Systems Integration

2 - Extending the Architecting Paradigm

Pere Palà

iTIC http://itic.cat

v1.1 September 2021

Source: A significant part is from Mark W. Maier and Eberhardt Rechtin's *The Art of Systems Engineering 3rd Ed*

Classical architecting

Classical architecting paradigm

- Egypt, >4000 years ago
- Ambitious system
- Interrelationships grow much faster than the number of elements
- Relationships not only technical: A burial site plus:
  - demonstration of political and religious power
  - secure repository of ruler's wealth ...
- Technical, financial, political and social complications

Complexity

Responding to complexity

- Complex: Composed of interconnected or interwoven parts
- System: Set of elements so connected or related as to perform a unique function not performable by the elements alone

Different techniques for different complexity levels

- Analytic tools for simple problems: circuit analysis works and gives insight for simple circuits (a few nodes, low-order transfer functions)
- Analytic tools fail for complex problems!
- Complex problems require experience-based heuristics, abstraction, modeling
- Key idea: simplify the problem concentrating on the essentials
- Reponse to complexity: during conceptual stage but also during development

ICT complexity

Change

- Advent of smart, software-intensive systems
- Past: Software was glue for hardware elements
- Now: Center of system
- Past: Hardware first
- Now: Software first
- Past: 70% hardware, 30% software
- Now: 30% hardware, 70% software (and growing!)

Objectives

- Simplifiy software development
- C vs ASM. AVR vs PIC
- Good software architectures (layers, information hiding ...)
- Good hardware architectures (hierarchy, locality of communication, interface transparency ...)

Foundations of Modern Systems Architecting

Even if real practice is different in each case, Systems Architecting has the following foundations:

- Systems approach
- Purpose orientation
- Modeling methodology
- Ultraquality
- Certification
- Insight

Systems Approach

- Focus on system as a whole. Strong links: *Value judgements* $\leftrightarrow$ *Design decisions*. *Problem* $\leftrightarrow$ *Solution*
- Systems: collection of things that *together* produce something unachievable by its elements *alone* (automobiles: transport)
- Arquitect does not assume “requirements” are fixed. Joint exploration of what system attributes are worth paying for.
- Arquitect: concentrate on the critical few details that matter (avoid overload by the rest)
- How deeply should the architect delve into each discipline and each subsystem?
- How to know what details are essential? (Experience, dialog with specialists, knowing the system's needs)

Exercise

Build a new system which you understand. Must outperform a version by a competitor. What do yo expect to be the critical details and in what disciplines or subsystems? What elements do you think you can safely leave to others? What do you need to learn the most about? Remind that you will be responsible for *all* aspects of the design!

Purpose Orientation

- Systems: driven by client's purposes
- President: send man to moon and back
- System: *useful* purpose at *affordable* cost in *acceptable* period of time

- Apolo project
- Douglas Aircraft DC-3
- Space Shuttle: low-cost transport to low earth orbit
- Supersonic transport
- Arpanet/Internet: structure with *good bones*

- Success is not (only) technical success but success in mission!

Modeling Methodology

- Model: Abstract representation of the system/subsystems to make predictions and analyze performance, costs, schedules and risks
  - Balsa wood and paper model of building
  - Software prototype
  - Computer model of a communication network
  - Mental model of users
- Conceptual models, engineering models, simulation models, test models, operational models...
- Models evolve as the system becomes more concrete
- Models as standard to compare against to verify consistency
- Failure models: cause and cure of problem

Ultraquality

- Ultraquality: Very high level of quality. Can not be measured or certified. There exists only the ability to measure the *absence* of ultraquality
- Zero defects in *any* part
- Automobile should not fail. Aircraft (100 times more complex) should not fail
- How can this be achieved?
  - Mixture of analytical and heuristic approaches
  - Measurements (indirect)
  - The view that each participant is a buyer and a supplier (do not buy defective items, try not delivering items to be refused)
- Flight control computers for statically unstable aircrafts: mtbf 10^{10} hours
- Redundancy? Triple redundancy plus voting system. More components!
- Ariane 5. Computer crash in both primary and secondary computers

Certification

- Avoid value judgements
  - The client should make them
  - Trap: Client asks “what would you do?”. You become the system owner (with rights and responsabilities)
- Clear avoidance of perceived conflict of interest
  - Example: Architect recommending a system that the architect will supply and profit from
- Do not mix with project management

Exercise

Pick a system for which the purpose is reasonably clear. What tests would you, as a client, demand? (to pay) What tests would you, as a builder, contract to pass (to be paid). Whose word would you accept that the tests have or have not been passed?

Insight and heuristics

- One insight is worth a thousand analyses
- Insight: Ability to structure a complex situation in a way that increases the understanding of it
- Comes from experience
- Heuristics = guide

Think about it:

- Success comes from wisdom
- Wisdom comes from experience
- Experience comes from mistakes

- Client won't pay for mistakes!
- Try profiting from collective wisdom!