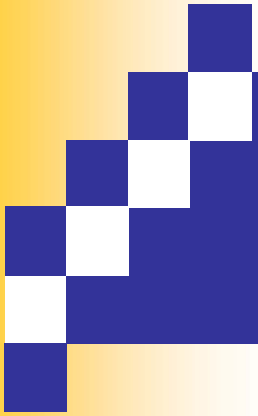


Information Systems Architecture

Stakeholders, Viewpoints, Perspectives

OOPSLA 2006, 22nd October 2006

Eoin Woods
eoin@eoinwoods.info
www.eoinwoods.info

1

Content

- Defining Software Architecture
- Stakeholders
- The Software Architecture Problem
- Viewpoints to Guide Structure
- Perspectives to Guide Qualities
- Example Application
- Uses for Viewpoints and Perspectives

2

Defining Software Architecture

- A common definition …
  - *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

3

Defining Software Architecture

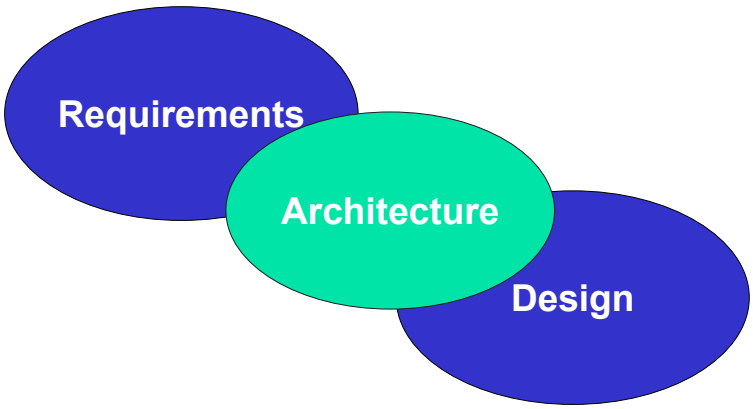
- An alternative definition …
  - *The set of system design decisions that dictate the fundamental structure and properties of a system*

    Thus, the set of decisions that will cause the system to fail if made incorrectly

4

Role of Software Architecture

Crucial bridge between requirements and design



5

Architecture & Requirements

- Requirements are an input to architecture
  - Requirements frame the architectural problem
  - Stakeholder needs and desires
- Architecture must influence requirements
  - "The art of the possible"
  - Stakeholder understanding of risk/cost
  - Stakeholder understanding of possibilities

6

Architecture & Design

- Architecture frames design
  - architecture *is* part of the design process
- Captures the system-wide decisions
  - what has to be consistent or constant
- Importance of role increases with scale
- Perfectly compatible with agile
  - even XP with the right approach

7

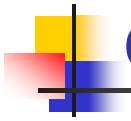
Just Design, Surely?

Architecture	Global	Intentional
Design	Local	Intentional
Implementation	Local	Extensional

"*Architecture, Design, Implementation*",
Amnon Eden And Rick Kazman, ICSE 2003

Intentional: infinitely many possible ways of satisfying the statement (i.e. constraint rather than instruction)
Local: satisfied in design "d" => satisfied in all possible extensions of "d"

8



Quality Properties

- Non-functional characteristics ("-illities")
  - Performance, Security, Availability, …
- Often crucial to stakeholders
  - Slow functions don't get used
  - Unavailable systems stop the business
  - Security problems cause headlines
- Yet often an after-thought

9

Quality Properties

- Addressing quality properties is a key architectural task
  - Understanding real stakeholder needs
  - Understanding what is possible
  - Making the key trade-offs to allow delivery
  - Avoiding expensive "retro-fit"

10

Stakeholders

- Identifying Stakeholders
  - People, Groups, Entities
  - Those who have an interest in or concerns about the realisation of the architecture
- Importance of Stakeholders
  - Architectures are built for stakeholders
  - Decisions must reflect stakeholder needs
  - Involving a wide stakeholder community increases your chances of success

11

Talking Point: Stakeholders

- Who cares whether systems you're creating get built?
- Why do they care?
- Are they positively or negatively impacted?

Draw up a short list of the stakeholders for your system(s)

12

Stakeholders

- Attributes of a good stakeholder
  - *Informed*, to allow them to make good decisions
  - *Committed*, to the process and willing to make themselves available in a constructive manner, even if decisions are hard
  - *Authorised*, to make decisions
  - *Representative*, of their stakeholder group so that they present its views validly

13

Stakeholder Groups

- *Acquirers* pay for the system
- *Assessors* check for compliance
- *Communicators* create documents and training
- *Developers* create it
- *Maintainers* evolve and fix it
- *Suppliers* provide parts of the system
- *Support Staff* help people to use the system
- *System Administrators*, keep it running
- *Testers* verify that it works
- *Users* have to use the system directly

14

The Challenge

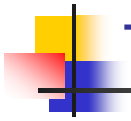
- Essential difficulties
  - Multi-dimensional problem
  - Highly complex mix of people and technology
  - Diverse stakeholder community to serve
  - Making trade-offs is essential but hard
  - Often no "right" answer

15

The Challenge

- Accidental difficulties
  - Little standardisation in description
  - Difficult to compare and discuss alternatives
  - Little standardisation in architectural activities
  - Little sharing of proven practice and known problems and their solutions
  - No framework for handling quality properties

16



The Challenge

- To help meet the challenge
  - Organise the architectural design process
    - roles & activities, relationship to requirements & design
  - Define the use of architecture artefacts
    - which models? when? why?
  - Capture, classify and share knowledge
    - best practice, problems and pitfalls, proven solutions

17

Architectural Viewpoints

- Help to deal with architectural structure
  - Decompose arch. description into views
    - each view addresses one aspect of the system
    - functional view, deployment view, …
  - Guide development of views via viewpoints
    - viewpoint contains proven practice, pitfalls, …
    - each view defined by one viewpoint
  - Organises the process and the artefacts

18

Architectural Viewpoints

- Well understood, widely applied
  - RUP/Kruchten "4+1" set (1995)
  - RM-ODP set (1995)
  - Siemens set (1999)
  - Garland and Anthony set (2003)
  - Rozanski & Woods set (2005)
  - Conceptual basis in IEEE 1471 (2000)

