Software Engineering

Chapter 2:

Computer-based System Engineering

Objectives

- To define what is system engineering
- To introduce the concept of **emergent system properties** such as **reliability and security**
- To explain why the **systems environment** must be considered in the system design process
- To explain system engineering processes
- To explain system procurement processes

What is Systems Engineering

- A system may include:
  - **Hardware**
  - **Software**
  - **People**

  - Examples: Banking system, Air Traffic control system
    Aircrafts, Trains, Vehicles

- Systems Engineering:
  - Designing, implementing, deploying and operating systems

Software Engineering

- Software Engineering: concerned **only** with S/W
- **Embedded** S/W:
  - Special purpose S/W to operate a H/W
  - Could be replaced by H/W
  - For **flexibility reasons**, S/W solution is adopted instead of H/W solution

What is a system?

- A collection of
  - ➡ **inter-related components**
  - ➡ **working together**
  - ➡ **to achieve a predefined common objective.**

System and Sub-systems

- A system may include sub-systems
- Example: A car system includes many subsystems:
  - Engine sub-system (fuel burning)
  - Transmission sub-system (gear)
  - Braking sub-system
  - Sound sub-system
  - Electrical sub-system (lights, electrical windows)
  - Security sub-system (alarm, center lock)
  - Safety sub-system (air bags)

- ***Divide and conquer*** strategy

Example: HMS "Health Management System" & Sub-systems

- Example of a system including sub-systems at many levels (hierarchy)
- HMS "Health Management System"
- HIS " Health Information System"

- HMS sub-systems include:
  - **In-patient sub-system**:
    - Admission (sub-sub-system)
    - Care services
    - Medical treatment
    - Medical record
    - Operation Theatre
    - Accounting…

Example: HMS “Health Management System” & Subsystems (cont.)

- **Out-patient**
  - Patient appointment
  - Staff scheduling
- **Drug store**
- **Maintenance of equipment**
- **Laboratories**
- **etc...**

Emergent properties

- Properties of the system **as a whole** rather than properties that can be derived from the properties of each component
- 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

Non-functional properties

- You impose a **constraint** on **non-functional** properties:
  - **Speed/performance**
  - **Reliability**
  - **Safety**
  - **Security**

System reliability engineering

- Reliability is to be considered at the **system level** rather than at individual component level

System overall reliability

System overall reliability is function of:

- *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 (but hardware does)
- *Operator reliability*
  - How likely is it that the operator of a system will make an error? (stress conditions)

