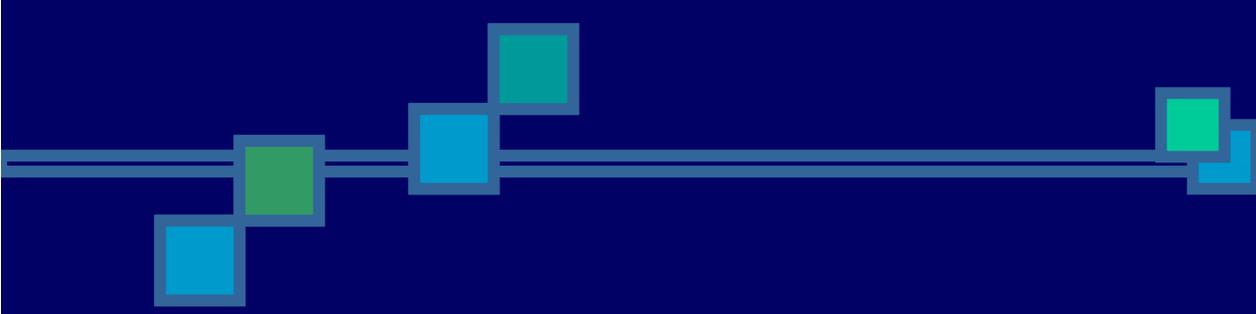


Using the Recycling Bin The Reusable Asset Specification



Tony Grout
Technical Director
FMI Solutions



Agenda



- Small word, big challenge
 - The reusable asset specification
 - Extensibility mechanisms
 - An example reusable asset (RAsset)
 - Potential reuse workflows
 - Benefits for us
- 



Small Word, Big Challenge

- Political Issues
 - Team focus is on immediate project success
 - Lack of process focus on reuse
 - Reuse is not just about code
 - Lack reward for reuse



Small Word, Big Challenge

- Financial Issues
 - Lack of funding to enable reuse
- Technical Issues
 - Lack of a common way to describing assets
 - Lack of common architectural contexts



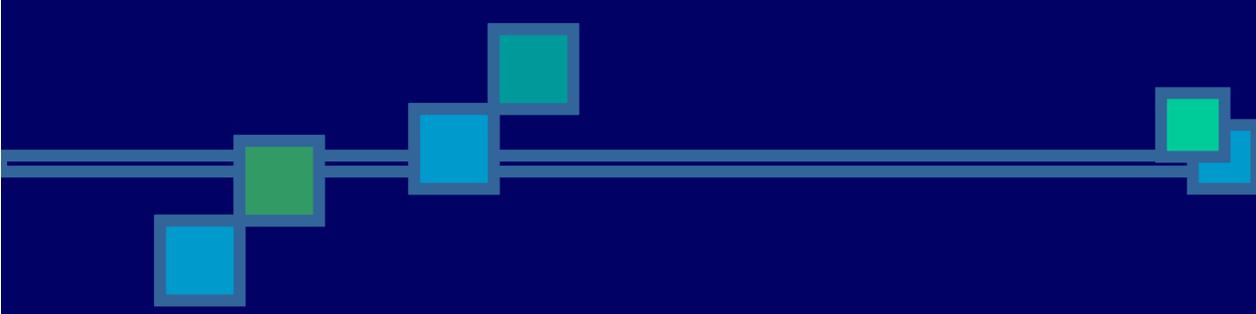
Overcoming the Challenges

- Create reuse resource
- ★ Use a process that has reuse built in
 - Use “cost plus” accounting to cover reuse
- ★ Adopt a standard describing mechanism
 - Govern architectural styles in order to standardise



The Reusable Asset Specification

- Common way of describing assets
- Covers more than just code
- Covers more than just executables
- Can be automated



What Is an Asset?