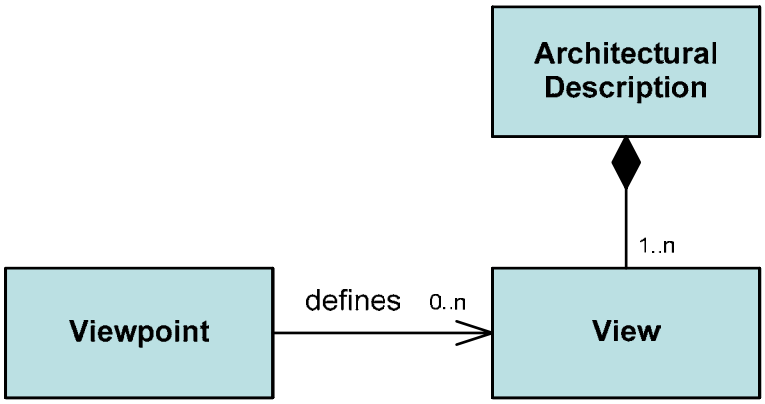
19

Viewpoints and Views

- IEEE 1471 provides standard definitions
  - A viewpoint is a collection of patterns, templates and conventions for constructing one type of view. It defines the stakeholders whose concerns are reflected in the viewpoint, and guidelines and principles and template models for constructing its views.
  - A view is a representation of all or part of an architecture, from the perspective of one or more concerns which are held by one or more of its stakeholders.
  - from IEEE Standard 1471 – Recommended Practice for Architectural Description (2000)

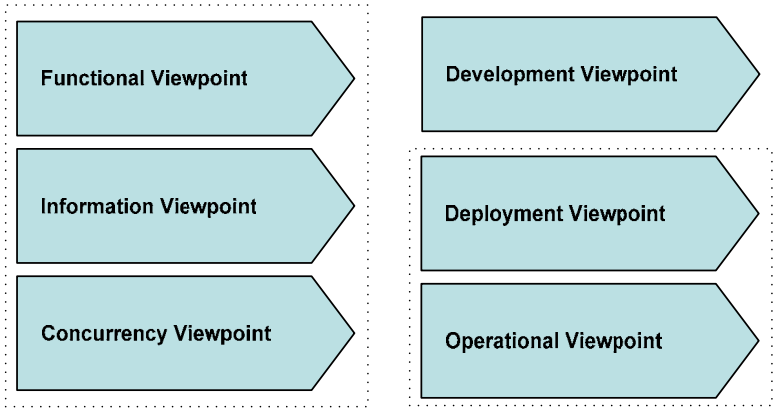
20

Viewpoints and Views



21

Example Viewpoint Set



[Rozanski & Woods, 2005]

22

Example Viewpoint Set

- Core architectural structures
  - Functional
    - elements, connectors, interfaces, responsibilities, interactions
  - Information
    - entities, constraints, relationships, timeliness, usage, ownership
  - Concurrency
    - processes, threads, coordination, element to process mapping

23

Example Viewpoint Set

- Working with developers
  - Development
    - layers, module structure, standard design, codeline
- Moving towards deployment
  - Deployment
    - hardware, network, software dependencies, process to node mapping
  - Operational
    - installation, migration, administration, support

24

Example Viewpoint Set

- Rozanski/Woods Viewpoint Set
  - Aimed at large scale information systems
  - Extension and refinement of Philippe Kruchten's "4+1" set
    - renamed "Logical", "Process" and "Physical"
    - added "Information" and "Operational"
  - Standard content for viewpoints
    - applicability, concerns, models, stakeholders, problems & pitfalls, solutions, checklists

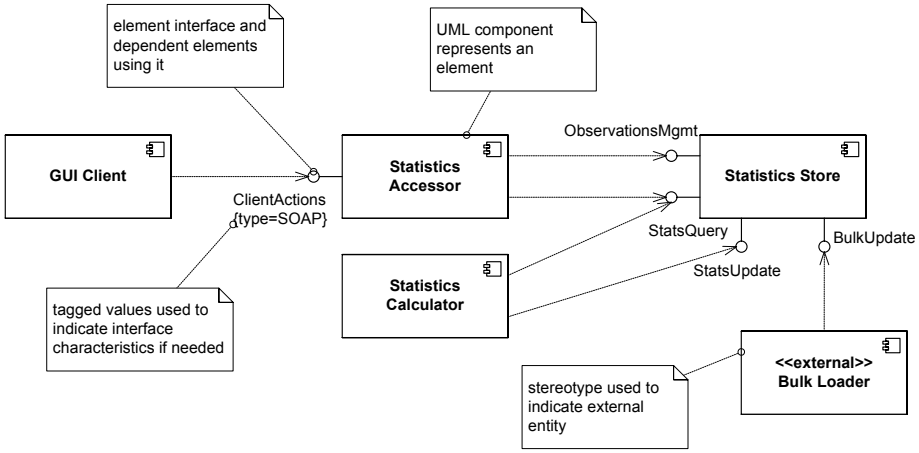
25

Functional Viewpoint

Focus	functional structure of the system
Content	design of runtime functional elements and their responsibilities, interfaces, and primary interactions
Concerns	▪ functional capabilities ▪ external interfaces ▪ internal structure ▪ design qualities
Models	▪ functional structure model
Pitfalls	▪ poorly defined interfaces / responsibilities ▪ infrastructure modelled as functional elements ▪ overloaded view ▪ just drawing pictures …

26

Functional View Fragment



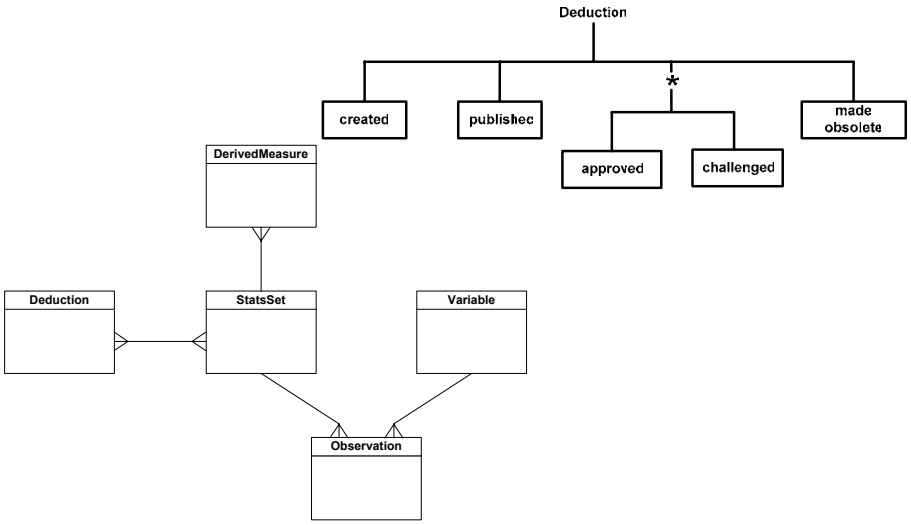
27

Information Viewpoint

Focus	information structure, ownership and processing
Content	design of storage, manipulation, management, and distribution of information
Concerns	▪ information structure, content and flow ▪ data ownership and quality ▪ timeliness, latency, and age ▪ …
Models	▪ static data and metadata structure models ▪ information flow models ▪ information lifecycle models ▪ data ownership and access models ▪ volumetric models
Pitfalls	▪ data incompatibilities ▪ poor data quality ▪ unavoidable multiple updaters ▪ key matching deficiencies ▪ …

