EA01

Understanding Architectural Assets

Peter Eeles
Executive IT Architect, IBM
peter.eeles@uk.ibm.com

IBM Rational Software Development Conference 2008

WHERE TEAMS ARE R-HEROES

IBM

Rational. software

Agenda

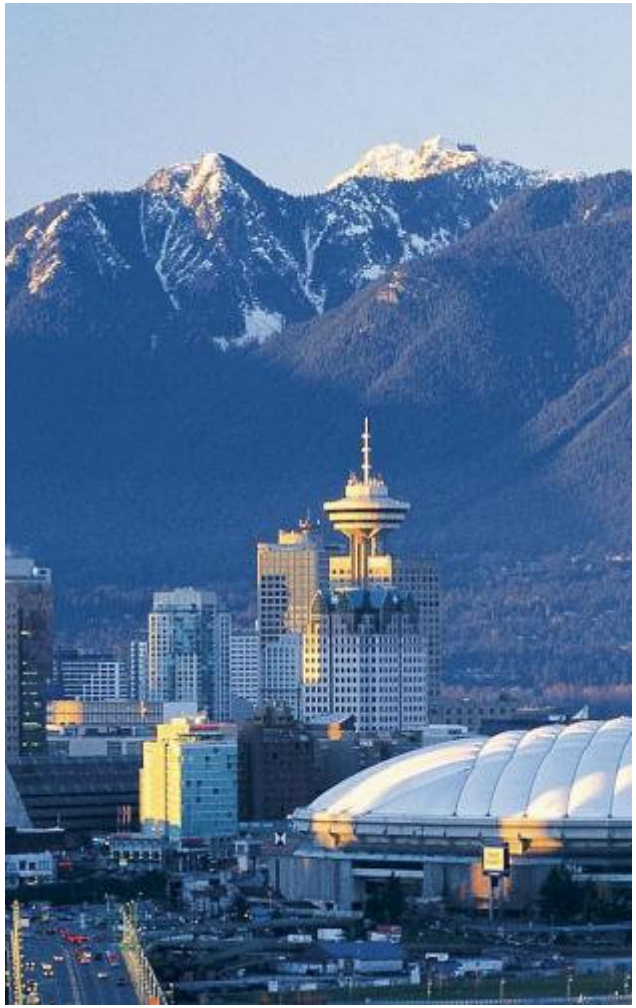
- Introduction
  - Sources of architecture
  - Types of architectural asset
  - Characterizing architectural assets
  - Automating asset reuse
  - Conclusion

Introduction

- “The life of a software architect is a long and rapid succession of suboptimal design decisions taken partly in the dark” - Philippe Kruchten

Inputs into this presentation

- Working IEEE/IFIP Conference on Software Architecture (WICSA) 2008
  - 18 – 22 February 2008, Vancouver, BC, Canada
  - Working session: Architectural Knowledge
- IBM Asset Architecture Board
- Reusable Asset Specification
- Rational Asset Manager
- RUP for Asset-based Development

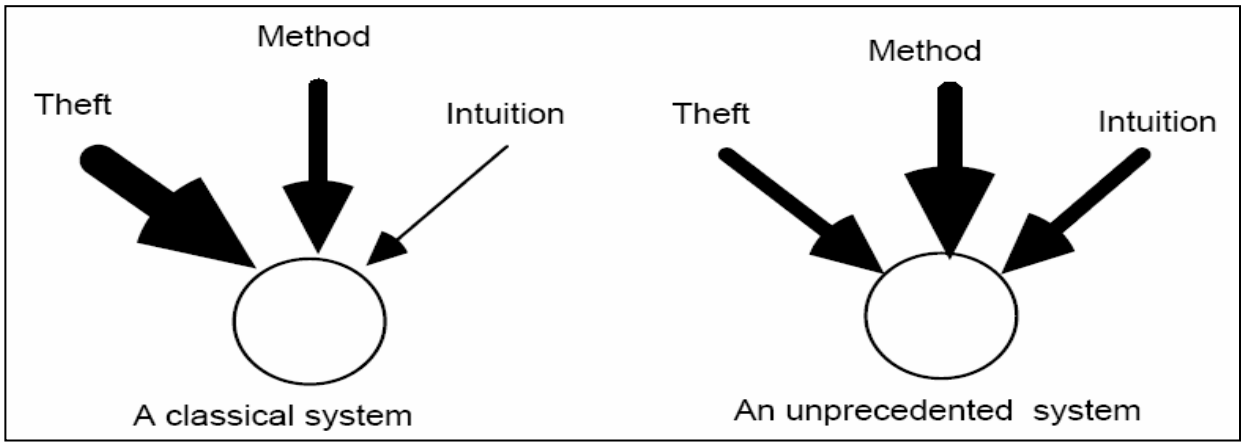


Agenda

- Introduction
- Sources of architecture
- Types of architectural asset
- Characterizing architectural assets
- Automating asset reuse
- Conclusion

Sources of Architecture

- Theft
  - From a previous system or from technical literature
- Method
  - An approach to deriving the architecture from the requirements
- Intuition
  - The experience of the architect

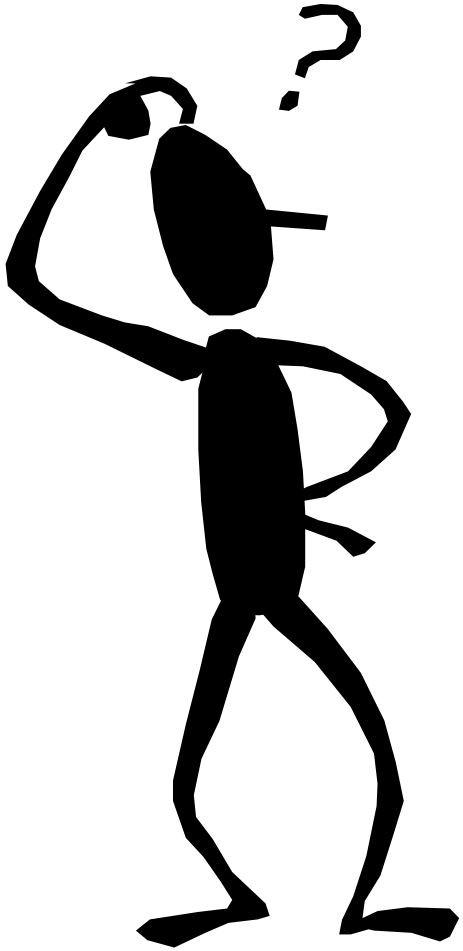


From "*Mommy, Where Do Software Architectures Come From?",* Philippe Kruchten
1st International Workshop on Architectures for Software Systems, Seattle, 1995

Agenda

- Introduction
- Sources of architecture
- Types of architectural asset
- Characterizing architectural assets
- Automating asset reuse
- Conclusion

What Types of Architectural Asset are there?



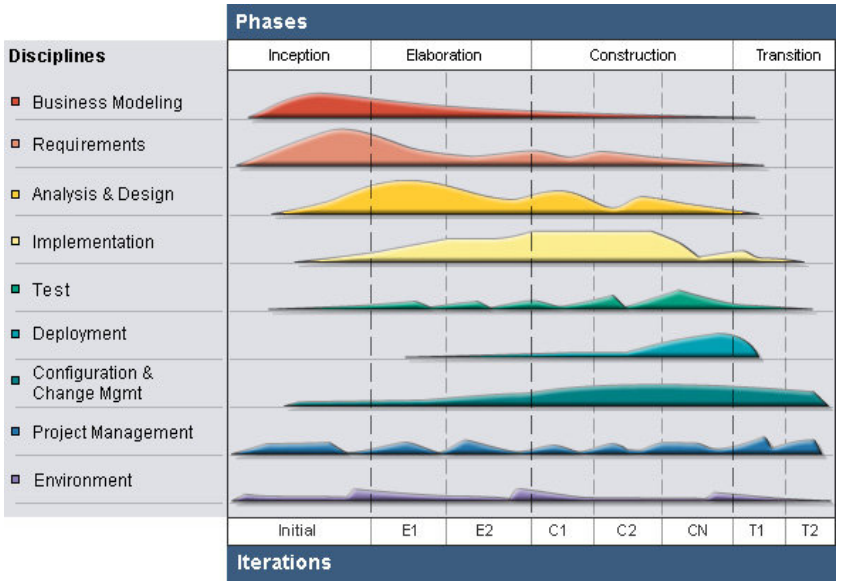
- Reference Architecture
- Design Pattern
- Legacy Application
- Architectural Mechanism
- Pattern Language
- Packaged Application
- Development Method
- Reference Model
- Architectural Decision
- Programming Pattern
- Pattern
- Component Library
- Component
- Architectural Pattern
- Architectural Style
- Application Framework

Development Method

- Best practices
- Guidance (techniques)
- Work product templates (e.g. architecture description template)
- Work product examples

IBM Global Services Method

Rational Unified Process



Pattern

- *[A pattern is] a common solution to a common problem in a given context.* [UML User Guide]
- Pattern types
  - Architectural Patterns
    - Distribution patterns, Security Patterns, …
  - Design Patterns
  - Programming Patterns
  - Requirements Patterns
  - Testing Patterns
  - Project Management Patterns
  - Process Patterns
  - Organizational Patterns
  - …

