OSLC PLM Workgroup

Working meeting

May 25th 2010

open-services.net

V0.3

Organisers today

- Workgroup lead: Rainer Ersch, Siemens
- Coordinator: Gray Bachelor, IBM

Today’s agenda

- Roll call and brief introductions- welcome new members
- Objective for today's meeting – Scenario analysis
- Detailed analysis of the workflow of Systems Engineering scenario link
- Update on the meeting at Innovate 2010 June
- Proposal for next meetings
- Any other business
- Next steps
- Close

Introduction

May 11th Attendees

- Rainer Ersch
- Gray Bachelor
- Mike Loeffler
- Charles Krueger
- Brenda Ellis
- Scott Bosworth
- Steve Speicher
- Pascal Vera
- Kieran McEnery

Roll call

- Please briefly state your name, organisation, role

- If you are in the web conference put your details in a chat
  - Send to “All”
- If not please send an email to gray_bachelor@uk.ibm.com

Aims of an OSLC PLM workgroup

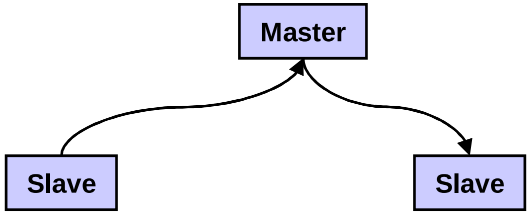
- Evaluate applicability of existing OSLC specifications towards use in an ALM/PLM setting
- Contribute towards extension or new OSLC specifications base upon need or ALM/PLM collaborations

Today’s objectives

1. To conduct a detailed analysis and walk through of the selected Systems Engineering scenario so that we have an Activity diagram and description to support subsequent alignment to OSLC specs

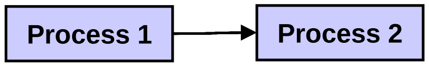
Initial discussion around the IT systems environment topology for the scenario

- Case 1: Master-Slave



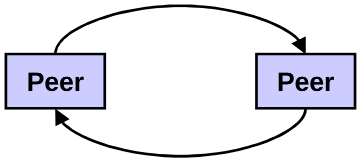
- All / Major CRs are propagated up
- CRs passed “down” only as required
- The Master is the reference

- Case 2: Segmented by process



- CRs are propagated according to process flow (e.g. discipline or phase)
- Detailed Process capability is tailored to organisation or context

- Case 3: Collaborating



- CRs are shared
- Each Peer is updated to reflect transitions and updates

System Engineering scenario

PLM Scenario 1

Systems Engineering scenario

Systems Engineer Reacts to Changed Requirements

PLM Scenario 1 – outline

1. SE receives input of change to existing product needed by marketing (via Change Request)
2. SE locates existing requirement(s) affected by desired change, either directly in requirements manager or SysML requirements diagram view
3. SE traces affected requirements to determine impacted behavior and physical design artifacts, adding them to CR as part of solution
4. SE adds or removes requirements as necessary
5. SE searches for behaviors and physical designs to meet new and revised requirements based on previous uses and related requirements
6. SE proposes changes of requirements, behaviors and physical designs as solution to CR, while collaboratively, all affected suppliers and downstream engineers analyze, review and approve changes

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

Systems Engineering scenario

Systems Engineer Reacts to Changed Requirements

PLM Scenario 1 – more insight

1. Marketing user (change requestor) opens change request tool (probably connected to enterprise PLM repository) and creates a change request to change the (already released) requirement that defines behavior of the **optional** power window feature on 2012 Ultra platform to add automatic open and close on all six windows. Existing required behavior is automatic open only on driver window. The requirements at this level are likely in the PLM repository.
2. Systems Engineer (SE) receives notification of the change request by email, including a link to the change request.
3. SE opens the change request by clicking the link in the email, this opens a (thin or rich client) portal into the enterprise PLM system that shows many aspects of the data for the Ultra platform
4. SE clicks the link in the change request that points to the requirement, opening the requirement in a SysML? Requirements Diagram editing tool
5. SE creates new working revision of the Ultra platform requirements context so the requirement can be changed, the new revision opens in the requirements editing tool
6. SE locates an existing generic power window feature implementation in a related architecture that matches the change requestor's needs by searching for existing requirements using the requirements editing client or related query tool
7. IFF EXISTS:
    1. SE compares the new implementation with the existing one, specifically looking at differences between requirements decompositions (spanning PLM and ALM), models (probably in ALM only), hardware designs (probably in PLM only) and software source code (in ALM) plus calibrations (probably in PLM)
    2. SE adds the new requirement and implementation to the change request as the solution
    3. SE swaps out the old implementation and replaces it with the new one in the working revision of the context
    4. SE fixes any loose ends left when the new implementation was swapped in, the loose ends (potentially in both PLM and ALM) are added to the CR as solutions
8. ELSE:
    1. SE traces all links from the existing requirement to determine what decomposed requirements (spanning PLM and ALM), behavior models (probably in ALM only) and physical implementations (hardware in PLM, software and/or calibrations in ALM and PLM) are impacted by the change
    2. SE adds all effected items to the change request by linking them
    3. SE modifies each of the impacted items in the context, either swapping it out with another existing one that fits the need, or creating a new one
    4. As the SE is making changes the working context is being updated as edits are saved, thus allowing interested collaborators to view and comment in real time
    5. Any further detailed design changes that the SE identifies cause a detailed CR to the appropriate subject matter expert (recursively), with attached links to the specific items that need to be changed. These changes must be closed out before the complete design can be finished.
    6. As each edited sub-requirement, model and physical design is finished, it is reviewed and approved for release. After all sub-elements are released the next higher assembly is updated with the new released revision if necessary.
9. ENDIFF:
10. SE submits the revised working context for review and approvals
11. After all validations and reviews are signed off the new context is automatically moved to released state, becoming mainstream design intent, with some effectivity statement (effective immediately, effective as of a certain date, a certain build event or lot, etc.)
12. Release of revisions automatically closes out the change request

