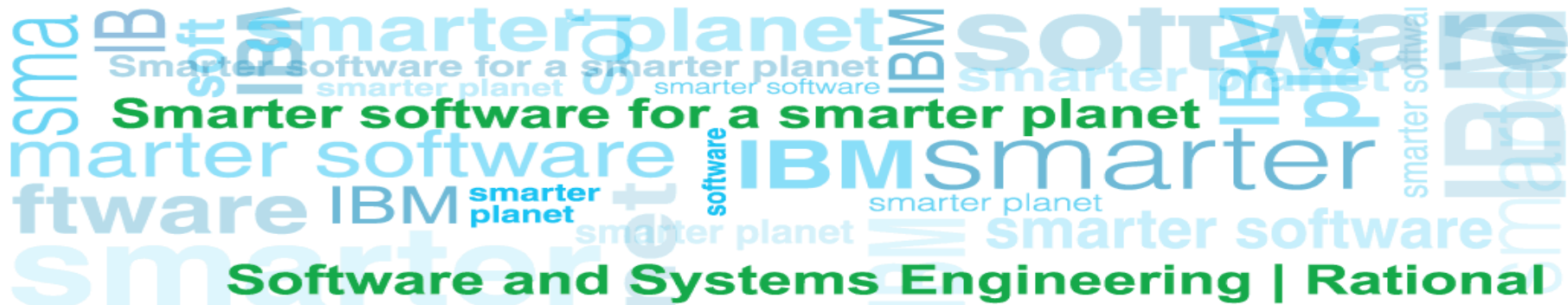


INCOSE IW 2014

MBSE track, Model Management WG

Amit Fisher

Program Director, Systems Technical Client Relationship Manager,
IBM Software Group, Rational



Agenda

- [Linked Lifecycle Data and OSLC](#)
- Rhapsody and Rhapsody Design manager
- Model Management in the context of Product Line Engineering

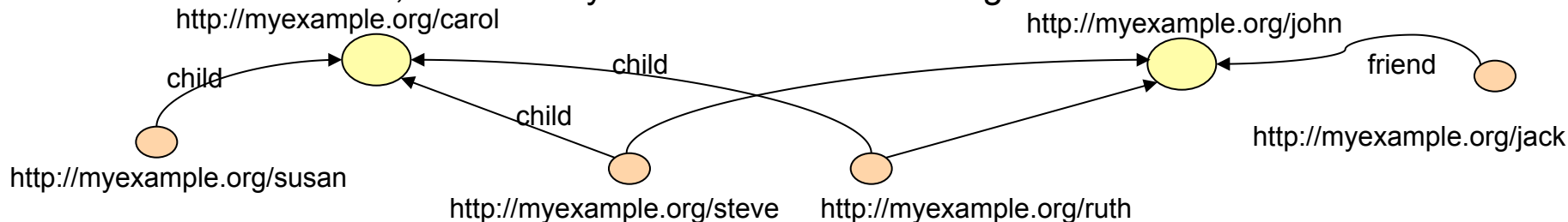
Linked Data



<http://www.w3.org/DesignIssues/LinkedData>

Four simple principals :

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names
3. When someone looks up a URI, provide useful information, using standards (e.g. RDF*, SPARQL**, ***REST)
4. Include links to other URIs, so that they can discover more things

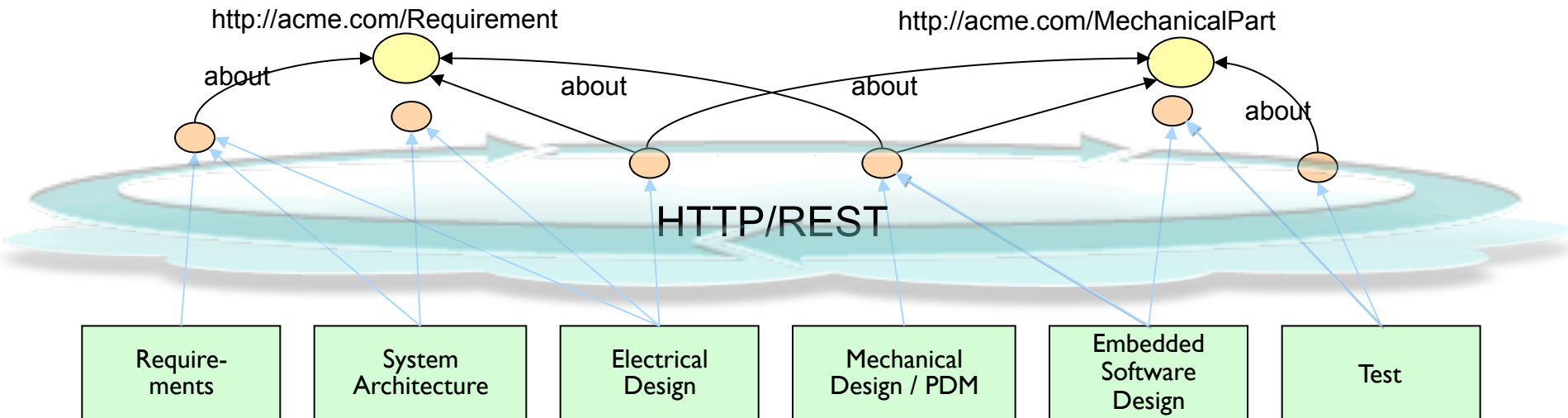


*RDF, the Resource Description Framework provides a generic graph-based data model for describing things, including their relationships with other things.