Architectural Pattern

- *An architectural pattern expresses a fundamental structural organization schema for software systems. It provides a set of predefined subsystems, specifies their responsibilities, and includes rules and guidelines for organizing the relationships between them.* [Buschmann]
- Example:

Pattern
Layers

Context
A system that requires decomposition

Problem
High-level elements rely on lower-level elements and the following forces must be balanced:

- Interfaces should be stable
- Parts of the system should be exchangeable
- Source code changes should not ripple through the system

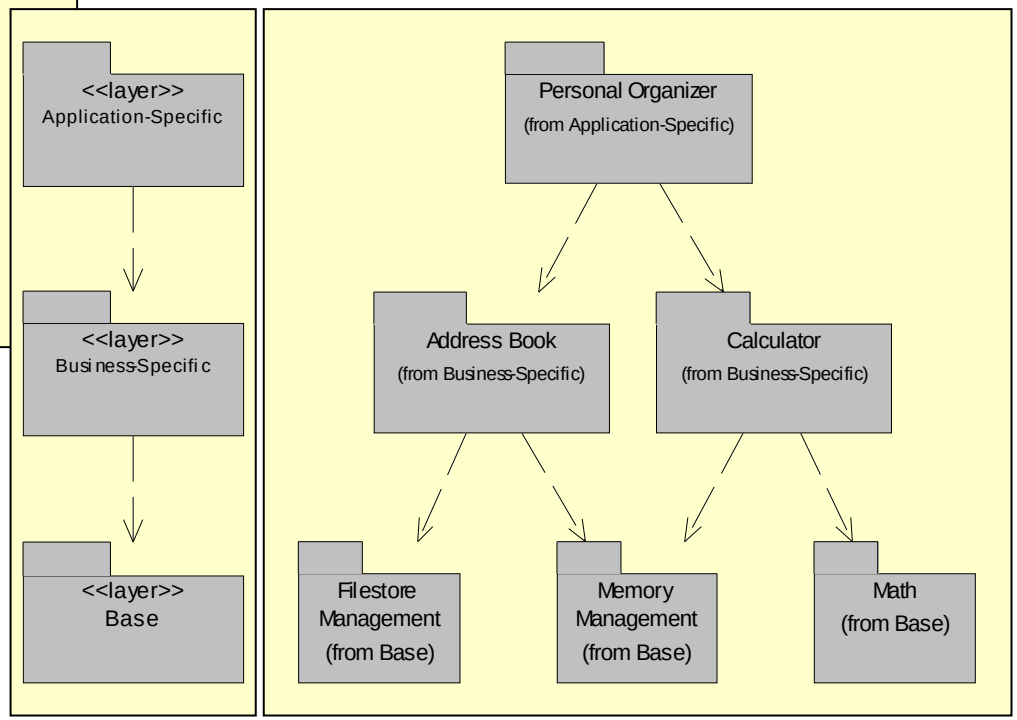
Solution
Structure the system into layers

Architectural Pattern - Layers

ISO OSI 7-Layer Model

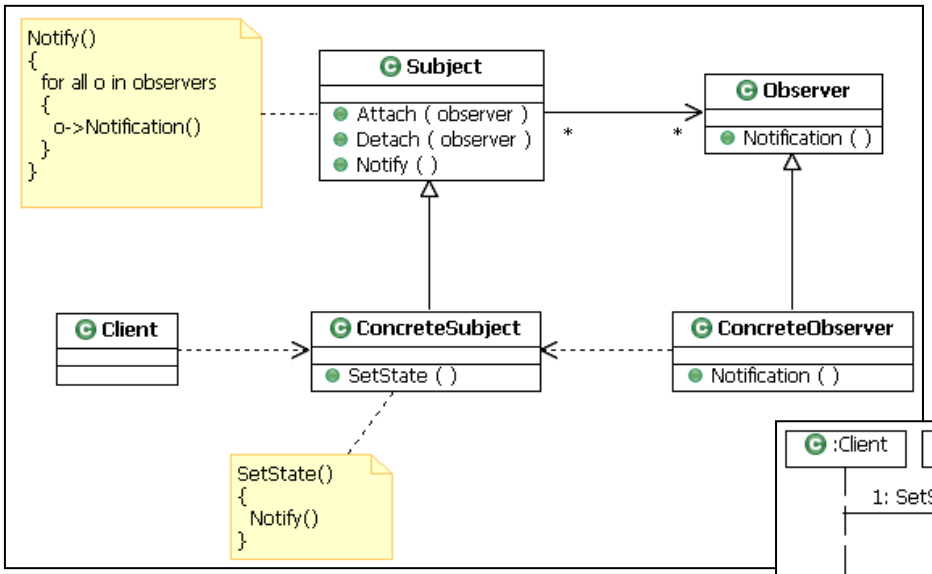
Layer 7	Application	Provides application facilities
Layer 6	Presentation	Structures information as required
Layer 5	Session	Manages the connection
Layer 4	Transport	Creates packets of data
Layer 3	Network	Routes packets of data
Layer 2	Data Link	Detects and corrects errors
Layer 1	Physical	Transmits bits

Personal Organizer

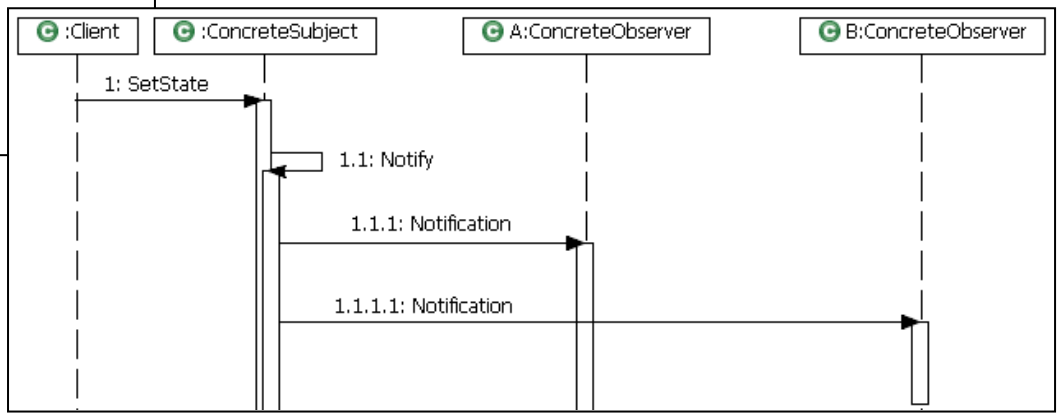


Design Pattern

- *A design pattern provides a scheme for refining the subsystems or components of a software system, or the relationships between them. It describes a commonly-recurring structure of communicating components that solves a general design problem within a particular context.* [Gamma]



Observer Pattern



Programming Pattern

- *An idiom is a low-level pattern specific to a programming language. An idiom describes how to implement particular aspects of components or the relationships between them using the features of the given language.* [Buschmann]

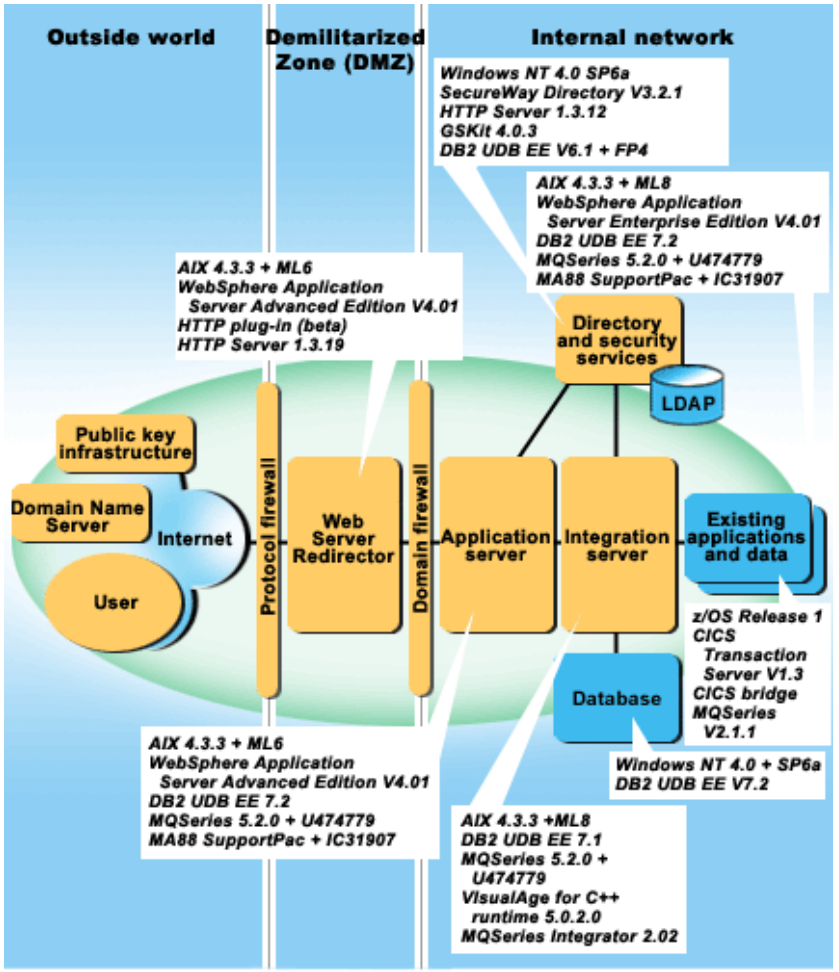
```
// Swap the values of 2 variables
temp = a;
a = b;
b = temp;
```

