

Achieving System Qualities Through Software Architecture

What is “software architecture?”
Role in determining system qualities
Architectural views

CIS 422/522 Fall 2012 1

Working Definition

“The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them.”

From *Software Architecture in Practice*, Bass, Clements, Kazman

Remember as: **Components, Interfaces, and Relations**

CIS 422/522 Fall 2012 2

Examples

- **An architecture comprises a set of**
 - Software components
 - Component interfaces
 - Relationships among them
- **Examples**

Structure	Components	Interfaces	Relationships
Calls Structure	Programs	Program interface and parameter declarations.	Invokes with parameters (A calls B)
Data Flow	Functional tasks	Data types or structures	Sends-data-to
Process	Sequential program (process, thread, task)	Scheduling and synchronization constraints	Runs-concurrently-with, excludes, precedes

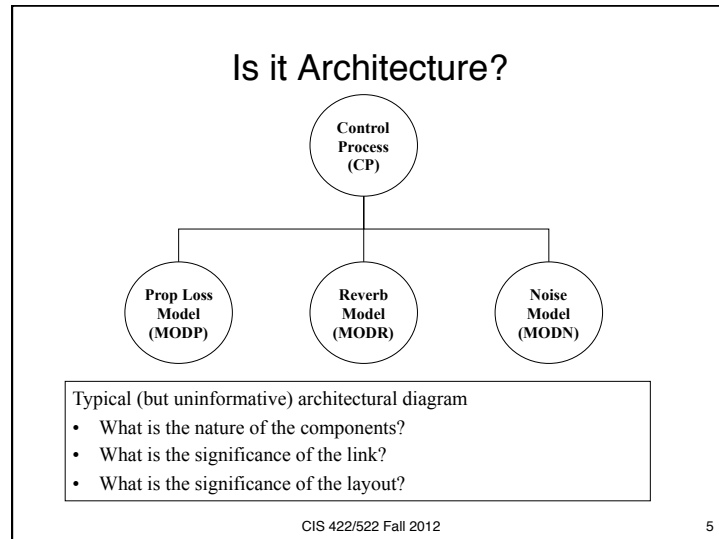
CIS 422/522 Fall 2012 3

Implications of the Definition

“The software architecture of a program or computing system is the structure or structures of the system, which comprise software components, the externally visible properties of those components, and the relationships among them.” - Bass, Clements, Kazman

- **Systems typically comprise more than one architecture**
 - There is more than one useful decomposition into components and relationships
 - Each addresses different system properties or design goals
- **It exists whether any thought goes into it or not!**
 - Decisions are necessarily made if only implicitly
 - Issue is who makes them and when
- **Many “architectural specifications” aren’t**

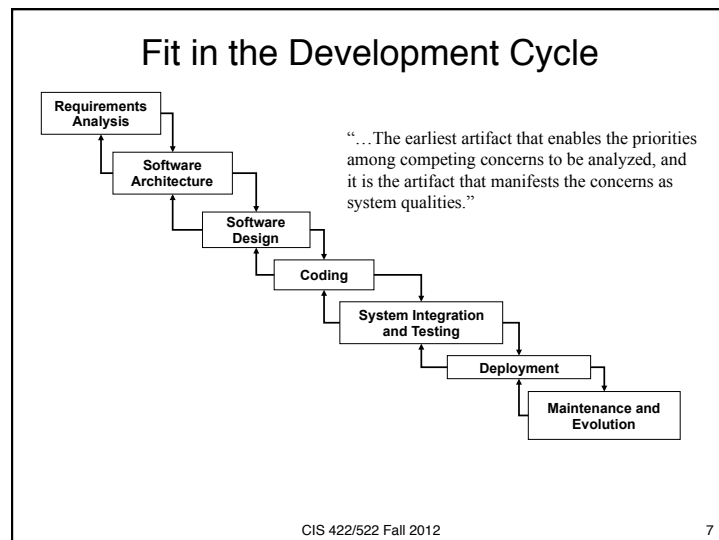
CIS 422/522 Fall 2012 4



The Role of Architecture

Which system or development characteristics are determined by architecture?
What is the source of requirements?

