This is the published version of a paper presented at CIRP CMS 2015 - 48th CIRP Conference on MANUFACTURING SYSTEMS.

Citation for the original published paper:

Shariatzadeh, N., Gürdür, D., El-khoury, J. (2015)
Using Linked Data with Information Standards for Interoperability in Production Engineering.
In:

N.B. When citing this work, cite the original published paper.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-182649
Using linked data with information standards for interoperability in production engineering

Navid Shariatzadeh, Didem Gurdur, Jad El-Khoury, Lars Lindberg, Gunilla Sivard

Abstract

Design and verification of factory layout and material flow is a multidisciplinary, knowledge-intensive task which requires a collaborative framework where all specialists involved can communicate, interact, manage and visualize different models. However, the communication of digital models comes with challenges. The information resides in various systems and applications, in different formats and with various levels of detail and viewpoints. This makes the communication and sharing of information among different actors and application, challenging.

To deal with the data exchange and integration problem, the information standards ISO 10303 (STEP) has shown a strong capability to represent rich information models in a wide variety of industrial domains for the purpose of exchanging data. On the other hand, the Open Services for Lifecycle Collaboration (OSLC) initiative provides a minimalistic set of standardized information models, focusing on the most common concepts within a particular domain. Assuming a loosely-coupled distributed architecture of tools and services, OSLC adopts the Linked Data (LD) approach to ensure data consistency across the data resources.

How can we combine STEP's rich information model for data exchange, with OSLC's minimalistic approach for data integration?

The aim of this work is to show the applicability of using these two complementary paradigms – and their corresponding standards to support interoperability and data integration in a heterogeneous IT environment for material flow analysis and layout design.

Keywords: Open communication standard; Integration; Information modelling

1. Introduction

Digital factories [1] have shown a great potential to support companies for analysis and simulation of industrial processes. Among digital manufacturing techniques Discrete Event Simulation (DES) and digital layout designs have been widely accepted in the manufacturing industry for material flow analysis and consequently plant layout design. The multi-disciplinary and knowledge-intensive nature of DES models and layout design requires collecting and structures data and information regarding the manufacturing processes and product routing as well as manufacturing facts and rules [2].

However, today this information is normally not integrated in one place, but is heterogeneous and stored in different IT applications in the factory plant. The lack of interoperability among these systems results in inefficiency of utilizing IT tools in terms of cost and time. Information standards can facilitate the communication and sharing of information between relevant applications. Many research initiatives have been conducted to develop ontologies and information standards to cope with the interoperability problem. Among the different available information standard ISO 10303 STEP has shown a rich capability to represent information in a wide variety of industrial domains including layout design and material flow analysis [3]. The applicability of the representation and integration of layout data and, material flow data using STEP has been shown in [4]. However, information can be integrated in other ways than sharing a common model for instance via the Linked Data (LD) approach of OSLC. OSLC advocates a service-oriented, RESTful architecture and uses LD for integrating tool data [5].

LD refers to a style of publishing and interlinking structured data on the web. Assuming a loosely-coupled distributed architecture of tools and services, OSLC [6] adopts the LD
approach to ensure data sharing across the different data resources. Light-weight integration of data helps companies to be more agile in reconfiguring their IT system architecture in case of introducing a change such as adding a new IT tool. Therefore, in practice there should be an optimal mix of STEP information modelling and OSLC LD that enables information access to a user with a minimum of effort for system integrators.

This paper explains the idea of data exchange, sharing and change management in a heterogenous IT environment via both standards to support interoperability and data integration. The idea is to manage heterogeneity by incorporating two approaches: (1) The information standard (STEP) for information modeling and mapping, and (2) The OSLC’s light-weight approach for linking data. This paper is structured as follows: Section 2 summarizes the preliminaries and background information by describing the data integration approach and technologies that is relevant to understand OSLC. Section 3 discusses the comparison of the two paradigms with detailed information about their characteristics. Section 4 explains the guidelines for better integration. The case study and application of incorporation both approaches is explained in Section 5. The paper is concluded with suggestions for future work.