Architectural Style

- *[An architectural style] defines a family of systems in terms of a pattern of structural organization. More specifically, an architectural style defines a vocabulary of components and connector types, and a set of constraints on how they can be combined. [Shaw]*
- Client-server
  - Supports the physical separation of client-side processing (such as a browser) and server-side processing (such as an application server that accesses a database)
- Event-based
  - Promotes a publish-subscribe way of working, applied strategically across large areas of the architecture
- Pipes-and-filters
  - A series of filters that provide data transformation, and pipes that connect the filters. Examples include compilers, signal processing, Straight Through Processing (STP) and trading of electricity, oil and gas

Pattern Language

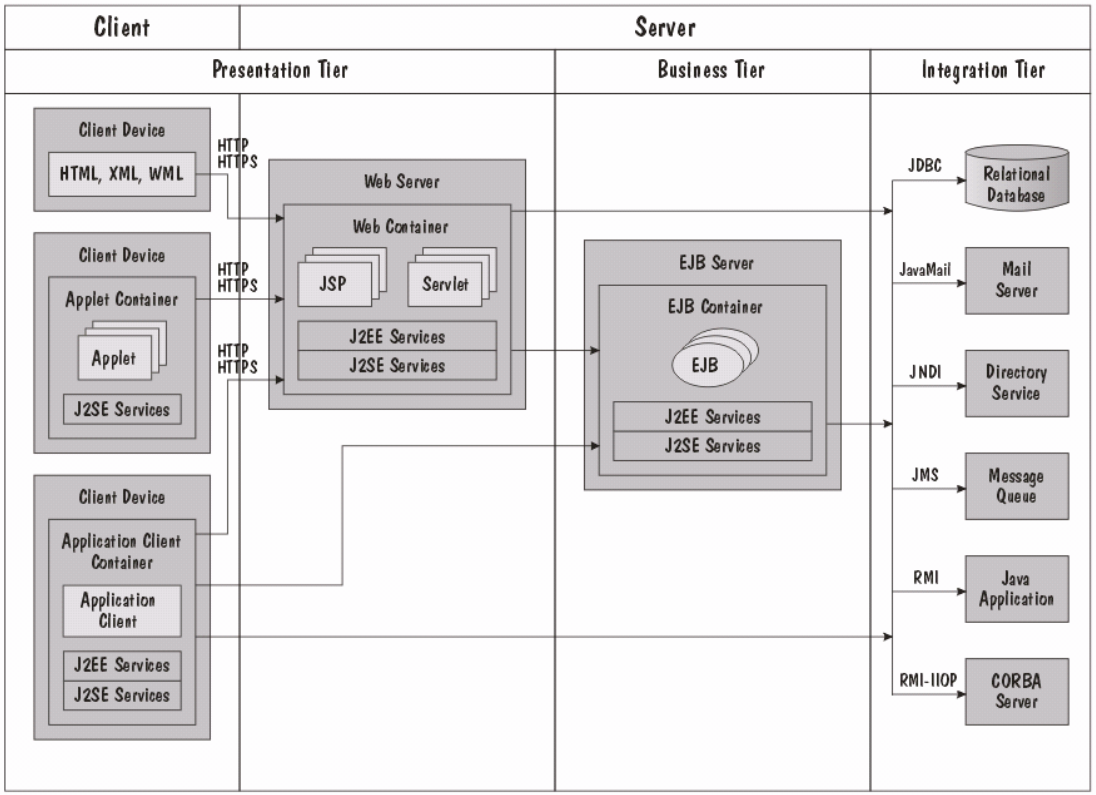
- *A pattern language defines a collection of patterns and the rules to combine them. Pattern languages are often used to describe a family of systems*
- IBM Patterns for e-Business
  - A set of architectural patterns that describe various web-based applications
  - Includes a pattern selection process that drives:
    - Selection of a business, integration or composite pattern
    - Selection of application patterns
    - Selection of runtime patterns
    - Identification of product mappings



*See http://www.ibm.com/developerworks/patterns

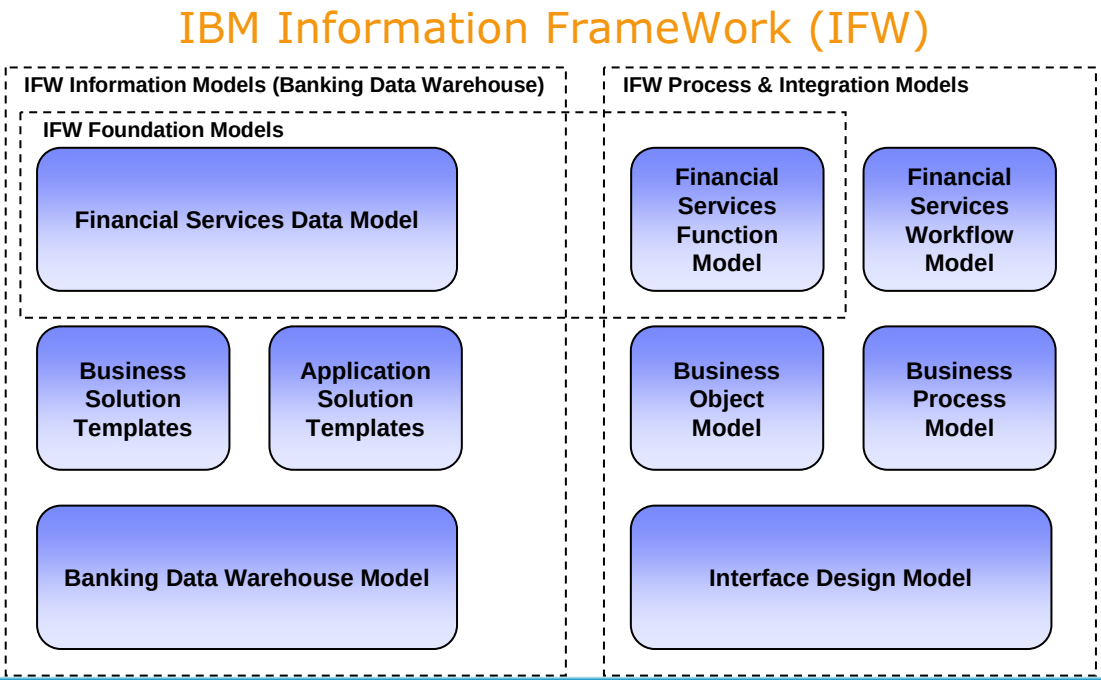
Reference Architecture

- *A reference architecture is an architecture representation of a particular domain of interest. It typically includes many different architectural patterns, applied in different areas of its structure*
- Examples include J2EE and .NET



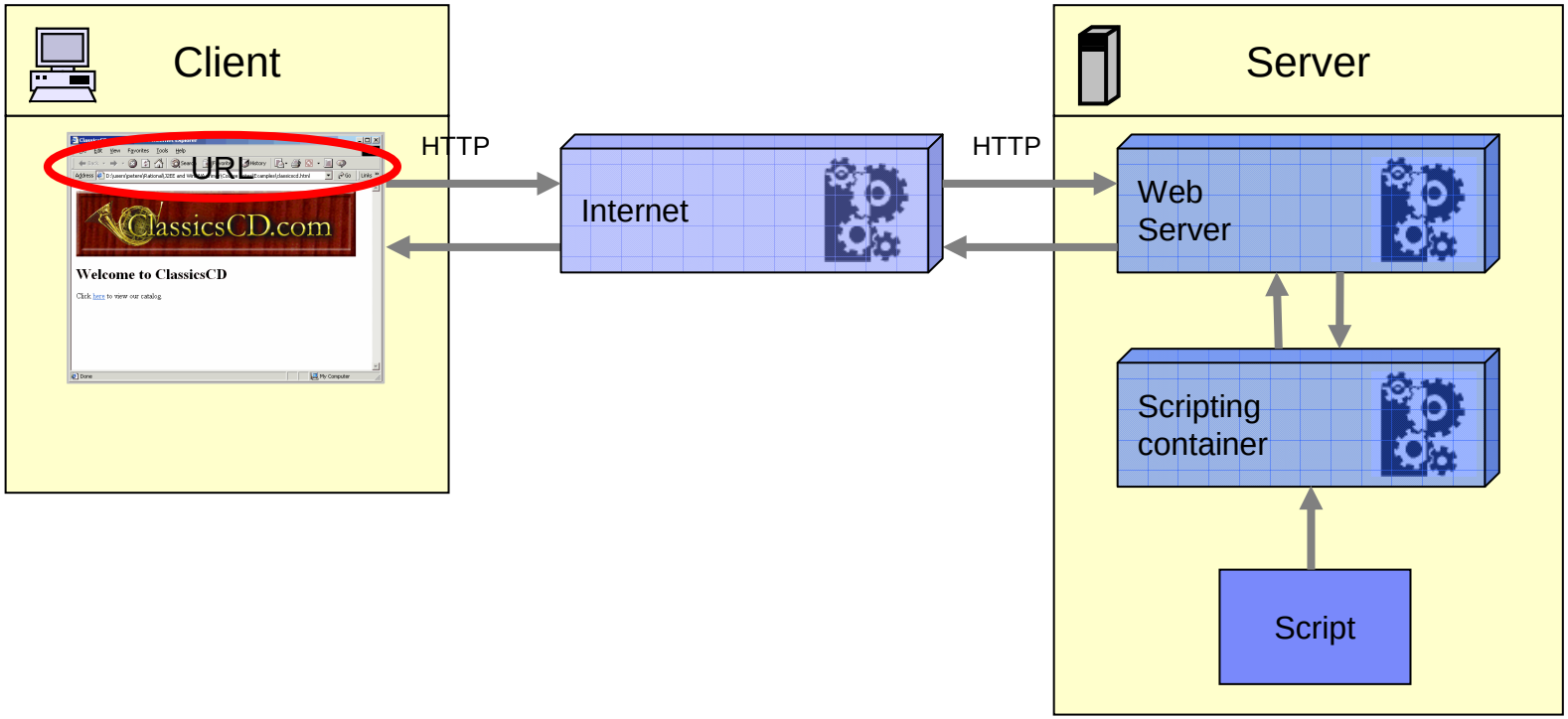
Reference Model

- *A reference model is an abstract representation of entities, their relationships and behavior, in a given domain of interest, and which typically forms the conceptual basis for the development of more concrete elements*
- Examples include a business model, an information model and a glossary of terms