CIS 422/522 Fall 2012 6



Effects of Architectural Decisions

- What kinds of system and development properties are and are not affected by architecture?
- System run-time properties
 -
- System static properties
 -
- Production properties? (effects on project)
 -
- Business/Organizational properties?
- What is not affected?

CIS 422/522 Fall 2012 8

Effects of Architectural Decisions

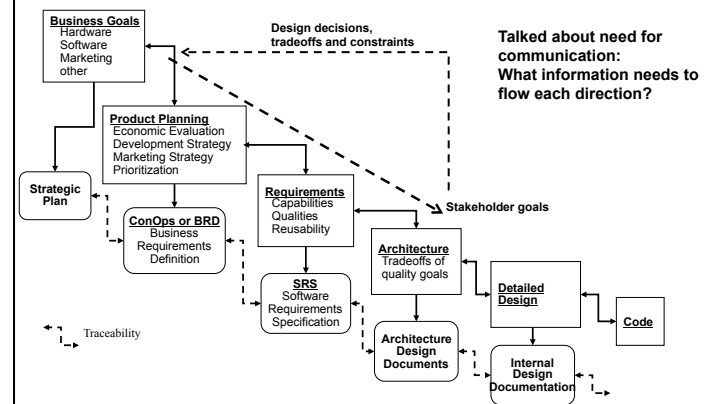
- What kinds of system and development properties are and are not affected by architecture?
- System run-time properties
 - Performance, Security, Availability, Usability
- System static properties
 - Modifiability, Portability, Reusability, Testability
- Production properties? (effects on project)
 - Work Breakdown Structure, Scheduling, time to market
- Business/Organizational properties?
 - Lifespan, Versioning, Interoperability

Importance to Stakeholders

- Which stakeholders have a vested interest in the architectural design?
 - Management, marketing, end users
 - Maintenance organization, IV&V, Customers
 - Regulatory agencies (e.g., FAA)
- There are many interested parties (stakeholders) with many diverse and often conflicting interests
- Important because their interests defy mutual satisfaction
 - There are inherently tradeoffs in most architectural choices
 - E.g. Performance vs. security, initial cost vs. maintainability
- Making successful tradeoffs requires understanding the nature, source and priority of these constraints

Role of Architecture in Disciplined Development

Product Development Cycle and Architecture



SW Engineering of Software Architecture

- What are we trying to gain/maintain control of in the Architectural Design phase?
 - Profoundly effect system and business qualities
 - Requires making tradeoffs
- Control implies achieving system qualities by choice not chance
 - Understanding what the tradeoffs are
 - Understanding the consequences of each choice
 - Making appropriate choices at appropriate times

CIS 422/522 Fall 2012

13

Implications for the Development Process

Implies need to address architectural concerns throughout the development process:

- Understanding the “business case” for the system
- Understanding the quality requirements
- Designing the architecture to meet quality goals
- Representing and communicating the architecture
- Analyzing or evaluating the architecture
- Implementing the system based on the architecture
- Ensuring the implementation conforms to the architecture

CIS 422/522 Fall 2012

14

Related Design Questions

- Create business case for the system
 - What is the “business” rationale or goal?
- Understanding the requirements
 - What are the design goals?
- Creating or selecting the architecture
 - What are appropriate components and relations?
 - What are the decomposition principles?
- Representing and communicating the architecture
 - How are the components and relations represented?
- Analyzing or evaluating the architecture
 - How do we decide if the architecture is any good?

CIS 422/522 Fall 2012

15

Quality Requirements

CIS 422/522 Fall 2012

16

Terminology

- Avoid “functional” and non-functional” classification
- Behavioral Requirements – any information necessary to determine if the run-time behavior of a given implementation constitutes an acceptable system
 - All quantitative constraints on the system's run-time behavior
 - Other objective measures (safety, performance, fault-tolerance)
 - In theory all can be validated by observing the running system and measuring the results
- Developmental Quality Attributes - any constraints on the system's static construction
 - Maintainability, reusability, ease of change (mutability)
 - Measures of these qualities are necessarily relativistic (i.e., in comparison to something else)

CIS 422/522 Fall 2012

17

Behavioral vs. Developmental

Behavioral (observable)

- Performance
- Security
- Availability
- Reliability
- Usability

Properties resulting from the properties of components, connectors and interfaces that exist at run time.

