System Architecture Approaches

18-849b Dependable Embedded Systems

Michael Scheinholtz

Tuesday, April 6th 1999

Required Reading: The Synthesis of Complex Systems, Rechtin

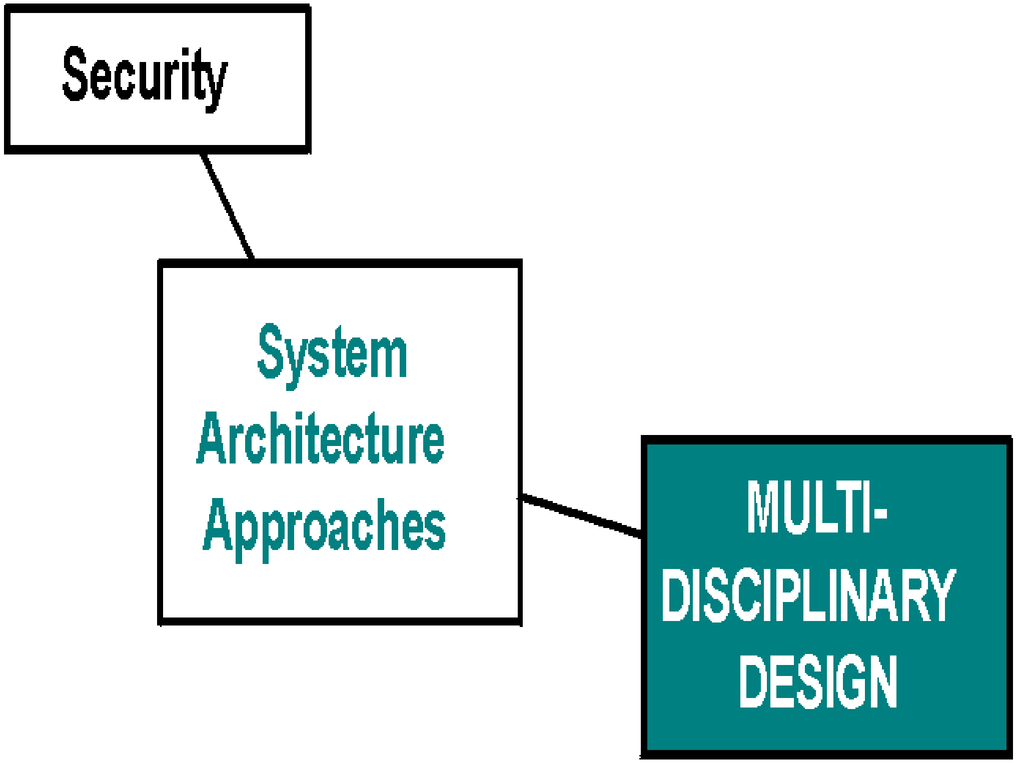
Authoritative Books: The Art of Systems Architecting, Rechtin, Eberhardt

Carnegie Mellon

Overview: System Architecture Approaches

- **Introduction**
  - Multi-Disciplinary Design
- **Key concepts**
  - The Systems approach
  - Heuristics, Models, and Metaphors
  - Creativity and Innovation
- **Tools / techniques / metrics**
  - CASE tools
  - Modeling Languages
- **Relationship to other topics**
  - Security
  - Multi-disciplinary Design
- **Conclusions & future work**