** SPARQL is a query language able to retrieve and manipulate data stored in RDF format

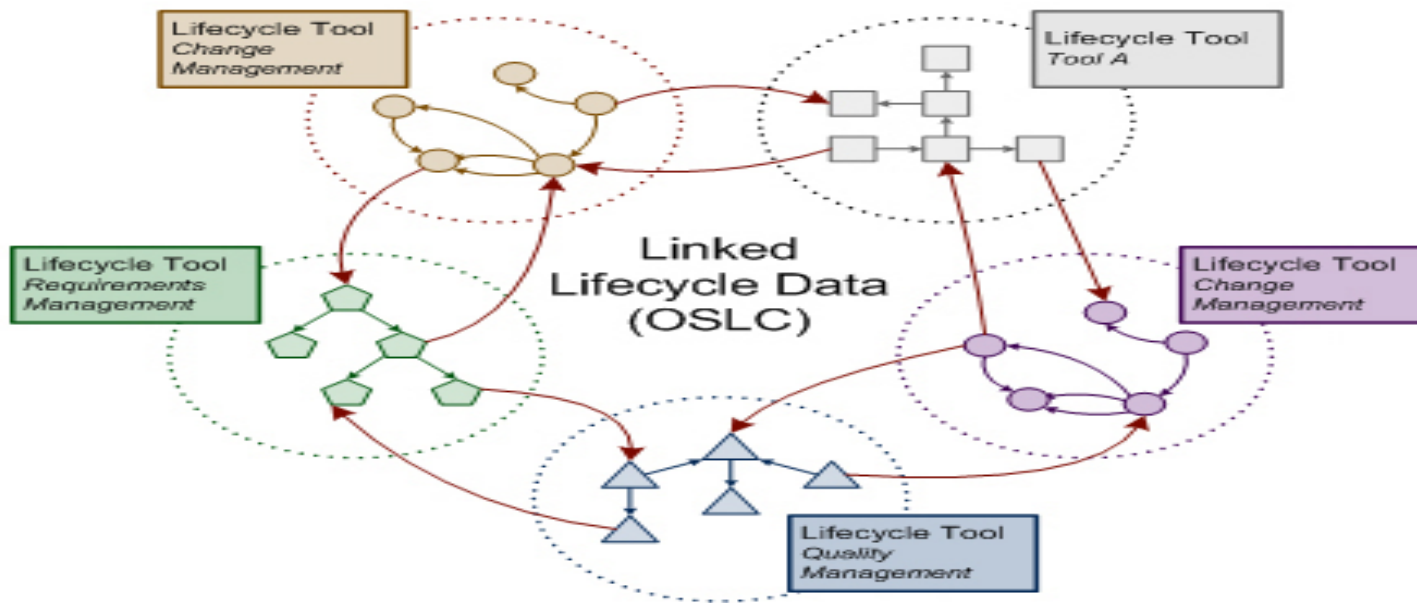
***REST, REpresentational State Transfer (REST) is a style of software architecture for distributed systems where requests and responses are built around the transfer of representations of addressable resources