- *"a reusable software asset is a software artifact or a set of related artifacts that have been created or harvested with an explicit purpose of applying it repeatedly in subsequent development efforts. An asset has a description of the way(s) in which it should be used and applied."*
- 

From Strategy to Platter



Business Strategy

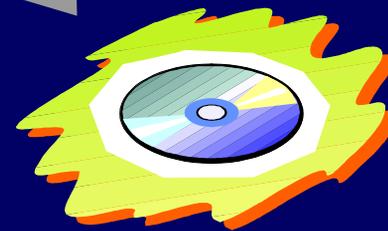
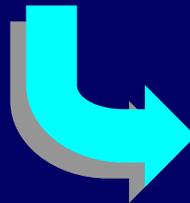
Software Requirements

Analysis & Design

Implementation

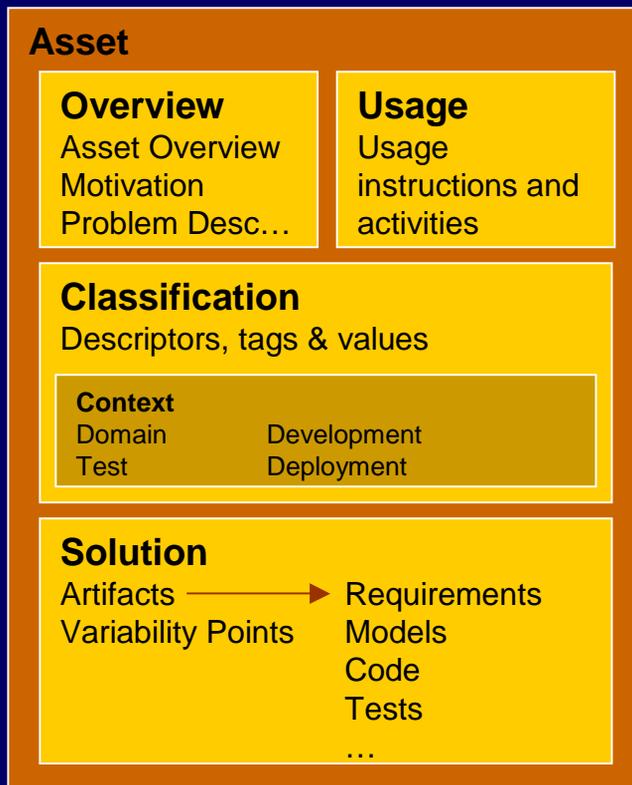
Physical Information

Using traceability
between artifacts
across workflows
to create assets
enables reuse at a
strategic level

