Solving Product Configuration and Traceability Challenges by Leveraging OSLC and the Jazz Platform
Agenda

Global Product Data Interoperability Summit | 2016

• Background – What does OSLC and the Jazz platform provide for development activities
• Motivations for Apache Systems Engineering
• Platform Enablers
• Change Management Process
• Lessons Learned and Next Steps
• Summary
No unified systems/software development platform leads to significant challenges with the following …
Open Services for Lifecycle Collaboration (OSLC) and Jazz

OSLC

- a community of software developers and organizations working to standardize the way that software lifecycle tools share data (for example, requirements, defects, test cases, plans, or code) with one another.
- provides open specifications for service descriptions and cross-tool communication ([http://open-services.net/bin/view/Main/WebHome](http://open-services.net/bin/view/Main/WebHome))
  - Under formal control by OASIS stds group.

Jazz

- An open collaboration platform designed to support systems and software tool integrations
  - Integration architecture that allows various tool components to be configured together
- A set of products that support this platform.
Tool Environment Leveraging the Jazz Framework
Jazz Investigation and Deployment Roadmap…

2012-13

- Define targeted toolset for Integrated Solution
- Research integrations and capabilities and supported workflows
- Develop User Documentation and Guidelines
- Engage vendors and the OSLC community
- Deploy sandbox environment testbed
- Outreach and disseminate information

2014

- Deploy environment for early adopters
- Engage and support early adopters
- OSLC StC Membership
- Mature technology by aggressively engaging vendors

2015

- Two Jazz Production Platforms brought online
- Focused Project Support and Engagement
- Design Manager Basic Training Course (LTD)
- Increased involvement in OSLC Specifications

2016

- Focus on helping DES team to investigate DOORS Next Generation and to help programs get started with DNG
- Leverage Jazz Global Configuration Control across tools in support of project PLE requirements
- Work with BCA MBSA project to integrate Jazz platform with Mentor Graphics Context SDM solution in support of cross-discipline use cases
Agenda

Global Product Data Interoperability Summit | 2016

• Background – What does OSLC and the Jazz platform provide for development activities
• Motivations for Apache Systems Engineering
• Platform Enablers
• Change Management Process
• Lessons Learned and Next Steps
• Summary
Apache Program Relationships

- Child program(s) spawn from a single parent program baseline
  - Designs are reused from parent to child programs where possible
- Child Programs may require a subset of content from the parent baseline
  - Include unique content from parent program
  - Require different access controls than parent program
Need for Change Management

**Problem:**
- Reuse of functional architecture models across programs requires change management process, including traceability.

**Plan:**
- Leverage established tools/processes
- Develop documented process to authorize architecture changes
- Trace changes to impacted model elements
How to Trace Design to Requirements?

- Traceability between subsystem and system level solved:
  - Requirements: managed in DOORS
  - Design: Managed in Rhapsody
  - Requirements <-> Design: ???

DOORS 9.X

Rhapsody

System Requirements Database (DOORS)

Trace (Activities -> Sys Reqt)

Subsystem Requirements Database (DOORS)

Trace (SubSys Reqt -> Sys Reqt)

System Design (SysML)

Allocation (Activities -> Subsys Block)

Subsystem Design (SysML)

Trace (Activities -> SubSys Reqt)
Goal is to Leverage Existing Tools.....

- **DOORS**
  - Stand-alone Requirements Management tool
- **Rhapsody**
  - Client Architecture Development app
…..and New Capabilities to meet needs

- **DOORS**
  - Stand-alone Requirements Management tool
- **Rhapsody**
  - Client Architecture Development app
- **DOORS Web Access (DWA)**
  - Web based server add-on for DOORS. Supports OSLC
- **Design Manager**
- **Team Concert (RTC)**
  - Jazz offering providing Change Management, workflow planning, interface to other Rational tools.
Resulting Integrated Tool Environment Benefits

- **Data Connectivity between Tools**
  - Enabled by OSLC linking
  - Improved traceability
  - Access controls managed by tool per project

- **Jazz platform is scalable** – can add tools for new capabilities

- Data Connectivity
  - Between Tools
  - Enabled by OSLC linking
  - Improved traceability
  - Access controls managed by tool per project

- Jazz platform is scalable – can add tools for new capabilities
• Background – What does OSLC and the Jazz platform provide for development activities
• Motivations for Apache Systems Engineering
• Platform Enablers
• Change Management Process
• Lessons Learned and Next Steps
• Summary
Rhapsody Design Manager Features

- System-wide Repository for Design Reuse and Cross-Domain Querying
- Linking and Tracing of modeling artifacts to other lifecycle artifacts and also between systems and software architecture/design
- Manage Design Configuration directly on DM Server
- Document Generation and Reporting across all OSLC assets
- Analyze Data (relationships and dependencies)
- Collaboration (including reviews and approvals)

Rhapsody Design Manager"
Traceability

You can create links between artifacts from Rhapsody or from the DM web interface (shown here).