Leveraging Linked Data concepts in engineering



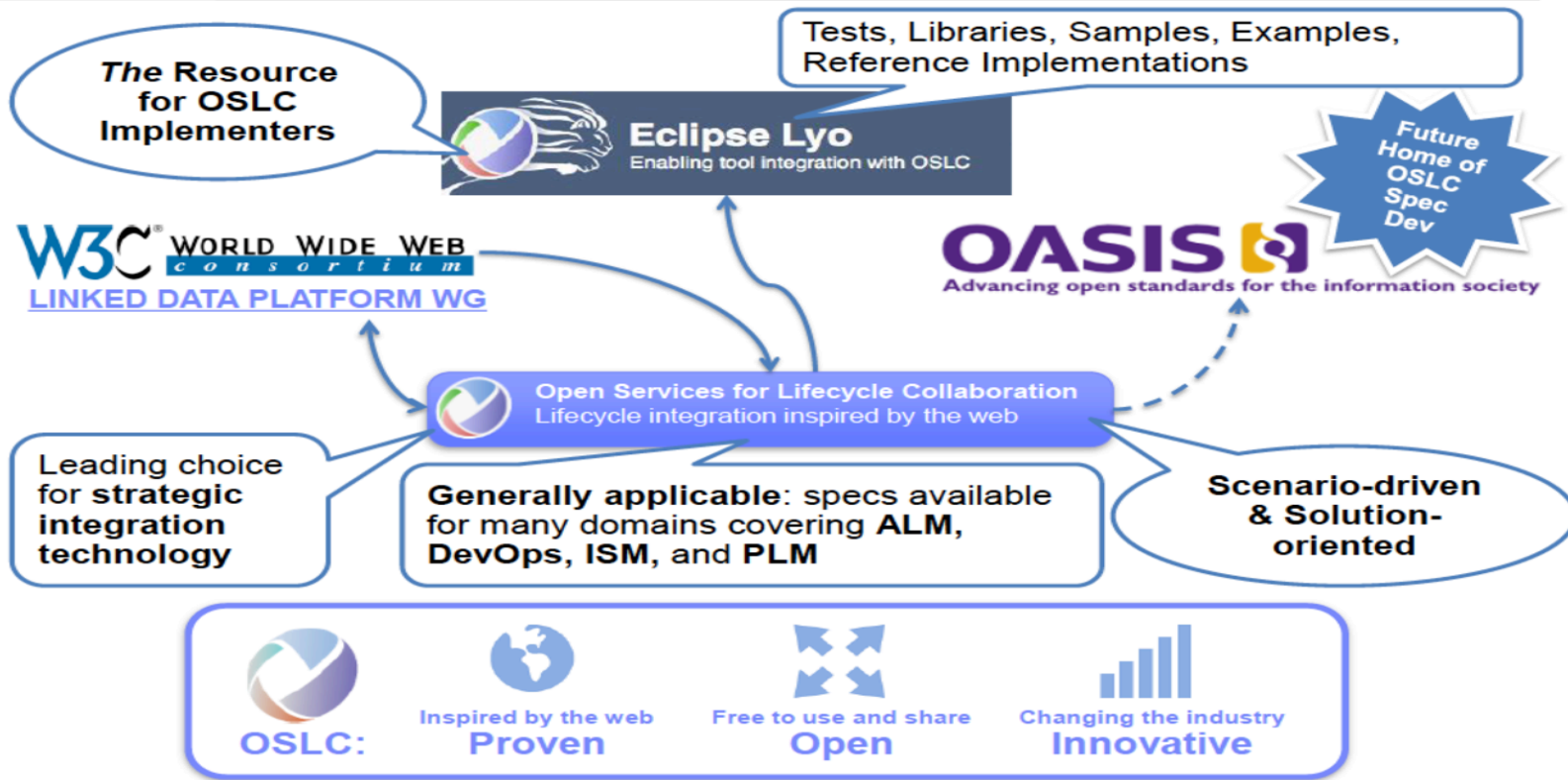
Linking Lifecycle Data via OSLC

Resources from different domain tools are linked together using OSLC

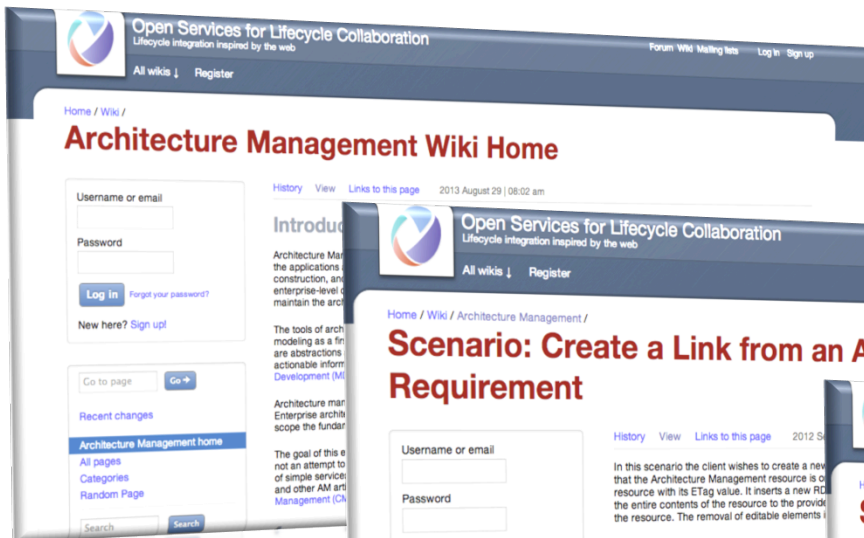


OSLC website at <http://open-services.net>

OSLC: Intersecting Open Standards and Open Source



OSLC Architecture Management



Open Services for Lifecycle Collaboration
Lifecycle integration inspired by the web

Forum Wiki Mailing lists Log in Sign up

All wikis Register

Home / Wiki /

Architecture Management Wiki Home

History View Links to this page 2013 August 29 | 08:02 am

Introduction

Architecture Management is the applications construction, and enterprise-level maintenance of the architecture.

The tools of architecture modeling as a firm are abstractions, actionable information, and development (IM).

Architecture management Enterprise architecture scope the fundamental goal of this effort is not an attempt to provide a simple service, and other AM artifacts Management (CA).

Username or email:

Password:

Log in Forgot your password?

New here? Sign up!

Go to page: Go →

Recent changes

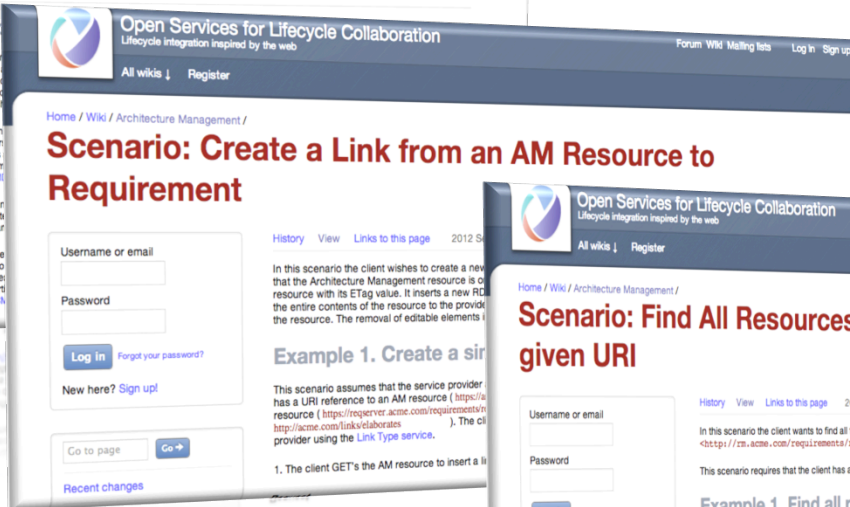
Architecture Management home

All pages

Categories

Random Page

Search: Search



Open Services for Lifecycle Collaboration
Lifecycle integration inspired by the web