Developmental Qualities

- Modifiability(ease of change)
- Portability
- Reusability
- Ease of integration
- Understandability
- Provide independent work assignments

Properties resulting from the properties components, connectors and interfaces that exist at design time *whether or not they have any distinct run-time manifestation.*

CIS 422/522 Fall 2012

18

Specifying Quality Requirements

- Traditionally done in natural language, usually badly
 - “The system shall be easy to maintain.”
 - “The system shall maximize the number of user transactions.”
- What’s wrong with these?

CIS 422/522 Fall 2012

19

Specifying Quality Requirements

- When using natural language, write *objectively verifiable* requirements when possible
 - Maintainability: “The following kinds of requirement changes will require changes in no more than one module of the system...”
 - Performance:
 - “System output X has a deadline of 5 ms from the input event.”
 - “System output Y must be updated at a frequency of no less than 20 ms.”
- Consider: what do we really mean by “maintainable?”

CIS 422/522 Fall 2012

20

Example Timing Requirements

5.2. TIMING REQUIREMENTS FOR DEMAND FUNCTIONS

For all the demand functions, the rate of demand is so low that it will not constitute a significant CPU-load.

For the starred entries, the desired maximum delay is not known; the entry is the maximum delay in the current OFP, which we will use as an approximation. In one case, both the current and desired values are given. The current value would be good enough to satisfy requirements, but the desired rate would be preferred.

Function name	Maximum delay to completion
IMS:	
Switch AUTOCAL light on/off	*200 ms
Switch computer control on/off	*200 ms
Issue computer failure	not significant
Change scale factor	*200 ms
Switch X slewing on/off	*200 ms
Switch Y slewing on/off	*200 ms
Switch Z slewing on/off	*200 ms
Change latitude-greater-than-70-degrees	*200 ms
Switch INA light on/off	*200 ms
FLR:	
Enable radar cursor	200 ms
Slave or release slave	40 ms

Architectural Views

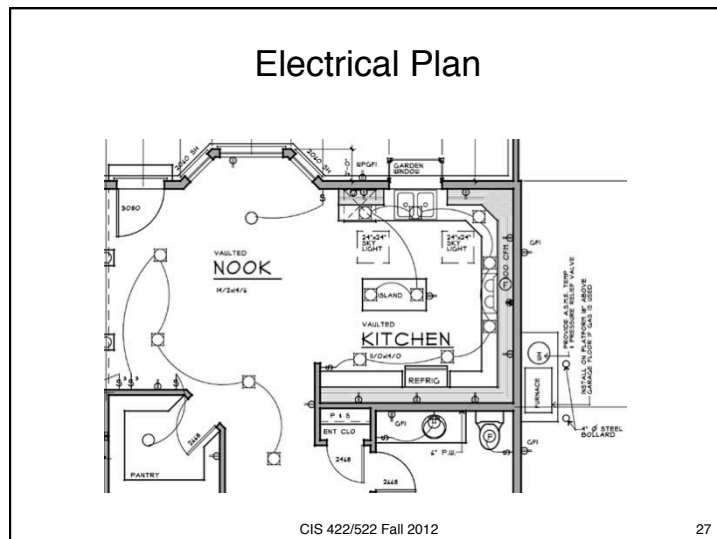
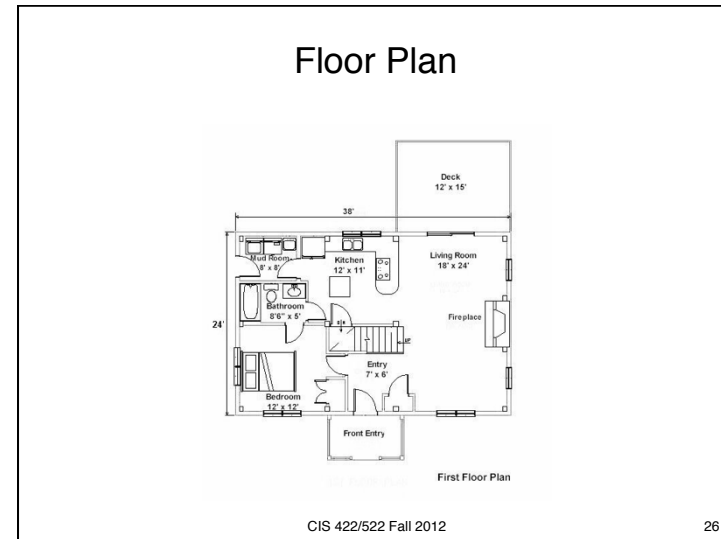
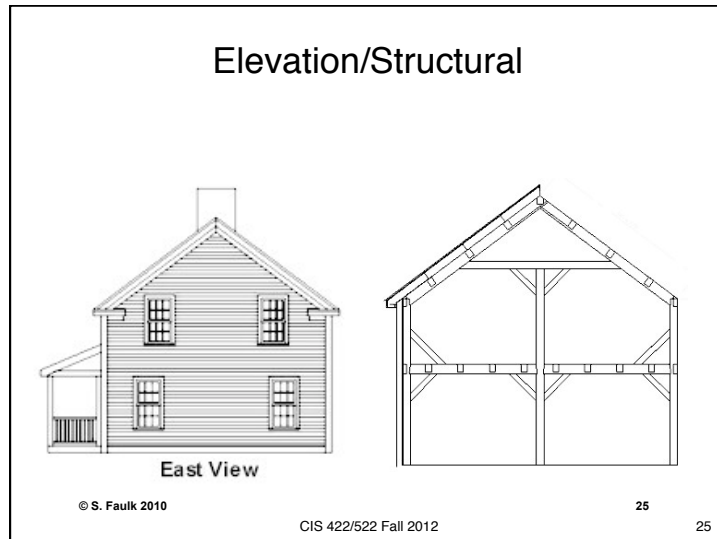
Architectural Development Process

