Understanding Architectural Assets

Peter Eeles
Understanding Architectural Assets

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

Agenda

- Introduction
  - Sources of architecture
  - An architecture asset metamodel
  - Asset types
  - Attributes of an architecture asset
  - Other reuse considerations
  - Summary
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
  - An architecture asset metamodel
  - Asset types
  - Attributes of an architecture asset
  - Other reuse considerations
- Summary

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
- An architecture asset metamodel
  - Asset types
  - Attributes of an architecture asset
  - Other reuse considerations
- Summary

What Types of Architecture Assets are there?

- Reference Architecture
- Design Pattern
- Existing Application
- Architectural Mechanism
- Viewpoint Catalog
- Packaged Application
- Development Method
- Reference Model
- Architectural Decision
- Idiom
- Pattern
- Component Library
- Component
- Architectural Pattern
- Architectural Style
- Application Framework
A Metamodel of Architecture Assets

Agenda

- Introduction
- Sources of architecture
- An architecture asset metamodel
- Asset types
- Attributes of an architecture asset
- Other reuse considerations
- Summary
Reference Architecture

- A reference architecture is associated with a particular domain of interest. It typically includes many different architectural patterns, applied in different areas of its structure.

- A reference architecture is an architecture
  - It is therefore described in the same manner as any other architecture
  - It may contain concrete (implementation) elements

- Concrete reference examples may be available
  - In 1999 IBM published a series of technical reference architectures
    - e-business
    - Life Sciences
    - Wireless

Development Method

- Best practices, guidance, techniques and standards
- Templates (e.g. architecture description template) and examples
- Augmented by appropriate tooling

Rational Unified Process
Viewpoint Catalog

- An architect will typically describe their architecture using a set of viewpoints that have been selected from a viewpoint catalog
- Referenced by the development method

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
Architectural Mechanism

- Architectural mechanisms represent common concrete solutions to frequently encountered problems. They may be patterns of structure, patterns of behavior, or both. [SPEM]

- 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

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
  - Idioms
  - 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**

- **Personal Organizer** (from Business Specific)
- **Address Book** (from Business Specific)
- **Calculator** (from Application-Specific)
- **Filestore Management** (from Base)
- **Memory Management** (from Base)
- **Math** (from Base)
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]

Idiom

- 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]

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

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.
Architecture 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.

Existing Application

- A legacy application is a system that continues to be used because the owning organization cannot or will not replace or redesign it.
  - “A cynic is a man who knows the price of everything but the value of nothing.” [Oscar Wilde]

- The focus is on integration rather than custom development.
  - Referred to as legacy integration or enterprise application integration (EAI).
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.

- “Software as a Service” (SaaS) is a possible deployment model.

---

Application Framework

- An **application framework** represents the partial implementation of a specific area of an application.

- Vary widely in scale:
  - Java EE, .NET
  - Hibernate, Java Server Pages, ASP.NET

---
Component & Component Library

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

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

Agenda

- Introduction
- Sources of architecture
- An architecture asset metamodel
- Asset types
- Attributes of an architecture asset
- Other reuse considerations
- Summary
Attributes of an Architecture Asset

Asset attributes

- General attributes
  - Articulation
  - Author
  - Concerns addressed
  - Contained artifacts
  - Granularity
  - Name
  - Prerequisites
  - Related assets
  - State
  - Type
  - Usage instructions
  - Variability
  - Version
Asset attributes (2)

- Usage-related attributes
  - Feedback
  - Rating
  - Usage count
- Application-related attributes
  - Business domain
  - Development discipline
  - Development method
  - Development phase
  - Development scope
  - Technical domain
- Non-functional attribute properties
  - Cost, performance, scalability, etc.

Attributes in Action

- Finding assets by matching against 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
- An architecture asset metamodel
- Asset types
- Attributes of an architecture asset
- Other reuse considerations
- Summary

Standards – 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
Process – An Asset Lifecycle

Tools (e.g. Rational Asset Manager)

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

- Roles, tasks, work products (e.g. asset catalog)
- Embedding of a reuse “culture”
- Determine the level of reuse that is practical or beneficial to the organization

Summary

- 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
Coming Soon!

THE PROCESS OF
SOFTWARE ARCHITECTING

PETER LELES
PETER CRIPPS

Questions