Forum Wiki Mailing lists Log in Sign up

All wikis Register

Home / Wiki / Architecture Management /

Scenario: Create a Link from an AM Resource to Requirement

History View Links to this page 2012 September 7 | 01:10 pm

Introduction

In this scenario the client wishes to create a new link from an Architecture Management resource to a Requirement resource with its ETag value. It inserts a new Resource into the entire contents of the resource to provide the link. The removal of editable elements is not required.

Example 1. Create a link

This scenario assumes that the service provider has a URI reference to an AM resource (<https://reqserver.acme.com/requirements/req123>). The client uses the Link Type service to create a link from the AM resource to the Requirement resource using the Link Type service.

1. The client GET's the AM resource to insert a link.

Username or email:

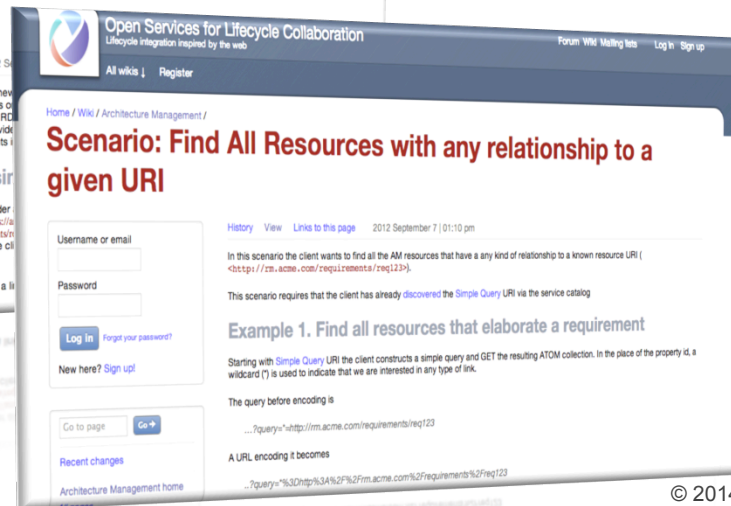
Password:

Log in Forgot your password?

New here? Sign up!

Go to page: Go →

Recent changes



Open Services for Lifecycle Collaboration
Lifecycle integration inspired by the web

Forum Wiki Mailing lists Log in Sign up

All wikis Register

Home / Wiki / Architecture Management /

Scenario: Find All Resources with any relationship to a given URI

History View Links to this page 2012 September 7 | 01:10 pm

Introduction

In this scenario the client wants to find all the AM resources that have a any kind of relationship to a known resource URI (<http://cm.acme.com/requirements/req123>).

This scenario requires that the client has already discovered the Simple Query URI via the service catalog.

Example 1. Find all resources that elaborate a requirement

Starting with Simple Query URI the client constructs a simple query and GET the resulting ATOM collection. In the place of the property id, a wildcard (*) is used to indicate that we are interested in any type of link.

The query before encoding is

```
...?query="http://cm.acme.com/requirements/req123"
```

A URL encoding it becomes

```
...?query="http%3A%2F%2Fcm.acme.com%2Frequirements%2Freq123"
```

Username or email:

Password:

Log in Forgot your password?

New here? Sign up!

Go to page: Go →

Recent changes

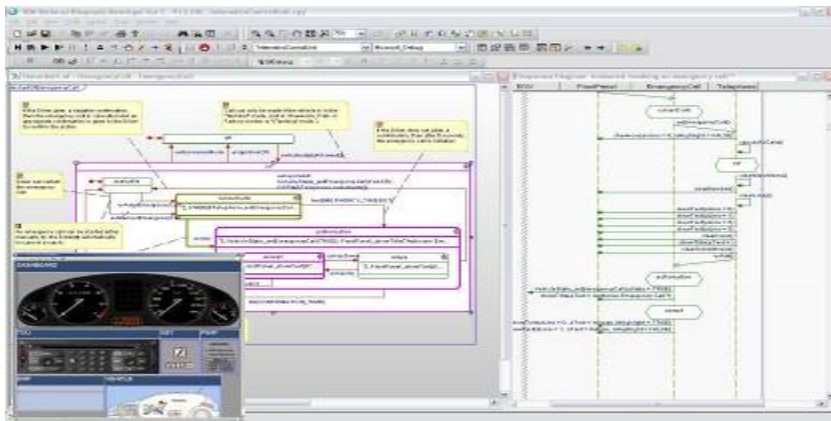
Architecture Management home

Agenda

- Linked Lifecycle Data and OSLC
- Rhapsody and Rhapsody Design manager
- Model Management in the context of Product Line Engineering

Model-Driven Development for Systems & Software

Rational Rhapsody®



Capabilities

- Specify, design and develop systems and software for technical, embedded and real-time solutions, including those based on *multi-core* architectures
- Validate and verify designs with model based simulation and test throughout the product lifecycle
- Develop complete C, C++, Java and Ada applications, working in either the code or model while ensuring the two remain in sync

