# OSLC4MBSE

## OSLC for Model-Based Systems Engineering Interoperability

This document presents the work of the OSLC4MBSE working group, which has been initiated as a collaborative effort between members of the OMG Systems Engineering and OSLC communities as part of the OMG SE DSIG. The aim of this workgroup is to investigate and develop an approach for multidisciplinary lifecycle integration in systems engineering, in order to support interoperability between the domains and their data. The graphical modeling language OMG SysML™ and its concepts represent a subset of the systems engineering domain which supports general purpose modeling of systems. The Open Services for Lifecycle Collaboration (OSLC) is a tool interoperability approach based on standardized and open Web technologies that enables common interoperability across various domains. The initial focus of this working group is to investigate how SysML concepts can be implemented using OSLC to achieve lifecycle integration. Demonstrations are being planned to illustrate the approach.

### Group Members


| | | |
|---|---|---|
| Parham Vasaiely \| Chairman | EADS | Parham.Vasaiely@eads.com |
| Axel Reichwein \| Co-Chair | Koneksys | axel.reichwein@koneksys.com |
| Yves Bernard | Airbus | Yves.Bernard@airbus.com |
| Roger Burkhart | John Deere | BurkhartRogerM@JohnDeere.com |
| Harald Eisenmann | Astrium | Harald.Eisenmann@astrium.eads.net |
| Amit Fisher | IBM | amfisher@us.ibm.com |
| Sanford Friedenthal | OMG | safriedenthal@gmail.com |
| Eldad Palachi | IBM | eldad.palachi@il.ibm.com |
| Chris Paredis | Georgia Institute of Technology | chris.paredis@me.gatech.edu |


An OMG Systems Engineering and OSLC working group as part of the OMG SE DSIG.

# Contents



# Introduction

## Background

The increasing **complexity** of modern technical systems, such as those used in the aerospace or equipment industry, and the fact that these systems are designed and manufactured in a distributed manner, represent a significant challenge to the traditional engineering discipline.

Systems engineering (SE) is a multidisciplinary field of engineering that focuses on ensuring the pieces of the system work together to accomplish the objectives of the whole, and that the functional, performance, physical, cost, and other lifecycle concerns are properly considered to achieve a balanced system solution. It also integrates engineering activities under common processes and methods to achieve these outcomes.

The International Council on Systems Engineering (INCOSE) has identified **Model-Based Systems Engineering (MBSE)** as a key practice for successful systems engineering in the future. Traditional systems engineering practices are often document centric, where the information about the system specification and design is described in documents. The content in such documents is often ambiguous because it has not been defined according to formal system modeling practices. As a result, miscommunication among stakeholders can occur due to different interpretations of the same content. Furthermore, the content of traditional documents is not easily machine-readable, which hinders reuse and discovery of inconsistencies.

In contrast, MBSE promotes a model-centric systems engineering approach supporting formal and machine-readable models. Another key feature is full traceability, the representation of dependencies between engineering data. Consequently, MBSE can enable systems engineering and change management to be performed more efficiently than with traditional documents.

The **OMG Systems Modeling Language (OMG SysML™)** was developed to enable MBSE by providing a standardized graphical modeling language for representing systems. OMG SysML is a well known systems modeling language and is increasingly being used to model complex systems.

However, as the intention of SysML is not to capture all system aspects as a single modeling language, other specialized modeling languages used in other domains, such as electrical, mechanical, thermal, and aerodynamics, have to be integrated. Changes to the system model may have an impact on other downstream modeling activities. For example a requirement change may not only require a change in a SysML Block, but also may require a change of a CAD Part and a Simulink Block. The **capability to link** systems engineering information across multiple lifecycle domains is therefore crucial.

MBSE is intended to enable the integration of systems engineering information across multiple domains and lifecycle phases. Integration is key for maintaining system integrity, satisfying system requirements, achieving full traceability, reducing rework and improving quality at the same time. However, integration of systems engineering information across several disciplines and tools is currently very difficult to achieve due to the lack of an open and common interoperability standard.

The big **challenge** lies in identifying the most appropriate standard for a common systems engineering data format. Several common data formats have been developed in single engineering disciplines such as STEP in mechanical engineering, XMI for software and systems modeling, and FMI for dynamic simulations. In parallel to these developments, common data formats have been developed for knowledge sharing. The **Resource Description Framework (RDF)** is a World Wide Web Consortium (W3C) standard for data interchange on the Web. It is discipline-neutral, widely adopted and increasingly used for Semantic Web applications.

