Living Architectures - from eclipse to jazz

Erich Gamma
IBM Distinguished Engineer
IBM Rational Zurich Research Lab
Outline

- First assignment
  - A tools platform – Eclipse

- Second assignment
  - A tools integration platform – Jazz

- Comparison and Conclusion
First Assignment: Eclipse

- A tools platform
  - Seamless integration
  - Easy to extend
  - Scalable to many extensions
  - Java APIs
Eclipse Architecture Layers: how buildings last

- **Stewart Brand: how buildings learn** – what happens after they're built
  - stuff: furniture
  - services: electrical, plumbing (7-15y)
  - structure: foundation, load bearing walls (30-300y)
  - site: geographical setting (forever)

- **layers:**
  - evolve at different rates during the life of a building
  - shear against each other as they change at different rates
  - an adaptive building must allow slippage
  - a building that lasts is adaptive and can change over time
structure foundation

- the eclipse plug-in architecture
- everything is a plug-in
  - simple and consistent
eclipse plug-in architecture

- **plug-in == component**
  - set of contributions
  - smallest unit of Eclipse function
  - details spelled out in plug-in manifest
- **extension point** – named entity for collecting contributions
- **extension** – a contribution
  - Example: a specific spam filter tool
- **runtime** – controls and manages contributions
scalability

user visible appearance

<action
  toolbarPath="search"
  icon="icons/opentype.gif"
  toolTip="Open Type"
  class="org.eclipse.jdt.OpenTypeAction"/>

lazily instantiated using reflection

org/eclipse/jdt/OpenTypeAction.class

Declarative Definition (manifest)

Procedural Implementation (Java JAR)

contribution implementation
services plumbing: APIs

- Plug-in dependencies through APIs

- But… APIs don’t just happen; we need to design them
  - specifications with precisely defined behavior
    - what you can assume (and what you cannot)
    - it works ≠ API compliant
    - documented classes ≠ API

- Must not break existing clients when evolving APIs
API Stability Techniques

- compatibility layer
- eclipse extension interface support: IAdaptable
- I*2 extensions interfaces
- restart in a new name space/package
extension interfaces: IAdaptable

- adding interfaces to existing types
- Interface negotiation

```xml
<extension point="org.eclipse.core.runtime.adapters">
  <factory>
    <class>org.eclipse.jdt.internal.ui.JavaElementAdapterFactory</class>
    <adaptableType>org.eclipse.jdt.core.IJavaElement</adaptableType>
    <adapter type="org.eclipse.ui.IPersistableElement"/>
  </factory>
  ...
</extension>
```
I*2 extension interfaces

- add new methods in extending API interface with extension interfaces
  - avoids breaking existing implementors of an interface

```java
public interface IActionDelegate { … } // original interface

public interface IActionDelegate2 extends IActionDelegate {
    void dispose();
}
```

```java
if (d instanceof IActionDelegate2) {
    IActionDelegate2 d2 = (IActionDelegate2) d;
    d2.dispose(); // call new method
}
```
Key Lessons

- APIs are a huge commitment
  - the tyranny of stable APIs
    - API layers
    - I*2… I*7

- Version challenge for product developers
  - which API level does our product require and support
    - n–1, n-2
  - Lockstep version upgrades
Next assignment: A Tools **Integration** Platform

- **Common goal**
  - Rich integration - loose coupling

- **New goal**
  - Avoid lockstep version upgrade
  - Independent upgrade - customers must be able to upgrade their products **one at a time** in the order of their choice
Traditional Tools Integration
Traditional Tools Integration…

- **Point-to-point integrations**
  - Limited coverage: there are too many tools to cover more than a small fraction of possibilities
  - Tight dependencies between tools require lockstep upgrades
  - Proprietary APIs create vendor lock-in

- **State of the Art: shared repository**
  - Hard to add existing (legacy) tools
  - Difficult to evolve tools individually
  - Limited to a single vendor’s tools or affiliates
Enter “Linked Data”

- Linked Data is an approach, defined by Tim Berners-Lee, to data integration on the Web
  - [http://www.w3.org/DesignIssues/LinkedData.html](http://www.w3.org/DesignIssues/LinkedData.html)

- Linked data principles
  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 the standards (RDF, SPARQL).
  4. Include links to other URIs so that they can discover more things.
Linked Lifecycle Data
Finding and Analyzing Linked Lifecycle Data
From Linked Data to an Integration Toolbox

- UI mash-ups – UI integration
  - Provide and consume Open Social gadgets
- Linked data – Data integration
  - Open services for life cycle collaboration
    - Creating linked data
    - Delegated UIs
    - Exploring linked data
    - UI previews
- Leverage Jazz Foundation integration services
- Utilize Jazz Foundation functional services
Linked Data is not sufficient – Integration Toolbox

- UI mash-ups
  - Provide and consume Open Social gadgets
- Linked data
  - Open services for life cycle collaboration
    - Creating linked data
    - Delegated UIs
    - Exploring linked data
    - UI previews
- Leverage Jazz Foundation integration services
- Utilize Jazz Foundation functional services
OpenSocial defines a common API for social applications across multiple websites. With standard JavaScript and HTML, developers can create apps that can embed and be embedded within a social network itself, or access a social network's data from anywhere on the web.