Benefits

- Build the right product through optimized communication and collaboration
- Eliminate defects early and increase quality by continually testing the design
- Reduce development time by automatically generating applications and documentation

Collaborative Design Management

Enhance cross-team collaboration on software and systems design

Central Design Hub

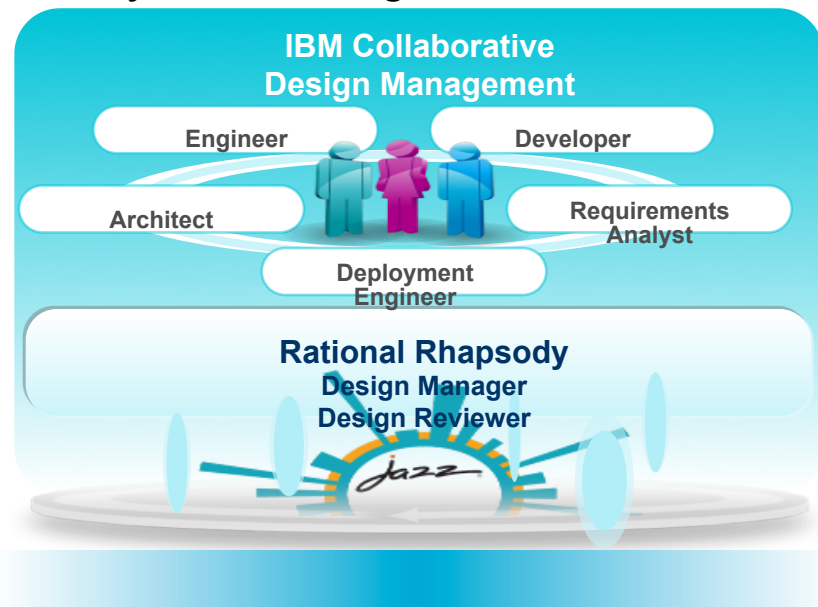
- ✓ *Enterprise-wide design storage for search, review, analysis, and reuse*
- ✓ *Links design elements to lifecycle artifacts*
- ✓ *Navigate and visualize relationships*
- ✓ *Simplify design collaboration through Jazz-based model management*

Stakeholder Collaboration

- ✓ *Automated design reviews at all stages of development*
- ✓ *Intuitive extended team web client for broader access to designs*
- ✓ *Unify requirements and design with single-source of truth workflow utilizing OSLC*

Document Generation and Reporting

- ✓ *Create documents directly from the development lifecycle*
- ✓ *Draw from information and assets linked through OSLC*



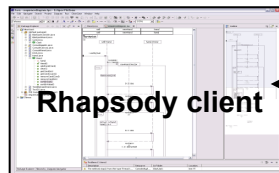
“The ability to review and comment on models from the Web client encourages feedback from a wide array of stakeholders... leading to faster consensus and improved quality of solution designs.”

Server Based Model Management



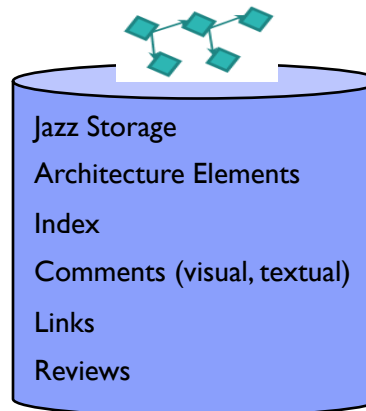
Design change control and versioning (model-based)

Design creation, editing, MDD/MBSE/MBT, search, query, validate, analyze, report



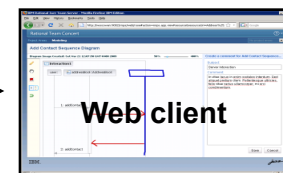
OSLC + DM REST APIs

Design Management services on Jazz Team Server (JTS)



OSLC + DM REST APIs

Design search, query, view, comment, review, link, report validate, analyze, limited editing



Benefits

- Direct editing of designs and change control on server providing a more simplified environment
- Change control (locking, history) at the model resource level providing more granularity
- No duplication or design synchronization issues

Rational Rhapsody Design Manager Web Client

The screenshot displays the Rational Rhapsody Design Manager Web Client interface. On the left, a tree view shows the project structure, including packages like 'Capture Usage Data' and 'WaterMeter'. The main area shows an activity diagram with nodes such as 'wakeUp', 'runHealthChecks', 'decision', 'assessFaults', 'readMeterUsageData', 'recordFaultData', and 'storeMeterUsageData'. A red box labeled 'REQ' is drawn over the 'assessFaults' node. A red oval highlights the 'readMeterUsageData' node and its associated 'WaterMeter' component. On the right, a 'Comments' panel lists several review comments with timestamps and user names like 'Sarah Reviewer' and 'Bill Lee'. The browser's address bar shows the URL: [AMR_System] FunctionalAnalysisPkg::CaptureUsageDataPkg::Capture Usage Data::Capture Usage DataBlackBoxView::activity_0::ActivityDiagram - Design Management - Mozilla Firefox: IBM Edition.