28

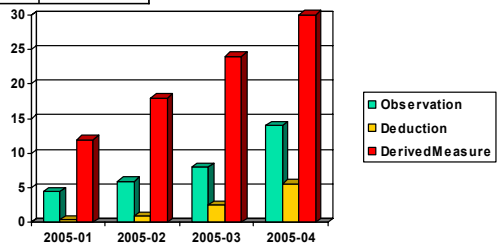
Information View Fragments



29

Information View Fragments (ii)

	Observation	Deduction	Derived Measure
Statistics Accessor	R	C,R,U,D	R
Statistics Calculator	-	-	C,U,D
Bulk Loader	C,U,D	-	-



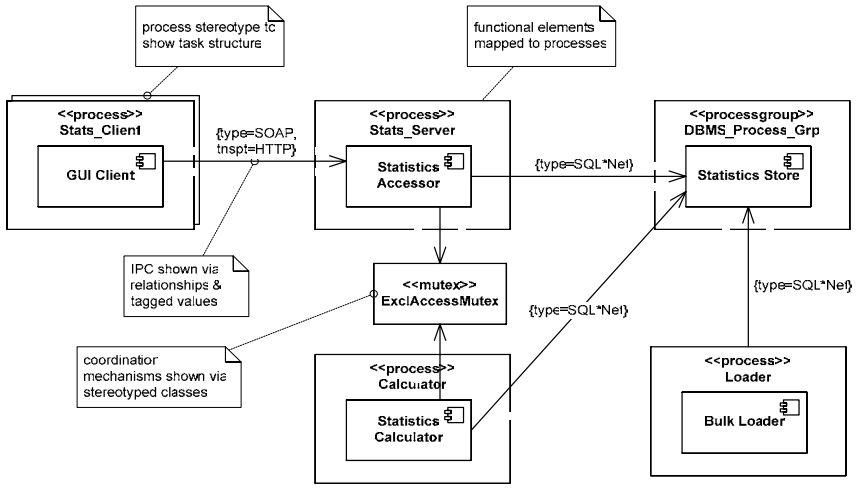
30

Concurrency Viewpoint

Focus	packaging elements into processes and threads
Content	the concurrency structure, mapping functional elements to concurrency units to clearly identify the parts of the system that can execute concurrently, and how this is coordinated and controlled
Concerns	▪ task structure and mapping of functional elements to tasks ▪ inter-process communication & re-entrancy ▪ state management ▪ synchronization and integrity ▪ task startup, shutdown and recovery from failure
Models	▪ system-level concurrency model ▪ system-level state model
Pitfalls	▪ modelling of the wrong concurrency ▪ excessive complexity ▪ resource contention ▪ deadlock and race conditions

31

Concurrency View Fragment



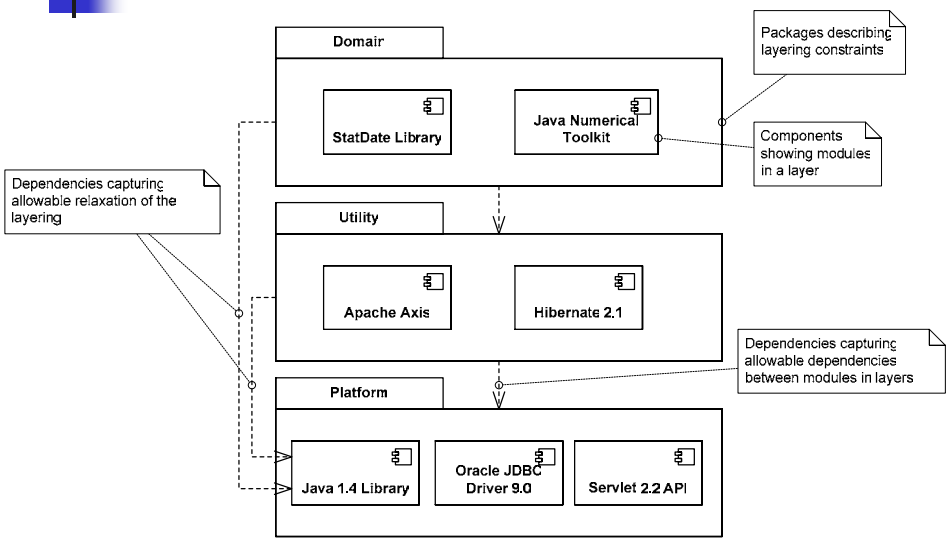
32

Development Viewpoint

Focus	architectural constraints on the software development process
Content	architectural design that supports and constraints the software development process
Concerns	▪ module organization ▪ codeline organization ▪ common processing ▪ standardization of design and testing ▪ instrumentation
Models	▪ module structure models ▪ common design models ▪ codeline models
Pitfalls	▪ too much detail ▪ overburdening the architectural description ▪ uneven focus or lack of developer focus ▪ lack of precision ▪ problems with the specified environment

33

Development View Fragment



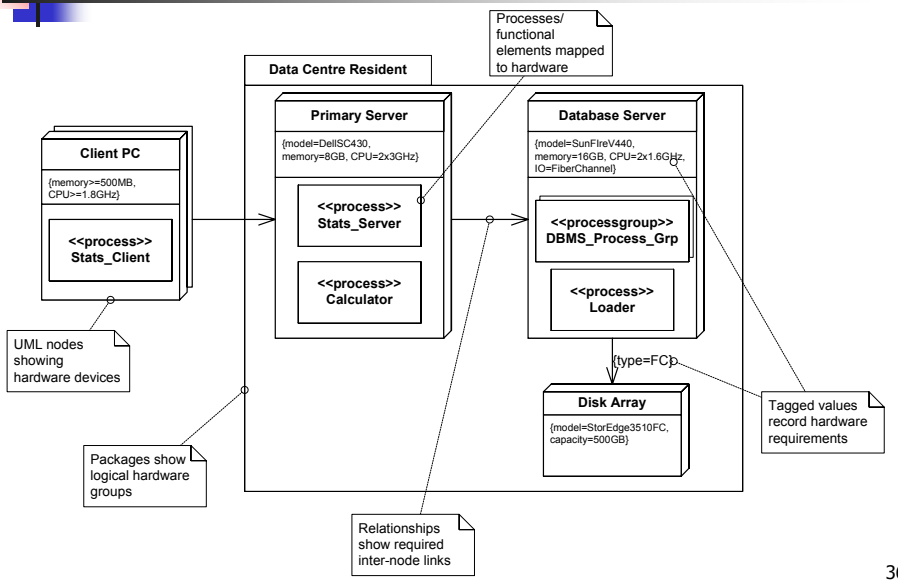
34

Deployment Viewpoint

Focus	runtime environment structure and the distribution of software across it
Content	design of the environment into which the system will be deployed, including the system's runtime dependencies
Concerns	▪ types, specification and quantity of hardware required ▪ third-party software requirements ▪ technology compatibility ▪ network requirements and capacity ▪ physical constraints
Models	▪ runtime platform models ▪ network models ▪ technology dependency models
Pitfalls	▪ unclear or inaccurate dependencies ▪ unproven technology ▪ lack of specialist technical knowledge ▪ late consideration of the deployment environment

