

LECTURE-9

Basic Concepts of Software Architecture

What Is Architecture?

- Software architecture encompasses the set of significant decisions about the organization of a software system
 - Selection of the structural elements and their interfaces by which a system is composed
 - Behavior as specified in collaborations among those elements
 - Composition of these structural and behavioral elements into larger subsystems
 - Architectural style that guides this organization

*Grady Booch, Philippe Kruchten, Rich Reitman, Kurt Bittner; Rational
(derived from Mary Shaw)*

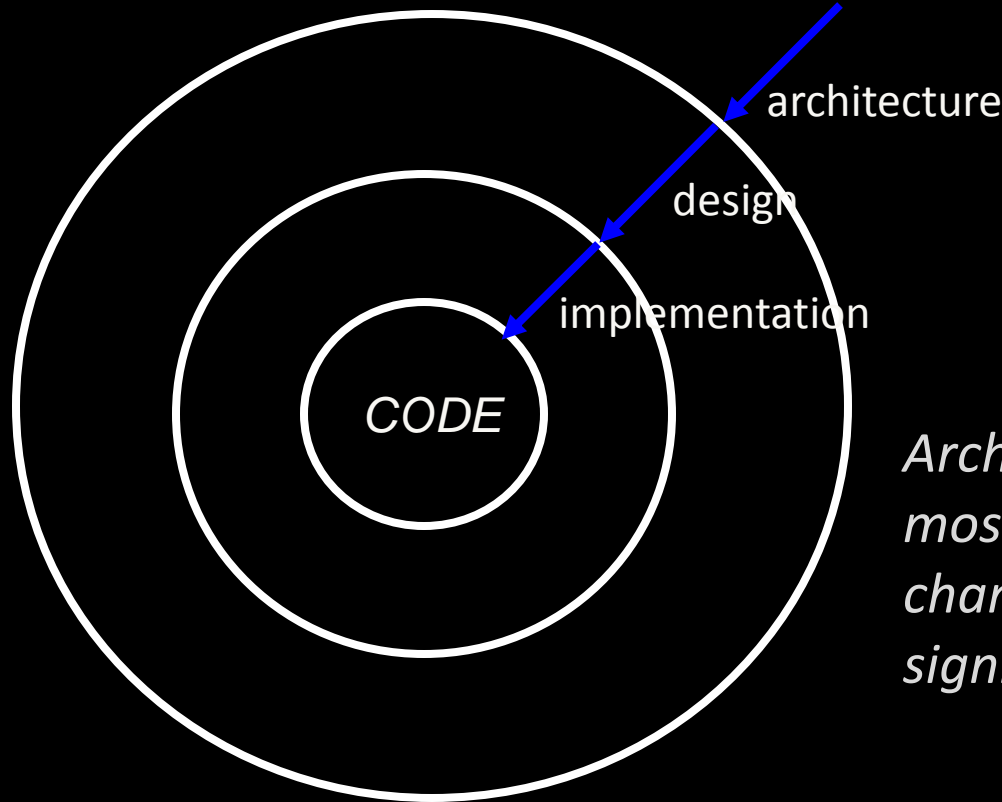
What is Software Architecture ?

- Software architecture also involves
 - usage
 - functionality
 - performance
 - resilience
 - reuse
 - Comprehensibility
 - economic and technology constraints and tradeoffs
 - aesthetic concerns

Architecture Constrains Design and

Implementation

- Architecture involves a set of strategic design decisions, rules or patterns that constrain design and construction