View design over web

Collaborate with stakeholders with commenting

Browse design information

Mark-up diagrams to elaborate comments

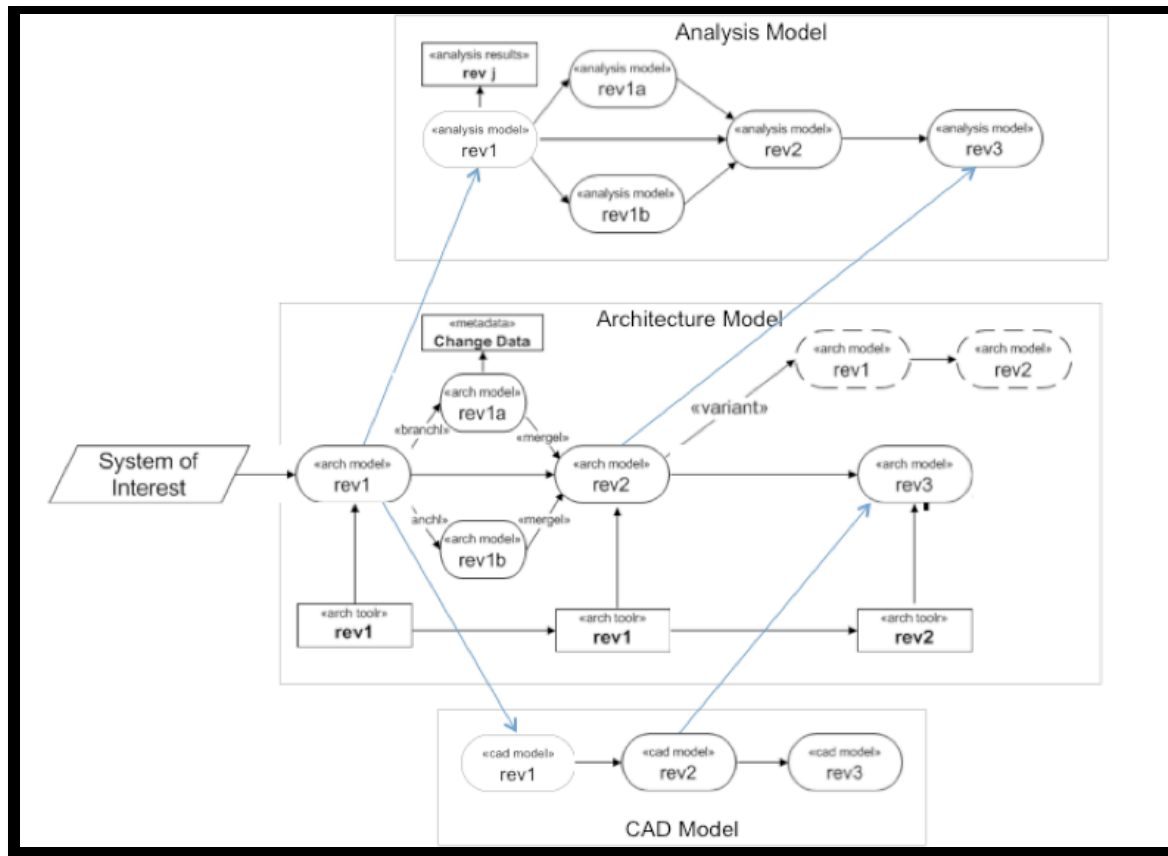
Model Integration: Rhapsody and The Mathworks Simulink

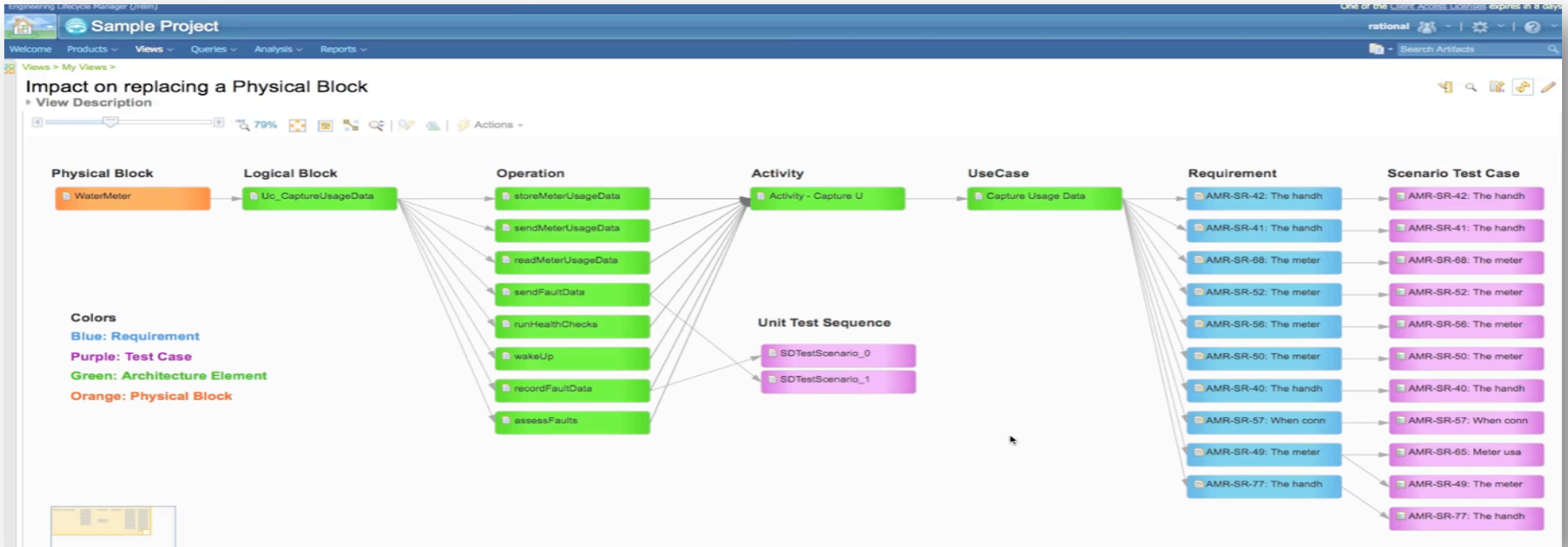
- Collaborate on multiple IBM Rational Rhapsody and/or The Mathworks Simulink projects using the Jazz platform
- Breakthrough combination of domain-specific design solutions enabling
 - ▶ Look at the system as a whole and not individual parts across the architecture, algorithms and environment
 - ▶ View Simulink design information with a web browser or Rational Rhapsody desk top client providing easy access for project stakeholders
 - ▶ Search and navigate across projects and lifecycle artifacts to help find key information from multiple sources quickly
 - ▶ Trace, via Open Services for Lifecycle Collaboration (OSLC), from designs to other lifecycle artifacts such as requirements, test cases, work items or other aspects of the engineering process



