Open Services for Lifecycle Collaboration

OSLC PLM Workgroup

Systems Engineering Scenario #1

Systems Engineer Reacts to Changed Requirements

V1.0 release July 30th 2010

Contents

OSLC PLM Workgroup - Introduction

- Open Services for Lifecycle Collaboration (OSLC) is a community effort to help product and software delivery teams by making it easier to use lifecycle tools in combination

- The OSLC PLM workgroup aims to:
  - Evaluate applicability of existing OSLC specifications towards use in an ALM/PLM setting
  - Contribute towards extension or new OSLC specifications based upon the need for ALM/PLM collaborations
  - Engage with the workgroup at http://open-services.net/bin/view/Main/PlmHome

- The Workgroup proposes that scenarios provide a way of focusing consideration of the usage of existing OSLC Specifications and build out, by way of a shared understanding of the wide and growing areas of concerns across ALM/PLM

Scenario #1: Business context

- Business setting
  - Systems engineering responsibles are responding to a change in requirements for an existing product or system implementation; they need to make updates across different content types and stakeholders in a controlled way
- Business problem
  - Today, organizations experience delays, wasted effort, actual errors or lost opportunities. These arise from the difficulties of establishing and maintaining linkage between different stakeholders, activities and the various product, system and software representations, e.g. during handling of changes to requirements for existing products or system components
- Business goals
  - To increase responsiveness, reduce cost & waste
  - To reduce the cost of managing complexity
- Stakeholders
  - Customer Responsibles e.g. Sales, Market, Field Engineers
  - System or Product Responsibles e.g. Product Managers, Systems Engineers, System Architects
  - System or Product component Responsibles e.g. Designers

Scenario #1: OSLC concerns

- Lifecycle Collaboration interests
  - Discovery and visibility across heterogeneous environments
  - Establishment of a relevant context for change
  - Maintenance of coherency during change
- Business problem
  - Fragmented support and control along life-cycle
  - Diverse and multiple tools and information stores
  - Expensive to build and maintain integrated tool and information flows
- Business goals
  - To simplify tool integration
  - To increase lifecycle process support
  - To reduce the cost and time to manage complexity
- Stakeholders
  - Capability or process owners
  - Research & Development operations
  - IT Governance, Architecture and Operations
  - Tool developers and suppliers

Scenario #1 addresses key activities of system responsibles

1. A system responsible like a Systems Engineer needs to respond quickly and accurately to requests for changes to meet responsiveness goals and objectives

2. A Systems Engineer needs to assess the impact of a change on the affected system definition, which is a combination of the relevant agreed requirements, specifications and implementation descriptions

3. System responsibles operate in various projects, communities and organizations for different systems, products and projects

4. A Systems Engineer needs to prepare an update to the system definition as a solution to the change request, working on the appropriate areas, re-using or designing relevant content and calling upon other contributors, as needed, to meet the system objectives

Outline of Scenario #1

1. Assign & communicate the change request (a1, a2, a3)
    - Assign change request context
    - Communicate change request
    - Locate change request from notification
2. Apply request context to establish impacted requirements & implementation (a4, a5, a6)
    - Locate requirements in change request context
    - Create new revision of requirements context and reserve for editing
    - Open new revision of context
3. Locate re-usable implementations to meet changed requirements (a7)
    - Located reusable implementation to satisfy change? (A decision that drives alternative flows)

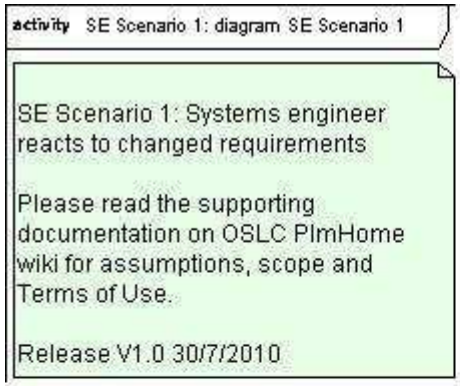
4a. Either update solution by way of adaption of re-usable implementations (a8, a9, a10, a13, a14, a15)
    - Add selected implementation to change request as solution
    - Merge selected implementation into context
    - Trace to discipline responsibility
    - Analyse detailed requirements & existing implementation
    - Design minor updates to existing implementation
    - Design by sub-team needed ?

