Software Architecture
Using Viewpoints and Perspectives

SET2005 - Zurich
11th May 2005

Eoin Wood
Zuhlke Engineering Ltd
49 Great Cumberland Place
London W1H 7TS
ewo@zuhlke.com
Content

- Defining Software Architecture
- The Software Architecture Problem
- Viewpoints to Guide Structure
- Perspectives to Guide Qualities
- Summary
The software architecture of a program or computing system is the **structure** or structures of the system, which comprise software **elements** the externally visible **qualities** of those elements, and the **relationships** among them

Len Bass, Paul Clements and Rick Kazman (SEI)
*Software Architecture in Practice, 2nd Edition*
Role of Software Architecture

A crucial bridge between requirements and design
The non-functional system characteristics ("-illities")
- Performance, Efficiency, Security, Availability, …

Quality properties are crucial to stakeholders
- Slow functions don’t get used
- Unavailable systems cause business interruption
- Security problems cause headlines
Yet quality properties are often an after-thought

Addressing quality properties is a key architectural task
- Understanding “real” stakeholder needs & required tradeoffs
- Often expensive to “retro-fit”
The Software Architecture Problem

Why software architecture is difficult
- Multi-dimensional problem
- Diverse stakeholder community to serve
- Making trade-offs inherent in the process
- Often no one “right” answer

Architecture practice today is largely ad-hoc
- Little standardisation in description
- Difficult to compare and discuss alternatives
- Unclear how to structure architectural activities
- No framework for handling quality properties
Dealing with architectural structure

- Decompose the architectural description into views
  - Each view addresses one aspect of the architectural structure
- Guide the development of each view via a viewpoint
  - The viewpoint contains proven practice, pitfalls, etc.
- Well understood approach
  - RUP/Kruchten “4+1”
  - Siemens set
  - RM-ODP set
  - Rozanski & Woods set
Viewpoints and Views

Inter-relationships

- Architectural Description
  - View
    - Viewpoint
      - Defines
        - 0..*
Viewpoints and Views

Example viewpoint set

- **Functional**: elements, connectors, interfaces
- **Information**: entities, constraints, relationship, ownership, usage
- **Concurrency**: processes, threads, coordination, element mapping
- **Development**: layers, module structure, standard design, codeline
- **Deployment**: hardware, network, dependencies, process mapping
- **Operational**: installation, migration, administration, support

[Rozanski and Woods; “Software Systems Architecture” – Addison Wesley, 2005]
Viewpoints and Views

**Viewpoints provide**

- A store of knowledge and experience
- A guide to the architect
- Templates to guide the process

**Views provide**

- A structure for description
- A separation of concerns
- Improved stakeholder communication
Limitations of viewpoints

- Quality properties are critical
- Viewpoints typically don’t consider quality properties
- Quality properties usually need cross-viewpoint consideration
- Viewpoints may lead to late consideration of quality properties
Architectural Perspectives

Dealing with quality properties

- Use *perspectives* to guide the architect in achieving the required quality properties
  - Each perspective addresses one major quality property
- The perspectives guide changes to the *views*
- A new approach, compatible with viewpoints
  - Related to SEI’s “tactics” work
Defining perspectives

Architectural *perspective* is a collection of *activities, checklists, tactics and guidelines* to *guide the process* of ensuring that a system *exhibits* a particular set of closely related *quality properties* that require consideration across a number of the system’s architectural views.

Rozanski and Woods, 2005
Architectural Perspectives

Adding perspectives to the inter-relationships

1..* defines 0..*

Architectural Description

applied to 0..*

View

defines 0..*

Viewpoint

Perspective
Using Viewpoints with Perspectives

- Security Perspective
- Accessibility Perspective
- Performance Perspective
- Location Perspective
- Availability Perspective
- Regulation Perspective
- Evolution Perspective
- etc.

Architecture

- Functional View
- Information View
- Information View
- Development View
- Deployment View
- Operational View

Stakeholders ➔ Architecture
Our initial core set

- Performance and Scalability
- Security
- Availability and Resilience
- Evolution
- Also: Location, I18N, Usability, Regulation, …

Different sets in different domains
Architectural Perspectives

Performance and Scalability

- **Concerns**: processing volume, response time, responsiveness, throughput, predictability
- **Techniques**: performance requirements definition, performance modelling, workload characterisation

Security

- **Concerns**: authentication, authorisation, confidentiality, integrity, accountability, availability, intrusion detection, recovery
- **Techniques**: threat identification, threat assessment, vulnerability analysis, application of security technology
Availability and Resilience

- **Concerns**: classes of service, planned / unplanned downtime, mean time between failures, mean time to repair, disaster recovery, redundancy, clustering, failover

- **Techniques**: MTBF and MTTR prediction, availability schedules, availability models, availability technology application

Evolution

- **Concerns**: flexibility, extensibility, functional evolution, deployment evolution, integration evolution

- **Techniques**: design for change, architectural assessment, configuration management, automated testing, build and release management
Viewpoints and Perspectives can

- Provide a framework for sharing knowledge
  - Viewpoints guide design of structures
  - Perspectives guide design for quality properties
- Act as a store of architectural knowledge
  - Document proven practice
  - Help standardise language and approach
    - Help to standardise languages and approaches
- Act as a tutorial for new architects
- Act as a guide for working architects
- Act as aide-memoir for experienced architects
More Information

Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives

Nick Rozanski and Eoin Woods
Addison Wesley 2005

http://www.viewpoints-and-perspectives.info