Simulink design information can be viewed, shared, marked up, reviewed and linked using a web client through Rhapsody Design Manager

From the INCOSE IS 2014 Model Management paper

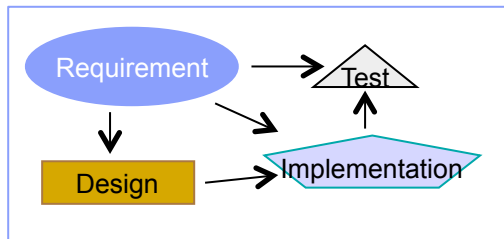




Agenda

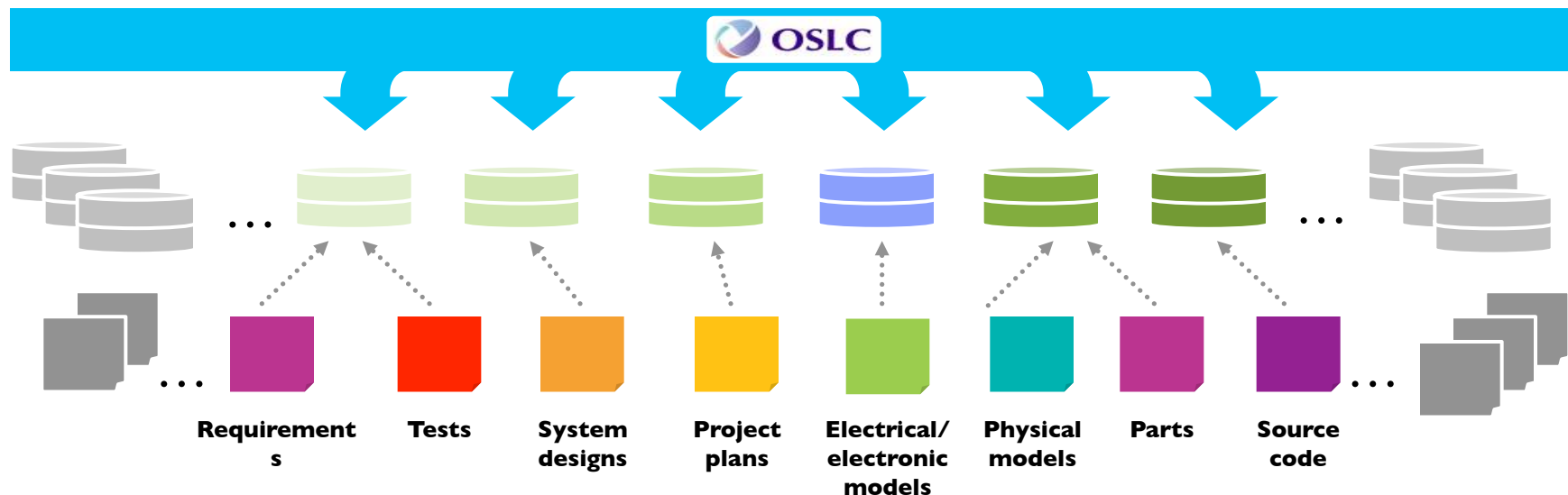
- Linked Lifecycle Data and OSLC
- Rhapsody and Rhapsody Design manager
- **Model Management in the context of Product Line Engineering**