1. You choose the appropriate link type from the available links which have been defined by the Project Administrator on your Project. The example is representative.
2. From the pulldown you will find a number of possible source repositories that you can use as the target.
3. In this example, you could select from Lifecycle Project Testing (Requirements).
Traceability (cont.)

4. Then using the interface, navigate and find the requirement(s) you wish to select for the link. Select them and click OK.

5. The resulting hyperlink will be reflected on the Links tab.
Traceability can be established by linking model artifacts to many different artifacts in other lifecycle tools or in other Design Management models. Some examples of linked data include IBM ClearQuest, DOORS, SysML, and external documents.

- IBM ClearQuest
- SysML model blocks
- External documents
- DOORS (/dwa)

Show that this diagram elaborates an RTC work item task.

Show that this diagram satisfies the following requirements to DOORS.

Show that this diagram is derived from another requirements to 'Fly UAV'.
Model Management on the Jazz Platform

• Actively Managed Model

When your Rhapsody/DM project is actively managed, you can move Rhapsody models to the Design Management Server or create new models directly on the server and open them from Rational Rhapsody.

A Rhapsody user can open a model on Design Manager, make changes to the model in Rhapsody, and save the changes back to the Design Management Server.

Other users, reviewers, or customers who may access the model through the DM web client, will get the latest updates to the model that have been shared by the development team.
**Change Management**

Global Product Data Interoperability Summit | 2016

- **Change Sets**
  - Use change sets to group related sets of changes. To make the changes in the change set visible to other users, you must *share* them.

  1. The changes to the Context Decomposition Diagram are now part of the Project Area.
  2. All Users can see the modified diagram.
  3. Any of the users can further modify it.

---

**Change Set Dialog**

- **Changeset Name**: [Changeset for Rhapsody Model Traffic Light8 341]
- **Status**: Active
- **[Traffic Light8] Level0:Context Decomposition**: Locked

**Level0 Diagram**

- **Block Definition Diagrams**: Context Decomposition, Context Diagram, External Interfaces Block Diagram, Initial Context Diagram, Interfaces

---

*Elysium*  
*[Boeing]*  
*Northrop Grumman*  
*[Parker]*

Global Product Data Interoperability Summit 2016

BOEING is a trademark of Boeing Management Company.  
Copyright © 2016 Boeing. All rights reserved.  
Copyright © 2014 Northrop Grumman Corporation. All rights reserved.
Change Management – Multiple Change Sets

- Multiple change sets can be active at once

1. Change set 341 has been shared with the Project Area
2. That change set modified the Context Decomposition Diagram.
Collaboration

- Improved workflows and lifecycle management
- Collaboration (including reviews and approvals)
Comments are viewable from the Rhapsody Client
Configuration Management on the Jazz Platform

- **Configurations (Streams and Baselines)**
  - A *stream* is a mutable working set of artifacts that make up a particular version of the model.
  - A baseline is an immutable (snapshot) set of the artifacts that made up the model at a particular point in time.

Stream contains all artifacts for an initial product

Product variant stream

baseline
• Background – What does OSLC and the Jazz platform provide for development activities
• Motivations for Apache Systems Engineering
• Platform Enablers
• Change Management Process
• Lessons Learned and Next Steps
• Summary
- Developed change management process flow
  - Leverages selected tools
  - Manages FA changes using records and approvals
RTC for Change Management

- Provides customized change request form and workflow options
- Customized Apache work item workflow

LEGEND
- Action state
- Review state

Ready for SERB Approval to Implement
Ready for SERB Approval to Close
Close
Deferred
Implement
Assign
Defer
Initialize

SERB Review to Implement
SERB Review to Close
RTC: Change Request Form

- **Change Request State**
- **Magnitude of the change**
- **How urgent is the change?**
- **Which program(s) are impacted?**
- **Which product(s) are impacted?**
- **Team member assigned to work item**
- **Auto-populated fields based on RTC project properties**
- **Program phase the change was identified**
- **Free text description of the change details**
RTC: Change Request Form - Linking

OSLC linked impacted requirements

OSLC linked impacted functional architecture elements
RTC: Change Request Form Fields

- **Type, Creation Date, Created By, Project Area, and Team Area:**
  These fields are predefined and will already be populated when the CR is initiated. The Project Area is the RTC project containing the change.

- **Owned By:**
  Select the name of the person who is responsible for implementing the requested change.

- **Severity:**
  Select Minor, Normal, or Major to indicate the importance of the change.

- **Priority:**
  Select Low, Medium, or High to indicate how urgent the change needs to be incorporated.

- **Program:**
  Check the box of the program that this requested change will impact. Since each CR is written for a specific program, only one box should be checked here.