2. OSLC Background and Preliminaries

The OSLC initiative is an open community that is an industrial effort supported by several enterprises. OSLC is a member of OASIS Open Standards Network (Advancing Open Standards for the Information Society), which is a non-profit consortium that drives the development, convergence, and adoption of open standards for the global information society to solve communications challenges. OSLC develops standards that make it easy and practical for software lifecycle tools to share data with one another. OSLC standards apply the principles of the LD and REST protocol to provide the interoperable web standards-based environment [7].

Linked Data: The Web has evolved in recent years in a direction of not only linking documents but of also linking data itself. The term Linked Data refers to a set of best practices for publishing and connecting structured data on the Web [8].

Berners-Lee m.,) defined four ‘Linked Data Principles’ for publishing data on the Web in such a way that all published data constitutes a single global data space:

- Use URIs as names for things.
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).
- Include links to other URIs, so that they can discover more things.

It is possible to create links between data from heterogeneous systems with the help of LD and it is considered as flexible approach that provides support to integrating data through different tools. OSLC built on Web specifications and uses Resource Description Framework (RDF) as a fundamental data model. RDF is a framework that represents the LD and it provides a generic graph-based data model for describing resources, including their relationships with other resources. RDF uses links to connect the resources and makes it possible to navigate through these resources. Linked Data consists of two technologies; Uniform Resource Identifiers (URIs) and the HyperText Transfer Protocol (HTTP). URIs identifies the resources or entities that exist, and it is possible to retrieve these entities through the HTTP. Hereby, the HTTP provides a modest mechanism for receiving either the entities themselves or the descriptions of them. LD provides the methods to connect data with its resources using HTTP links and URIs.

Representational State Transfer (RESTful): a RESTful architecture is based of HTTP and consists of a set of architectural principles that supports Web services that focus on a system’s resources [9]. REST also looks how resources are addressed and transferred over HTTP. The REST design principles establishes mapping between create, read, update, and delete (CRUD) operations and HTTP methods with POST, GET, PUT, and DELETE methods.

OSLC tool adapters are the software that exposes the data of the integrated tools for data consistency. An OSLC adapter represents tool data in the form of an OSLC Resource and makes these resources available to other tools. These resources are available through web services and could be accessed and manipulated with the GET, PUT, POST and DELETE HTTP methods by using a RESTful (Fielding, 2000) architecture. This means that resources are addressable on the Web, so they can be created, changed, queried and retrieved by clients –such as other tool adapters.

OSLC standards define Core Specifications for setting out the common features that every OSLC Services can be expected to support and different domains such as Architecture Management (AM), Change Management (CM) and Requirements Management (RM) to deliver an environment for integrating IT-tools. The specification for each domain defines the basic concepts and rules and ensures consistency among different domain specifications.

OSLC defines the concept of ServiceProvider for each tool adapters to allow adapters to expose containers of resource that is hosted by a tool for integration. These containers are smaller parts of the overall space of artifacts in the tool which allows clients to navigate through the resources. The provider posts and provides RESTful services[10].

3. Comparison of STEP to OSLC

In this section we will focus on the applicability of STEP and OSLC according to different characteristics of each one. Import/export functionality of data: when a model developed in one IT tool needs to be transferred to another IT tool for further processing. An example is importing a 3D machine model into a layout CAD system for layout design. To achieve this, the internal data structure of the source IT tool must be mapped to the other tool. Using STEP standard as a system-neutral format eliminates the need to develop point-to-point translators among IT tools. On the other hand, the OSLC is not intended to support import/export functionalities.
Product data archiving capability: the long term storage of product data usually requires a rich metadata [11]. As mentioned before, using the STEP standard includes this rich data model to represent a digital product model with all properties including the detail geometry. In contrast, persistent storage of the data is not the focus of the OSLC.

Data integration mechanism: in some cases, data needs to be integrated instead of import/export. OSLC is about integrating applications by providing a loosely coupled integration of data (minimalistic set of data to be linked). It interrelate at least the concepts, which are common in different domains. On the other hand, STEP defines a form of data that is to be transferred between a pair of applications.

Web services: is a method to support interoperable machine-to-machine interaction over a network. OSLC specifies a common tool protocol for CRUD lifecycle data based on Internet standards in different domains. Some OSLC domains such ALM/PLM (application/product lifecycle management and Change Management) intrinsically include the behavior/actions, which must be taken into account when an engineering change occurs. In contrast STEP does not define any web services.

