Embedded Software & Systems Engineering

Unit 1: Embedded Software & Systems Engineering

Objectives

– To provide an introduction to systems engineering and the embedding of software in a systems context.

Everyday Words

"it is in the system", "the system failed", "rage against the system", "you can't buck the system", "the system is down", "the economic system", "in-car stereo system", "biological system", "paperwork system", "the financial environment", "closed system", "open system", "dynamic system", "in equilibrium"

We use these "system words" a lot. What do they mean

System

- "A system is an organised or complex whole: an assemblage or combination of things or parts forming a complex or unitary whole." (Kast & Rosenzweig)
- "A system is a set of interrelated elements" (Ackoff)

Example: an air traffic control system

Communications
Display
Radar
Flight Plan Database

System

Example: an air traffic control system

Systems are not just about technology

Controllers
ATC Procedures
Aircraft
Airports
Training

System

- An abstract system is one all of whose elements are concepts

Examples: languages, philosophic systems, number systems

- A concrete system includes elements which are physical *things*

Examples: aeroplanes, financial markets,

Sub-systems

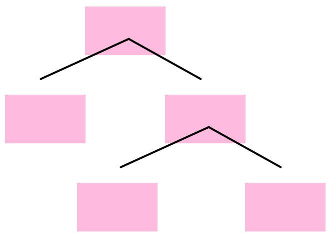
- If the elements of a system are of a reasonable degree of complexity - that is, in general, are themselves systems, we call them sub-systems.

Example: an aircraft

Engine
Avionic
Cockpit
Airframe
Control

Organisation

- The predominant mode of organisation is hierarchical. Systems are composed of sub-systems, sub-systems are composed of sub-sub-systems and so on.



- In very complex cases we talk of “systems of systems”

Example: integrated battlespace command and control

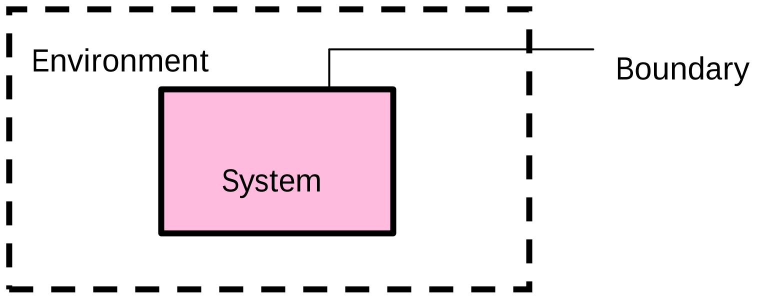
State

- The state of a system at a moment in time is the set of values of *relevant* properties which that system has at that time.
- Any system has an unlimited set of properties - only some of which are relevant for any particular set of purposes.

Examples: mass=10g, colour=red, market liquidity=high

Environment

- The environment of a system is the set of elements (and their relevant properties) which are NOT part of the system - but a change in any of which can produce a change in the state of the system



Environment

- The choice of the boundary is subjective. Different people may divide a domain of discourse into different systems and environments.

An architect views a house as the system comprised of mechanical, electrical, heating and water sub-systems. The electrical supply system is in the environment.
The electrician may view the electrical sub-system together with the electrical supply system as the system with the house as its environment.

Environment

- Setting boundaries is very important when analysing and designing a system. It limits your investigation and problem solving "space".

Example: Imagine you are designing a new electrical car. Are the repair shops, refuelling stations and parts supply part of the system you are designing or not? How much Do they affect the design of the car? Can you change them? How much would changes in them affect your design (robustness)?

Closed and Open

- Systems can be considered closed or open.
- Closed systems do not interact with their environment. Any closed system tends to move towards a chaotic or random state in which there is no further potential for energy transformation or work - an increase in entropy.
- Open systems have a dynamic relationship with their environment, receiving inputs, transforming these inputs and exporting outputs. They adapt to their environment by changing the structure and nature of their components - they can be said to be in equilibrium.

Goals

- Systems generally have purpose (sometimes the word mission is used). Their purpose may be to maintain a particular state or to achieve a goal, that is some preferred outcome. It is not always clear what the purpose of a complex system is.

Example: The National Health Service is a system. What is its purpose? Is this purpose one shared by all its sub-systems?

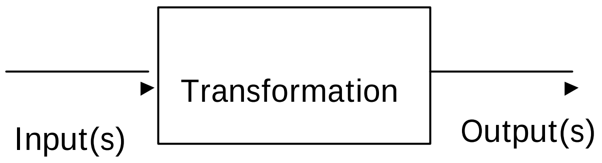
Behaviour

- In achieving their purpose systems exhibit dynamic characteristics - behaviour.
- It may not be easy to understand this behaviour - systems can behave counter-intuitively.

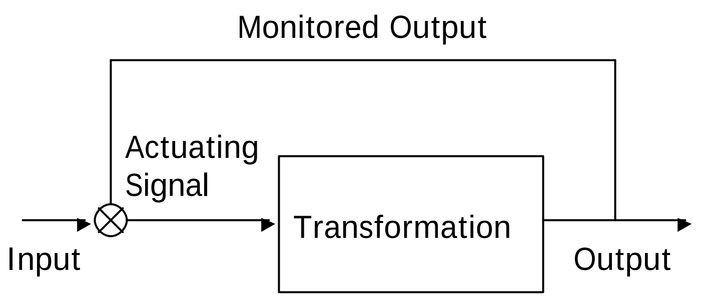
Example: Arrest of dealers in hard drugs?

Inputs and Outputs

- A general view of a system



Feedback and Control



A very large class of systems look like this, with feedback control to provide stability.

Compositionality & Emergence

- If you know something about the behaviour of a sub-system and you compose that sub-system with other sub-systems to constitute a new system the behaviour of that sub-system may change
- New behaviours - desired and undesired - may arise because of the composition. These behaviours are known as emergent.

Example: Feature interaction in telephone systems. Ring Back When Free and Call Divert.

Systems Engineering

- Software is an important part of many large and complex real-world systems but only a part!
- It is embedded in a broader systems context!
- This means that you have to “engineer” the system not just the software.

Implications

- This means you need to:
  - Work with engineers from multiple disciplines
  - Avoid “software mindset” think carefully about the different characteristics of the constituent technologies
  - Configure a systems engineering process, approaches which work for software will not work in other settings

Partitioning

- Partitioning is dividing the system according to implementation technology
- In general delay partitioning till as late as possible
  - Early partitioning can result in non-optimal solutions
  - Once partitioning decisions have been made they can be very costly to change

Key Points

- Throughout the Units which follow you should “think systems” not just software.
- Very few *real problems* are amenable to being solved by software alone they require *systems* solutions.