Application Framework

- *An application framework represents the partial implementation of a specific area of an application*
- Most widely-known frameworks are those supporting user interfaces
  - Java Server Pages, ASP.NET



Architectural Mechanism

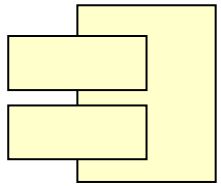
- *Architectural mechanisms represent common concrete solutions to frequently encountered problems. They may be patterns of structure, patterns of behavior, or both.* [RUP]
- Often characterized as
  - “the mechanism for achieving X”
  - “this element is underpinned by mechanism Y”
- Examples
  - Persistency mechanism
  - Error logging mechanism
  - Communication mechanism
  - Shopping cart

Packaged Application

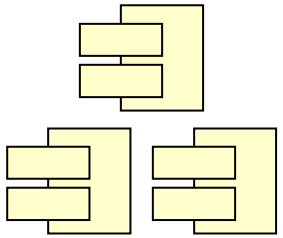
- *A packaged application is a large-grained Commercial-Off-The-Shelf (COTS) product that provides a significant amount of capability (and reuse)*
- Examples
  - Customer Relationship Management (CRM) application (e.g. Siebel)
  - Enterprise Resource Planning (ERP) application (e.g. SAP)
- The amount of custom development required is greatly reduced
- Primary focus is on configuring the application

Component & component library

- Component examples
  - GUI widget (such as a table)
  - Service

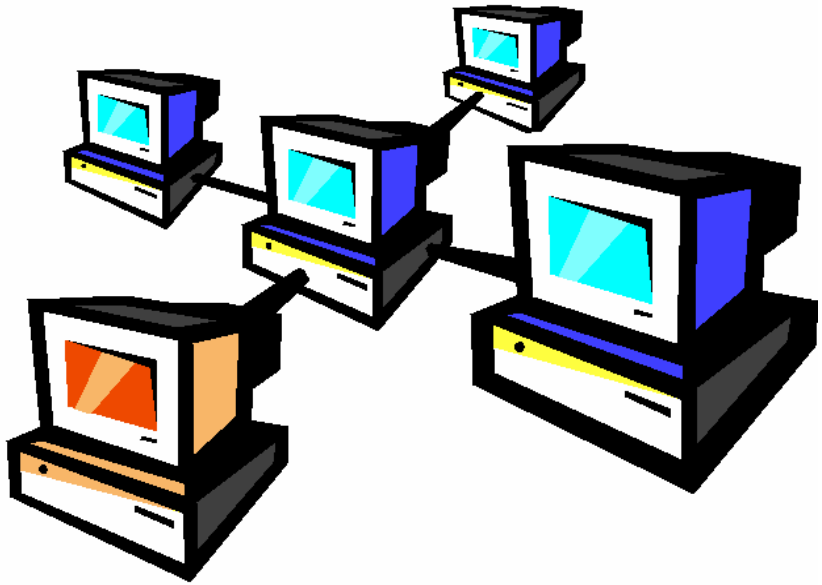


- Component library examples
  - Class libraries (e.g. Java class library)
  - Procedure libraries



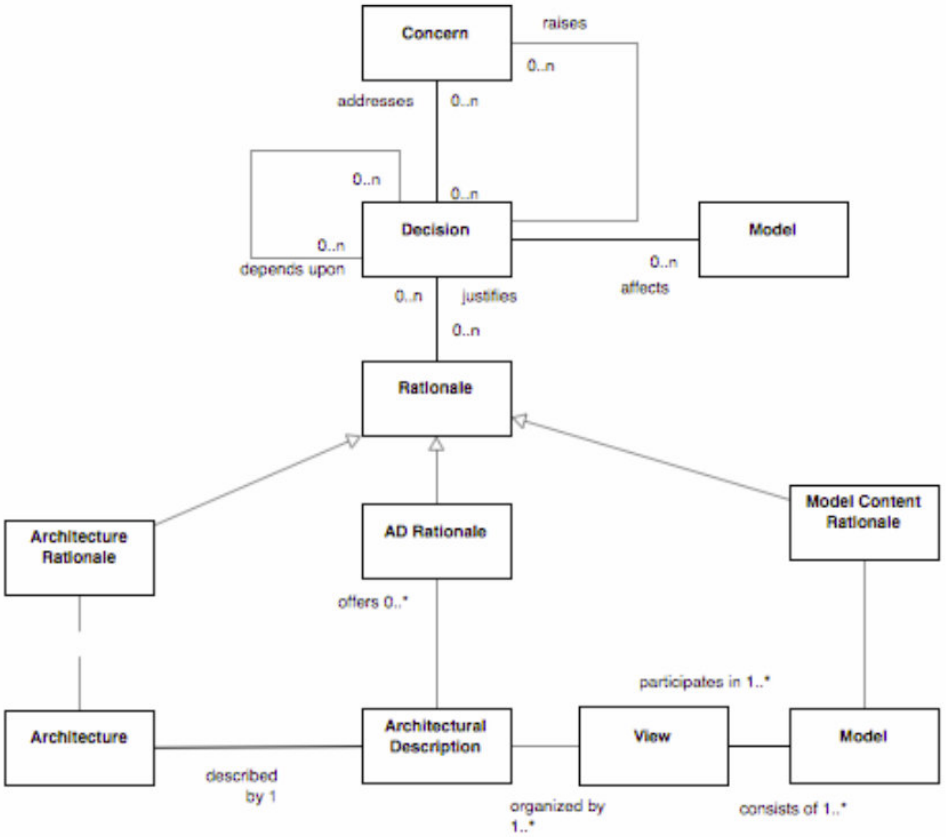
Legacy Application

- *A legacy application is a system that continues to be used because the owning organization cannot replace or redesign it*
- Tends to be a focus on integration rather than new development
- Often results in a focus on enterprise application integration (EAI)



Architectural Decision

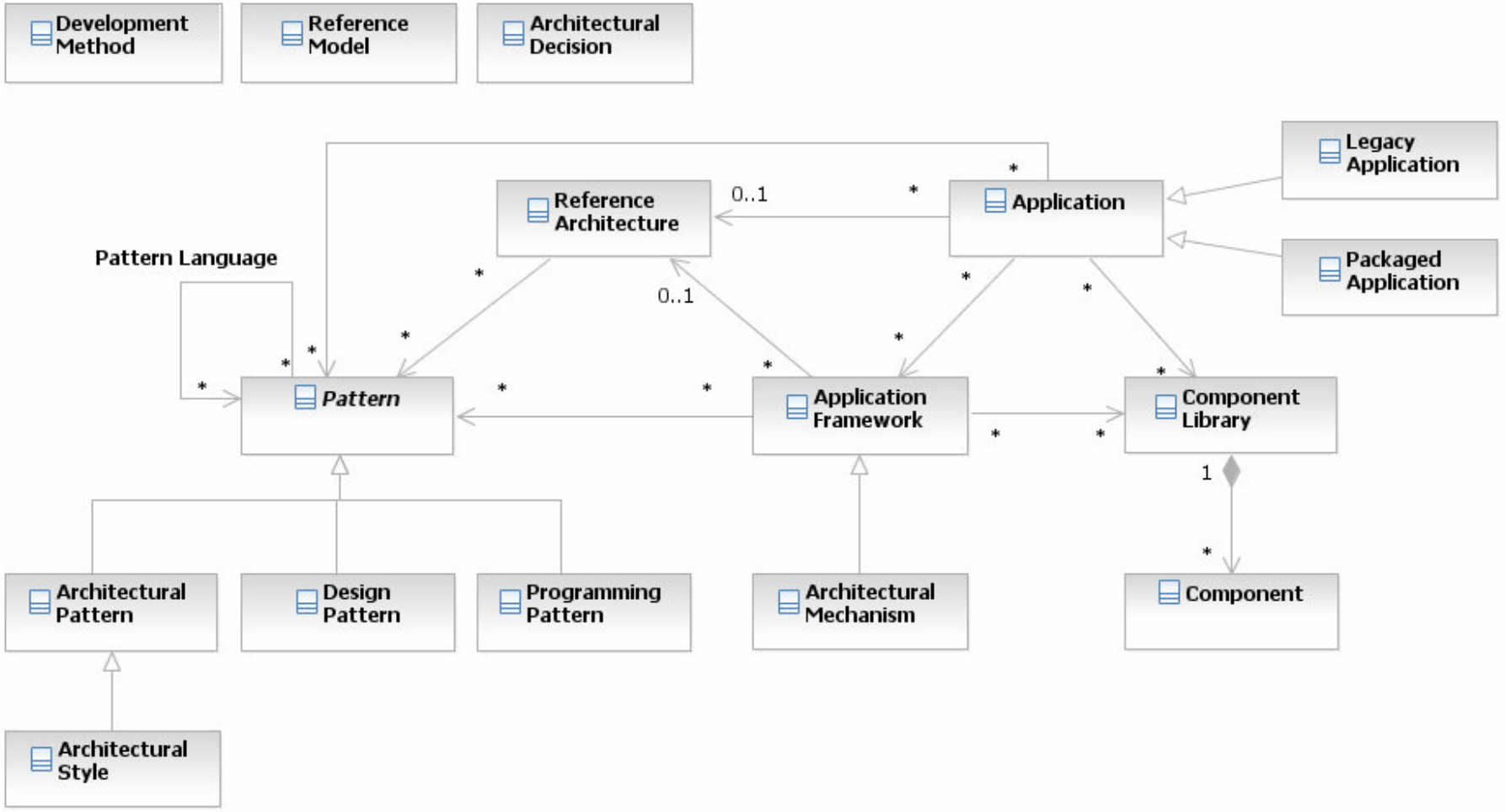
- *[Architectural decisions are] conscious design decisions concerning a software system as a whole, or one or more of its core components. These decisions determine the non-functional characteristics and quality factors of the system. [Zimmermann]*
- Decision rationale may come from experience, method or some other asset