35

Deployment View Fragment



36

Deployment View Fragment (ii)

Client PC	▪ Windows XP SP1 ▪ Java JRE 1.4.2_06 or later ▪ Internet Explorer 6.0 SP1
Primary Server	▪ Windows 2003 server, w/sec patches ▪ Java SDK 1.4.2_06 or later ▪ Apache Tomcat 5.5.9 or later

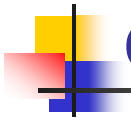
Database Server	▪ Solaris 9.0 w/Aug05 patch cluster
	▪ Oracle 9.2.0.2 Std Edition
	  ▪ 10GB buffer cache, auto sized SGA
	  ▪ auto storage management, 2 table spaces
	▪ OEM 9.2.0.2 installed and working

37

Operational Viewpoint

Focus	system installation, migration, operation & support
Content	defines strategies for how the system will be operated, administered, and supported when in its production environment
Concerns	▪ installation, upgrade and migration ▪ operational monitoring, control and configuration management ▪ performance monitoring ▪ support responsibilities and procedures ▪ backup and restore
Models	▪ installation and migration models ▪ configuration management models ▪ administration, support and escalation models
Pitfalls	▪ lack of engagement with the operational staff ▪ lack of migration and backout planning ▪ insufficient migration window ▪ missing management tools ▪ …

38



Operational View Content

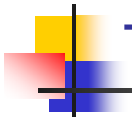
- Installation Model
  - Installation groups
  - Dependencies and constraints
  - Backout strategy
- Operational CM Model
  - Configuration groups and dependencies
  - Configuration parameter sets
  - Operational control (switching between sets)

39

Operational View Content (ii)

- Administration Model
  - Monitoring and control facilities required and provided
  - Required routine operational procedures
  - Required operational action in case of error conditions

40



Talking Point: Views

- Which views would be useful for your systems?
- Who would be interested in each view?
- Which views wouldn't be useful – why?

Draw up a list of which views you would use and who would be interested in each.

Can you sketch fragments of one or two?

41

Viewpoints and Views Recap

- Viewpoints
  - A store of knowledge and experience
  - A guide to the architect
  - Templates to guide the process
- Views
  - A structure for description
  - A separation of concerns
  - Aid to stakeholder communication

42

Break

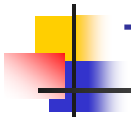
30 minutes

43

Limitations of Viewpoints

- Quality properties are critical
  - existing viewpoint sets don't explicitly consider quality properties
- Quality properties usually need cross-view consideration
  - viewpoints are relatively independent
- Viewpoint focus may lead to late consideration of quality properties
  - qualities are often expensive to add later

44



Talking Point: Qualities

- Which quality properties are crucial to your systems?
- Which views are going to be impacted by considering each such quality property?

Create a matrix showing which qualities are likely to impact which of your views

45

Handling Quality Properties

- A new concept could help
  - Allowing cross-view focus
  - Being quality rather than structure oriented
  - Providing similar organisation and guidance to a viewpoint but for a quality property
  - That can be used in tandem with viewpoints
- We call this new concept a "perspective"
  - or "architectural perspective" in full

46

Architectural Perspectives

*An 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

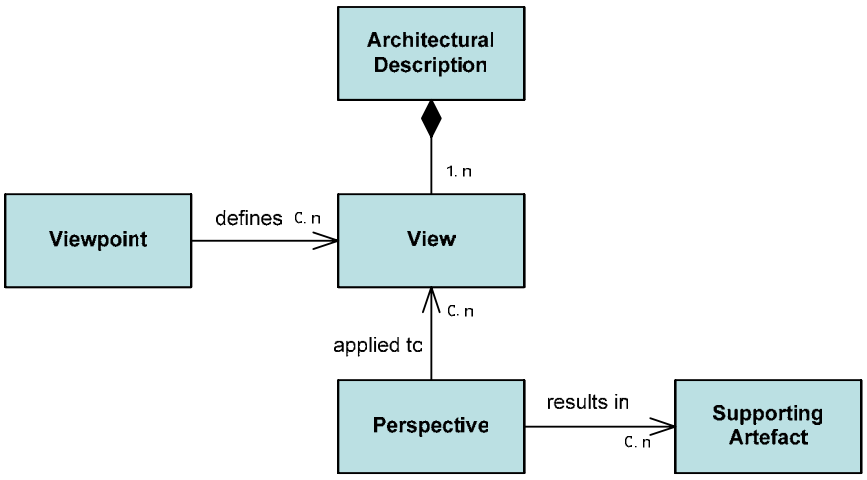
47

Architectural Perspectives

- A guide for dealing with quality properties
  - Guide the architect in achieving the required quality properties
  - Suggest changes to the existing views
  - Avoid possible redundancy between quality and structural views
- A new concept to use with viewpoints
  - Related to and extends SEI tactics work
  - Adds more context and advice to tactics

48

Adding Perspectives



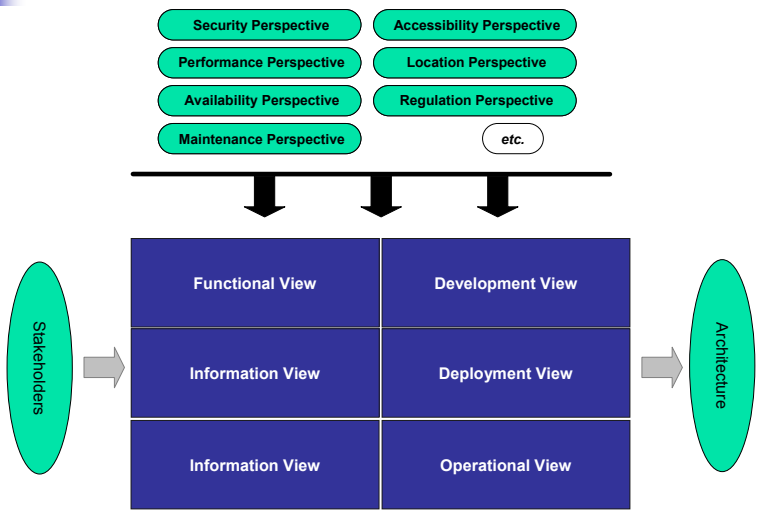
49

Architectural Perspectives

- A simple but effective idea
  - A store of knowledge and experience
  - A guide to the architect
  - Templates to guide the process
- Analogous to viewpoints but for quality properties, rather than structures
- Perspectives "applied" to views to assess qualities and guide changes needed

50

Perspectives and Views



51

Architectural Perspectives

- Our initial core set for information systems
  - Performance and Scalability
  - Security
  - Availability and Resilience
  - Evolution
  - Also: Location, I18N, Usability, Regulation, …
- Different sets in different domains