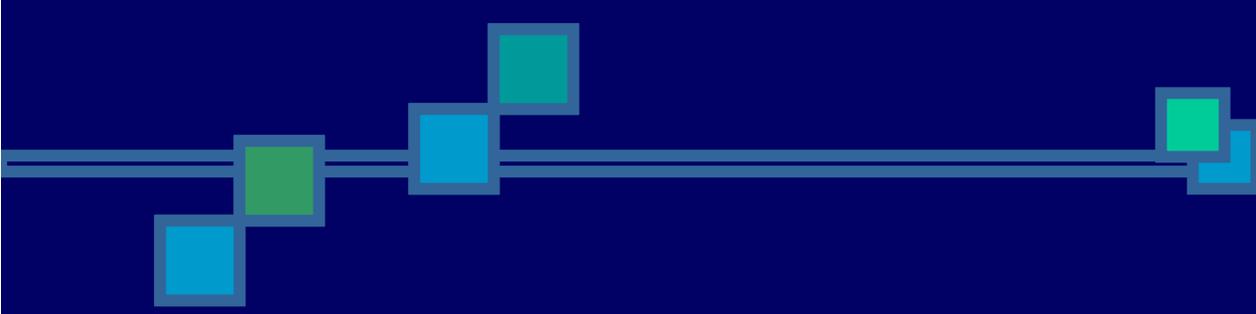




Describing a Reusable Asset

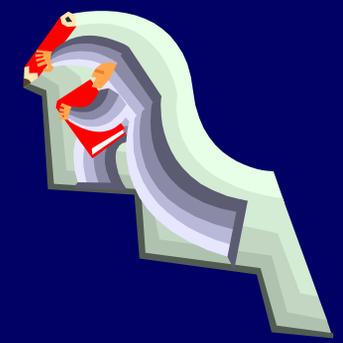
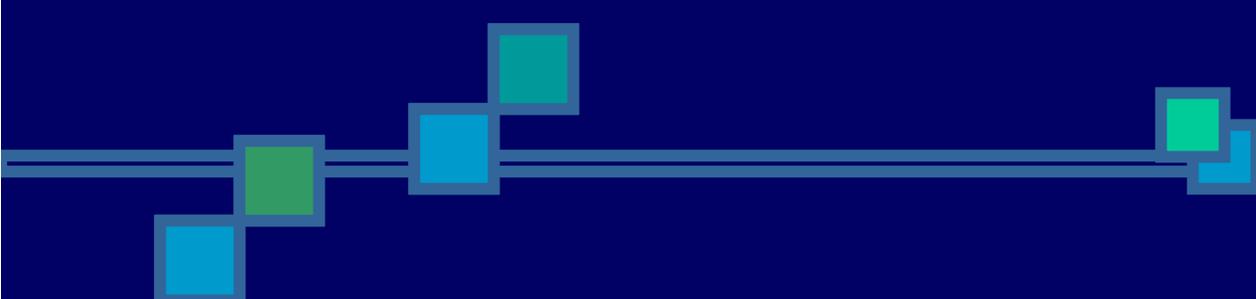


When defining an asset using the Reusable Asset Specification these sections are core



Overview

- Gives a general overview of the RAsset
 - Describes the RAssets purpose
 - Enlightens on the problem being solved
 - Describes the motivation for the RAsset
- 



Usage

- Activities
 - Steps and direction on how to use the asset with a focus on using variability points
 - Artifact Usage Instructions
 - Activities can be assigned to specific artifacts
 - Context Activities
 - Activities can be filtered based on the context the asset is being used in
- 



Classification – Descriptors

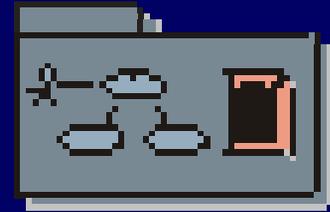
- Tags for searching and generally applying the asset
- Specification has suggested descriptors
- Examples
 - Asset Type - component
 - Author
 - Keywords
 - Version
 - Technologies



Classification – Context

- Domain
 - Business area covered - Financial Services
- Development
 - The environment to develop within – J2EE
- Test
 - Test configurations to test the asset
- Deployment
 - Runtime environment for the asset

Solution

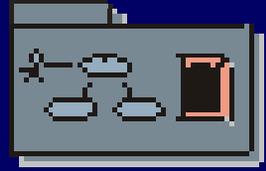
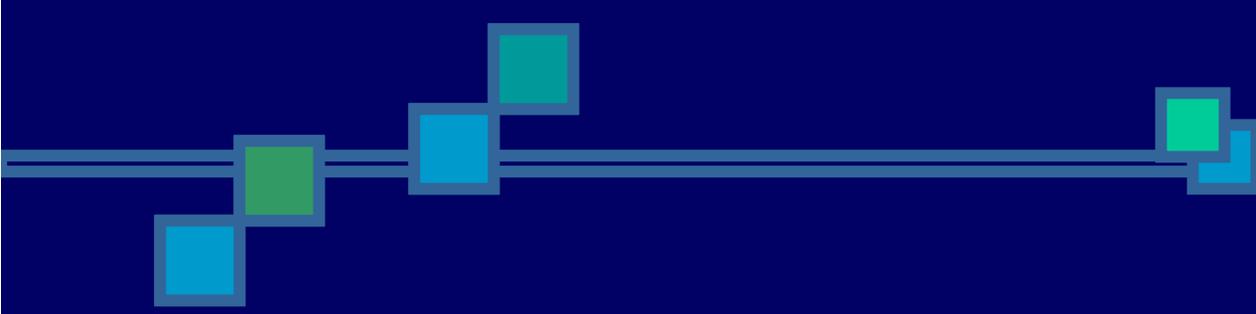


- Contains the artifacts making the asset
- An asset has only one solution section
- An artifact can have variability points
 - Where customisation can take place
- All assets have a root context
 - The starting point for the artifacts