Domain: the topic area in the product lifecycle. In OSLC each domain (AM, CM, RM, etc.) specifies a common set of resources, formats and services that can be used by tools in this domain. From the information modeling perspective OSLC domains are covered in the STEP standard but sometimes with different names. Application of STEP standard for the purpose of Requirement Management, Modelling & Simulation, Product configuration, Maintenance feedback has been tested by the Organization for the Advancement of Structured Information Standards (OASIS) by development of templates and reference data libraries on top of STEP AP239 to represent concepts in the afore-mentioned domains [12,13]. However, OSLC does not include any particular domain considering production engineering yet such as manufacturing process representation, layout design.

Openness: is the ability of referencing resources from different specification or referenced specifications. In that sense STEP does not have an “open model”, by which we mean that the data instantiation must be done according to an information schema in different protocols and application modules. However, OSLC generally avoid constraining a resource reference to be a resource in another OSLC specification. OSLC specifications do not constrain the value of a property in one OSLC specification to be the URL of a resource in a different OSLC specification either. That means any resource can be contained in a ServiceProvider, not just resources defined in OSLC specifications. A reference in one OSLC resource may in general point to any HTTP resource, not just an OSLC resource.

The STEP advantages can be summarized as:

- Involvement in the ISO community.
- Existence of templates, reference data library and domain ontologies in some disciplines in STEP, which can be reused to define new OSLC resource.

The OSLC advantages can be summarized as:

- The simplicity of the specification.
- The OSLC Core which specifies the primary integration techniques and services for integrating lifecycle tools.
- The lightweight integration mechanism (RESTful services)
- The possibility of using internet-based devices to share data.
- The ability of RDF to represent properties which are not part of the OSLC specification. It is expected that any particular resource may have many more properties that are not defined in an OSLC specification.

4. Integration approach

Decision making concerning the integration approaches and technologies is a sub process of the design of the IT system architecture. The design of the IT system architecture mainly includes making decisions concerning the IT tools that must be used to support business processes (within the scope) and how these IT tools are integrated and how they communicate. The following sections summarize the most important actions to identify the integration strategies, scenarios, technologies, and file formats.

4.1. Define what flow to support? What information is required?

In this step three questions must be answered. 1. What do we want to achieve? (Goal and scope) 2. What processes should be carried out? 3. Which tools are involved to support these processes to accomplish the defined goals? In this step overall problem is partitioned into the domain specific ones, for instance our case (Section 5) includes process planning, material flow simulation and plant layout design. Hence, product, process and resource properties are spread among a set of disparate models. Basically in this step sources and requirements to creation, deletion, update and read of data must be identified. This requires both an understanding of the domain and application software. Developing activity models or sequence diagrams will help to identify goal, scope and business process sequences and their interaction. Different viewpoints involved in the development process must be taken into account to identify information requirements and level of granularity.

4.2. Identify standards for integrations

Using the identified requirements from the previous step (4.1) and OSLCSTEP comparison in Section 3, users can identify if light weight integration is enough or STEP standard is required. For instance; if import/export functionalities is required or whenever detail geometry and geometry closely related data such kinematic media connection point of
machines should be exchanged. OSLC is preferred when sharing a minimalistic set of data is preferred rather than exchanging a whole data model of a particular domain. It is important to note that the decision concerning sharing data versus exchanging data is influenced by the users’ viewpoints which in turn is affected by business process and requirements identified in the previous step. Figure 1 illustrates an example different integration approaches for plant layout and material flow data integration. Full integration of the 3D layout model with material flow data is not necessary for analysis purposes but it is important for the purpose of visualization.

4.3. Using two standards complementarily

In this section tool chain integration via STEP information model is studied by creating OSLC resources and OSLC core for creating services. The idea is to use the STEP information model in order to define OSLC resources, their interrelation and resource shape. ResourceShape allows the specification of a list of properties with allowed values and the association of that list with a resource (RDFs Class). The following steps can help to implement this idea.