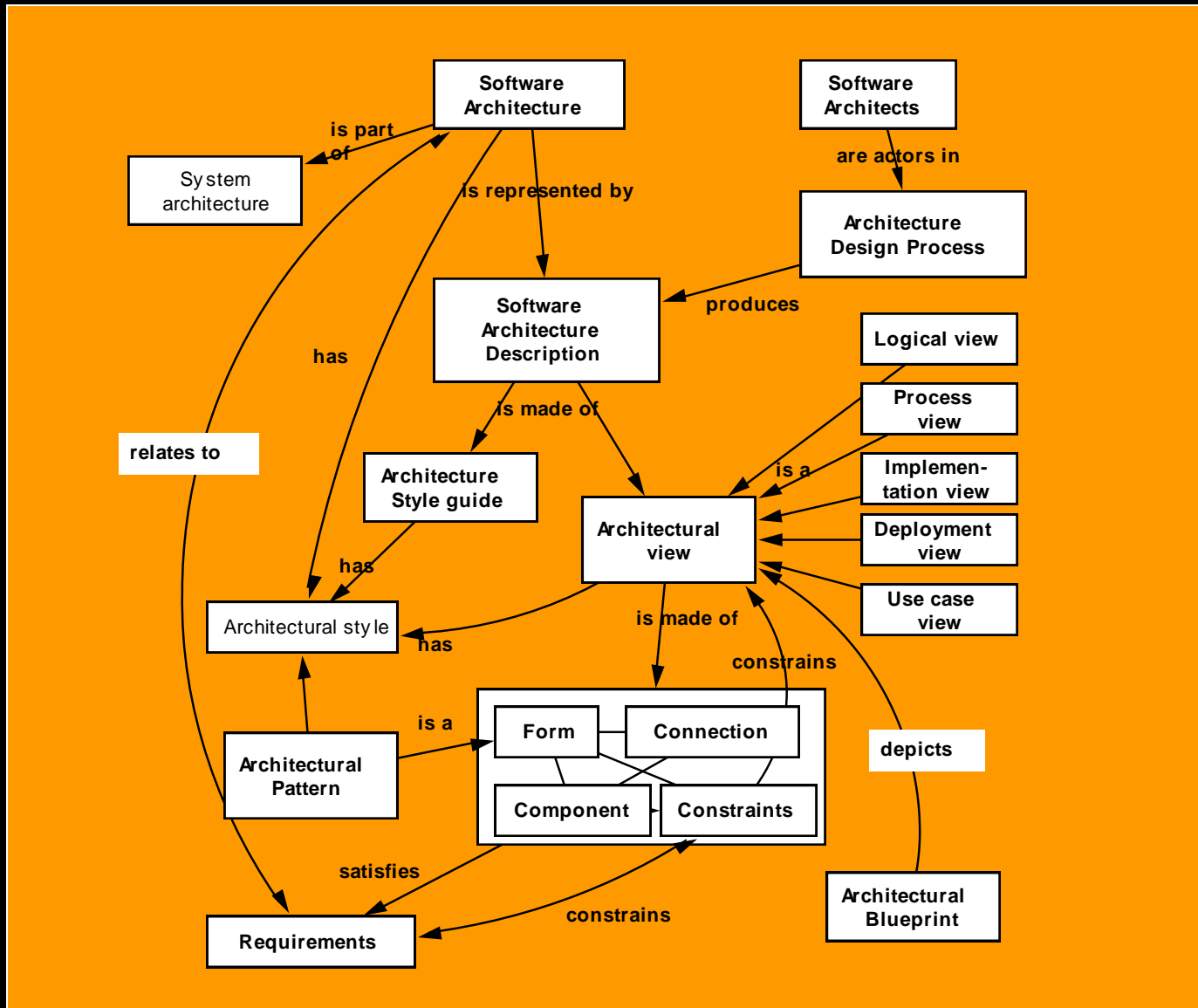


Architecture decisions are the most fundamental decisions and changing them will have significant ripple effects.

The common theme in all software architecture definitions

- Regardless of the definition (and there are many) the common theme in all software architecture definitions is that it has to do with the large scale —
 - the Big Ideas in the forces,
 - organization,
 - styles,
 - patterns, responsibilities,
 - collaborations,
 - connections,
 - and motivations of a system (or a system of systems),
 - and major subsystems.