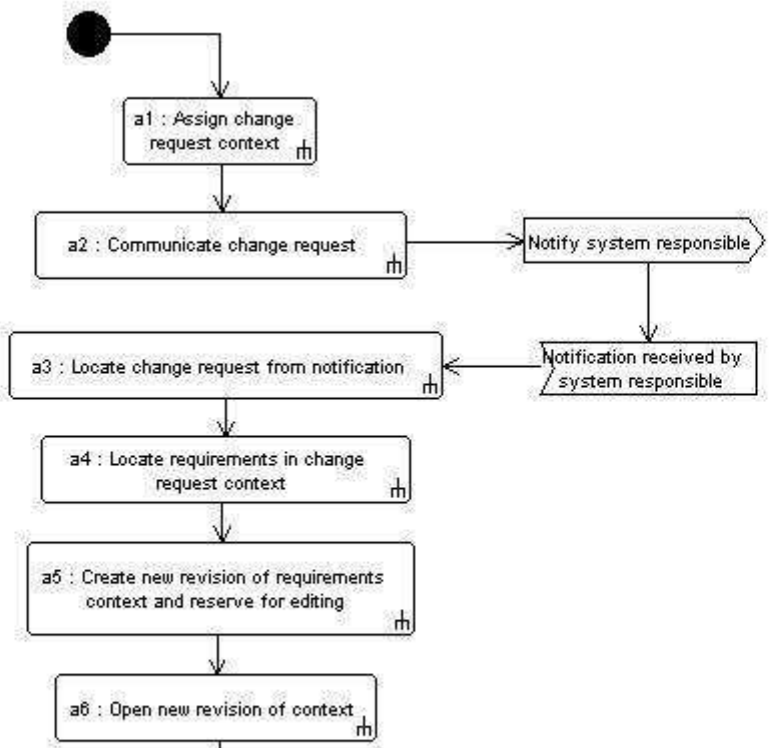
4b. Or design solution by original design (a10, a11, a12, a15)
    - Trace to discipline responsibility
    - Design new implementation
    - Add new design to customer request solution
    - Design by sub-team needed ?

5. Approve change request solution (a16, a17)
    - Passed review of implementation for customer request closure?
    - Close customer request