The huge **success of the World Wide Web** provides an opportunity for the systems engineering community to rethink the approach to a common systems engineering data format. Since systems engineering data originates in various engineering disciplines, it would be meaningful to choose a common data format for systems engineering which is discipline-neutral and compatible with a widely adopted standard for sharing data on the Web. Here the **Linked Data (LD)** approach, based on HTTP access to web resources that describe their state using RDF, seems extremely promising to a future-proof and scalable solution. Furthermore implementing a solution based on Web principles supports a resource-oriented architecture, focusing on information and documents and their links or relationships. A tool exposes its data and services through a resource-oriented Web service, in order to make its data and services available to the widest possible range of clients. The intent of the Web service is to provide a stable long-term web-based interface for directly accessing the facilities and data offered by the tool. These web services use an arrangement of URIs, HTTP methods, and standard representation languages such as XML and RDF, that work like the rest of the Web.

Therefore, by reusing RDF as a common systems engineering data format, the systems engineering community could take advantage of the currently available tool support for RDF as well as Semantic Web technologies for consistency checking, classification and queries.

## Open Services for Lifecycle Collaboration (OSLC)

The Open Services for Lifecycle Collaboration (OSLC) is an open community with **the objective to create specifications on how tools, involved in different engineering lifecycles, can be integrated to share data and information**. In addition to the open community, OSLC is also a technical approach to support tool integration based on open and standard Web technologies like Linked Data, RDF, REST, and HTTP. The application of OSLC will ensure the implementation of integration of domains and their corresponding tools based on a common interoperability approach. In this work the term interoperability specifically relates to the challenges to ***share data*** rather than exchange. The difference between sharing and exchanging of data is related to the underlying implementation approach. When implementing interoperability to exchange data, the work and information flows supported by the system are predefined, whereas the implementation of data sharing, allows a more generic access which is work and information flow independent. By adopting the W3C Linked Data approach information can be linked, in order to achieve full traceability, as illustrated in Figure 0-1.

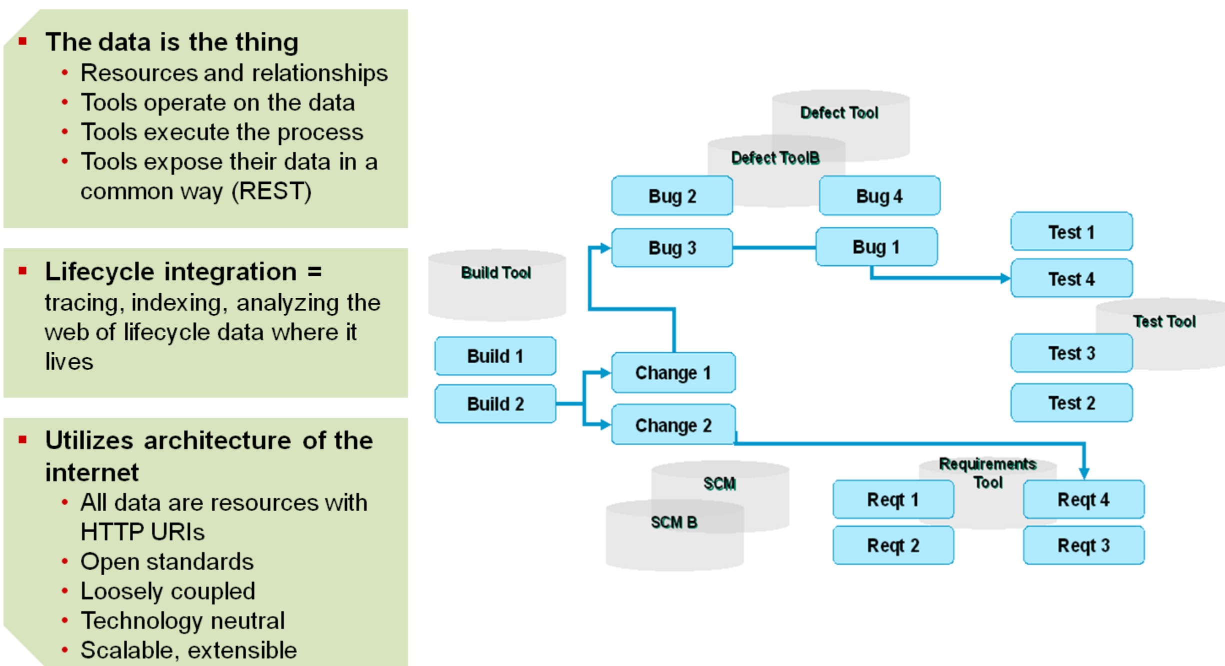


Figure 0-1 Linked Lifecycle Data used by OSLC

The technical features of OSLC are summarized below and more information can be found at [open-services.net]:

**Tool- and platform-independent**

- OSLC tool integrations rely on the **Hypertext Transfer Protocol (HTTP)** which is a request-response protocol for client-server communication on the Web. HTTP is independent of a tool or platform. OSLC tool integrations can therefore be developed without having to rely on tool- or platform-specific API's.
- OSLC services follow the **REpresentational State Transfer (REST)** architectural pattern. OSLC services can create and link new resources, describe properties of existing resources, and query existing resources. REST is an architecture style for designing networked applications which provides the underlying scalability and resiliency of the World Wide Web. In practical terms, it means that rather than relying on tightly coupled client-server mechanisms such as Common Object Request Broker Architecture (CORBA), Remote Procedure Calls (RPC), or Simple Object Access Protocol (SOAP) to connect between applications, these connections can be based on the protocols of HTTP, which are designed for decentralized, distributed access across the World Wide Web.

**User-friendly and secure**

- OSLC services can support **delegated user-interface dialogs** to interact with resources via a Web browser.
- Access to OSLC service providers can be authenticated by the **OAuth protocol**, allowing secure authorization by a simple and standard method from web, mobile and desktop applications.

**Ready for the Semantic Web**

- OSLC publishes its resource specifications according to the principles of **Linked Data**, which has established success as a general-purpose method for publishing data on the World Wide Web.
- OSLC resources are represented in the **Resource Description Framework (RDF)**, a knowledge representation format widely used for the Semantic Web. RDF has features that enable inconsistency checking, classification, and merging of data even if their underlying schemas differ.

**Developer-friendly**

- OSLC-compliant tools can be built using the **Eclipse Lyo** software development kit (SDK), which consists of a Java SDK for both providers and consumers called OSLC4J, as well as reference implementations and test suites. OSLC4J builds on several more widely used Java libraries, such as Apache Jena for reading, querying and writing RDF data, and Apache Wink for building RESTful Web services.

## OSLC Workgroups

OSLC Workgroups specify a common vocabulary in RDF for several systems engineering domains such as requirements management, change management, and automation. Most OSLC Workgroups currently define vocabularies for domains in Application Lifecycle Management (ALM). However, the OSLC approach is domain-neutral and OSLC vocabularies in RDF can also be defined for other domains such as MBSE. An OSLC Workgroup is, for example, currently drafting a vocabulary for ALM-PLM Interoperability.

Before defining a vocabulary, each workgroup first agrees on tool interoperability scenarios which shall be supported by the new vocabulary. Most workgroups then go through several iterations in order to identify the vocabulary which will find the largest consensus. Once the vocabulary is final, it is called an OSLC specification. The steps for defining an OSLC specification are shown in Fig. X.

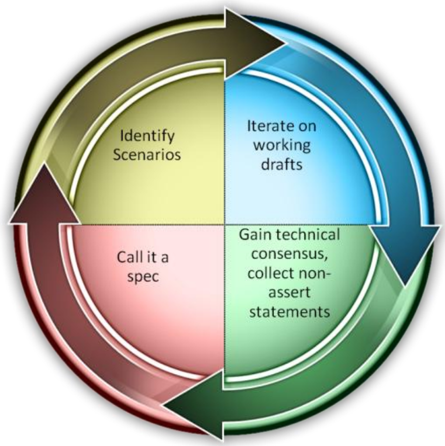


**Figure 0-2 Iterative Specification Authoring**

# The Workgroup

This section introduces the purpose and scope of the workgroup, as well as main milestones and planned dissemination activities.

## Purpose and Goal of the initiative

By developing an OSLC specification for MBSE, including a vocabulary and services , information from a SysML model may be linked with information from other models belonging to other domains (see Figure). This would enable traceability between tools across different engineering disciplines and lifecycle phases.