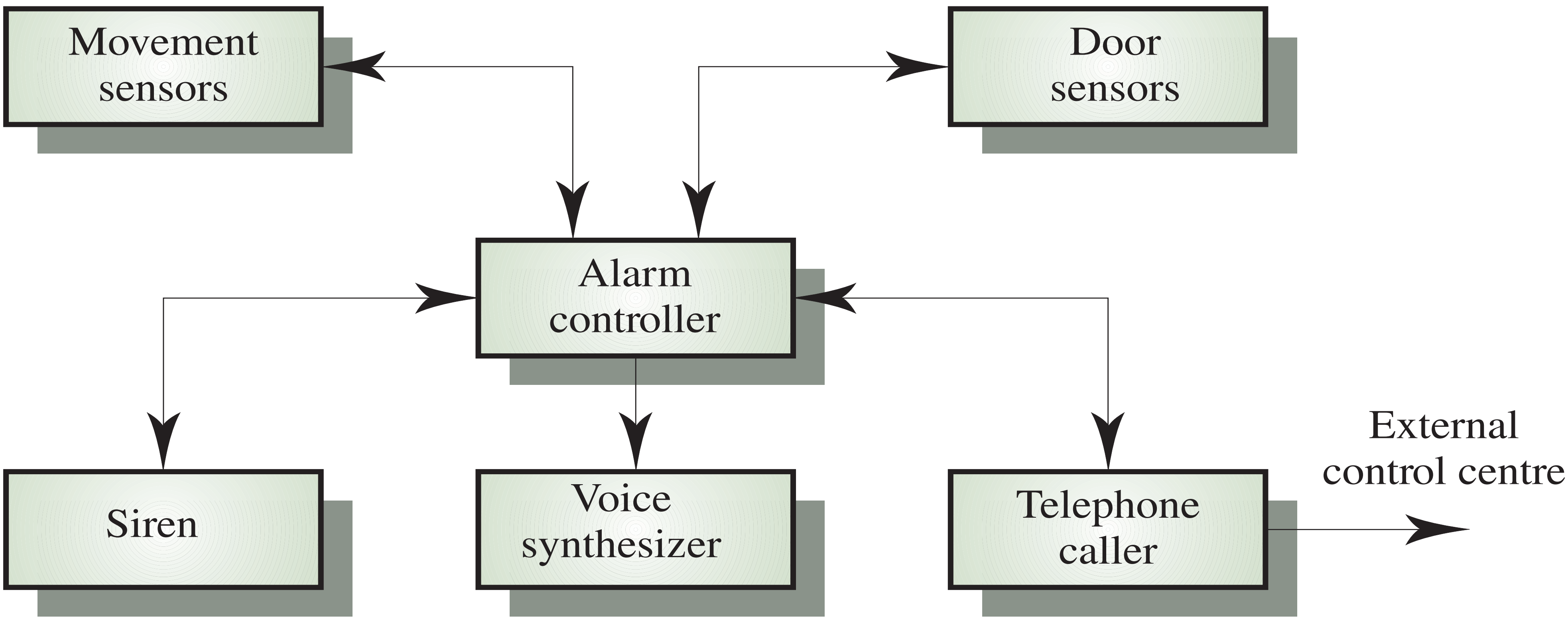
The ‘shall-not’ list

- Properties that the system **shall NOT** exhibit
- Examples:
  - Safety issues - the system shall **NOT** behave in an unsafe way
  - Security issues - the system shall **NOT** permit unauthorised use
- The ‘shall-not’ properties must also be defined in the system requirements

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 ‘interface’
- Usually presented as a block diagram