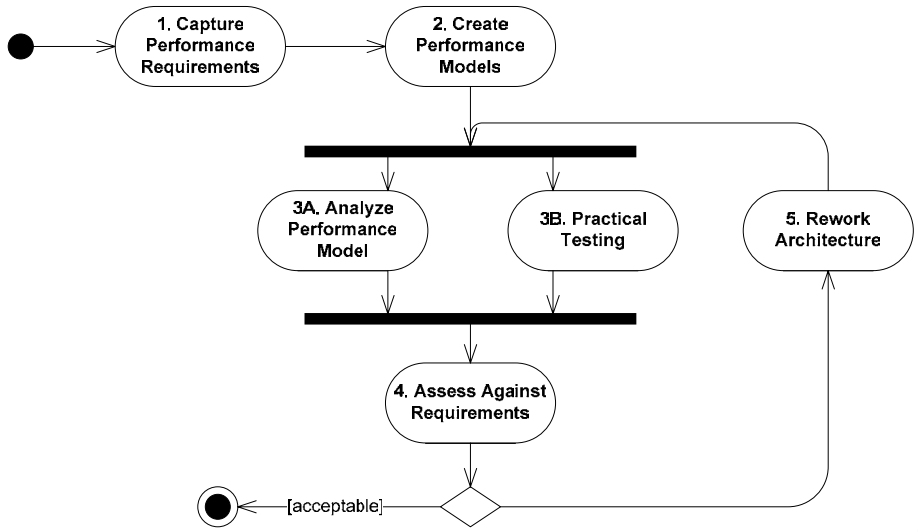
52

Performance and Scalability

Quality	ability of the system to predictably execute within its mandated performance profile and to handle increased processing volumes
Concerns	▪ processing volume ▪ response time ▪ responsiveness ▪ throughput ▪ predictability
Tactics	▪ optimize repeated processing ▪ reduce contention via replication ▪ prioritize processing ▪ consolidate related workloads ▪ …
Pitfalls	▪ imprecise goals ▪ unrealistic models or use of simple measures for complex cases ▪ inappropriate partitioning ▪ …

53

P & S Perspective Activities



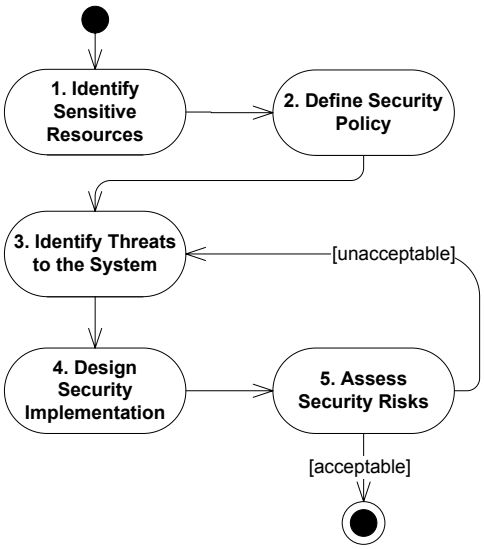
54

Security

Quality	ability of the system to reliably control, monitor, and audit who can operate on resources and to detect and recover from breaches
Concerns	▪ policies ▪ threats ▪ mechanisms ▪ accountability ▪ availability ▪ detection and recovery
Tactics	▪ apply recognised security principles ▪ authenticate the principles ▪ authorise access ▪ …
Pitfalls	▪ no clear requirements or models ▪ complex security policies ▪ unproven or ad-hoc security technologies ▪ …

55

Security Perspective Activities



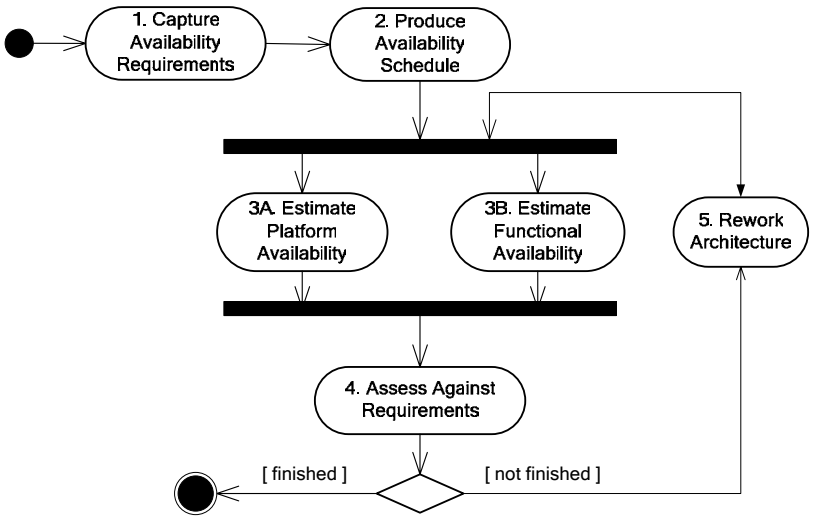
56

Availability and Resilience

Quality	ability of the system to be fully or partly operational as and when required and to effectively handle failures that affect availability
Concerns	▪ classes of service ▪ planned / unplanned downtime ▪ mean time between failures & mean time to repair ▪ disaster recovery ▪ redundancy, clustering, failover
Tactics	▪ MTBF and MTTR prediction ▪ availability schedules and models ▪ application of high availability technology ▪ …
Pitfalls	▪ single point of failure ▪ overambitious availability requirements ▪ ineffective error detection ▪ overlooked global availability requirements ▪ incompatible technologies

57

A & R Perspective Activities



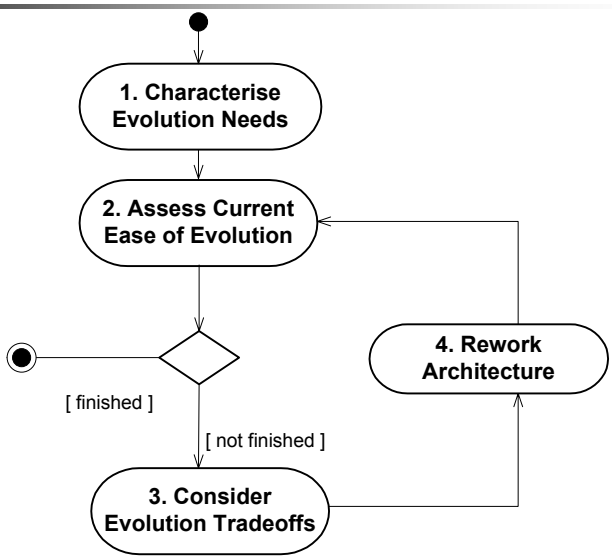
58

Evolution

Quality	ability of the system to be flexible in the face of change, balanced against the cost of providing that flexibility
Concerns	▪ flexibility ▪ extensibility ▪ functional, deployment and integration evolution
Tactics	▪ design for change ▪ architectural assessment ▪configuration management ▪ automated testing ▪ build and release management ▪ …
Pitfalls	▪ prioritization of the wrong dimensions ▪ changes that never happen ▪ impact of evolution on critical quality properties ▪ lost development environments ▪ ad hoc release management

59

Evolution Perspective Activities



60

Other Perspectives

Accessibility	Can the system be used by people with disabilities?
Development Resource	Can the system be built within people, time and budget constraints?
Internationalisation	Is the system independent of language, country and culture?
Location	Will the system work, given its required geographical constraints?
Regulation	Does the system meet any required regulatory constraints?
Usability	Can people use the system effectively?