- **Work Product:**
  This is the area against which the change is being recorded. Check only one box here since each CR is written for one specific work product.

- **Phase:**
  Select System Design, Subsystem Design, Detailed Design, Production, Integration, or Verification. This is the phase in the lifecycle of the project, and only one box should be checked here.

- **Description:**
  Enter a detailed description of the change being requested.
Design Manager: Changesets

- Automatically created upon editing the model
- Accumulate changes over multiple sessions
- Each Changeset can have exactly one user (owner)
- Users may own one or more Changesets
Design Manager: Changeset Linking

• Design Manager can link Changesets to Change Request
  • Provides Traceability
  • ‘Link to Work Item’: changeset option in Rhapsody model
Design Manager: Peer Reviews

- Peer Review of Model Edits
  - Requires Change Request to be Approved to Implement
  - Model Elements have been updated

- Design Manager allows the peer review owner to specify a Changeset to review
  - Automatically populates model elements
  - Allows other users to view pending changes
Design Manager: Peer Review Form

1. Log into DM
2. Choose project area
3. Create new review
4. Assign review title
5. Set due date
6. Choose changeset
7. Provide instructions for reviewers
8. Select reviewers
9. This populates with artifacts when the changeset is selected
10. Save
RTC: Change Request Closure

• Two possible outcomes for Change Request submitted to SERB for closure:

**Approved:** CR state advanced to “CLOSED”
  - Changeset is incorporated and model is updated

**Rejected:** CR state is reverted to “Implement”
  – The CR owner must complete the Implementation process correctly
    – make any adjustments requested
  – Resubmit CR to SERB for closure review
Agenda

• Background – What does OSLC and the Jazz platform provide for development activities
• Motivations for Apache Systems Engineering
• Platform Enablers
• Change Management Process
• Lessons Learned and Next Steps
• Summary
Advantages of Change Process & Tools

- Process provides consistency for changes made to the functional architecture model
- Electronic history of changes made to functional architecture database
- OSLC enables creating traceability from a change request record to impacted model elements/requirements
- Opportunity for proposed changes to be reviewed and authorized before implementing
- In-Process changes are hidden from other users to avoid using incorrect or incomplete data
- Web based tools support geographically diverse teams working on common model
Lessons Learned

• Process
  – Keep Change Request form and process steps minimal
  – Leverage existing processes for managing changes to other work products to help team adapt to new process for managing functional architecture

• Tools
  – Need to deliver changes from Rhapsody rather than the DM client in order to avoid inconsistencies and to take advantage of the DiffMerge capability for conflicting changes.
  – Access Control Consideration
    – Each project area is completely readable by all users
    – In 6.0.2
      • Will be able to define teams with different write access to different parts of the model
      • Will be able to assign specific permissions to users to deliver changes with and across streams.

  – Consider separation of project data (can you use one RTC project to support all DM projects?)
Next Steps

- **Transition from DOORS 9.X + DWA to DOORS Next generation (DNG)**
  - Seamless requirements change process transition when DNG switchover occurs
  - Addresses limitations when linking from DOORS object to Rhapsody model currently experienced in DOORS + DWA

- **Design Manager 6.0.2**
  - Will be able to define teams with different write access to different parts of the model (addresses access control lesson learned on previous slide)
  - Will be able to assign specific permissions to users to deliver changes with and across streams.
  - Expand use of parallel configurations (streams and baselines) to manage work and develop PL assets

- **Organize multi-discipline models (i.e. testing and software engineering) to expand change process and traceability (next slide)**
Next Steps (Cont’d): Model Management for Multiple Disciplines

System Requirements Database (DOORS)

Subsystem Requirements Database (DOORS)

Software Requirements Database

System Design (SysML)

Subsystem Design (SysML)

Software Design (UML)

Software Test Model (UML)

Trace
(Activities -> Sys Reqt)

Trace
(SubSys Reqt -> Sys Reqt)

Trace
(Activities -> SubSys Reqt)

Trace
(SW reqt -> SubSys Reqt)

Trace
(Seq Dgm -> SW reqt)

Trace
(Test Cases -> SW reqt)

Trace
(Test Cases -> Design)

Trace
(Activities -> SubSys Block)

Trace
(Activities -> Partition Block)

Trace
(System Reqt -> SubSys Reqt)

Trace
(SubSys Reqt -> System Reqt)

Trace
(Software Design (UML))
Agenda

- Background – What does OSLC and the Jazz platform provide for development activities
- Motivations for Apache Systems Engineering
- Platform Enablers
- Change Management Process
- Lessons Learned and Next Steps
- Summary
Summary

Projects adopting this platform and approach are addressing some of the key integration challenges mentioned at the outset using open interoperability standards on production Jazz development platforms currently available to Boeing engineering teams.