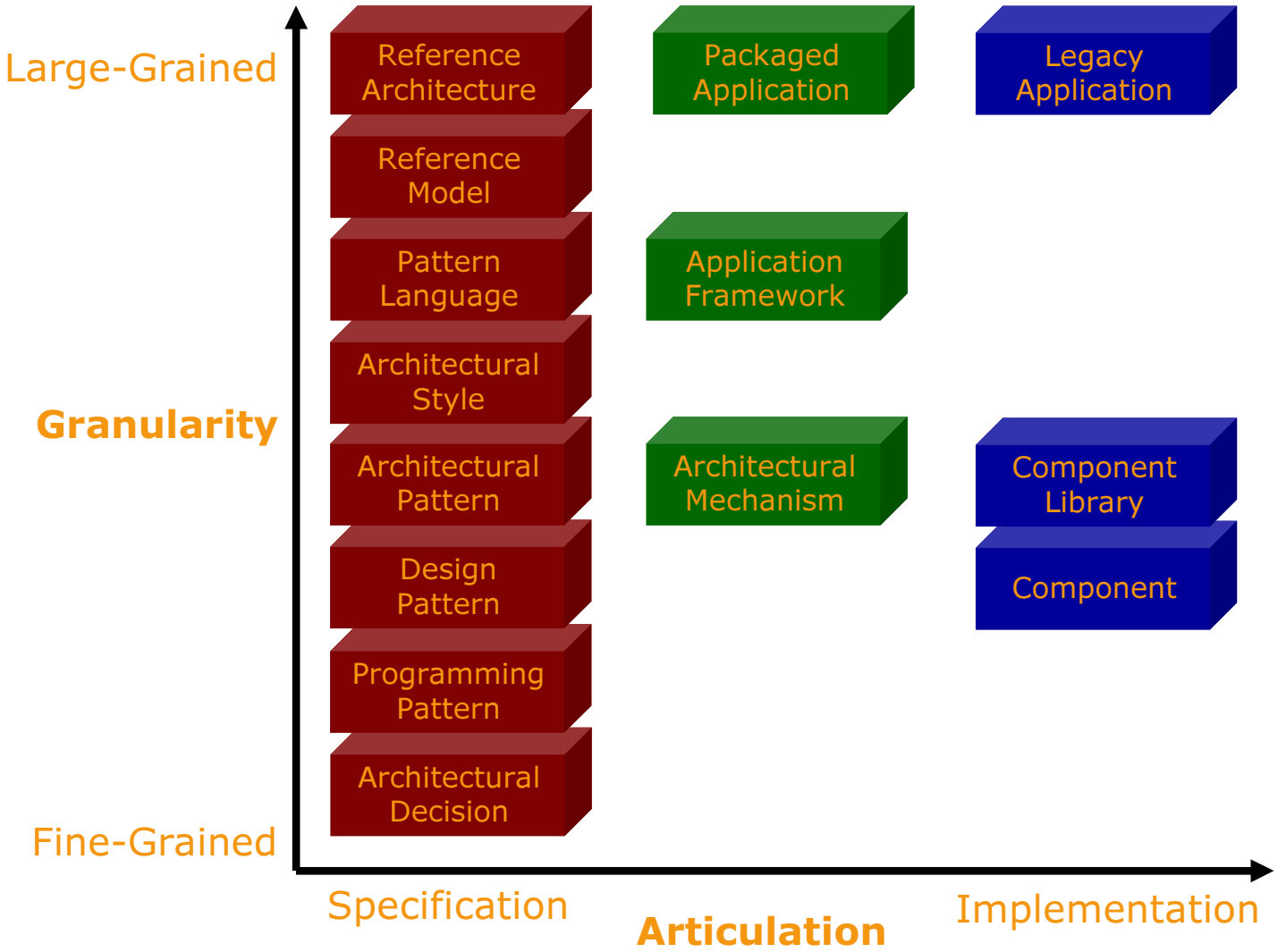
Agenda

- Introduction
- Sources of architecture
- Types of architectural asset
- Characterizing architectural assets
- Automating asset reuse
- Conclusion

An Architectural Asset Metamodel



Attributes of an Architectural Asset



Asset attributes

- General attributes
  - Contained artifacts
  - Name
  - Related assets
  - Usage instructions
  - Version
- Process-related attributes
  - Author
  - Feedback
  - Rating
  - Reviewer
  - State
- Architecture-related attributes
  - Application type (e.g. custom app.)
  - Articulation (e.g. specification)
  - Asset type (e.g. design pattern)
  - Business domain (e.g. telecoms)
  - Development discipline (e.g. testing)
  - Development process (e.g. RUP)
  - Granularity (e.g. fine-grained)
  - Level of abstraction (e.g. logical)
  - Lifecycle phase (e.g. inception)
  - Non-functional properties (e.g. cost)
  - Scope (e.g. systems engineering)
  - Technical domain (e.g. embedded)
  - Variability (e.g. limited)
  - Visibility (e.g. public scope)

Asset Categories

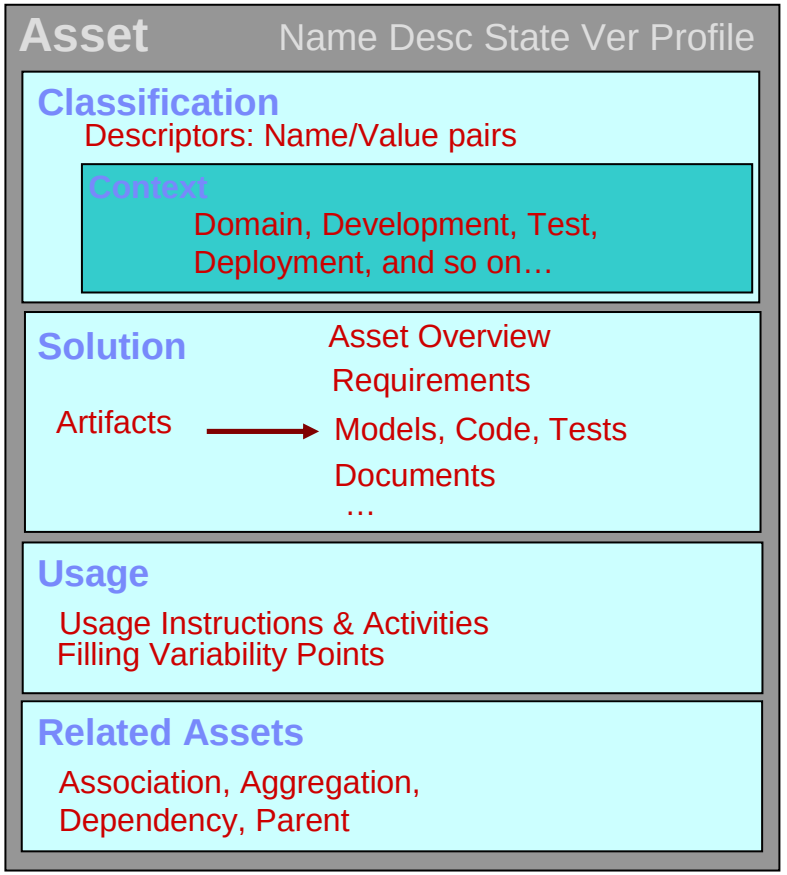
- A category is a search mechanism
- It provides a match on assets with specific attribute values
  - E.g. All assets associated with the telecoms business domain
  - E.g. All assets whose cost is less than $100

Agenda

- Introduction
- Sources of architecture
- Types of architectural asset
- Characterizing architectural assets
- Automating asset reuse
- Conclusion