Models are part of a web of related engineering artifacts

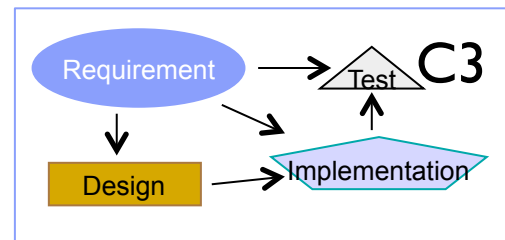
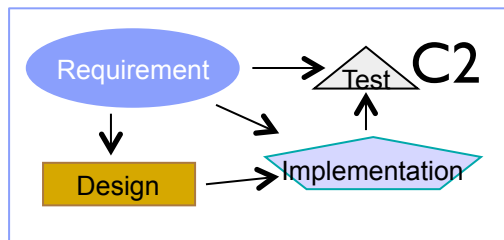
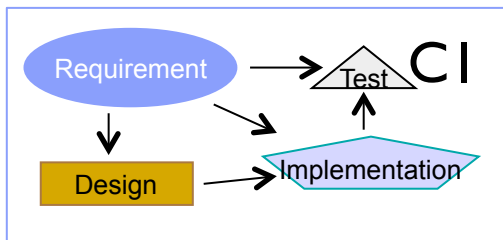


These artifacts may not live in the same tool or database

In fact, in faithfulness to the real world,
we should assume they do not



These sets of engineering artifacts can be constructed as reusable components in a product line catalog



Where products consist of ...
 Different combinations of components
 Components at different versions

What do we mean by Product Line Engineering?

Maximizing reuse of engineering assets across product variants

- Reuse architecture based on configurable components
- Avoid duplication -> maintenance nightmares across hundreds of product variants

Specifying a product definition

- The product consists of which components [sets of versioned engineering artifacts]?

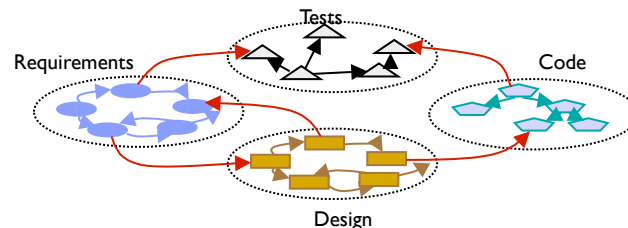
Consistently manage artifact versions and product variants across all lifecycle disciplines

- Create cross-component, cross-tool baselines
- Query and do analysis in multi-version/multi-variant environments

Effectively handling change propagation to a multitude of variants

- “Where does this change need to go?”

Effectively creating new product variant based on functional parameters (“features”)



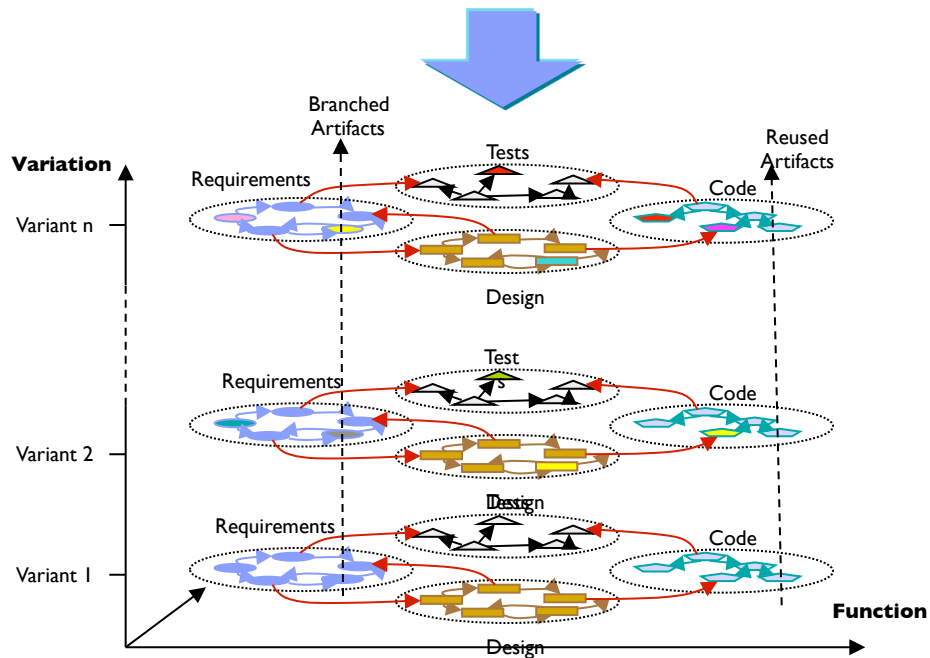
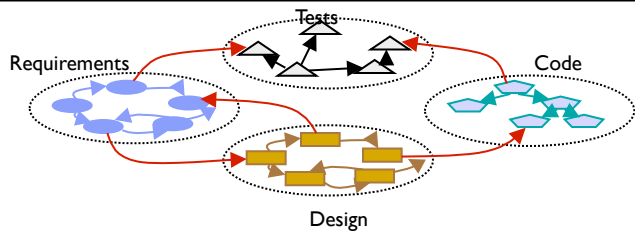
More automation – less manual bookkeeping

- **Tools manage the product definition – not spreadsheets that you manually maintain**
- **Tools present to the engineers the right artifacts at the right versions – and the right links between artifacts**

From “single system” to “multi-variant system”

Some key capabilities

1. **Configuration management** of *requirements* and across engineering domains
2. **Parallel development** of more than code
3. **Hierarchical, global baselines**
4. **Definition of products** as sets of reusable “components” containing the respective engineering artifacts
5. **Automation** to select product variant’s components *via parameters* or feature model
6. Automation to enable *parameters* defined in product definition to be **used in component’s engineering artifacts** (requirements, designs, tests, software, ...)



The multi-dimensions of product line development: temporal and functional dimensions