1- Analyse individual domains to create set of domain-specific concepts with properties and relations.
2- Identify of what information is common among domains that must be shared.
3- Decide about the level of granulaty of shared data according to the requirements.
4- Use the STEP information model and the required reference data library to structure the data, achieve an agreement on the data model and the semantic of data among different disciplines involved in the development process.
5- Identify OSLC resource and resource shape according to previous steps.
6- Use the OSLC core model to define the services for the identified resources that must be shared.
7- Use the structured data in Step 4 for persistent storage of data or reporting when it is required. Figure 2 is an illustration of the suggested method.

5. Case study

The aim is to illustrate the proposed integration concepts presented, in layout design and material flow analysis, as the case study. The idea is to integrate data and IT-tools in a way that various applications (services) can act on the data in parallel, enabling faster feedback between activities in layout design. The IT environment in this case is heterogeneous including Autodesk Inventor for plant layout development, SimJava for material flow analysis and ARAS Innovator for coordination the data. In the first step an activity model was developed. The activity model defines required processes and their interactions in terms of input/output data. The activity model forms the basis for the system architecture used in the case study, shown in Figure 3. At the top we have the three authoring applications (AAs) including Autodesk Inventor for layout design, SimJava for material flow simulation and Siemens Process Simulate for 3D assembly simulation. Next we have an intermediate layer, Application File Managers (AFM) which manages versioning the files from the AAs. Since 3D CAD models intrinsically include complex relationship between applications files it is more convenient for modellers to use vendor’s AFMs which are tied to the AAs. In the case 3D CAD model files such as plant layout are coordinated through Autodesk Vault to manage revisions and states. The PLM layer (ARAS Innovator) manages the complete factory model, and including files and items and their attributes and relationships. Here the persistent data is stored: a database for item data, and a vault (generally a protected file area) for the files. The consolidated CAD model in STEP standard format is stored in Aras. This allows users to work concurrently on data. From analysis of the domains of process planning, layout, and material flow simulation, the ontologies of domain-specific concepts with properties and relations are created, as well as an identification of what information is required form other domains. Consequently a set of minimalistic concepts with properties and relations to be shared has been identified which is outlined in Figure 4.
The next step is to decide on the level of granularity of shared data. As mentioned earlier this is decided according to scope and requirements of particular users in a domain. For instance, whether the flow analyst needs cycle times for each of the processes of a part feature or just the total cycle time of one set up; or if the footpath of an operator in the layout is required for the flow analysis or not. In this case study, the sequence of operations, the cycle time of each operation, and set up times were selected.

Another issue concerns managing the use of diverse vocabularies in different domains. For example; using terms such as operation, task and process differently when representing a transformation activity on a work piece may result in misinterpretation among users. Hence, the involved disciplines need to agree on the meaning of the shared concept and the terminology to facilitate communication and sharing data.

In the next step the data structure is identified according to STEP. Identification of the data structure is used to identify Resource and ResourceShape in OSLC for tool integration. It can also be used for persistence storage of data (see Figure 2). In our case STEP AP 214 has been selected [14]. Figure 5 illustrates a very simplified representation of product, process and resource, process plan, sequence of manufacturing processes and their properties in STEP application protocol 214.

Finally to define the services and subsequently develop tool adaptors in the tool chain of the case, the scenarios have been identified. Figure 6 is a simplified UML sequence diagram which illustrates the interaction and order of data creation, read and update in the case study. In other word it identifies where and when data are created, updated, deleted or read. Aras Adaptor ServiceProvider is a collection of product_process_plans, their corresponding manufacturing operations, and equipment and required properties. These three resources are RDF classes, which by definition are rdf: type in RDF specification. The tool adaptor provides read and query, services for all three resources. Flow analysis tool is the consumers of these services. For example, the flow analyst gets the process plan of the desired products, then performs the analysis and updates the key performance indicator (KPI) of the equipment such as utilization. Figure 7 demonstrates a small part of the interface of the tool adaptor.
Subsequently, the ServiceProvider was developed for Autodesk Inventor. Two important types of factory layouts are; machinery layout and media layout. These two layouts are developed by different disciplines. A machinery layout usually includes the geometry models of manufacturing resources including dimensions, outer shape, position and orientation of ports for media. The machinery media data are the key information for developing different media layouts (ventilation, electricity etc.). The tool adaptor in this case represents the machine technical data which is integrated with the machinery geometry in Inventor.