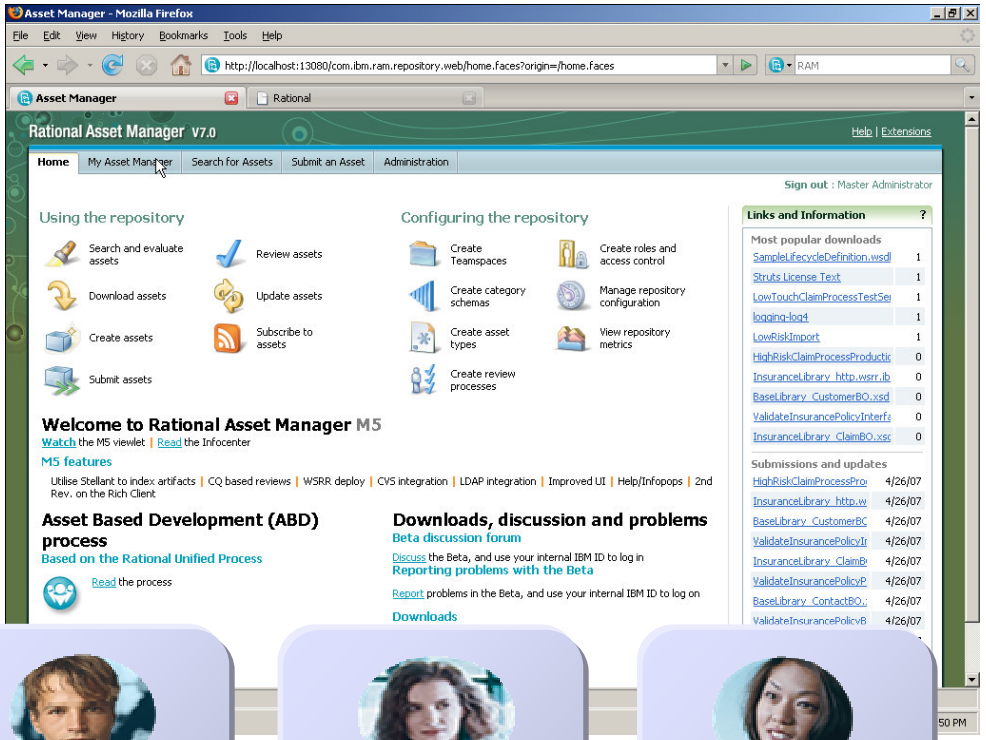
The Reusable Asset Specification (RAS)

- An OMG standard
- Defines a standard way to describe and package assets
- Defines the interface to a RAS repository
- RAS is used to package many kinds of assets including components, services, patterns, and so on



Rational Asset Manager (RAM)

- Manages assets across their lifecycle from design/creation to consumption/change
- Leverages an extensive library of process best practices for asset creation & reuse



