Socio-technical Systems
Objectives

- To explain what a socio-technical system is and the distinction between this and a computer-based system
- To introduce the concept of emergent system properties such as reliability and security
- To explain system engineering and system procurement processes
- To explain why the organisational context of a system affects its design and use
- To discuss legacy systems and why these are critical to many businesses
Topics covered

- Emergent system properties
- Systems engineering
- Organizations, people and computer systems
- Legacy systems
What is a system?

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

- Technical computer-based systems
  - Systems that include hardware and software but where the operators and operational processes are not normally considered to be part of the system. The system is not self-aware.

- Socio-technical systems
  - Systems that include technical systems but also operational processes and people who use and interact with the technical system. Socio-technical systems are governed by organisational policies and rules.
Socio-technical system characteristics

- Emergent properties
  - Properties of the system of a whole that depend on the system components and their relationships.

- Non-deterministic
  - They do not always produce the same output when presented with the same input because the systems’s behaviour is partially dependent on human operators.

- Complex relationships with organisational objectives
  - The extent to which the system supports organisational objectives does not just depend on the system itself.
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 system
Examples of emergent properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volume</td>
<td>The volume of a system (the total space occupied) varies depending on how the component assemblies are arranged and connected.</td>
</tr>
<tr>
<td>Reliability</td>
<td>System reliability depends on component reliability but unexpected interactions can cause new types of failure and therefore affect the reliability of the system.</td>
</tr>
<tr>
<td>Security</td>
<td>The security of the system (its ability to resist attack) is a complex property that cannot be easily measured. Attacks may be devised that were not anticipated by the system designers and so may defeat built-in safeguards.</td>
</tr>
<tr>
<td>Repairability</td>
<td>This property reflects how easy it is to fix a problem with the system once it has been discovered. It depends on being able to diagnose the problem, access the components that are faulty and modify or replace these components.</td>
</tr>
<tr>
<td>Usability</td>
<td>This property reflects how easy it is to use the system. It depends on the technical system components, its operators and its operating environment.</td>
</tr>
</tbody>
</table>
Types of emergent property

● Functional properties
  • These appear when all the parts of a system work together to achieve some objective. For example, a bicycle has the functional property of being a transportation device once it has been assembled from its components.

● Non-functional emergent properties
  • Examples are reliability, performance, safety, and security. These relate to the behaviour of the system in its operational environment. They are often critical for computer-based systems as failure to achieve some minimal defined level in these properties may make the system unusable.
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.
- Software reliability measures may give a false picture of the system reliability.
Influences on reliability

- **Hardware reliability**
  - What is the probability of a hardware component failing and how long does it take to repair that component?

- **Software 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?
Reliability relationships

- Hardware failure can generate spurious signals that are outside the range of inputs expected by the software.
- Software errors can cause alarms to be activated which cause operator stress and lead to operator errors.
- The environment in which a system is installed can affect its reliability.
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.
Systems engineering

- Specifying, designing, implementing, validating, deploying and maintaining socio-technical systems.
- Concerned with the services provided by the system, constraints on its construction and operation and the ways in which it is used.
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 systems engineering process

- Requirements definition
- System design
- Sub-system development
- System integration
- System installation
- System evolution
- System decommissioning
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

- Should define why a system is being procured for a particular environment.
- 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

- Complex systems are usually developed to address wicked problems
  - Problems that are not fully understood;
  - 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 knowing the 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

Partition requirements

Identify sub-systems

Specify sub-system functionality

Assign requirements to sub-systems

Define sub-system interfaces
System design problems

- Requirements partitioning to hardware, software and human components may involve a lot of negotiation.
- Difficult design problems are often assumed to be readily solved using software.
- Hardware platforms may be inappropriate for software requirements so software must compensate for this.
Requirements and design

- Requirements engineering and system design are inextricably linked.
- Constraints posed by the system’s environment and other systems limit design choices so the actual design to be used may be a requirement.
- Initial design may be necessary to structure the requirements.
- As you do design, you learn more about the requirements.
Spiral model of requirements/design
System 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.
Burglar alarm system
## Sub-system description