- Understand the goals for the system (e.g., business case or mission)
- Understand/communicate the quality requirements
- Design architecture(s) that satisfy quality requirements
 - Choose appropriate architectural structures
 - Design structures to satisfy qualities
 - Document to communicate design decisions
- Evaluate the architecture
- Implement the system based on the architecture

Which structures should we use?

Structure	Components	Interfaces	Relationships
Calls Structure	Programs (methods, services)	Program interface and parameter declarations	Invokes with parameters (A calls B)
Data Flow	Functional tasks	Data types or structures	Sends-data-to
Process	Sequential program (process, thread, task)	Scheduling and synchronization constraints	Runs-concurrently-with, excludes, precedes

- Choice of structure depends the *specific design goals*
- Compare to architectural blueprints
 - Different blueprint for load-bearing structures, electrical, mechanical, plumbing



Models/Views

- Each is a view of the same house
- Different views answer different kinds of questions
 - How many electrical outlets are available in the kitchen?
 - What happens if we put a window here?
- Designing for particular software qualities also requires the right architectural model or “view”
 - Any model can present only a subset of system structures and properties
 - Different models allows us to answer different kinds of questions about system properties
 - Need a model that makes the properties of interest and the consequences of design choices visible to the designer, e.g.
 - Process structure for run-time property like performance
 - Module structure for development property like maintainability

CIS 422/522 Fall 2012

28

For Your Projects

Behavioral Qualities

- Performance
- Security
- Availability
- Reliability
- Security
- Usability

Developmental Qualities

- Modifiability(ease of change)
- Portability
- Reusability
- Ease of integration
- Understandability
- Independent work assignments
- Subsetability

Which qualities are of interest for your projects?
Which structures should you use?

Examples of Key Architectural Structures

- **Module Structure**
 - Decomposition of the system into work assignments or information hiding modules
 - Most influential design time structure
 - Modifiability, work assignments, maintainability, reusability, understandability, etc.
- **Uses Structure**
 - Determine which modules may use one another's services
 - Determines subsetability, ease of integration

Summary

- Earliest set of design decisions – hence, most influential and hardest to change
- Determines a wide range of critical system, production, and business properties
- A product of tradeoffs between conflicting demands by different stakeholders
- Requirements come from product/business goals and subsequently affect them
- Realized at design time in different views
- Next look at specific views and their application

Questions?

Assignments

- Set up Instructor Meeting
 - Review Project 1
 - *Requirements for Project 2*