61

Talking Point: Perspectives

- Going back to your qualities/views matrix
- Which perspectives would help you to achieve your quality properties?
  - Are they in the primary or secondary sets?
  - Useful feedback for us if in secondary
- Where may you have conflicts between advice in different relevant perspectives?

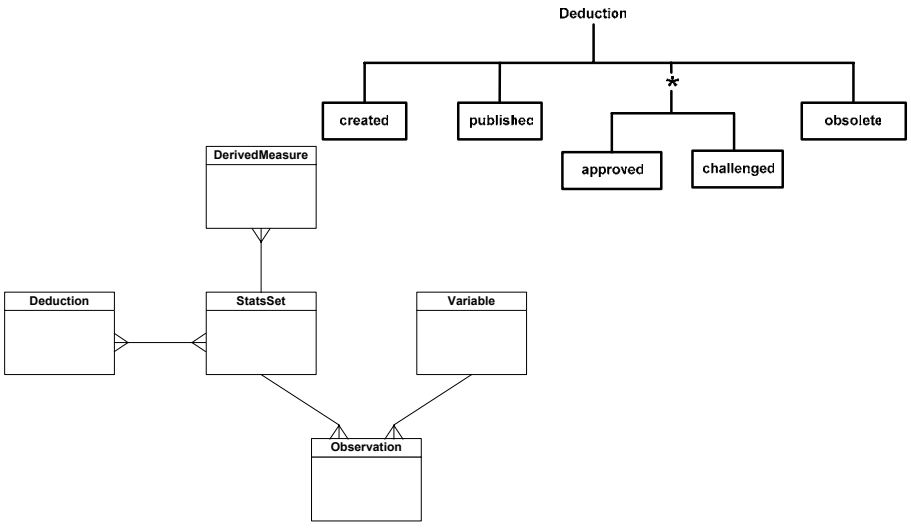
62

Example Application

- Simple example of viewpoints and perspectives
- Used throughout the tutorial materials
- Statistics storage and processing system
  - Data loaded into the database
  - Derived measures calculated automatically
  - Statisticians view and report on the data
  - Deductions recorded and reviewed manually