<table>
<thead>
<tr>
<th>Sub-system</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Movement sensors</td>
<td>Detects movement in the rooms monitored by the system</td>
</tr>
<tr>
<td>Door sensors</td>
<td>Detects door opening in the external doors of the building</td>
</tr>
<tr>
<td>Alarm controller</td>
<td>Controls the operation of the system</td>
</tr>
<tr>
<td>Siren</td>
<td>Emits an audible warning when an intruder is suspected</td>
</tr>
<tr>
<td>Voice synthesizer</td>
<td>Synthesizes a voice message giving the location of the suspected intruder</td>
</tr>
<tr>
<td>Telephone caller</td>
<td>Makes external calls to notify security, the police, etc.</td>
</tr>
</tbody>
</table>
Sub-system development

- Typically parallel projects developing the hardware, software and communications.
- May involve some COTS (Commercial Off-the-Shelf) systems procurement.
- Lack of communication 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 hardware, software 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.
● After completion, the system has to be installed in the customer’s environment
  • Environmental assumptions may be incorrect;
  • May be human resistance to the introduction of a new system;
  • System may have to coexist with alternative systems for some time;
  • May be physical installation problems (e.g. cabling problems);
  • Operator training has to be identified.
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.
System decommissioning

- Taking the system out of service after its useful lifetime.
- May require removal of materials (e.g. dangerous chemicals) which pollute the environment
  - Should be planned for in the system design by encapsulation.
- May require data to be restructured and converted to be used in some other system.
Organisations/people/systems

- Socio-technical systems are organisational systems intended to help deliver some organisational or business goal.
- If you do not understand the organisational environment where a system is used, the system is less likely to meet the real needs of the business and its users.
Human and organisational factors

- Process changes
  - Does the system require changes to the work processes in the environment?

- Job changes
  - Does the system de-skill the users in an environment or cause them to change the way they work?

- Organisational changes
  - Does the system change the political power structure in an organisation?
Organisational processes

- The processes of systems engineering overlap and interact with organisational procurement processes.
- Operational processes are the processes involved in using the system for its intended purpose. For new systems, these have to be defined as part of the system design.
- Operational processes should be designed to be flexible and should not force operations to be done in a particular way. It is important that human operators can use their initiative if problems arise.
Procurement/development processes
System procurement

- Acquiring a system for an organization to meet some need
- Some system specification and architectural design is usually necessary before procurement
  - You need a specification to let a contract for system development
  - The specification may allow you to buy a commercial off-the-shelf (COTS) system. Almost always cheaper than developing a system from scratch
- Large complex systems usually consist of a mix of off the shelf and specially designed components. The procurement processes for these different types of component are usually different.
The system procurement process

1. **Off-the-shelf system available**
   - Adapt requirements
   - Choose system
   - Issue request for bids
   - Choose supplier

2. **Survey market for existing systems**
   - Issue request to tender
   - Select tender
   - Negotiate contract
   - Let contract for development

3. **Custom system required**
Procurement issues

- Requirements may have to be modified to match the capabilities of off-the-shelf components.
- The requirements specification may be part of the contract for the development of the system.
- There is usually a contract negotiation period to agree changes after the contractor to build a system has been selected.
Contractors and sub-contractors

- The procurement of large hardware/software systems is usually based around some principal contractor.
- Sub-contracts are issued to other suppliers to supply parts of the system.
- Customer liaises with the principal contractor and does not deal directly with sub-contractors.
Contractor/Sub-contractor model

- System customer
- Principal contractor
- Subcontractor 1
- Subcontractor 2
- Subcontractor 3
Legacy systems

- Socio-technical systems that have been developed using old or obsolete technology.

- Crucial to the operation of a business and it is often too risky to discard these systems
  - Bank customer accounting system;
  - Aircraft maintenance system.

- Legacy systems constrain new business processes and consume a high proportion of company budgets.
Legacy system components

- Hardware - may be obsolete mainframe hardware.
- Support software - may rely on support software from suppliers who are no longer in business.
- Application software - may be written in obsolete programming languages.
- Application data - often incomplete and inconsistent.
- Business processes - may be constrained by software structure and functionality.
- Business policies and rules - may be implicit and embedded in the system software.
Socio-technical system

- Business processes
- Application software
- Support software
- Hardware
Key points

- Socio-technical systems include computer hardware, software and people and are designed to meet some business goal.
- Emergent properties are properties that are characteristic of the system as a whole and not its component parts.
- The systems engineering process includes specification, design, development, integration and testing. System integration is particularly critical.
Key points

- Human and organisational factors have a significant effect on the operation of socio-technical systems.
- There are complex interactions between the processes of system procurement, development and operation.
- A legacy system is an old system that continues to provide essential services.
- Legacy systems include business processes, application software, support software and system hardware.