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

Activity diagram

- We will use the diagram provided by GM
- OSLC_Cases.Jpg attached to
  - http://open-services.net/bin/view/Main/PlmScenariosGm

Topics for discussion on the scenario Activity Diagram

- Are we able to identify a workflow of common interest ?
- Can we capture the essence of the main workflow ?
  - Partitions
  - Activities
  - Business entities
  - Transitions
  - Decisions
  - Forks/joins
  - Guard conditions
- What about conditional flows ?
- Are we clear on terminology ?
- Are we able to identify issues to address ?

Other topics

OSLC-PLM Workgroup Get Together at the Walt Disney Swan and Dolphin Hotel Innovate 2010

We have a room in the conference center at the Dolphin Hotel for

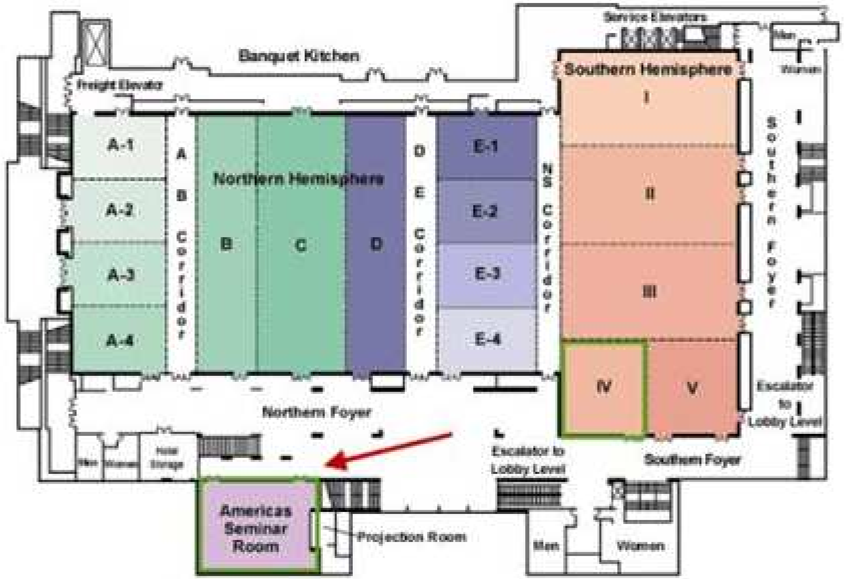
Wednesday, June 9th, 4pm - 5pm+
Americas Seminar Room (see map)

Please join us, meet workgroup members and hear about the work regarding:

Product Lifecycle Management and Application Lifecycle Management Integration

Please note that I will be talking about PLM – ALM Integration the hour before (see info below: **ALM-1633B).**

The talk could be a good start for the discussion in the Get Together session.



Rainer Ersch will also deliver a talk at Innovate

Talk Info:

ALM-1633B : An Open Services (OSLC) Approach to ALM and PLM Integration for Systems Development

*Show details...***Track: Collaborative Application Lifecycle Management**
Speaker(s): Rainer Ersch*, Siemens AG; Pascal Vera, Siemens PLM*

Abstract: For development of complex systems, the integration of mechanical, electrical and software artifacts is critical to success. This talk describes a pragmatic approach to ALM and PLM interoperability based on the OSLC (Open Services for Lifecycle Management) initiative. Together, we'll explore how companies from across multiple industries are coming together through OSLC to integrate tools from ALM vendors with tools from PLM vendors. Come both to learn and to contribute your ideas and expertise.

Session Type: Breakout Session
Level: All Levels (general knowledge)
When: Wed, 9/Jun, 03:00 PM - 04:00 PM
Where: Dolphin - Southern IV

Also at Innovate of potential interest

- Examples of OSLC in action for ALM-PLM
  - The Aero & Defence pedestal in the Industry demo area is hosted by Brian Nolan and will show two scenarios using the CM OSLC 1.0 Spec
    - Integrated Systems Engineering scenario showing field defects driving engineering change back into SW development and subsequent update to reduce lifecycle costs and down time
    - Product Collaboration Portal scenario showing Product Management and Development collaborating around product feature readiness and release for customer responsiveness
  - Bob Kennedy will discuss OSLC in action in his talk CCM 1133B
- Systems Engineering Practices
  - Dan Popescu, Peter Hoffman and Dave Brown will host a Systems Engineering panel
    - Session SDP-2291B
  - Dan will be available to show the IBM Rational Systems Engineering Practice library and preview of enactment in Rational Team Concert
    - Session SDP-2291B

Any other business ?

Summary

Thank you

rainer.ersch@siemens.com

gray_bachelor@uk.ibm.com