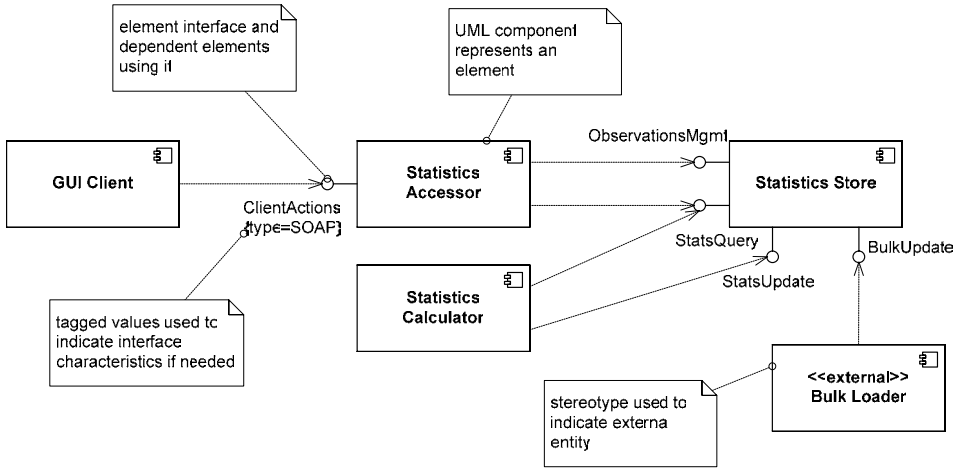
63

Information View



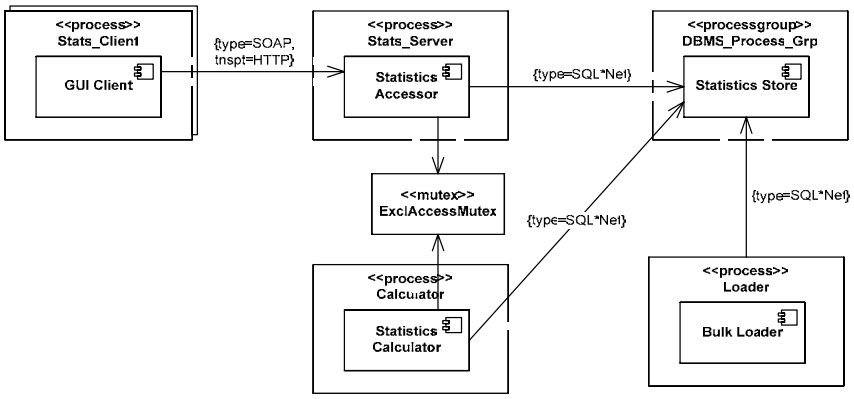
64

Functional View



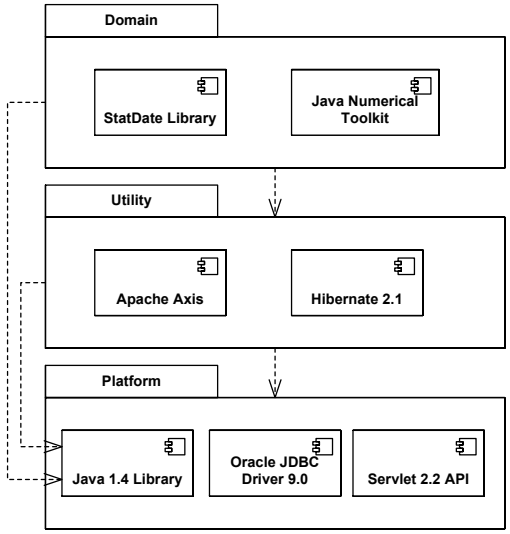
65

Concurrency View



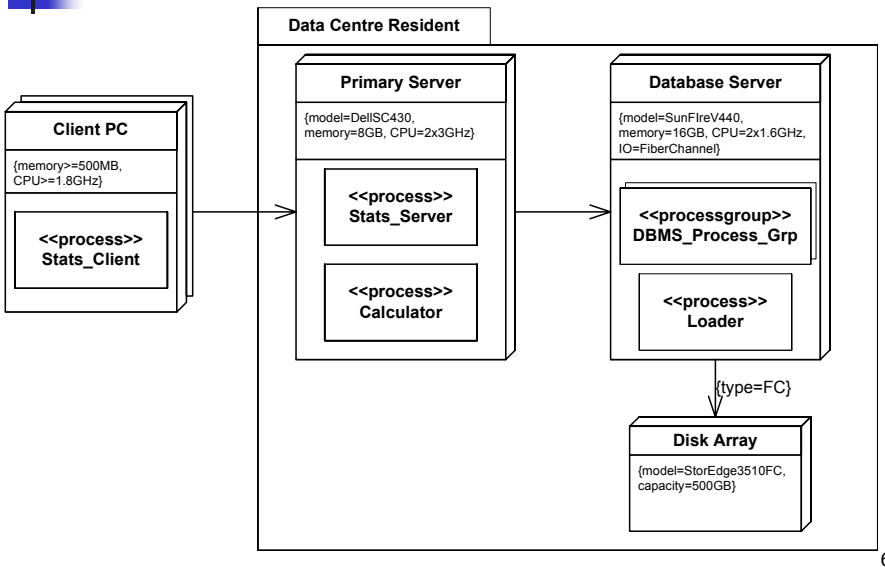
66

Development View



67

Deployment View



68

Deployment View (ii)

Client PC	▪ Windows XP SP1 ▪ Java JRE 1.4.2_06 or later ▪ Internet Explorer 6.0 SP1
Primary Server	▪ Windows 2003 server, w/sec patches ▪ Java SDK 1.4.2_06 or later ▪ Apache Tomcat 5.5.9 or later

Database Server	▪ Solaris 9.0 w/Aug05 patch cluster
	▪ Oracle 9.2.0.2 Std Edition
	  ▪ 10GB buffer cache, auto sized SGA
	  ▪ auto storage management, 2 table spaces
	▪ OEM 9.2.0.2 installed and working

69

Operational View

- Omitted from slides for space reasons.
- Would include:
  - Operational CM approach
  - Monitoring and Control
  - Operational Needs
  - Installation / migration / backout strategies

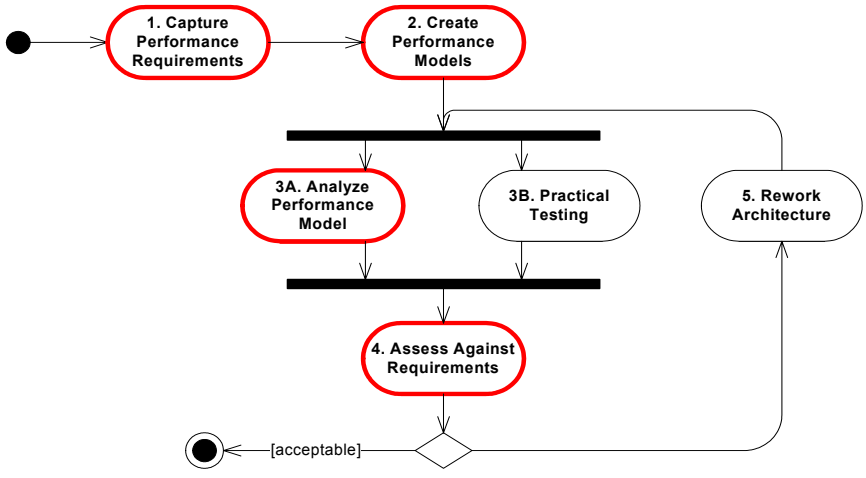
70

Example: Apply a Perspective

- Performance and Scalability
  - Capture P & S Requirements
  - Create Performance Models
  - Analyse Models
  - Perform Practical Testing
  - Assess Against Requirements
  - Rework Architecture (apply tactics)
- What affect will this have on our system?

71

Example: Apply a Perspective

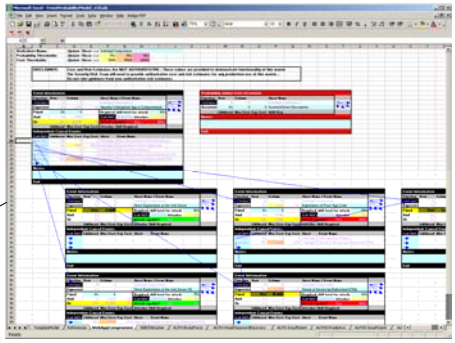


72

Example: Apply a Perspective

Measure	*Value*
Network A	100Mb
DB access (ro)	20ms
DB access (rw)	50ms
Single derived calc	1400ms
Client network	10Mb
Bulk load 100K	2500ms
Online load users	100
Memory per user	1MB

Calibration measures



Performance model

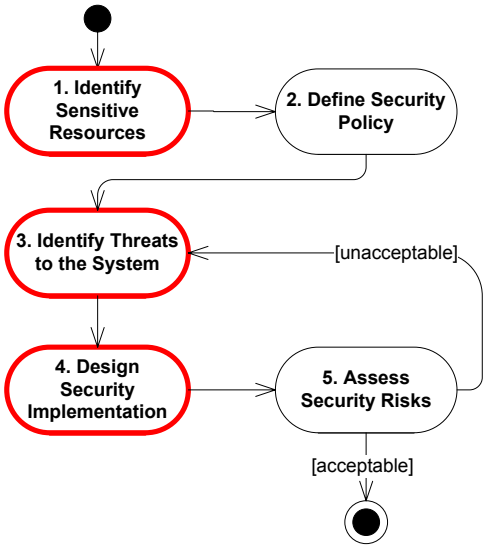
73

Example: Apply a Perspective

- Security
  - Identify Sensitive Resources
  - Define Security Policy
  - Identify Threats to the System
  - Design Security Implementation (apply tactics)
  - Assess Security Risks
- What affect will this have on our system?

74

Example: Apply a Perspective



75

Example: Apply a Perspective

- Sensitive Resources
  - The data in the database
- Security Threats
  - Operators stealing backups
  - Administrators querying data, seeing names
  - Bribing investigating officers
  - Internal attack on the database via network

76

Example: Apply a Perspective

- Security Countermeasures
  - Backups: encrypt data in the database
    - How about performance?
    - Does this make availability (DR) harder?
  - Seeing names: use codes instead of names, protect codes at higher security level
    - More development complexity
    - Possible performance impact