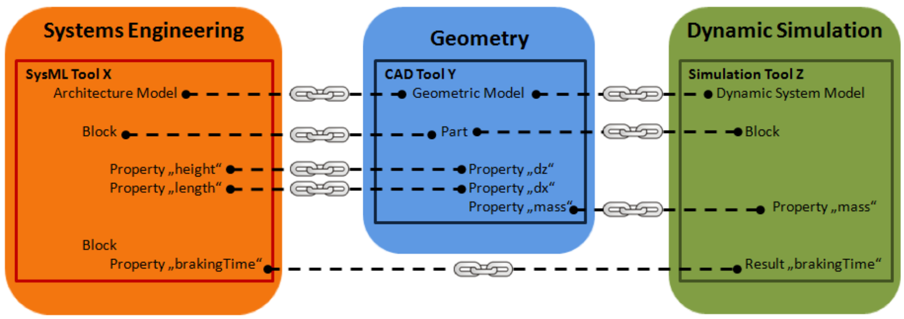


**Figure 0-1 Linked Lifecycle Data**

This initiative will identify how SysML concepts can be implemented using OSLC in order to support MBSE activities and achieve full traceability through the systems engineering lifecycle. The approach shall support collaborative system architecture and design management to be applied in the context of OMG SysML. This supports key activities between design teams and their stakeholders such as the sharing of design data, traceability, reviewing and managing system architecture and design.

Furthermore, this workgroup shall support the following aspects within the workgroup members, as well as the OMG and OSLC communities:

- Mutual education on Lifecycle Management using OSLC and how the MBSE domain, using SysML, can benefit from this technology.
- Development a business case for OSLC and Lifecycle Management to be applied in the MBSE domain
- Develop introduction and presentation material to this topic
- Develop a first demonstration, including tooling to be used to implement the approach as a first prototype

## Scope

The initiative shall investigate the minimum set of SysML concepts need to be shared during the engineering lifecycle to achieve MBSE. This may require the investigation of multiple adjacent and affected lifecycle domains (e.g. Change, Requirements, Quality and Test) related to existing SysML concepts. Addressing multiple lifecycle aspects, allows us to look at various domains of the OSLC Specification. Finally a proposal shall be made of how to develop the approach further and plans for future work.

## Work plan and main milestones

The general RTP work plan is divided into five phases, as described below. The phases go from the planning and organization through to the deployment and validation of the RTPv1.