* Note: 4 has alternative flows

Scenario Activity Diagram – 1 of 3

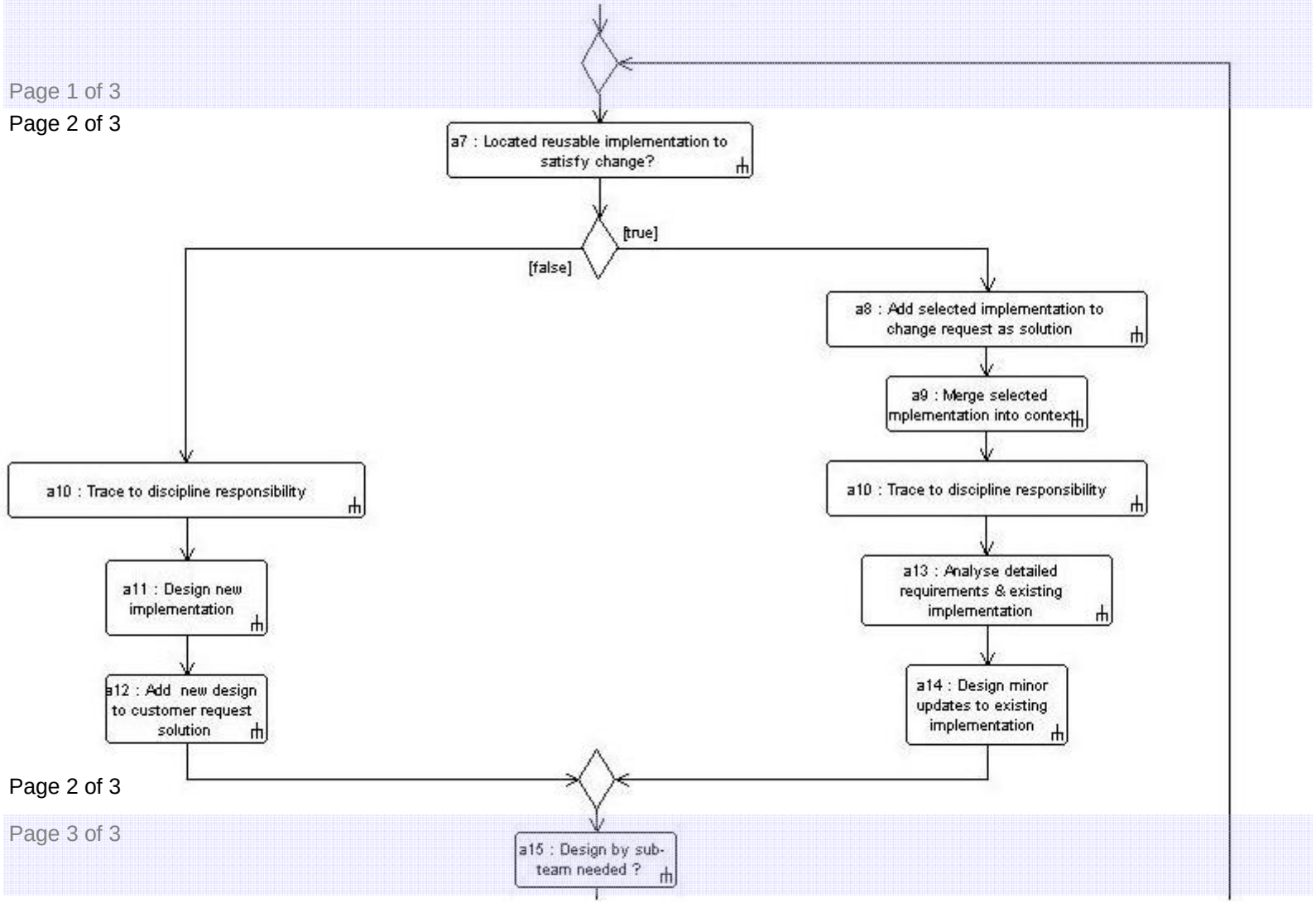




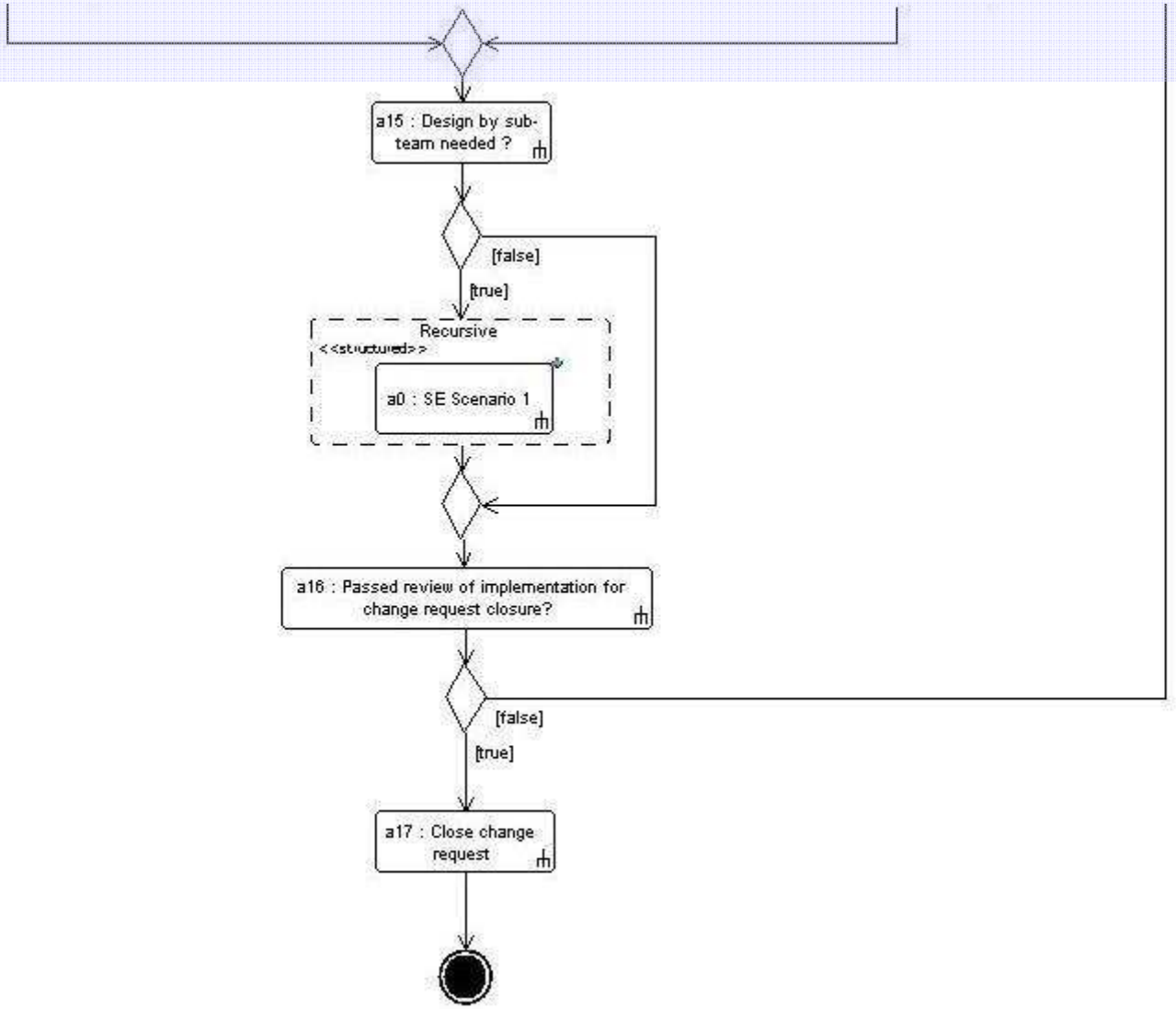
Page 1 of 3

Page 2 of 3

Scenario Activity Diagram – 2 of 3



Scenario Activity Diagram – 3 of 3



Scenario release deliverables

Ref	Item name	Description	Version	Format	Location
1	Scenario overview	Overview presentation	V1.0	Pdf, ppt	Link to scenario page
2	Scenario Activity Diagram	Document & graphical image	V1.0	Html, Jpg	Link to scenario page
3	Scenario Activity Diagram	SysML model	V1.0	SysML export zip	Link to scenario page
4	Scenario feedback wiki	Place to discuss and provide feedback on the Scenario	NA	Wiki on website	Link to scenario feedback page

Next steps

- Publicise for feedback
- Review with OSLC Workgroup leads
- Analyse opportunities to use existing OSLC Specifications
- Gap analysis
- Proposals to extend existing Specifications
- Select additional concerns to build out

Engaging and providing feedback

- You are welcome to join and contribute to the work effort
- Please provide direct feedback to the OSLC PLM Workgroup wiki and through our regular meetings
  - Scenario feedback page
    - http://open-services.net/bin/view/Main/PlmSystemsEngineeringScenarioSystemsEngineerReactstoChangedRequirementsFeedback
  - Meeting announcements are made via the workgroup Distribution list