77

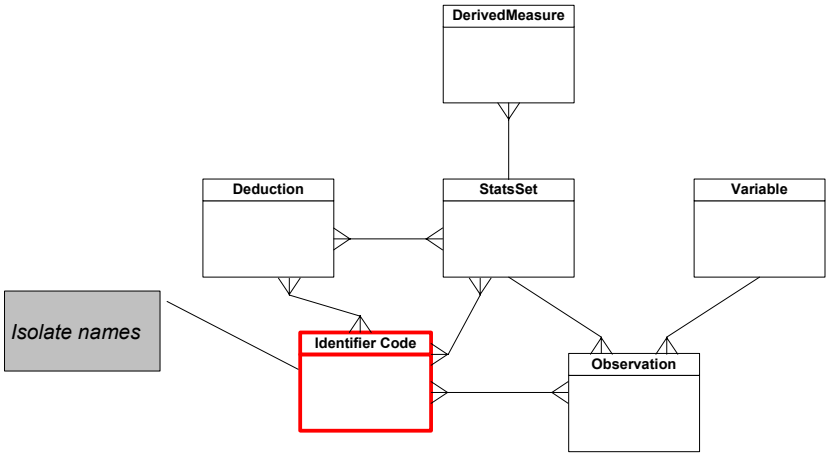
Example: Apply a Perspective

- Security Countermeasures
  - Network Attacks: firewalls, IDS
    - More cost
    - More deployment / administration complexity
    - Operational impact if IDS trips
  - Bribery: add audit trail for data access
    - Possible performance impact
    - More complexity
    - Protecting / using the audit trail

78

Example: Apply a Perspective

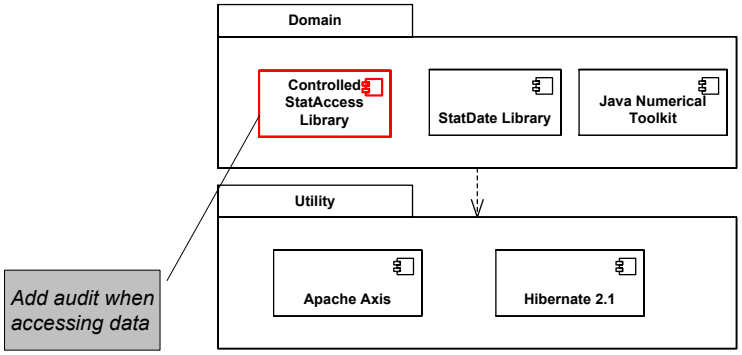
- Information View Impact



79

Example: Apply a Perspective

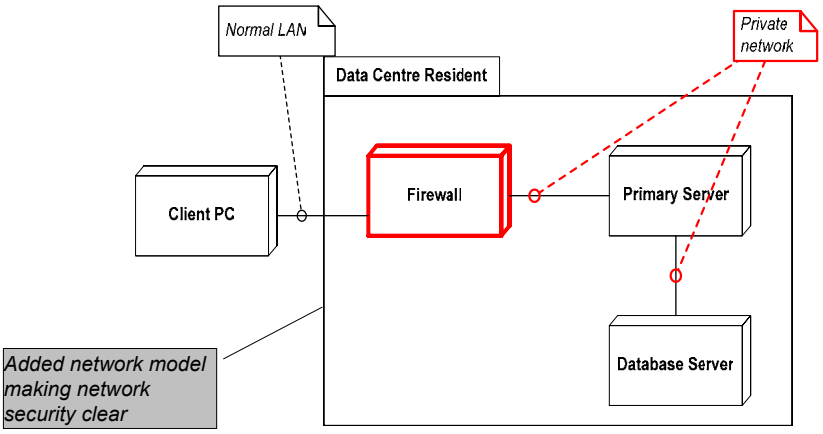
- Development View Impact



80

Example: Apply a Perspective

- Deployment View Impact



81

Example: Apply a Perspective

- Other Impact
  - Need IDS added to Development view
  - Need to capture impact on Operational view
  - Need to consider impact on availability
  - Need to re-work performance models to allow for database encryption, audit, …
- Note the need to change many views to address security needs

82

Use Viewpoints & Perspectives

- As a framework for organising work
- As a store of knowledge
  - Document proven practice
  - Help standardise language and approach
  - Help to standardise languages and approaches
- At different career stages
  - To mentor novice architects
  - To guide and focus experienced architects
  - To support expert architects

83

Use Viewpoints & Perspectives

- For Novice Architects
  - An introduction to each area of knowledge
  - A guide to what is important
  - A structure for the process
  - Definitions of standards and norms
  - Repository of proven practice and tactics
  - Pitfalls and solutions to avoid common errors
  - Checklist to ensure nothing is forgotten

84

Use Viewpoints & Perspectives

- For Working Architects
  - A reminder of what is important
  - A guide to new or rarely used areas of practice
  - Repository of proven practice and tactics
  - Pitfalls and solutions to avoid common errors
  - Checklist to ensure nothing is forgotten

85

Use Viewpoints & Perspectives

- For Expert Architects
  - A framework to allow knowledge sharing
  - An aid to tutoring and mentoring
  - Checklists to ensure nothing is forgotten

86

Summary

- Architecture Essential Difficulties
  - Multi-dimensional problem, no right answer
  - Stakeholder needs conflict
  - Complex mix of people and technology
  - Tradeoffs are inevitable
- Architecture Accidental Difficulties
  - Lack of standardisation (approach, artefacts)
  - Little sharing of knowledge and experience

87

Summary (ii)

- Viewpoints and Views
  - Views provide a convenient approach for effective architectural description
  - Viewpoints standardise views by defining their content
  - Viewpoints contain proven architectural knowledge for a particular domain
  - Viewpoints and views can address many accidental difficulties of software architecture

88

Summary (iii)

- Viewpoints for Information Systems
  - Functional
  - Information
  - Concurrency
  - Development
  - Deployment
  - Operational

89

Summary (iv)

- Perspectives
  - Viewpoints handle structure well, less convinced about quality properties
  - Perspectives provide similar guidance and knowledge sharing for quality properties
  - Perspectives suggest the design activities required to achieve a property
  - Perspectives provide proven practice, pitfalls, solutions and checklists to share experience
  - Applying perspectives modifies views

90

Summary (v)

- Perspectives for Information Systems
  - Availability and Resilience
  - Evolution
  - Performance and Scalability
  - Security
  - Others
    - Accessibility, Development Resource, Geographical Location, I18N, Regulation, Usability

91

Summary (iv)

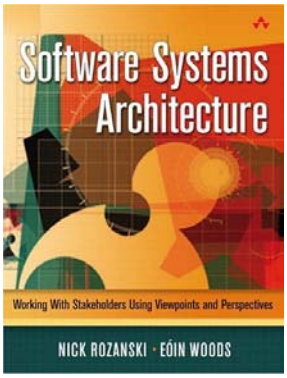
- Using Viewpoints and Perspectives
  - Novices
    - Overview, guides, focuses attention
    - Proven practice, pitfalls, solutions and checklists
  - Working Architects
    - Reminder of existing knowledge
    - Aid in new areas
  - Experts
    - Mentoring and communication vehicle
    - Reminders of hard won lessons

92

To Learn More

Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives

Nick Rozanski & Eoin Woods
Addison Wesley, 2005



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

93

Nick Rozanski
nick@rozanski.com
www.nick.rozanski.com

Eoin Woods
eoin@eoinwoods.info
www.eoinwoods.info

Comments and Questions?