- We focus on the mash up part
- You can provide new or consume existing gadgets
Open Social Gadgets

Rational Team Concert

Erich Gamma's Dashboard

Open Social Gadget

Remember The Milk

Add Task

Overdue

Mar 5  Make Build Input
May 2  make hotel reservation

Today

No tasks due today.

News Feed (82 new)

[2] Provide a native iPhone App for RTC/Work Items (113728) 10 minutes ago
[5] Import and Export from MS Project (26076) 1 hour ago
[16] Provide development environment guidelines and tools (113355) 5 hours ago
[9] Track C/ALM 2010 M6 build and install (May 1 - May 7) (113651) 8 hours ago

Page 1 of 11

Pending approvals for me (2)

101461: Don't deploy the deprecated Scrum template on new servers
66070: Template Work Item with Blue Prints for Approvals

Open assigned to me (3)

93642: Help mapping of MCIF practices into RTC
82012: Consider to rename resolution Won't Fix to Works as Designed on jazz.net
62601: RTC 2.0 plan approval tracking

Current Plans for Erich Gamma (10)

Team Concert iFix3
C/ALM 2010 *** Deprecated ***
Team Concert Developer
Work Item
Open Social Gadgets

Work items gadget in Open Social Containers (iGoogle, gmail)
Linked Data is not sufficient – Integration Toolbox

- UI mash-ups
  - Provide and consume Open Social gadgets
- Linked data
  - Open Services for Life Cycle collaboration (OSLC)
    - Creating linked data
      - Delegated UIs
    - Exploring linked data
      - UI previews
- Leverage Jazz Foundation integration services
- Utilize Jazz Foundation functional services
Open Services for Lifecycle Collaboration OSLC

- Applies Linked Data principles to Lifecycle artifacts
  - Provides specifications for sharing lifecycle data
- A minimalist scenario driven approach
- Effort is divided into domains
  - Change Management
  - Quality Management
  - Requirements Management
- Builds on a Common core specification
Open Services for Lifecycle Collaboration (also known as OSLC or Open Services) is a community effort to help software delivery teams by making it easier to use lifecycle tools in combination. The OSLC community is creating open, public descriptions of resources and interfaces for sharing the things that software delivery teams rely on, like change requests, test cases, defects, requirements and user stories.

By agreeing on common specifications for lifecycle resources and the services to access them, we can eliminate traditional barriers between tools and open the door to new forms of collaboration. OSLC can bring value to software delivery teams and tool providers alike, from the most Agile to the most ceremonial of projects, and for commercially-licensed, open source, and internally developed tools. None.

With OSLC’s open and scenario-based approach, businesses benefit from the ability to better integrate tools together. This collaborative approach gives our consultants the flexibility to make lifecycle tool choices based on specific client project demands.

Suppose tools exposed their data in a consistent way?

- Open community of individuals interested in improving lifecycle integration.

- Goals

1. Make life better for software and product delivery teams

2. Reduce the complexity and cost for tool providers in integrating tools together

3. Open up new possibilities in the marketplace by opening up the way lifecycle tools and data can be used in ALM, PLM and outside

- Creating open, public specifications that describe resources and interfaces for sharing the things that software and product delivery teams rely on.
OSLC Core concepts
OSLC Core provides Guidance for

- Resource representations
  - must provide an RDF/XML representations of a resource
  - may provide JSON, Turtle, ATOM
- Link modeling
- Partial updating of resources
- UI Previews
- Delegated UIs
- Specification versioning
**Specification Example: Change Management**

http://open-services.net/bin/view/Main/CmSpecificationV2

---

### Resource URIs and Methods

<table>
<thead>
<tr>
<th>Resource</th>
<th>URI</th>
<th>GET</th>
<th>POST</th>
<th>PUT</th>
<th>DELETE</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection of Change Requests</td>
<td>{CR Collection URI}</td>
<td>Y</td>
<td>*</td>
<td>N</td>
<td>N</td>
<td>A collection of change requests</td>
</tr>
<tr>
<td>Change Request</td>
<td>{CR URI}</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>An identifiable change request, by a permanent URI</td>
</tr>
</tbody>
</table>

* - the collection MAY support creation on its URI, see Create a new Change Request

N - in the HTTP verb column indicates that a Service Provider MUST return a 405 Not Supported response

For a complete list of HTTP Response Codes

---

### URIs for working with Change Requests

The following table outlines the key items that are exposed in the Change Management Service Discovery Document. Details of each of these capabilities will follow in subsequent sections.