    - http://open-services.net/mailman/listinfo/oslc-plm_open-services.net
  - PlmHome wiki page
    - http://open-services.net/bin/view/Main/PlmHome
- Please also review and satisfy yourself of your ability to meet the Terms of Use
    - http://open-services.net/html/Terms.html

Acknowledgements

- Particular thanks to the workgroup members

  Rainer Ersch (Siemens, lead)
  Gray Bachelor (IBM, organizer)
  Mike Loeffler (General Motors)
  Brenda Ellis (Northrop Grumman)
  Roch Bertucat (ENEA)
  Pascal Vera (Siemens)
  Scott Bosworth (IBM)
  Keith Collyer (IBM)
  Brent Feather (IBM)
  and others

Supporting information

Definition & usage of main terms

Name	Description used here
Change request (n)	A request to modify an existing product or system representation
Context (n)	The relevant combination and states of the business & technical environment including formal configuration
Design (v)	To define, verify and validate a solution
Discipline (n)	A stakeholder capability
Implementation (n)	A definition of realisation of the product or system (may be not have been realized)
Product	A commercial item
Requirement (n)	A statement of need and/or specification to be fulfilled
Solution	An implementation meeting requirements
Sub-team	A unit of organization of stakeholders and their resources
System	A combination of components to provide or realise some greater function
Trace	Locate through relationships and associations

