Systems and systems thinking

What is a system?

Definition:

A system is a set of related processes and components within a recognisable boundary that receives inputs and outputs stuff. The system has a purpose.

A group of interrelated components that work together to achieve a common purpose. It is recognisable as a system.

Characteristics:

- process/set of procedures/instructions - explicit/implicit
- parts,components
- inputs/outputs
- purpose
- boundary
- relationships between the different parts - (structural or behavioural)

Examples:

Health system
Ticketek
digestive system
human being
computer system
production cycle (manufacturing cycle)
city infrastructure
solar system
ecosystem
payroll system!

Four Types of Systems

1. **Biological** - e.g. organism/ a cat
2. **Social** - e.g economy, transport network
3. **Mechanical** - e.g car
4. **Natural** – e.g. weather

where do computer systems fit into this?

We can also categorise systems as Hard or Soft

Hard Systems (quantifiable and describable)

- the problems associated with such systems are well-defined
- they have a single, optimum solution
- a scientific approach to problem-solving will work well
- technical factors will tend to predominate

Hard Systems Thinking can be characterised as having an objective or an end to be achieved, and a system can be engineered to achieve the stated objective.

In computing terms these often built with software engineering methods
(often the puzzles or problems of IT development)

Soft Systems (fluid, people oriented, hard to describe accurately)

- associated problems hard to define
- system can be interpreted in different ways
- has social/cultural/human factors

Soft Systems Thinking can be characterised as having a desirable end, but the means to achieve it and the actual outcome are not easily quantified or agreed.

In computing terms these are often built with IS development methods
(often the problems or messes of IT development)

Systems thinking
Linear, complex or adaptive systems?

1. Linear Systems Thinking

From 17th century prevailing Western view was of **linear** systems
(world as a machine metaphor), i.e. mechanistic view (Newtonian physics)

- universe is knowable (realist ontology)
- the way to 'know' it is by reductive analysis (positivist epistemology)
- relationships flowed one way and worked on simple cause and effect
- systems have predictable behaviour

*(**great for puzzles**, ok for problems, no good for messes!)*

Early IT developments and methodologies followed this view - e.g structured methodologies like SSADM (Structured Systems Analysis and Design Methodology)

---

2. Systems Thinking

General Systems Theory (Bertalanffy 1950's)

"Collection of parts that interact with each other to function as a whole BUT is more than the sum of its parts – a system is **the product of the interactions** between its parts."

this resulted in a new way of thinking about systems

Systems Thinking -

- we may never know everything about the whole system
- interested in how all parts of a system interact and influence other parts
- interaction with environment is also important (see adaptive systems below)
- Attempting to understand systems by examining the interactions and linkages between the components (a holistic approach)
- Not interested in just understanding the components (reductionism)
- encourages a holistic problem solving approach.

*(ok for puzzles, **pretty good for problems**, not much help with messes)*

Information systems developments and methodologies (particularly OO based ones) adopted this view from around the 80's. The systems came to be seen as more 'complex' and less predictable.

As systems thinking matured it also recognised

3. Complex Adaptive Systems (Holland,Gell-Mann 1980's)

Complex Adaptive Systems (Complexity Theory - Santa Fe Institute)

- Can receive and act on feedback - it learns! (adaptive)
- will produce unpredictable **'emergent behaviour'** often as the result of simple interaction rules embedded in components. eg
    - Boids – Craig Reynolds - http://www.red3d.com/cwr/boids/
    - Massive – Weta Workshop http://en.wikipedia.org/wiki/Massive_(software) and http://www.massivesoftware.com/
    - https://killscreen.com/articles/the-software-behind-lord-of-the-rings-giant-battles-now-has-a-playable-demo/
    -

*(useless for puzzles, not helpful for problems, **good for messes?**)*

Recent methodologies are based on this!

"But another explanation for the shift lies in the complexity of the systems we build. The behavior of a simple system is often easy to understand as the sum of the behavior of its component parts; good engineering practice is to design components with well-defined and reliable behaviors for precisely this reason. As systems become more complex, this reductionist way of understanding them fails; they behave in ways that cannot feasibly be predicted from understanding of the individual parts, or were not expected by the system designer who assembled the parts, or both" (Jeffrey C. Mogul, 2005)

What does this mean for systems analysis?

System developers need to be aware of

- Mechanistic vs systemic view of systems
- Hard vs Soft systems
- Machine type vs organic type system (or hybrid?)
- Predictable vs emergent behaviour

Knowing what kind of system you are developing helps to determine how you will approach it and what tools might be useful.