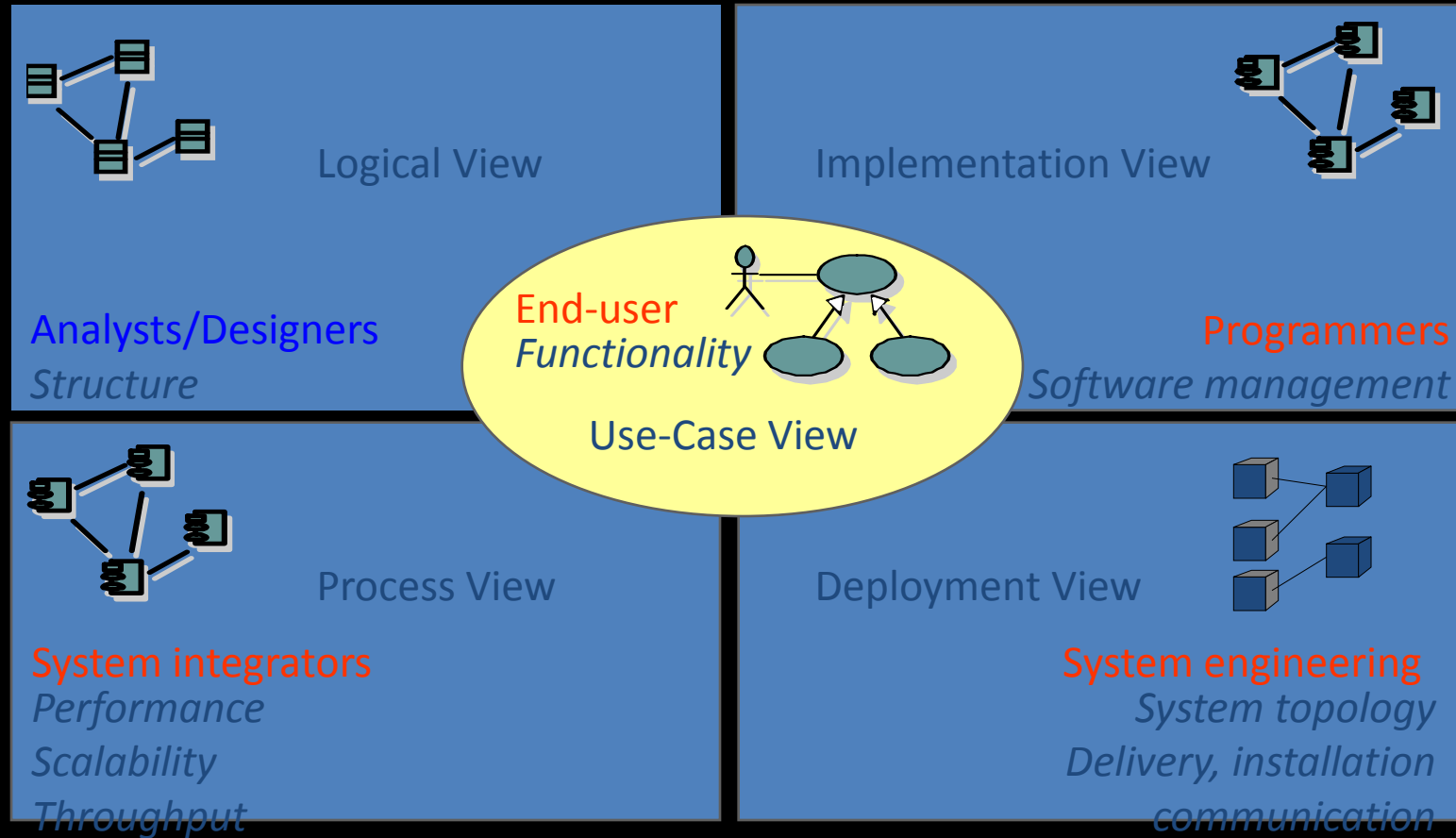
Architecture metamodel (Booch)



Architectural view

- An architectural view is a simplified description (an abstraction) of a system from a particular perspective or vantage point, covering particular concerns, and omitting entities that are not relevant to this perspective

Software Architecture: The 4+1 View Model



How many views?

- Simplified models to fit the context
- Not all systems require all views:
 - Single processor: drop deployment view
 - Single process: drop process view
 - Very Small program: drop implementation view
- Adding views:
 - Data view, security view

Architectural Style

- Non software examples
 - http://www.bc.edu/bc_org/avp/cas/fnart/fa267/amstyles.html
- An architecture style defines a family of systems in terms of a pattern of structural organization.
- An architectural style defines
 - a vocabulary of components and connector types
 - a set of constraints on how they can be combined
 - one or more semantic models that specify how a system's overall properties can be determined from the properties of its parts

Architecturally significant elements

- Not all design is architecture
- Main “business” classes
- Important mechanisms
- Processors and processes
- Layers and subsystems
- Architectural views = slices through models

Architectural Focus

- Although the views above could represent the whole design of a system, the architecture concerns itself only with some specific aspects:
 - *The **structure** of the model* - the organizational patterns, for example, layering.
 - *The **essential elements*** - critical use cases, main classes, common mechanisms, and so on, as opposed to all the elements present in the model.
 - A few key ***scenarios*** showing the main control flows throughout the system.
 - The ***services***, to capture modularity, optional features, product-line aspects.

Characteristics of a Good Architecture

- Resilient
- Simple
- Approachable
- Clear separation of concerns
- Balanced distribution of responsibilities
- Balances economic and technology constraints