Asset Administrator

Define Configuration

Analyst, Architect, Developer, Tester

Create & Modify Assets

Analyst, Architect, Developer, Tester

Search & Retrieve Assets

Project Manager, Reviewer, Approver

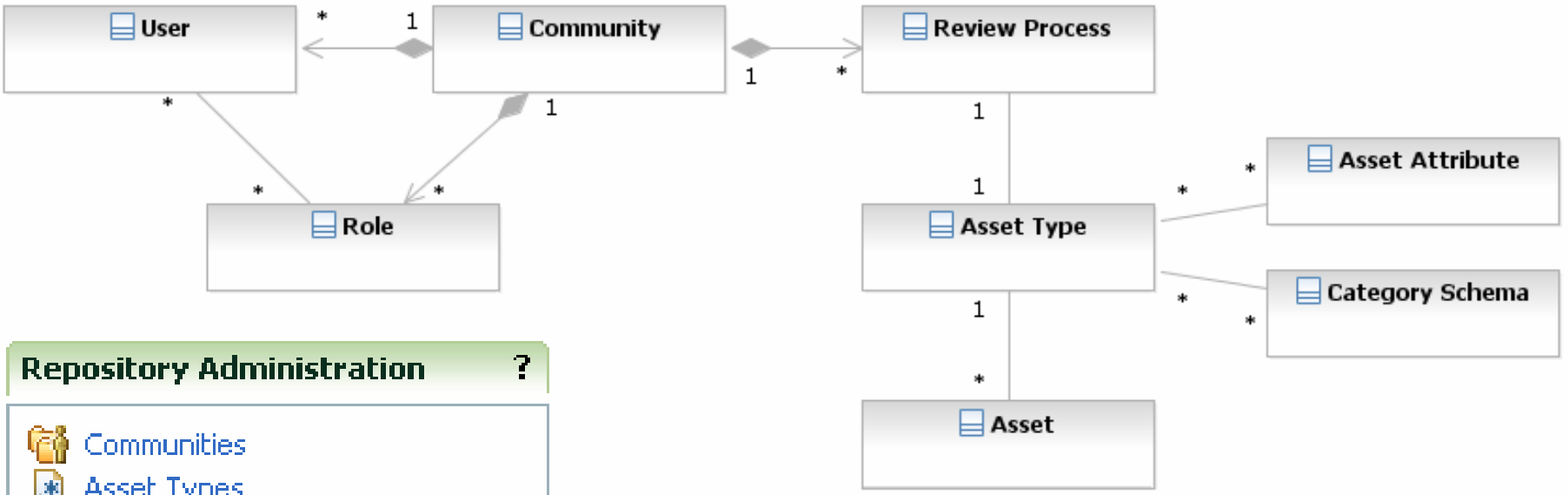
Measure

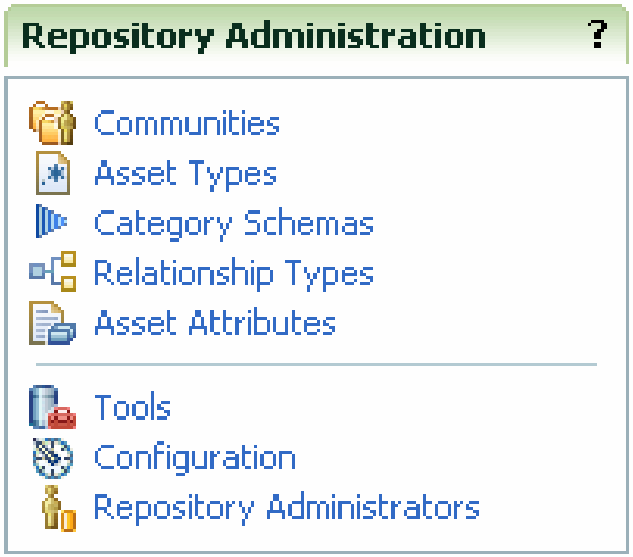
Project Manager, CIO/CTO

Govern

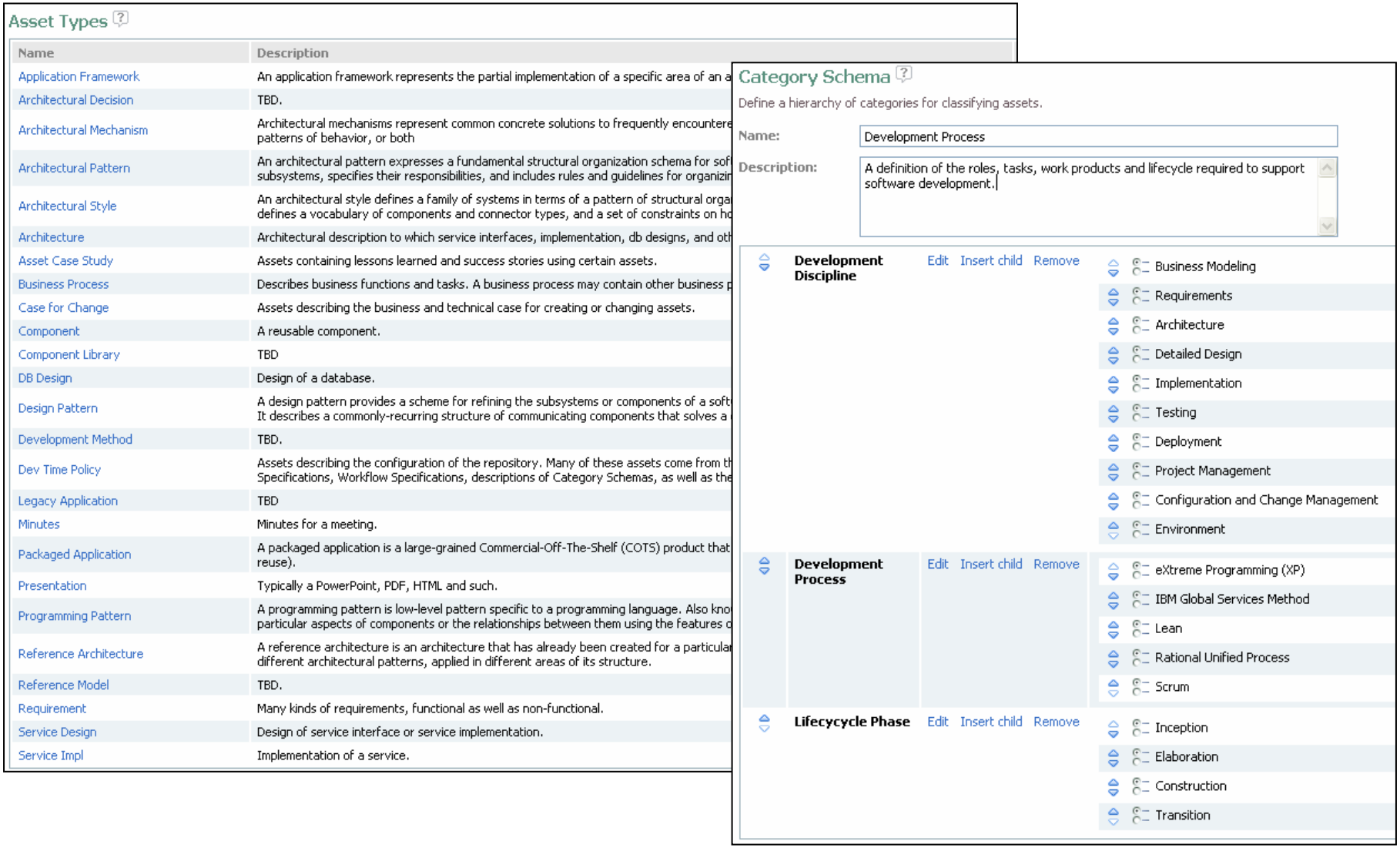
Asset Management Lifecycle

RAM - Configuration





RAM – Asset types, category schema



Asset Types

Name	Description
Application Framework	An application framework represents the partial implementation of a specific area of an a
Architectural Decision	TBD.
Architectural Mechanism	Architectural mechanisms represent common concrete solutions to frequently encountere patterns of behavior, or both
Architectural Pattern	An architectural pattern expresses a fundamental structural organization schema for soft subsystems, specifies their responsibilities, and includes rules and guidelines for organizin
Architectural Style	An architectural style defines a family of systems in terms of a pattern of structural orga defines a vocabulary of components and connector types, and a set of constraints on ho
Architecture	Architectural description to which service interfaces, implementation, db designs, and oth
Asset Case Study	Assets containing lessons learned and success stories using certain assets.
Business Process	Describes business functions and tasks. A business process may contain other business p
Case for Change	Assets describing the business and technical case for creating or changing assets.
Component	A reusable component.
Component Library	TBD
DB Design	Design of a database.
Design Pattern	A design pattern provides a scheme for refining the subsystems or components of a soft It describes a commonly-recurring structure of communicating components that solves a
Development Method	TBD.
Dev Time Policy	Assets describing the configuration of the repository. Many of these assets come from th Specifications, Workflow Specifications, descriptions of Category Schemas, as well as the
Legacy Application	TBD
Minutes	Minutes for a meeting.
Packaged Application	A packaged application is a large-grained Commercial-Off-The-Shelf (COTS) product that reuse).
Presentation	Typically a PowerPoint, PDF, HTML and such.
Programming Pattern	A programming pattern is low-level pattern specific to a programming language. Also kno particular aspects of components or the relationships between them using the features o
Reference Architecture	A reference architecture is an architecture that has already been created for a particular different architectural patterns, applied in different areas of its structure.
Reference Model	TBD.
Requirement	Many kinds of requirements, functional as well as non-functional.
Service Design	Design of service interface or service implementation.
Service Impl	Implementation of a service.

Category Schema

Define a hierarchy of categories for classifying assets.

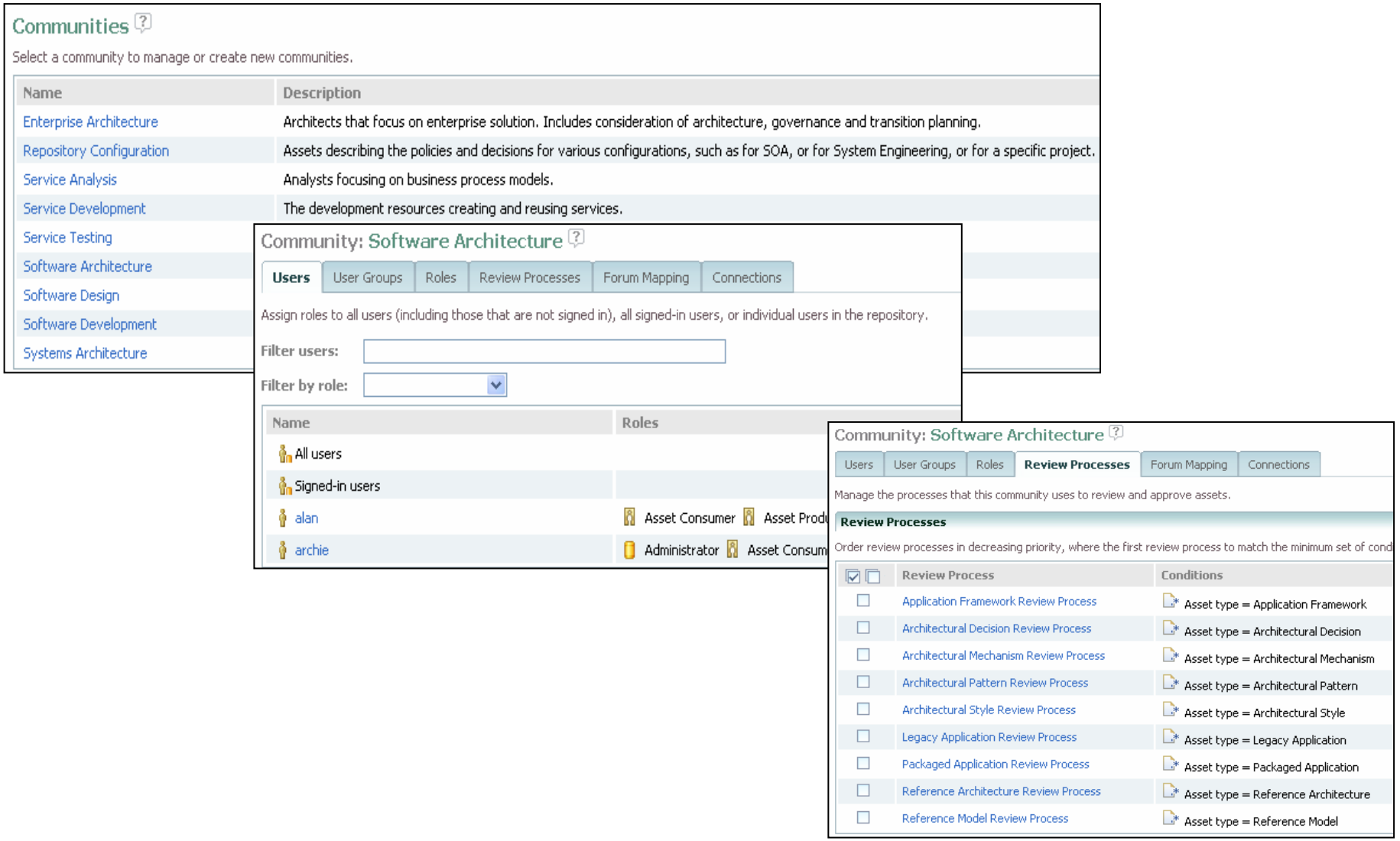
Name: Development Process

Description: A definition of the roles, tasks, work products and lifecycle required to support software development.

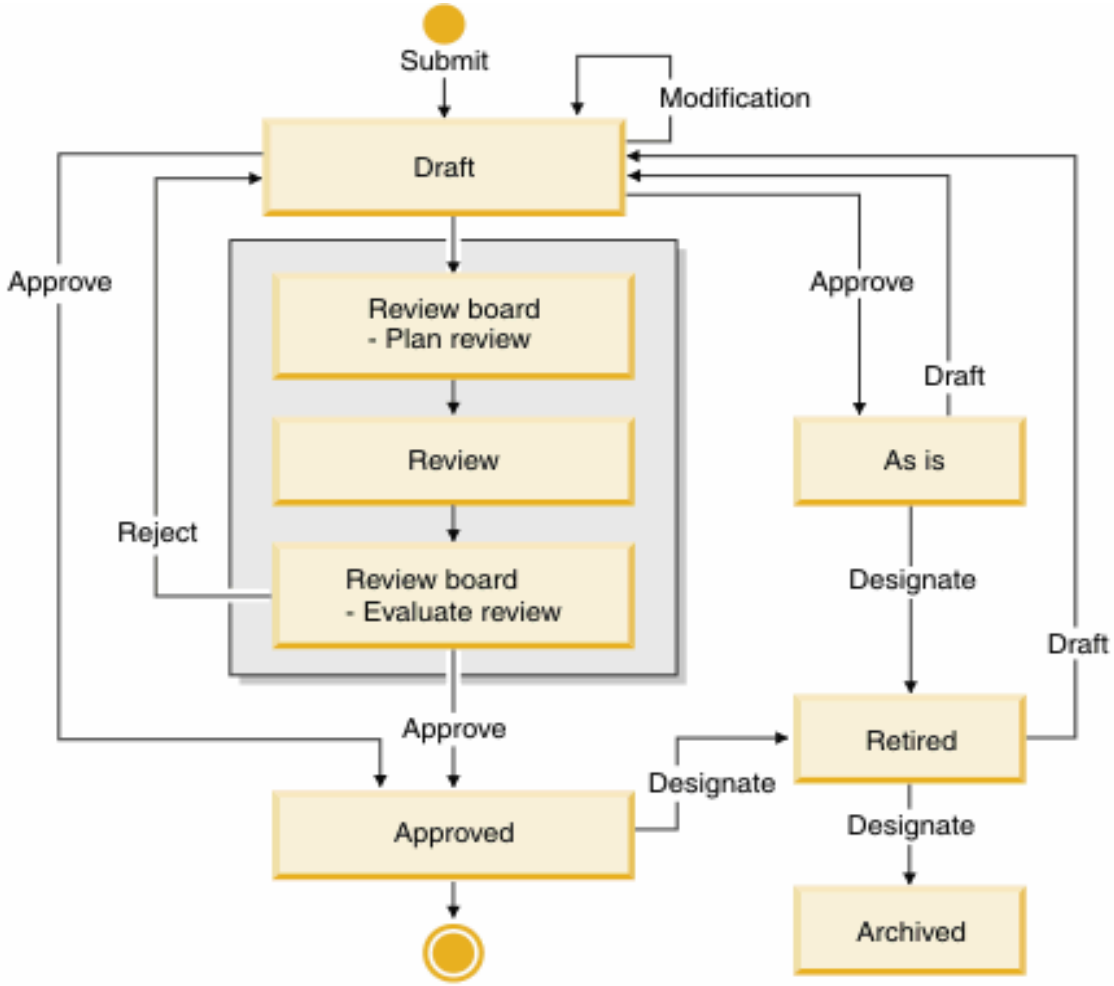
Development Discipline	Edit	Insert child	Remove
Development Process	Edit	Insert child	Remove
Lifecycle Phase	Edit	Insert child	Remove