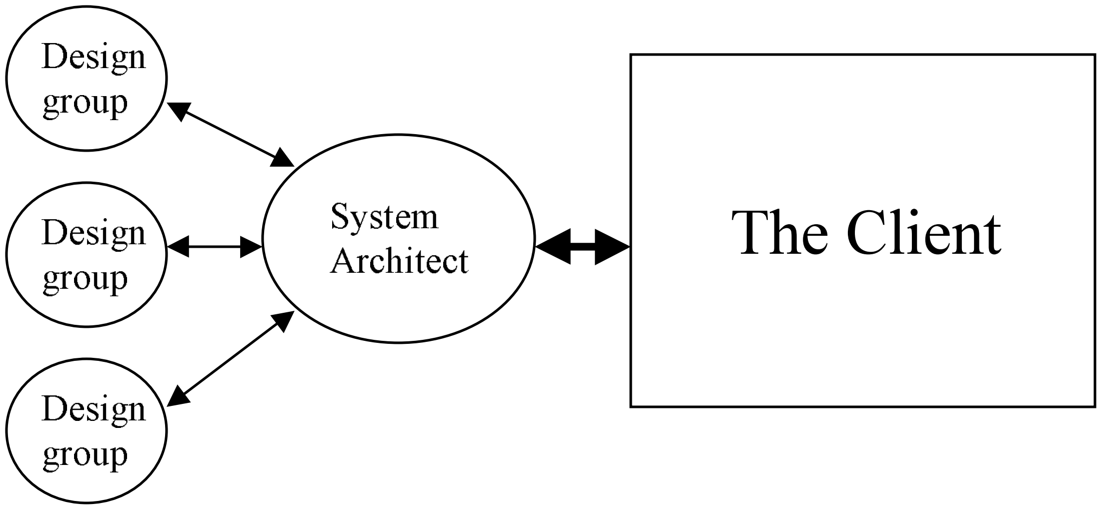
YOU ARE HERE MAP



System Architecture Approaches

- **Definitions**
  - System: A collection of different objects related in such a way that they can do more than what the sum of their parts separately could produce.
  - Systems Architecture: The underlying structure of a system.
  - Systems Architecture Approaches: How to make an architecture.
- **The Systems Approach**
  - A person or small group balancing the needs of client and designers while controlling system behavior.
- **Heuristics, Models, and Metaphors**
  - Ways of reducing complexity and give direction.
- **Creativity and Innovation**
  - How do you make a revolutionary system?

The Systems Approach



- **Usually one person or a small team is responsible**
  - Using one person’s overall vision seems to yield the best results
    - Cray Supercomputers, SR-71, DC-3, Sistine chapel ceiling, any symphony...
- **System architects deal with emergent properties**
  - safety, performance, quality
- **The Architect is driven by client**
  - little interaction between client and design groups

Heuristics, Models, and Metaphors

- **Complexity is the enemy of every project, and any way to abstract problems or provide direction is helpful.**
- **Heuristics: Ways to give direction for an architecture**
  - Example: Simplify, simplify, simplify
    - Complexity is the main adversary to any project
  - Actions are probably better if several heuristics agree
- **Metaphors**
  - A familiar context used to explain the unknown.
    - The “Desktop Metaphor” for the personal computer…
  - Can be dangerous if they abstract away important details
- **Models**
  - Essential for communicating the architects ideas to all parties
  - Problematic if they limit or exclude something important

Creativity and Innovation

- **Well understood design**
  - Requirements must be fought over with the customers….
  - Design document created from the requirements.
  - System implemented

- **What happens when the system is completely new?**
  - Must all ground breaking systems be created by one individual?
- **Synthesis**
  - When a system is completely new, requirements and design must be worked on together.
  - Analysis is almost useless.

Tools / Techniques

- **Traditional Design Techniques**
  - Pronouncement: Follow a set of rules to design a new version of something already in existence.
  - Rational: Formalize the design into a provably optimal solution
  - Consensual: Design by committee
  - Heuristics: Use general advice from the years past.
- **Can it be taught?**
  - Can system architecture ever be turned from art to science
- **CASE tools?**
  - Rational Rose, could one day help with transition from requirements to code.
- **Modeling Languages**
  - UML, can be good, clear ways of expressing a design

Metrics

- **Certification**
  - Client agrees to accept the system
  - The new product sells well
  - It requires little maintenance
- **Architectures can be evaluated, once the system is built**
  - User satisfaction is a good measure of the quality of an architecture, but it requires a complete system implementation.
  - This is, of course, too late for an easy fix.
  - Its easy to spot a good architecture, but hard to make one yourself
  - Another heuristic: The worst project mistakes are made on the first day.
- **Architectures should have "good bones".**
  - Only becomes obvious after time: IBM PC's 640K memory limit.

Relationship To Other Topic Areas

- **Security**
  - Security is an emergent property.
- **Multi-disciplinary Design**
  - Making the system architecture is a key phase of any design
- **System Architecting relates to any area involving design or emergent properties (properties not created by any one piece of a design).**
  - Safety
  - Reliability
  - Life Cycle
  - Ultradependability
  - Quality of Service
  - etc...

Conclusions & Future Work

- **The best approach has been to use one person or small team to control the overall development of the project.**
  - Average case: use a few talented people and a good process.
- **Any way to reduce complexity can help, up to a point.**
- **The Future**
  - How can systems architecting be improved?
    - Can you teach Salieri to compose operas like Mozart?
    - A certain amount of creativity is innate, but it can be improved.
    - Much of creativity is picking out the good ideas from the bad.
  - What can almost be taught is how to develop an idea.
    - Steve Jobs had the tools he needed to implement his new computer.
  - The single client model may no longer apply.
    - Many projects may be made without a central client (ie. Internet)

The Synthesis of Complex Systems

- **Synthesis comes first**
- **Major points**
  - Synthesis is a necessity when a new system is designed
  - Heuristics and metaphors provide common sense in context.
  - A unified model is important to the success of a project
- **Conclusions**
  - Systems synthesis is much more quantitative than system analysis or engineering ,and it requires a qualitative approach.
  - The synthesis phase may create some of the worst problems in a system because it holds the earliest decisions.