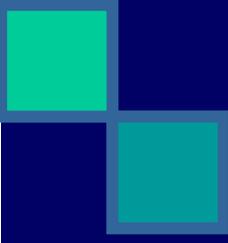


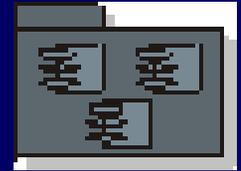
Suggested Solution Sections

- Requirements
- Design
- Realization
- Test



Solution – Requirements

- 
- 
- Requirements Documents
 - Business Object Model
 - Business Use Case Model
 - Analysis Model
 - System Use Case Model
 - Supplementary Specification



Solution – Design

- Use case realisations
- Component specification

Solution – Realization

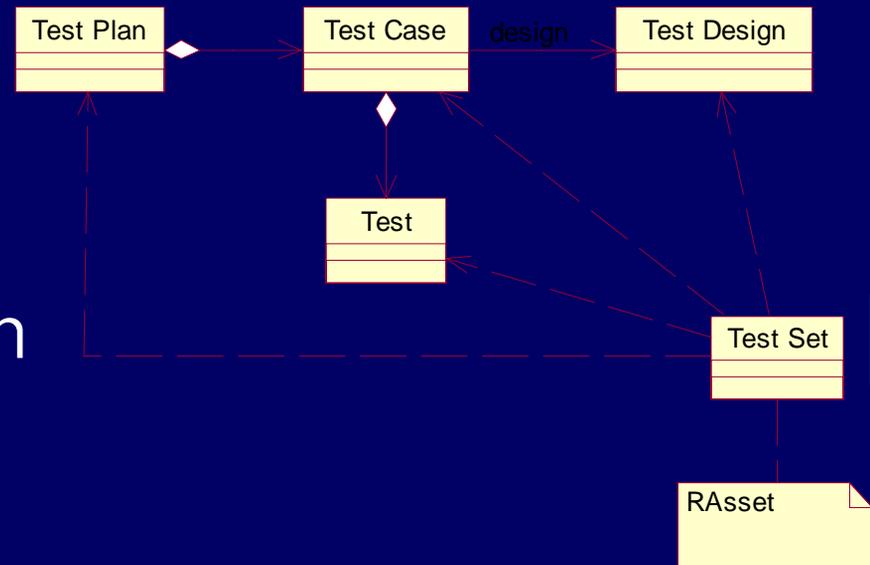
- Specify implementation content
- Implementation Model Specification
 - Specification of subsystems & components
- Deployment Model Specification
 - Describes distribution of components, processes and nodes