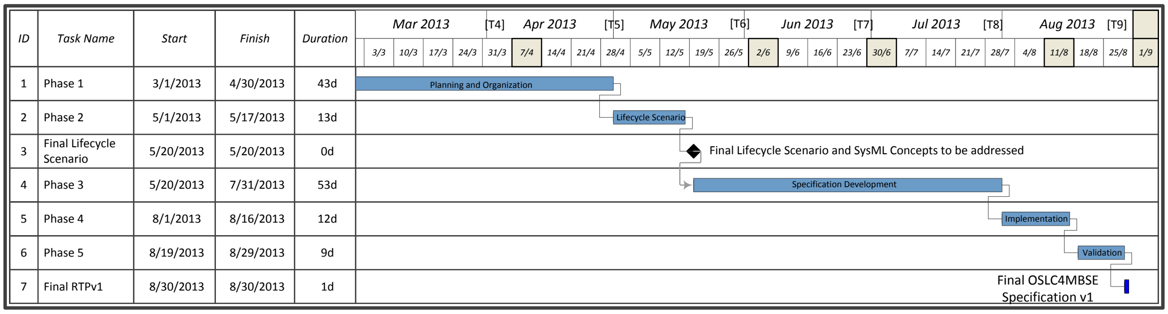


Figure 0-2 General Work Plan as Time and Milestone Diagram

| Phase | Name | Short Description | Deliverables/Milestone | Resp. | Start | End |
|---|---|---|---|---|---|---|
| **P1** | Planning and Organization | This phase includes the planning and organization of the working group, as well as the allocation of resources.<br>T1.1: Project Setup | Final work plan and communications document | | **M0**<br>01 Mar 2013 | **M2**<br>30 Apr 2013 |
| **P2** | Identify lifecycle scenario and requirements | Develop an integration scenario expressing the challenges of lifecycle interoperability in MBSE. Identify requirements set to such a lifecycle interoperability. The scenario shall cover a small subset of SysML concepts (i.e. BDD, IBD, Requirements, Dependencies, Ports).<br>T2.1: Develop Lifecycle Interoperability Scenario for MBSE | **M1: Lifecycle scenario to be covered by the approach.**<br>Subset of SysML concepts to be addressed. | | **M2**<br>01 May 2013 | **M2**<br>17 May 2013 |
| **P3** | Specification Development | The implementation of the specification is divided in three tasks:<br>T3.1: Identify how to express SysML concepts using OSLC<br>T3.2: SysML Resource Types Definition based OSLC Core principles<br>T3.3: Service Definition based OSLC Core principles. | OSLC4MBSE Specification v1<br>- Resource Type Definitions<br>- MBSE common Services | | **M4**<br>20 May 2013 | **M5**<br>31 Jul 2013 |
| **P4** | Implementation | The implementation of a demonstration shall prove the application of the developed specification covers the identified lifecycle scenario. This is done by the following tasks:<br>T4.1: Define transformation rules for translating SysML into RDF/XML using existing technologies.<br>T4.2: Implement a tool integration scenario to be demonstrated<br>T4.3: Develop presentation material | First Demonstration | | **M5**<br>01 Aug 2013 | **M6**<br>16 Aug 2013 |

| Phase | Name | Short Description | Deliverables/Milestone | Resp. | Start | End |
|---|---|---|---|---|---|---|
| **P5** | Validation | The first version of the OSLC4MBSE specification will be validated using the initial scenario from the first phase. In addition, all aspects of the scenario that are not covered shall be documented with the goal of addressing this in a future version of the specification.<br>T5.1: Validate specification<br>T5.2: Wrap-Up and future work | **M2: Validated and Final OSLC4MBSE Specification v1** | | **M6**<br>19 Aug 2013 | **M6**<br>30 Aug 2013 |

Figure 0-3 Development Phases of the OSLC4MBSE Specification

## Tasks and corresponding sub-tasks

**T1.1: Project Setup**

T1.1: Identify workgroup purpose and partners

T1.2: Develop Work Plan

T1.3: Communication Document

T1.4 Project Wiki: Start at OMG Wiki, migrate over to OSLC Wiki

T1.5 Mailing List

**T2.1: Develop Lifecycle Interoperability Scenario for MBSE**

**T3.1: Identify how to express SysML concepts using OSLC**

T3.1.1 Identify SysML concepts already addressed in the various OSLC specifications

T3.1.2 Identify SysML concepts which need to be defined using OSLC technologies

T3.1.3 Define Resource type

**T3.2: SysML Resource Types Definition based OSLC Core principles**

T3.2.1 Define naming conventions for HTTP URIs of SysML model elements

**T3.3: Service Definition, , based OSLC Core principles**

T3.3.1 Define basic services to be provided as common internal to the SysML domain, based OSLC Core principles.

T.3.3.2 Define basic services to be provided as common across the MBSE domain to other lifecycle domains

**T4.1: Define transformation rules for translating SysML into RDF/XML using existing technologies.**

**T4.2: Implement an tool integration scenario to be demonstrated**

**T5.1: Validate specification**

**T5.2: Wrap-Up and future work:** Document not covered aspects of the scenario to be addressed in the next version of the specification

**Tx.1 Develop Whitepaper and Presentation Material**

**Tx.2 Dissemination**

Tx.2.1 Dissemination to OMG Community: OMG, SE DISG Meeting in Berlin, on Tuesday, June 18.

Tx.2.2 Dissemination to OSLC Community: International OSLC Conference 2013, as being planned, http://open-services.net/wiki/communications/Conference2013Planning

Tx.2.3 Dissemination to INCOSE Community: TBD

# References


OSLC Website, http://Open-Services.net

OSLC Tutorial, http://open-services.net/resources/tutorials/oslc-primer/

Linked Data Platform Working Group, http://www.w3.org/2012/ldp/wiki/Main_Page

Linked Data Basic Profile, http://www.w3.org/Submission/2012/02/

Linked Data Design Issues, Tim Berners-Lee, http://www.w3.org/DesignIssues/LinkedData.html

AP233, http://homepages.nildram.co.uk/~esukpc20/exff2005_05/ap233/index.html

CESAR Interoperability Specification, http://www.cesarproject.eu/fileadmin/user_upload/CESAR_D_SP1_R1.6_M4_v1.000_PU.pdf

The Open Group, SOA Source Book, http://www.opengroup.org/projects/soa-book/

The World Wide Web Consortium (W3C), Resource Description Framework (RDF), http://www.w3.org/RDF/