Open Services for Lifecycle Collaboration (OSLC) - Extending REST APIs to Connect Data

Axel Reichwein[1] and Fernando Lopez[2]

[1] Koneksys, San Francisco CA 94117, USA
[2] Koneksys, Tultitlan 54955, Mexico
{axel.reichwein, fernando.lopez}@koneksys.com

Abstract. Organizations designing complex technical systems need to understand the relationships between various engineering artifacts such as requirements, design and manufacturing information. These relationships should be captured in a global graph, which can then be queried by different stakeholders. However, engineering data is authored in hundreds of different applications with different APIs. Open Services for Lifecycle Collaboration (OSLC) provides a standard approach to expose data in the RDF format with REST APIs, which in turn supports data interconnectivity and graph-based data processing. The global graph can quickly be built if heterogeneous engineering data is available in a neutral data format and accessible through a standard API, as provided by OSLC. Without the adoption of standards, it is practically impossible to build a fully comprehensive global graph for engineering. Without such a graph, the impact of design decisions can lead to unforeseen errors, which in the worst case can lead to fatal accidents.

Keywords: Data Integration · Global Graph · PLM · Systems Engineering · OSLC · Digital Thread · REST APIs

1 Traceability in Engineering

Design errors are not just the result of negligence but also of poor traceability across design data. Traceability allows engineers to understand the impact of design changes and to see if design changes still meet the original requirements. For example, if the design of an aircraft is modified, it is necessary to check if it still meets all the requirements, especially the safety-critical requirements such as sensor data redundancy and control system behavior. Unfortunately, design data exists in many data silos, and cross-disciplinary traceability is currently just described in documents, in non-machine-readable formats, which prevents the execution of intelligent queries or automatic consistency checks. Even a world leading aircraft manufacturer, with a formidable reputation for designing safe aircrafts, has recently identified design errors late, only after two of its aircrafts crashed. As technical systems become more complex, organizations are reaching the limits of document-based traceability. Traceability is the key to understand the impact of design changes and to manage the complexity in design.

Typically, virtual computer simulations are performed to verify requirements compliance because physical tests on prototypes are very expensive. The virtual computer simulations usually require input data from many other sources. For example, a simulation of the aircraft control system requires information from a test case document such as environment conditions to use for the simulation, from a geometric model such as the center of gravity position from other simulation models such as a plant model computing how the aircraft behaves under certain forces. The computer simulation will create results which can then determine if a test case was successful or not, and ultimately if a requirement was satisfied. A design change may impact multiple system aspects and may require the re-execution of many test cases and computer simulations. Traceability helps engineers understand which simulation results were created by which test case, using which simulation model, using which simulation model inputs, and for verifying which requirement. Based on a digital representation of these traceability relationships, engineers can navigate through these relationships, perform queries on these links and efficiently assess the impact of design changes.

2 Data Heterogeneity and Global Graphs

Engineering organizations use different software applications to describe different aspects of a system, such as requirements, test cases, architecture models, simulation models, 3D geometric models, etc. There are over 500 different engineering applications to describe and simulate various system aspects. Each software application uses a specific data format and API. This data heterogeneity currently prevents an organization from efficiently analyzing all their data as a whole. Data integration solutions for engineering data exist but are limited to specific engineering disciplines, such as software engineering in the form of Application Lifecycle Management (ALM) solutions, mechanical engineering in the form of Product Lifecycle Management (PLM) solutions, systems engineering in the form of Model-Based Systems Engineering (MBSE) solutions, manufacturing data in the form of Enterprise Resource Planning (ERP) solutions, and sensor data in the form of IoT (Internet of Things) solutions.

Engineering organizations need a global graph to capture in a machine-readable format the traceability links describing cross-cutting concerns. Unfortunately, due to the data heterogeneity, the Extract-Transform-Load (ETL) process to merge data from all engineering applications into one global graph currently represents a huge effort in time and money and is practically impossible to realize. Open Services for Lifecycle Collaboration (OSLC) [1–3] provides standards to simplify the ETL process, and for subsequently linking data within a global graph.

3 Open Services for Lifecycle Collaboration (OSLC)

Engineering applications are increasingly supporting REST APIs. However, many generic aspects of engineering data such as its version, the project it belongs to, its change history, or the data model it conforms to, are currently described in a multitude of different ways. The OSLC Core Specification [2] defines RDF vocabularies to describe these discipline-independent generic data aspects in order to support data interoperability. OSLC APIs [2] expose engineering data in RDF in serializations such as JSON-LD, Turtle, and RDF/XML. OSLC [1–3] also reuses W3C standards such as HTTP, RDF, URI, and Linked Data Platform (LDP). OSLC [1] has been adopted by several software vendors, most notably IBM by a better integration of software engineering data in their ALM solution. OSLC [1–3] is supported by products of many other vendors such as Mentor Graphics, PTC, Tasktop, Kovair, Sodius, Maplessoft, Smartfacts, and many more. These solutions among others improve requirements traceability and collaboration in the design of complex systems.

OSLC APIs [2] contribute to decoupling data from applications, similarly to the Solid initiative of Sir Tim Berners Lee. By decoupling data from applications, applications can consume data from more sources, and provide value across more domains. For example, a global discovery and full text search capability to find data stored in many different sources, similar to the search engine to find documents on the Web, can only be developed if data can be decoupled from applications through a standard API such as an OSLC API [2].

References

1. OSLC Homepage, https://open-services.net/. Last accessed Jun 28th, 2019.
2. OSLC Core Version 3.0, https://docs.oasis-open.org/oslc-core/oslc-core/v3.0/csprd03/part1-overview/oslc-core-v3.0-csprd03-part1-overview.html. Last accessed Jun 28th, 2019.
3. OSLC GitHub, https://github.com/oslc. Last accessed August 29th, 2019.