the probability of a infrastructure component failing and how long does it take to repair that component?
- *Control reliability*
  - How likely is it that a software component will produce an incorrect output. Software failure is usually distinct from hardware failure in that software does not wear out.
- *Operator reliability*
  - How likely is it that the operator of a system will make an error?

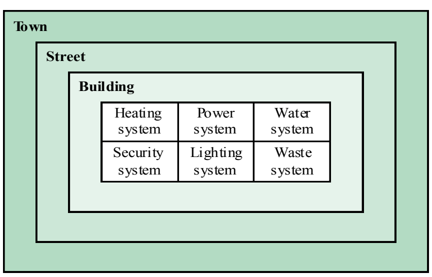
The ‘shall-not’ properties

- Properties such as performance and reliability can be measured
- However, some properties are properties that the system should not exhibit
  - Safety - the system should not behave in an unsafe way
  - Security - the system should not permit unauthorised use
- Measuring or assessing these properties is very hard, yet it has to be an integral part of the project development process

Systems and their environment

- Systems are not independent but exist in an environment
- System’s function may be to change its environment
- Environment affects the functioning of the system e.g. system may require electrical supply from its environment
- The organizational as well as the physical environment may be important

System hierarchies - Example



Human and organisational factors

- *Process changes*
  - Does the proposed system require changes to the work processes in the environment?
- *Job changes*
  - Does the proposed system de-skill the users in an environment or cause them to change the way they work?
- *Organisational changes*
  - Does the proposed system change the political power structure in an organisation?

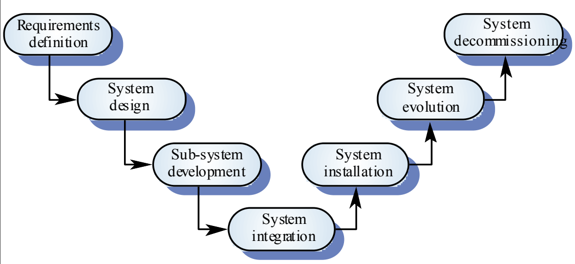
System architecture modelling

- An architectural model presents an abstract view of the sub-systems making up a system
- May include major information flows between sub-systems
- Usually presented as a block diagram
- May identify different types of functional component in the model

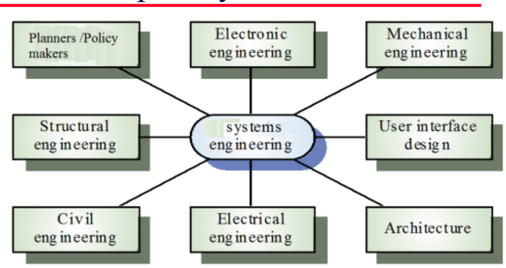
The system engineering process

- Usually follows a 'waterfall' model because of the need for parallel development of different parts of the system
  - Little scope for iteration between phases because hardware changes are very expensive. Software may have to compensate for hardware problems
- Inevitably involves engineers from different disciplines who must work together
  - Much scope for misunderstanding here. Different disciplines use a different vocabulary and much negotiation is required. Engineers may have personal agendas to fulfil

The system engineering process



Inter-disciplinary involvement



System requirements definition

- Three types of requirement defined at this stage
  - Abstract functional requirements. System functions are defined in an abstract way
  - System properties. Non-functional requirements for the system in general are defined
  - Undesirable characteristics. Unacceptable system behaviour is specified
- Should also define overall organisational objectives for the system

System objectives

- Functional objectives
  - To provide a fire and intruder alarm system for the building which will provide internal and external warning of fire or unauthorized intrusion
- Organisational objectives
  - To ensure that the normal functioning of work carried out in the building is not seriously disrupted by events such as fire and unauthorized intrusion

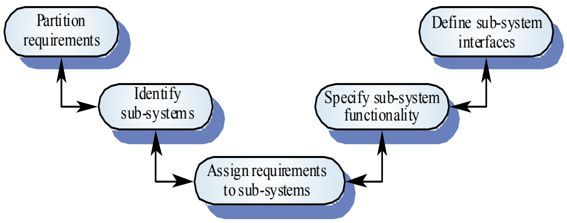
System requirements problems

- Changing as the system is being specified
- Must anticipate hardware/communications developments over the lifetime of the system
- Hard to define non-functional requirements (particularly) without an impression of component structure of the system.

The system design process

- Partition requirements
  - Organise requirements into related groups
- Identify sub-systems
  - Identify a set of sub-systems which collectively can meet the system requirements
- Assign requirements to sub-systems
  - Causes particular problems when COTS are integrated
- Specify sub-system functionality
- Define sub-system interfaces
  - Critical activity for parallel sub-system development

The system design process



System design problems

- Requirements partitioning to infrastructure, control, and human components may involve a lot of negotiation
- Difficult design problems are often assumed to be readily solved using integrated-solution strategies

Sub-system development

- Typically parallel projects developing other components such as power, control and communications
- Lack of communication/integration across implementation teams
- Bureaucratic and slow mechanism for proposing system changes means that the development schedule may be extended because of the need for rework

System integration

- The process of putting infrastructure, control, and people together to make a system
- Should be tackled incrementally so that sub-systems are integrated one at a time
- Interface problems between sub-systems are usually found at this stage
- May be problems with uncoordinated deliveries of system components

System Implementation

- Environmental assumptions may be incorrect
- May be human/institutional resistance to the introduction of
  a new system
- System may have to coexist with alternative systems for some time
- Operator training has to be identified

System operation

- Will bring unforeseen requirements to light
- Users may use the system in a way which is not anticipated by system designers
- May reveal problems in the interaction with other systems
  - Physical problems of incompatibility
  - Increased operator error rate because of inconsistent interfaces/integration

System evolution

- Large systems have a long lifetime. They must evolve to meet changing requirements
- Evolution is inherently costly
  - Changes must be analysed from a technical and business perspective
  - Sub-systems interact so unanticipated problems can arise
  - There is rarely a rationale for original design decisions
  - System structure is corrupted as changes are made to it
- Existing systems which must be maintained are sometimes called legacy systems

Key points

- System engineering involves input from a range of disciplines
- Emergent properties are properties that are characteristic of the system as a whole and not its component parts
- System architectural models show major sub-systems and inter-connections. They are usually described using block diagrams

Key points

- The systems engineering process is usually a waterfall model and includes specification, design, development and integration.
- System procurement is concerned with deciding which system to buy and who to buy it from

Conclusion

- Systems engineering is hard! There will never be an easy answer to the problems of complex system development
- Disciplines need to recognise each others strengths and actively rather than reluctantly cooperate in the systems engineering process