However representation of some geometry related properties such as the position and orientation of a media connection point must be visualized to be fully understood by the other disciplines, Hence Inventor adaptor links the Equipment resource to the URL of the Equipment in the cloud asset of the ARAS adaptor. Thus the user can also visualize the shape aspect of a resource. Figure 8 demonstrates the resource representation according to OSLC on the left that is linked to the resource in the cloud asset of the Inventor. Finally, a tool adaptor has been developed for SimJava. After running the simulation the KPI of the Equipment will be created and linked as a resource property in ARAS adaptor. OSLC offers an additional style of integration that solves based on the concept of a dialog. The idea of a dialog is that instead of the development of UI for creating or selecting a resource in a tool service provider the tool asks the resources of the other tool to display to the user a “dialog” from its own user interface. Figure 9 shows an example of this integration style. The flow simulation tool is provided by a dialog to create and a dialog to select equipment in ARAS Service Provider. This integration style not only prevents extra work for UI implementation but also helps users to find all detail about the current resources in a tool.

Fig. 8. Representation of Machine data card via OSLC.

Fig. 9. Delegated UI between SimJava and ARAS Adaptors.

6. Conclusion

The multi-disciplinary nature of layout development results in the creation of heterogeneous, yet interdependent information from various actors. Therefore, the task of communicating and integrating this information and the IT tools are important.

STEP standard has shown a strong capability for communication and exchange of the information in a system neutral format.

When integrating information with different scope, levels of detail and consequently IT tools using OSLC, the primary objective is to standardize the information models that represent the minimalistic set of data that needs to be shared.

From a practical perspective for integration, STEP and OSLC can be used in a complementary way to facilitate data communication and tool integration with a minimum of effort for system integrators or end users. Hence, this paper introduces the idea of incorporation of two approaches in the collaborative layout design. STEP can be used to incorporate a data structure that represents processes, product and manufacturing resources information, and their interrelations. This common model can be used to define OSLC resources and their structure. Then OSLC can be used when developing services to share this data. Hence system integrator and end user can benefit from such combination.

However to use these two approaches there is a need for development of a formal method to define OSLC resources by reusing STEP standard.

References

[1] Ackermann J. Approaches for planning and operation of adaptable factories. International Journal of Production Research; 2013. p. 4618-4629.
[2] Shariatzadeh N, Sivard G and Lindberg L. An approach for manufacturing process representation in product lifecycle management. Key Engineering Materials; 2014. p. 239-244.
[3] Shariatzadeh N. Information requirement and representation for concurrent material flow analysis and layout design. Stockholm : Licentiate thesis, Royal Institute of technology; 2014.
[4] Shariatzadeh N, Sivard G and Hedlund M. Material flow data representation and integration based on STEP. Interworking Indonesia Journal; 2013.
[5] Open-services.net. OSLC Primer - Open Services for Lifecycle Collaboration. Available at: http://open-services.net/resources/tutorials/oslc-primer/; 2015.
[6] Open Services for Lifecycle Collaboration, http://open-services.net/2015.
[7] Lee B. Linked data design issue. Available at www.3.org/designissues/linkeddata.html; 2006.
[8] Ryman A. Linked Data, RDF, and OSLC Resource Shapes: Define REST API Contracts for RDF Resource Representations; 2015.
[9] Fielding R. Representational state transfer, Architectural Styles and the Design of Network-based Software Architecture; 2000. P. 76-85.
[10] OSLC Core Specification Workgroup. OSLC core specification version 2.0. Technical report, Open Services for Lifecycle Collaboration; 2010.
[11] Raymond G. STEP: The grand experience. National Institute of Standards and technology; 1999.
[12] OASIS PLCS TC. DLXML for PLCS, Review release, available at http://www.plcs.org/plcxml/plcxml; 2012.
[13] ISO 10303-239: Industrial automation systems and integration: Application protocol: Product Life Cycle Support; 2 ed.; 2011.
[14] ISO 10303-214: Industrial Automation Systems and Integration— Product Data representation and exchange Introduction. J. TS 1848SC4184; 2009.