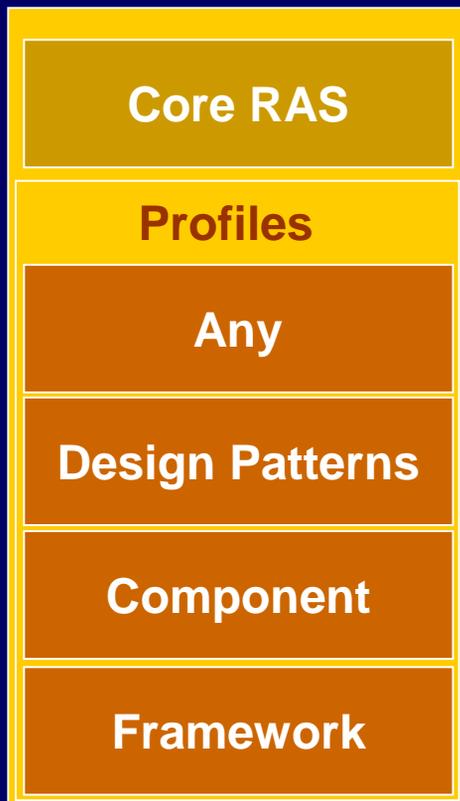


Solution - Test

- Asset test plan
- Asset test cases
- Asset test design



Extensibility Mechanisms

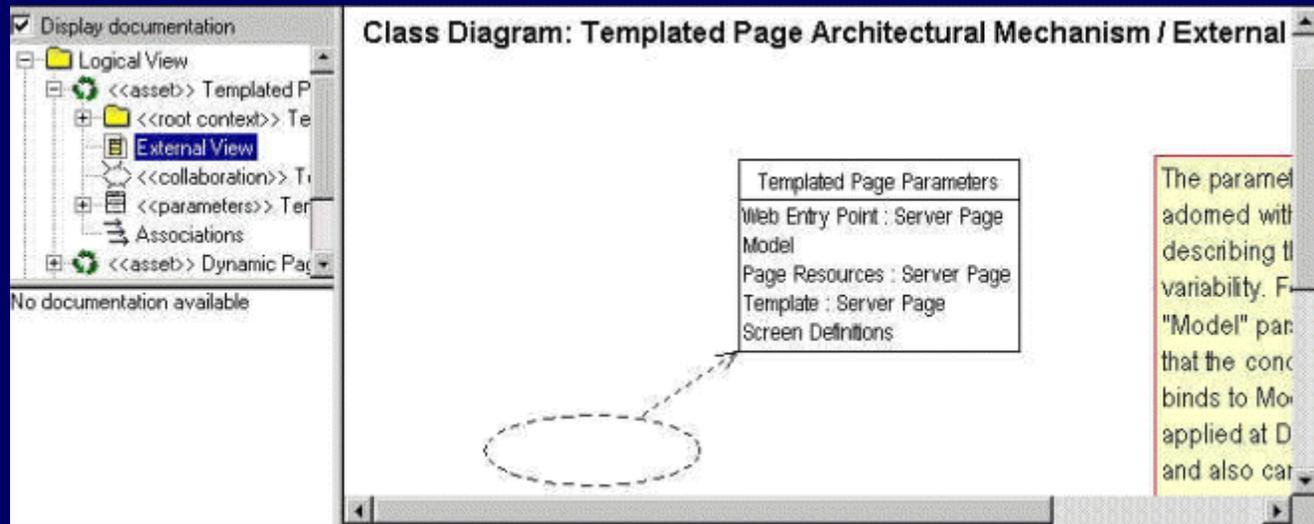


An XSD file has been created, capturing the semantics of the Core Specification

Introduce new elements to the core RAS using a profile

Example RAsset

The Reusable Assets Specification online



Example RAsset

← ↑ ↓ → Where am I | Glossary | Feedback

Implementation

This reference implementation of the Templated Page mechanism was developed and tested with the Tomcat version 3.2 Beta. [Tomcat](#) is an implementation of the Java Servlet 2.2 and JavaServer Pages 1.1 Specifications. Its development is organized under by [The Apache Software Foundation](#).

Topics

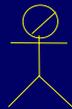
- [Build File](#)
- [Deployment Descriptors](#)
- [Java Classes Source](#)
- [HTML and Java Server Pages](#)

Build File ↑

Business Actors involved



System Developer



Asset Packager



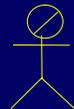
Asset Developer



Reuse Candidate Identifier



Artifact Sanitizer



Asset Reviewer



Asset Harvester



Asset Consumer

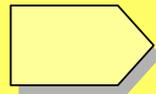


Asset Broker

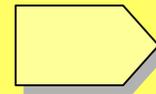
Potential Workflow Details



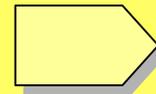
Capture Asset



Harvest Asset



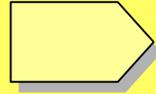
Sanitise Asset



Package Asset



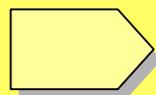
Search Asset



Search Asset



Apply Asset



Apply Asset



Benefits - Organisational

- Working extendable solution
- Consistency of strategic approach
- Reuse outside of the organisation
- Focus can be on unique extensions to industry standard assets

Benefits – Project

- Reduced risk – already proven
- Increased speed of delivery
- Ability to go outside the enterprise





Benefits – Team Member

- Focus on the challenging issues
- Knowledge transfer at the pattern level



More Information

- Rational web site
 - www.rational.com/eda
- FMI Solutions web site downloads
 - www.fmisolutions.com in Products / Whitepapers
 - Contact tony.grout@fmisolutions.com

Acknowledgements

- Grant Larsen (Rational)
- Paul D'Ambra (Barclays)