- Business Modeling
- Requirements
- Architecture
- Detailed Design
- Implementation
- Testing
- Deployment
- Project Management
- Configuration and Change Management
- Environment
- eXtreme Programming (XP)
- IBM Global Services Method
- Lean
- Rational Unified Process
- Scrum
- Inception
- Elaboration
- Construction
- Transition

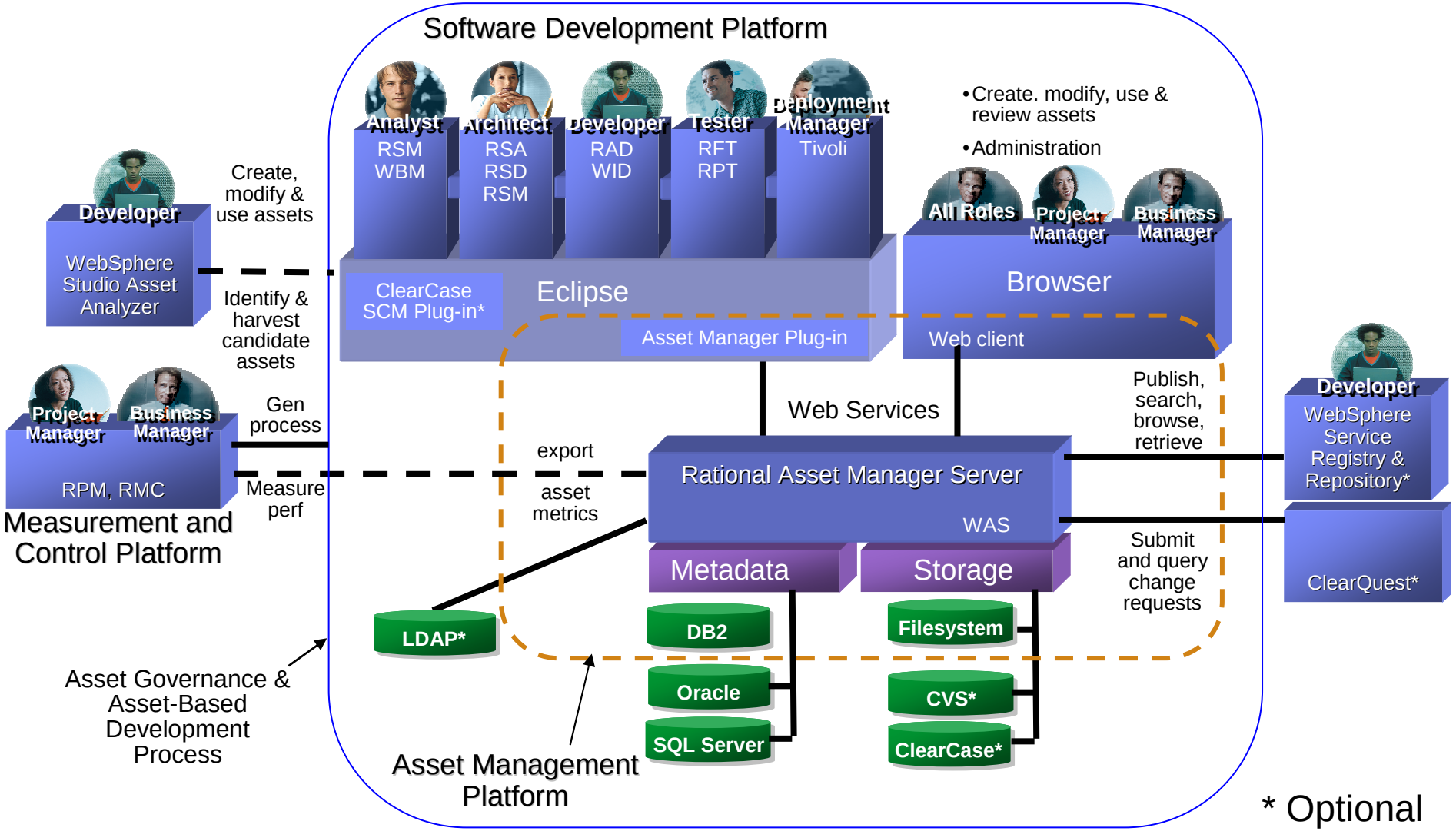
RAM – Communities, review processes



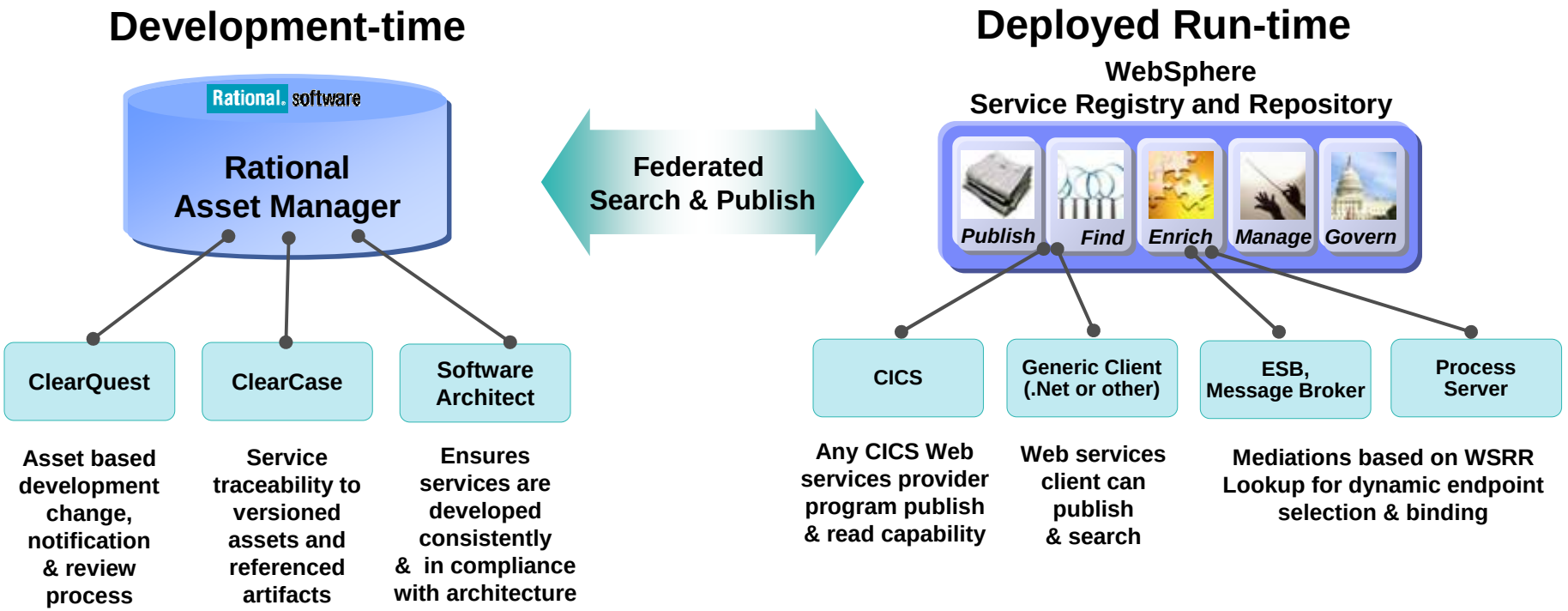
RAM – Asset Lifecycle



RAM - Architecture



RAM and WSRR



Manages information that is useful for developing, re-using and managing all types of *reusable assets*

Manages information that is useful for the runtime operation, management and development use of *services*

Conclusion

- Many types of architectural asset are at the disposal of the architect
- Understanding their characteristics and value can help the architect in their decision-making
- Application of appropriate automation is critical in ensuring the success of strategic reuse


QUESTIONS

THANK YOU

Learn more at:

- IBM Rational software
- IBM Rational Software Delivery Platform
- Process and portfolio management
- Change and release management
- Quality management
- Architecture management
- Rational trial downloads
- Leading Innovation Web site
- developerWorks Rational
- IBM Rational TV
- IBM Rational Business Partners