Example: Intruder alarm system
System Architecture Block Diagram



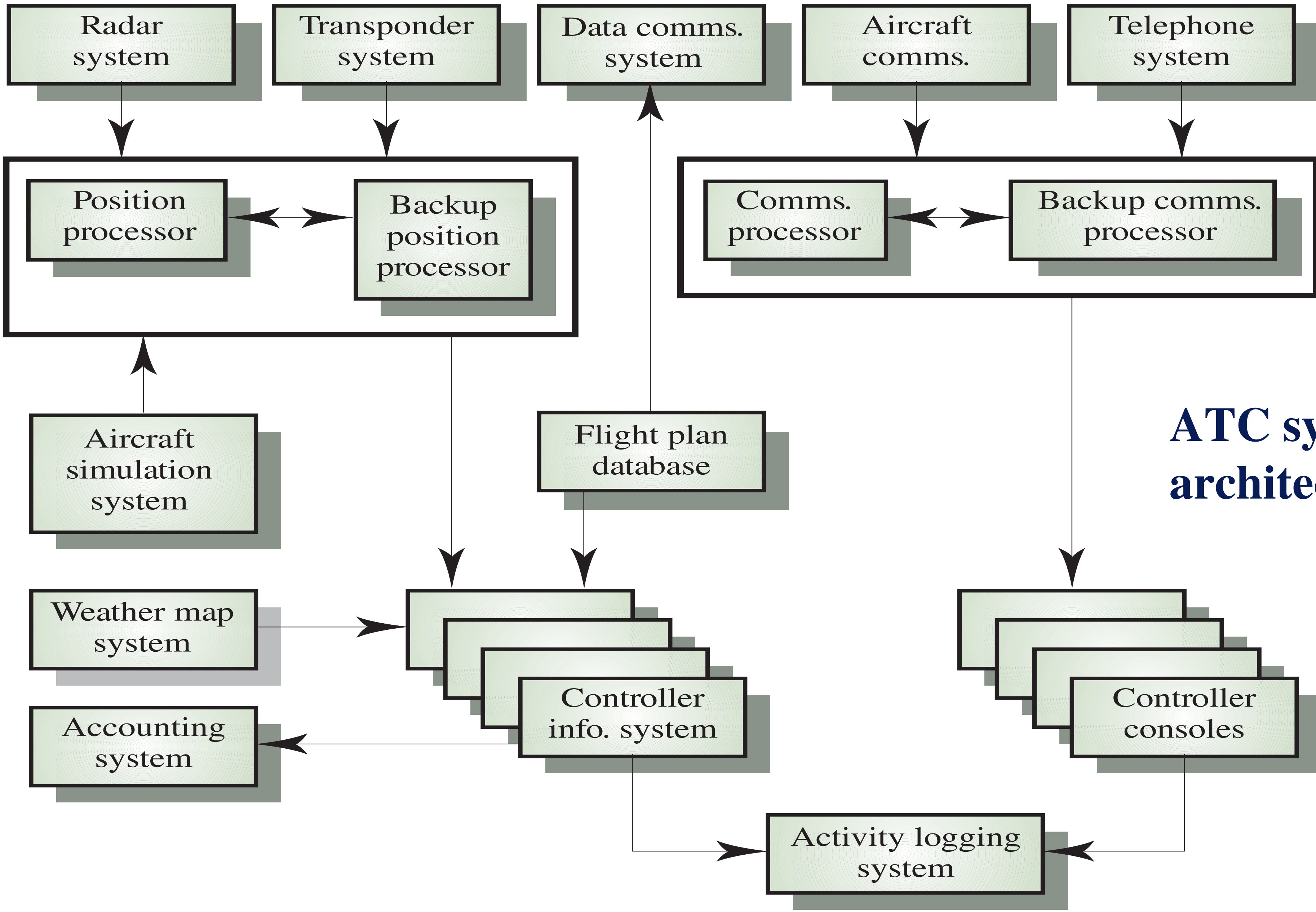
Component types in alarm system

- Sensor
  - Movement sensor, door sensor
- Actuator
  - Siren
- Communication
  - Telephone caller
- Co-ordination
  - Alarm controller
- Interface
  - Voice synthesizer

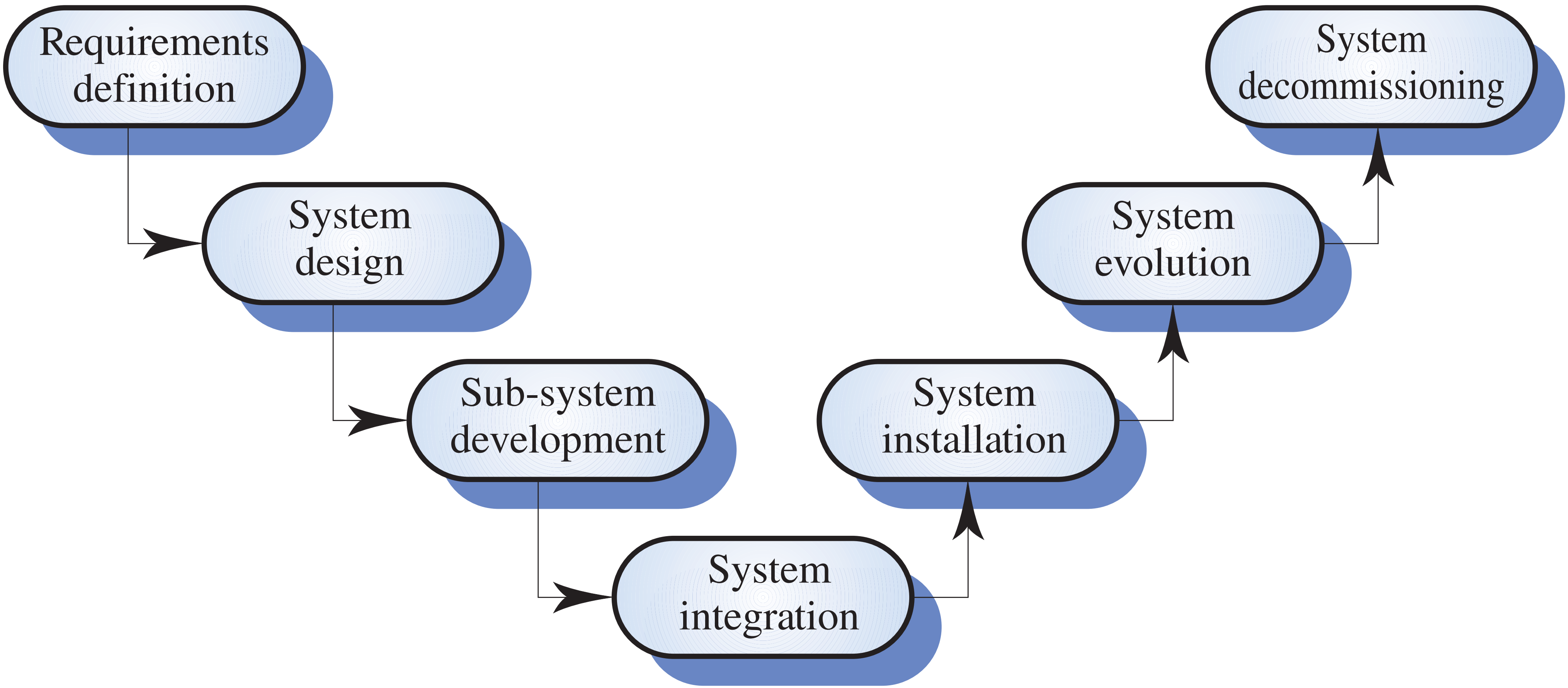
System architecture
H/W or S/W sub-systems

