Architectural Stability

Some Preliminary Comments

Anthony Finkelstein

Software Systems Engineering Group
Department of Computer Science
University College London

Outline

- Definitions
- Context
- Problem
- Example
- Goals
- Issues
- Assumptions

Definitions

- Software Architecture
  - A "structural" description of a complex software-intensive system in terms of components, connectors and configurations of components and connectors.
- Architectural Style
  - A generic description of a class of software architectures.

Definitions

- Requirements
  - The real-world goals for, services provided by, and constraints on a large and complex software-intensive system.

Context: Requirements & Architecture

- Requirements engineering and architectural design are intertwining processes
  - system conception
  - analysis
  - 'brownfield development'
  - component-based systems construction
- Processes that involve co-development of requirements and software architecture

Context: Stability & Change

- Requirements change
  - such change is inevitable and must be accommodated as it reflects the changing needs of the stakeholders and the environment in which the system works
- Stable architectures
  - it is expensive and may be difficult to change the architecture of the system, a stable architecture is a significant asset during development and in operation

Problem: Architectural Breakdown

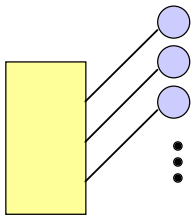
- Observation (1) - some requirements changes 'break' the architecture
  - in other words, some classes of requirements change entail large and disruptive changes to the architecture in order to be accommodated
- Observation (2) - these changes are mostly to 'non-functional requirements'
  - in other words, to global system properties such as security, performance, scaleability and similar

Problem: Architectural Breakdown

- Observation (3) - not all changes are equally likely
  - in other words, we may be able to identify the space of likely changes
- Observation (4) - some architectures accommodate some changes more readily than others
  - in other words some architectures are more robust wrt certain changes than others

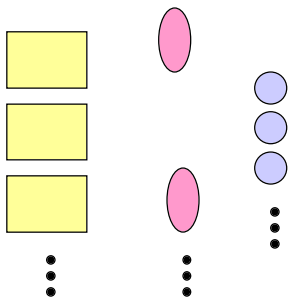
Example I

- Banking system
  - Architectural style
    - multiple client- single server
  - Service evolution
    - Scaleability
    - Security



Example II

- Banking system
  - Architectural style
    - multiple client- multiple server
  - Service evolution
    - Scaleability
    - Security



"Real Example"

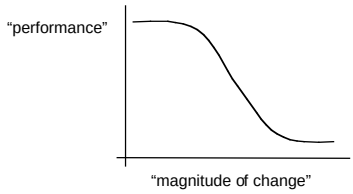
- Moving to mobility
  - Mobile e-shopping
  - Specialist client
  - Caching strategies
  - Narrow bandwidth connectivity
- Straight transition of existing application architecture will not work

Goals

- Would it be possible - given that we can identify the space of likely changes to a set of requirements - to identify an architectural style that will accommodate those changes with minimal disruption
- Would it be possible to do architectural impact analysis given a proposed change

Issues

- Are the informal observations correct?
- What is a disruptive change to an architecture, what are appropriate 'metrics'?
- Do architectures 'shatter' or do they exhibit other failure characteristics such as degraded performance with respect to the requirement prior to failure?



Issues

- How do particular architectural styles respond to different changes?
- How can we identify and document the likelihood of change in the requirements

Assumptions

- Goal-oriented Specification
  - Like KAOS
- Architectural Description Languages
  - Like Darwin

Conclusions

- A research and practical challenge
- Towards a treatment of volatility
- From clay to gold!