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Discovery Element</th>
<th>URL*</th>
<th>Section</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Creation</td>
<td>&lt;factory&gt;</td>
<td>{Resource Creation URL}</td>
<td>Create a new Change Request</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>Resource Query</td>
<td>&lt;simpleQuery&gt;</td>
<td>{Simple Query URL}</td>
<td>Get a Collection of Change Requests</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>Resource Selection UI</td>
<td>&lt;selectionDialog&gt;</td>
<td>{Selection Dialog URL}</td>
<td>Resource Selection</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>Resource Creation UI</td>
<td>&lt;creationDialog&gt;</td>
<td>{Creation Dialog URL}</td>
<td>Resource Creation</td>
<td>REQUIRED</td>
</tr>
</tbody>
</table>
Retrieving a Defect

Accept: application/x-oslc-cm-change-request+xml

<?xml version="1.0" encoding="UTF-8"?>
<oslc_cm:ChangeRequest
    xmlns:rtc_cm="http://jazz.net/xmlns/prod/jazz/rtc/cm/1.0/" xmlns:oslc_disc="http://open-services.net/xmlns/disc
    xmlns:jp="http://jazz.net/xmlns/prod/jazz/presentation/1.0/" xmlns:jd="http://jazz.net/xmlns/prod/jazz/discover
    xmlns:oslc_cm="http://open-services.net/xmlns/cm/1.0/" xmlns:atom="http://www.w3.org/2005/Atom
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:calm="http://jazz.net/xmlns/prod/jazz/calm/1.0/">
    <dc:type rdf:resource="https://rtc:9443/jazz/oslc/types/_gasc4Ju-Ed6cerS91b5AWw/defect"/>
    <dc:identifier>9</dc:identifier>
    <dc:created>2009-09-07T14:59:06.333Z</dc:created>
    <dc:creator rdf:resource="https://rtc:9443/jazz/oslc/users/_6I82MJu9Ed6cerS91b5AWw"/>
    <dc:title>My First Bug</dc:title>
    <dc:description>This is my first bug</dc:description>
    <dc:subject/>
    <dc:modified>2009-09-07T14:59:06.348Z</dc:modified>
    <oslc_cm:severity rdf:resource="https://rtc:9443/jazz/oslc/enumerations/_gasc4Ju-Ed6cerS91b5AWw/severity/severi"
Retrieving JSON Representation of a Defect


```json
{
  "dc:title":"My First Bug",
  "rdf:resource":"https://\!/\!/rtc:9443\!/\!/jazz\!/\!/resource\!/\!/itemOid\!/\!/com.ibm.team.workitem.WorkItem\!/\!/_0J39QJu-Ed6cerS91b5AWw",
  "rtc_cm:comments": [
    {
      "rdf:resource":"https://\!/\!/rtc:9443\!/\!/jazz\!/\!/oslc\!/\!/workitems\!/\!/_0J39QJu-Ed6cerS91b5AWw\!/\!/rtc_cm:comments\!/\!/0"
    },
    {
      "rdf:resource":"https://\!/\!/rtc:9443\!/\!/jazz\!/\!/oslc\!/\!/workitems\!/\!/_0J39QJu-Ed6cerS91b5AWw\!/\!/rtc_cm:comments\!/\!/1"
    }
  ]

  "rtc_cm:com.ibm.team.workitem.linktype.relatedworkitem.related": [
    {
      "rdf:resource":"https://\!/\!/rtc:9443\!/\!/jazz\!/\!/resource\!/\!/itemOid\!/\!/com.ibm.team.workitem.WorkItem\!/\!/_CD62QJu_Ed6cerS91b5AWw",
      "oslc_cm:label":"10: My Second Bug"
    },
    {
      "rdf:resource":"https://\!/\!/rtc:9443\!/\!/jazz\!/\!/resource\!/\!/itemOid\!/\!/com.ibm.team.workitem.WorkItem\!/\!/_7Z5cAJsCTEd6cerS91b5AWw",
      "oslc_cm:label":"11: My Third Bug"
    }
  ]
}
```
Linking

