

Architectural Design

Software Engineering
Tom Verhoeff

1

SW Architecture

- IEEE Std 1471
- Architecture vs. Architecture Description
- Place in development process

2

What is Architecture?

- the fundamental **organization** of a system
- embodied in its **components**,
- their **relationships** to each other and
- to the **environment**, and
- principles guiding its **design** and **evolution**

3

Why Architecture?

- Facilitates parallel construction
- Improves ability to make work plans
- Improves verifiability
 - Unit testing of separate components
 - Allows stepwise integration (no big bang)
- Most changes affect few components

4

Architecture Description

- Stakeholders
- Viewpoints
- Architectural views
- Inconsistencies among views
- Rationale

5

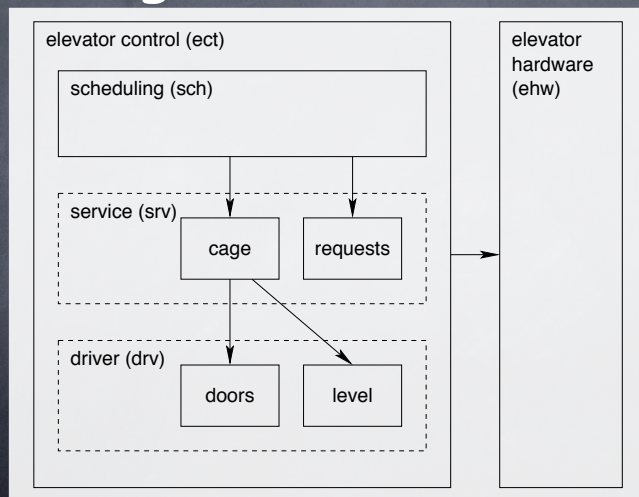
Kruchten's 4+1 Views

	Static	Dynamic
Abstract	Logical	Process
Concrete	Development	Deployment

+ Use case scenarios (traced through arch.)

6

Logical Structure



7

How to Evaluate

- Number and size of components
- Number of relations (less is better)
- Coupling: how components depend on others
- Cohesion: similar items in same component
- Complexity of interfaces
- Fan-in, fan-out

8

How to Design

- Top down, bottom up, yo-yo design
- KISS: Keep It Simple, Stupid
- Driven by extra-functional requirements
- Consider alternatives, and compare them

9

Standard Components

- Graphical User Interface
- Data Base Engine
- Container Classes
- Numerical Library
- Help Facility

10

Architectural Styles

- Layers
- Pipes and Filters
- Client-Server
- Blackboard

11

SW Design Description

- IEEE Std 1016-1998
- Recommended Practice for SDD
- SDD describes structure of SW solution
- Design entities & attributes
- Necessary, intrinsic attributes

12

Design Entity Attributes

- **Identification** (unique name, for reference)
- **Type** (nature of the component, e.g. library)
- **Purpose** (traced to requirements)
- **Function** (what it does)
- **Subordinates** (constituting components)

13

Design Entity Attributes (2)

- **Dependencies** (relation to other entities)
- **Interfaces** (provided to other entities)
- **Resources** (used from outside design)
- **Processing** (algorithmic details of function)
- **Data** (stored inside entity)

14

Cross-Cutting Issues

- Architectural Infrastructures
- Memory Management
- Initialization, Reset
- Exception Handling, Logging
- Internationalization
- Testing and Integration Facilities

15

Design Practices

- Stepwise Refinement
- Incremental Risk-Driven Design, Prototyping
- Structured Systems Analysis and Design
- Jackson Structured Programming
- Jackson System Development
- Designing with Objects
- Component-Based Design

16