- System architecture should be designed in terms of functional sub-systems (whether H/W or S/W subsystems)
- Deciding on **H/W or S/W** sub-system for providing a function?
- Decision is governed by **non-technical factors** e.g.
  - Availability of **COTS** components
  - Time constraint
  - Cost

ATC system architecture



The system engineering process



System Architecture
System Decomposition into subsystems

- Many possible alternative decomposition solutions

System requirements definition "WHAT"

- Types of requirement
  1. Abstract functional requirements.
     - System functions are defined in an abstract way
     - Details at sub system level
  2. Non-functional requirements:
     - Are defined for the system in general
     - EX: Performance, safety, etc … that affect the requirements of the integrated sys
  3. 'Shall-not' characteristics: Unacceptable system behaviour is specified
  4. Domain requirements: Specific and highly technical for a particular domain of application eg protocol Z39.50 for Library Info Sys

- Should also define
  - System environment (Physical, Organisational and Human)
  - **Business requirements:** Organizational objectives for the system

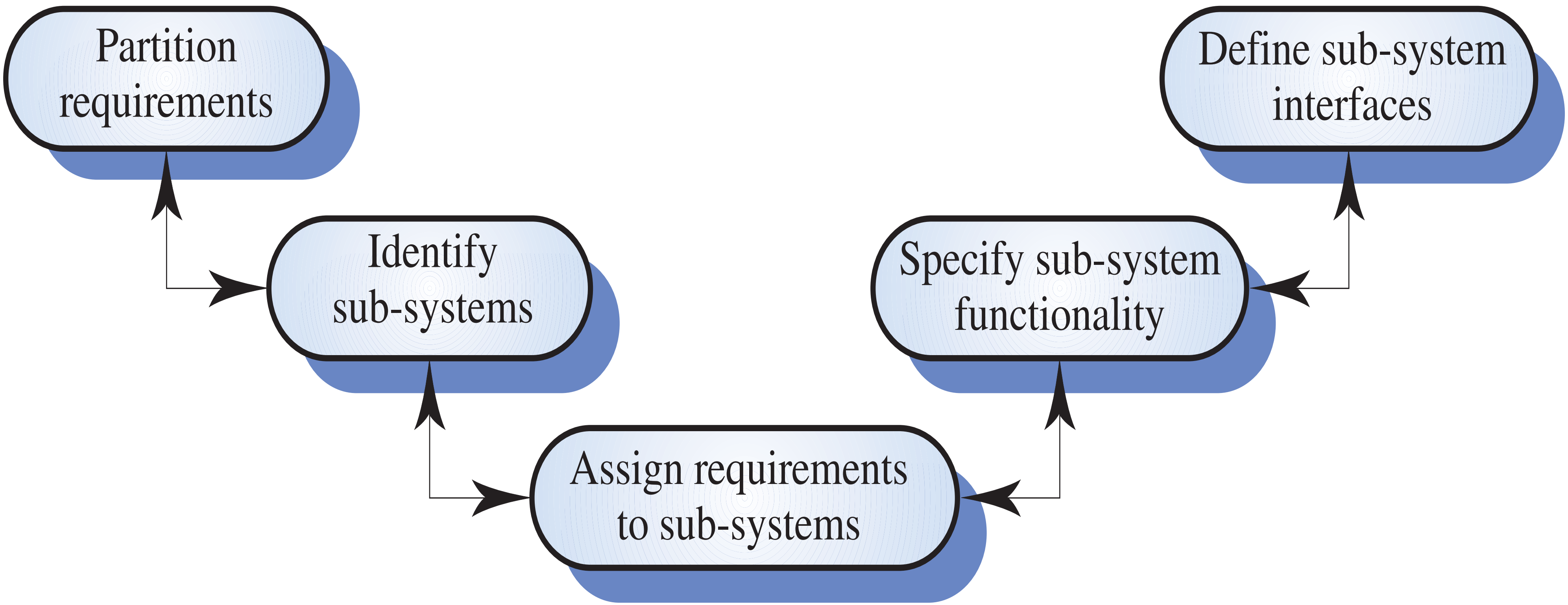
System requirements problems

requirements **Change !!!!!**

as the system is being specified

The system design process “HOW”

5 Design Activities



The system design process "HOW"

5 Design Activities:

1) Partition requirements
    - Organise requirements into related groups
2) Identify sub-systems
    - Identify a set of sub-systems which collectively can meet the system requirements
3) Assign requirements to sub-systems
    - Problems with externally purchased subsystems - **COTS** (Commercial Of The Shelf)
        - May impose modifications on the requirements because of non 100% compliance with requirements
        - Integration problems

The system design process "HOW" (cont.)

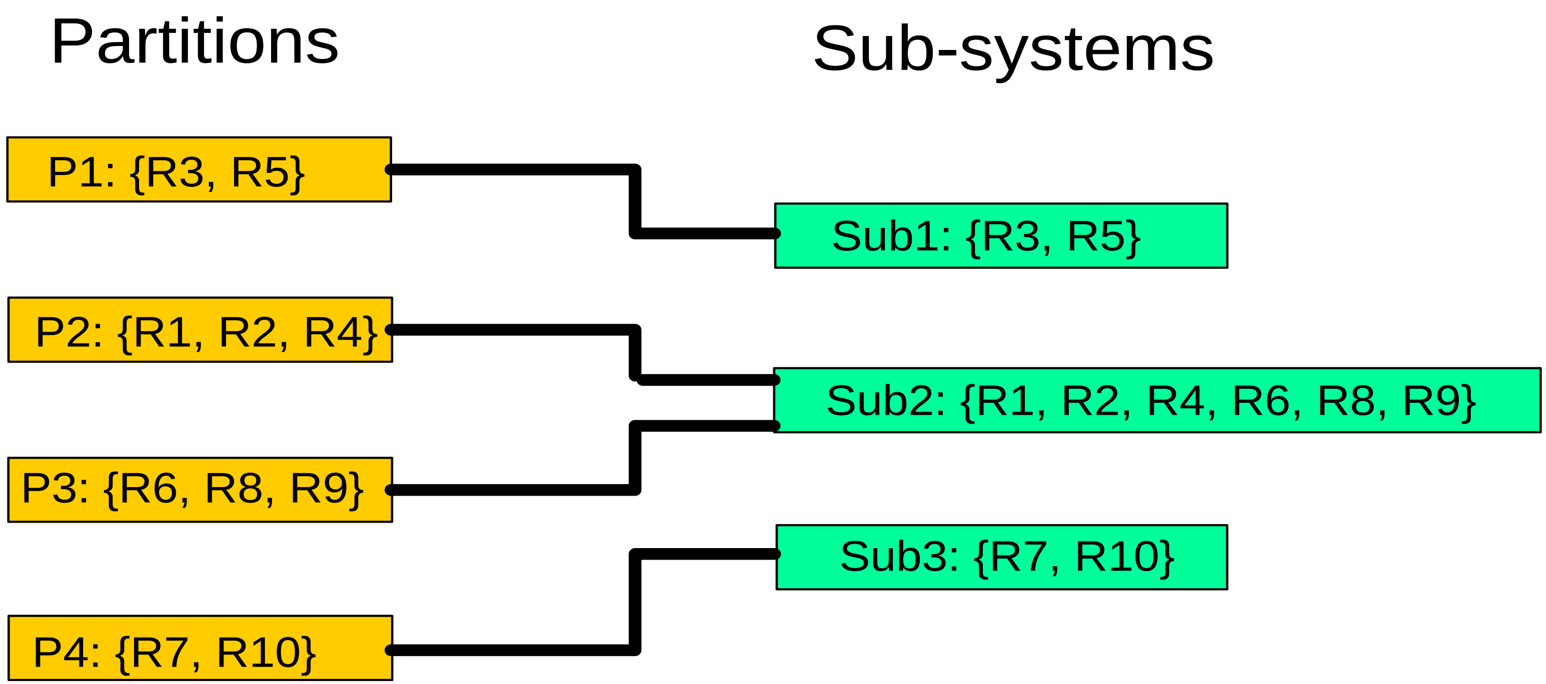
4) Specify sub-system functionality & inter-relationships
   - May be seen as part of sys Design phase for general sub-sys (H/W+S/W+Human/W)
   - OR as part of Analysis phase if the sub-sys is a S/W sub-sys
5) Define sub-system interfaces
   - Critical activity for parallel sub-system development