- Allow to establish relationships between your resources and resources provided by others
  - Consume resource pickers provided by others
  - Support OLSC linking protocols to establish links
  - Support resource pickers that can be consumed by others

picker dialog:

creation dialog:
Delegated UI Dialogs
For resource creation and selection

1. UI Consumer gets UI Provider's Service Resource, finds Dialog URI
2. UI Consumer embeds a Dialog from UI Provider via <iframe>
3. UI Provider informs UI Consumer of response in browser
4. User creates or selects resource(s) via embedded Dialog
Delegated UIs: Embedded Editors

- Using Open Social for composing UIs
UI Preview (was Compact Rendering)

- Allows to show information about linked resources
UI Preview

- GET with Accept header application/x-oslc-compact+xml

- Return is an RDF/XML document, return media type is application/x-oslc-compact+xml

- http://open-services.net/bin/view/Main/OsIcCoreUiPreview

```xml
<?xml version="1.0" encoding="UTF-8"?>

<oslc:Compact
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
   xmlns:dc="http://purl.org/dc/terms/
   xmlns:oslc="http://open-services.net/xmlns/oslc#"
   rdf:about="http://example.com/bugs/12345">

   <dc:title> 12345: &lt;s&gt;Null pointer exception during startup&lt;/s&gt; </dc:title>
   <dc:name> 12345 </dc:name>
   <oslc:icon rdf:resource="http://example.com/icons/defect.jpg" />

   <oslc:smallPreview>
     <oslc:Preview>
     </oslc:Preview>
   </oslc:smallPreview>

   <oslc:largePreview>
     <oslc:Preview>
       <oslc:hintWidth> 60em </oslc:hintWidth>
       <oslc:hintHeight> 20em </oslc:hintHeight>
     </oslc:Preview>
   </oslc:largePreview>

</oslc:Compact>
```
Service Discovery

1. Discover the existence of the Change Management system itself, known URL
   - E.g. https://rtc:9443/rtc/rootservices

2. Discover the contexts (e.g. projects) in which change requests may exist, e.g. project

3. Discover the services that are provided within that context

```
<rdf:Description rdf:about="https://rtc:9443/jazz/rootservices">
    ...
    <oslc_cm:cmServiceProviders rdf:resource="https://rtc:9443/jazz/oslc/workitems/catalog"/>
    ...
</rdf:Description>
```
Discovering the Creation Dialog

Service Providers Catalog
<oslc_disc:services rdf:resource=""/>  
<oslc_disc:services rdf:resource=""/>

Service Descriptors of Project A
<oslc_cm:...

Service Descriptors of Project B
<oslc_cm:creationDialog oslc_cm:hintWidth="740px"...>
  <dc:title>New Defect</dc:title>
  <oslc_cm:url></oslc_cm:url>
</oslc_cm:creationDialog>
<oslc_cm:factory oslc_cm:default="true">
  <dc:title>Creation of change requests</dc:title>
  <oslc_cm:url></oslc_cm:url>
</oslc_cm:factory>
<oslc_cm:simpleQuery>
  <dc:title>CQL based change request queries</dc:title>
  <oslc_cm:url></oslc_cm:url>
</oslc_cm:simpleQuery>
Jazz is a platform for transforming software delivery

Jazz is...

- Our vision of the future of systems and software delivery
- A scalable, extensible team collaboration platform
- An integration architecture enabling mashups and non-Jazz products to participate
- A community at Jazz.net where Jazz products are built

Jazz is a platform for *transforming how people work together* to deliver greater value and performance from their software investments.
Collaborative ALM

Rational DOORS Requirements Professional

Requirements project
- Requirements
- Requirements Collections

Rational Team Concert

Development Project
- Work items
- Plans
- Streams, Change Sets
- Builds

Rational Quality Manager

Test Project
- Test Cases
- Test Plans
- Test Executions
Eclipse vs. Jazz Integration Architecture

- **Eclipse**
  - a tools platform, implement new tools
  - Java APIs
  - Java
  - Integrate by writing new plug-ins
    - Plugin.xml
    - Factories
  - Desktop application
  - Fine grained integration
  - Independent upgrade difficult

- **Jazz integration architecture**
  - Integrate existing tools
  - REST based specifications
  - Many languages
  - Integrate by providing REST implementations
    - Service documents
    - Discovery, Factory URLs
  - Web (but can integrate with desktop apps)
  - Coarse grained integration
  - Supports independent upgrade
References

- OSLC
  - www.open-services.net
- Jazz.net
  - www.jazz.net