Under discussion
n – The Name is treated as a Noun

Scenario actions & descriptions

Ref	Action	Description
a1	Assign change request context	Align and assign the change request to product lines, products and system contexts e.g. identities, coding, classification, release, version, variants, effectivity. Out of scope in 1.0 release of the scenario - organisational assignment and change of context during change request processing within the scenario.
a2	Submit change request	Send change request to responsibles identified by way of the context. Out of scope in 1.0 release of the scenario: Definition of responsibles, Build up and validation of supporting information, notification of other stakeholders such as the original requester, capturing of change request metrics
a3	Locate change request from notification	A system responsible engages with a change request in their own environment customised by the change request context.
a4	Locate requirements in change request context	Using the change request context locate related requirements. Includes exploration of context and requirements to locate further relevant requirements.
a5	Create new revision of requirements context and reserve for editing	Revise the requirements context for its subsequent change.
a6	Open new revision of context	Access the newly revised context.
a7	Located reusable implementation to satisfy change?	Locate, analyse and decide upon existing implementations in the change request and requirements context for applicability (i.e. re-use) as content in e solution. Out of scope in 1.0 release of the scenario: Multiple solution build up, evaluation and trade off.
a8	Add selected implementation to change request as solution	Add (i.e. adopt) the selected existing implementation into the change request solution by building up a solution from abstracted implementation components descriptions.
a9	Merge selected implementation into context	Merge (i.e. add, replace, delete) the detailed implementation content into the change request solution and update the context.
a10	Trace to discipline responsibility	Apply the context to locate those responsibles for defining missing content, verifying & validating and approving the change request solution.
a11	Design new implementation	Design missing implementation content to meet the requirements in the change request context
a12	Add new design to customer request solution	Make the updates of the design implementation into the change request solution. Handle any conflicts. Out of scope in 1.0 release of the scenario: Loops or concession to deal with conflicts.
a13	Analyse detailed requirements & existing implementation	Detailed analysis of the requirement and implementation within the prevailing context
a14	Design minor updates to existing implementation	Make updates and changes to the re-used implementation to meet the requirement in the change request context. (i.e. minor updates). Out of scope in 1.0 release of the scenario: Governance of changes to re-usable content
a15	Design by sub-team needed ?	Evaluate if additional design, updates, verification and validation are needed by other system responsibiles (e.g. by discipline or due to division of system responsibilities). Out of scope in 1.0 release of the scenario: Workshare due to capacity or organisation (e.g. supplier)
a16	Passed review of implementation for customer request closure?	Evaluate, agree upon and approve implementation as meeting the requirements in context. Out of scope in 1.0 release of the scenario: Iteration towards agreement. Concessions or additional changes to derived / internal requirements. Establishment of criteria.
a17	Close customer request	Release (i.e. make available and mark as complete to some criteria) the updated change request context and solution. Communicate availability to stakeholders.
a0	Design customer request solution content by sub-team	Sub-team fulfils their design responsibilities towards the implementation build up of the change request solution

Scope – areas deemed out of scope

- This first scenario is indicative of real world challenges but is deliberately simplified
- The following concerns were identified out of scope for V1.0 as the work progressed
  - Definition of change request context
  - Pre-analysis of a change request
  - Establishment of change request evaluation criteria
  - Evaluation of alternative change request solutions
  - Detailed and ad-hoc recursion caused by re-work, re-design, re-approval, backtracking
  - Design activities associated with intended capability, behavior, function or performance
  - Handling of multiple concurrent flows (i.e. more varied recursion)

Main assumptions

- Product, systems, components, requirements and implementations are configured and under change control
  - Managed as a collection with defined relationships
  - Active change control rules and policies & evolution traced
- Multiple and overlapping change requests will be “in play”
  - Change requests have effectivity (when and to what they apply)

Thank you

For more information please contact

Rainer Ersch
rainer.ersch@siemens.com
Gray Bachelor
gray_bachelor@uk.ibm.com