Design comments

- Many possible design alternatives of H/W, S/W , Human operations
- The selected solution need not be the most appropriate technical solution: organizational and political issues may influence the choice of the solution (e.g. National supplier for a government sys)

System Design:
Assignment of requirements to sub-systems

Set of Requirements: {R1, R2, R3, ... , R10}



Sub-system development

- **If the sub-sys is a S/W**
- **Then** S/W process is started '**RDIV**'
- **RDIV**
  - Requirements,
  - Design,
  - Implementation,
  - Validation

Sub-system development

- **Make/Buy decision**
- **Make** → **Develop (RDIV)**
- **Buy: Get** COTS sub-sys and **integrate** into the sys
  - Cheaper
  - Faster
  - Tested
  - If COTS does not meet the requirements exactly, rethink the design for adaptation in order to use COTS

System integration

- Interface problems between sub-systems are usually found at this stage
- Integration approaches:
  - Bing Bang integration
  - Incremental integration

- Bing Bang integration: all sub-systems are integrated at the same time
- Incremental integration: sub-systems are integrated one at a time

System installation

- System is put in its environment

- Problems:
  - Different environment from that assumed by sys developers (e.g. Operating System version)
  - Human resistance to the introduction of the new system
  - System may have to coexist in parallel with alternative systems or sub-system components for some time
  - Physical installation problems (e.g. network cabling problems)
  - Operator training has to be identified

System operation

- Will bring unforeseen requirements to light
- Problems:
  - Incompatibility if coexisting with old systems
- May require
  - Users training
  - Data conversion

System evolution

- A systems has a long lifetime span
  - Embedded errors correction
  - Dynamic business environment yields new requirements
- Evolution is inherently costly
  - Changes must be analysed and approved
  - Changes to one sub-system may affect other sub-systems, these in turn need to be fixed
  - Cost of maintaining a sys increases with time - System structure becomes corrupted because of continuous maintenance changes
- Legacy systems:
  - Existing old systems that the organisation must continue to use because they are critical for the business operations.
  - Must be maintained and present a challenge to S/W Engineers aiming at reducing cost of maintenance

System decommissioning

- Taking the system out of service after its useful lifetime
- Physical H/W decommissioning problems
  - 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
- S/W has no physical decommissioning issues

Computer-based System Procurement

- Acquiring a system for an organization to meet predefined need

- Deciding about:
  - The best approach to acquire a system
  - The best supplier

- A system may be:
  - Bought as a whole
  - Bought as separate parts and then integrated
  - Especially designed and developed

System procurement (cont.)

- Some system specification and architectural design must be done before procurement decision
  - Provide the supplier or contractor with specification about the required system
  - The specification/ architectural design will help identify commercial off-the-shelf (COTS) sub-systems that might be bought.
  - Large systems consist of a mix of:
    - COTS
    - Especially developed components

System procurement (cont.)

Glue ware

- Use of S/W allows more use of H/W components
- The S/W act as a glue ware between the different components of H/W
- Cost of developing glue ware may counter balance savings from using COTS
- COTS is usually cheaper than developing a component from scratch

The system procurement process