Version handling (proposal).

Introduction

A version is a very important concept in managing the lifecycle of products and system, explicit OSLC support for versioning is lacking in today's OSLC Specifications(Core, CM, AM, RM), there is some coverage in the SCM Spec. Whilst noting it is possible to apply existing Core, CM, AM and RM OSLC properties, for say, version identification, the aim here is to create an explicit version handling approach for OSLC resources.

To support ALM or PLM, indication that a resource is a version of another resource is a common way of denoting some difference or variation, such as a change of state, between the two resources. The business purpose of which is to shift the focus of the work done by contributors, denote a specialisation of product or system content say for a given customer or market application or signal some significant event (like achieving a certain condition) to trigger alternate processing by the business. Here the concern is the product or system definition and so changes of state arise from work done or selection of compositional options; options can be logical and physical or some combination.

To achieve coherency within PLM typically all significant business items (products, systems, components, documents etc) undergo versioning as part of change and configuration management including variant handling. Companies build up business rules so that stakeholders know what conditions need to be fulfilled to declare a version as available to some criteria or how to react when such signals are received.

It is worth noting that typically for complex products and systems the exact state of a resource is more ambiguous and is usually designated as being available to some criteria by assessment, i.e. approval. This ambiguity allows for late customisation, thus reducing the effort to build up and maintain many similar configurations in parallel throughout the lifecycle. Whereas within ALM for software the availability of a version may be more readily signalled by way a simpler test, like an error free build. However for software with many options and parameters such simple tests are insufficient and then ALM and PLM needs become similar - that is version readiness for consumption is assessed by domain rules and criteria.

From an OSLC perspective alternate versions of OSLC resources, like a product, allow specific resource states, as product versions, to be linked to, and by, other resources like requirements, AM resources and SCM baselines. Similarly for
requirement versions to be associated with each other or AM resource versions or SCM baselines, and so on. In general most OSLC resources undergo changes which may:

1- require re-evaluation of the composition of the OSLC resource

2- invalidate the purpose of a link between OSLC resources

3- require new links to be established between OSLC resources

Therefore a common way to define the behaviour of an OSLC resource as a version of another OSLC resource is needed. This proposal formalises basic OSLC resource version behaviour, and proposes that it be made explicit in extensions to the OSLC Core Specification.

**Need for versioning with the target scenario**

Consistent versioning support is necessary for OSLC to be able to support the target PLM scenario [A systems engineer responds to a change in requirements.](#)

The scenario begins with a product (or system) and associated requirements and their associated, satisfying, implementations at some defined state; a change request is applied (i.e changes are made to relevant and impacted items) and their version identification are updated to signal completion of the change, with approvals. Version handling of the OSLC resource used in the scenario is needed, namely CM, AM and RM resources plus the new proposed PM resource that supports Products, Product Versions and Product Views.

![Diagram](#)

**Additional notes on the scenario:**

Typically all items are under version control including the Change Request which may evolve during its evolution and approval.

The system or product context is the prevailing configuration, including variants.

The state of requirements and implementation is simplified as multiple constituents will go through various and multiple revisions before State 2 is achieved.

**Resource Terminology and usage**

**Base resource**

A base resource is an optional resource which act as a reference for one or more versions and may be able to generate additional resources which are versions of this resource.

May be superceded by another base resource as the resource history evolves
May generate versions, and additionally optional domain views, from criteria (not specified yet here, see the Draft PLM Specification for examples.

A base resource may maintain and make available a full version history for resource versions that it has generated.

**Resource version**

A resource version is a resource which is a revision or variation of another resource, either or both, a base resource or another versioned resource.

So it may indicate some difference, or variance, from some other related version resources e.g. from a similar basis. It may be transient, or it may be worthy of sustaining for some time, or indeed forever.

**Case 1: Basic versioning capability**

![Diagram of resource versioning](image)

- Resource A may be a base resource
- Resource B is a version of Resource A

**Case 2: Multiple versions with parallel lives**

![Diagram of multiple resource versions](image)

- Resource A is a base resource
- Resource B is a version of Resource A
- Resource C is another version of Resource A

**Fig2: Overview of the OSLC Core versioning proposal**

**Additional notes on the examples in the diagram:**

In Fig2 Case 1 Resource A may or may not be a base resource. The advantage of the base resource concept are that the full version history can be readily provided and that resource A can provide configuration services to specify the versions, such as by variant determinants and expressions; which will be addressed later.

In Fig2 Case 2 Resource C may or may not supersede Resource A, this is a business decision and will be addressed further here.

**Specification of versions and their status**

The proposal here does not specify how one version is distinguished from another, other than by way of being a separate resource. It is up to the service provider or consumer to do this. Additional proposals are being prepared for managing variation of resources.

**Handling of succession**

Control of availability or designation of significance like "release", "latest", "use this one" is an orthogonal concern to creation and maintaining version history. For instance the designation of succession is a separate concern to denoting the basis of a version. The relationship needed is "isSuccessorOf" however it is proposed to adopt the dcterms:replaces as this includes succession and supplanting of one thing for another, in this case resource versions.
Properties

These are the proposed mandatory and optional properties to added to an OSLC Resource to make it a Version or View to associate it with the base Resource, examples of

OSLC resource type are e.g ChangeRequest, Requirement, (AM) Resource

Versions of a base resource are associated with that base resource by way of a new property adopted from dcterms:isVersionOf

A version can signify that it replaces one or more versions of the same base resource by the use of a new property adopted from dcterms: replaces, to show that this Resource version succeeds another version.

- **Name**: Resource
- **Type URI** [http://open-services.net/ns/oslcdomainname#](http://open-services.net/ns/oslcdomainname#) Resource

<table>
<thead>
<tr>
<th>Prefixed Name</th>
<th>Occurs</th>
<th>Read-only</th>
<th>Value-type</th>
<th>Representation</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rdf:type</td>
<td>one - or - many</td>
<td>unspecified</td>
<td>Resource</td>
<td>Reference</td>
<td>n/a</td>
<td>New resource type of Version and/or View added to existing OSLC Resource Types</td>
</tr>
<tr>
<td>dcterms:isVersionOf</td>
<td>zero - or - one</td>
<td>yes</td>
<td>Either</td>
<td>Either Reference of same type</td>
<td>A related resource of which the described resource is a version, edition, or adaptation. OSLC usage requires the target resource <strong>MUST</strong> be a resource of the same type as the owning resource.</td>
<td></td>
</tr>
<tr>
<td>dcterms:replaces</td>
<td>zero - or - many</td>
<td>unspecified</td>
<td>Either</td>
<td>Either Reference of the same type</td>
<td>A related resource that is supplanted, displaced, or superseded by the described resource. OSLC usage requires the target resource <strong>MUST</strong> be a resource of the same type as the owning resource.</td>
<td></td>
</tr>
</tbody>
</table>
## Version Creation

OSLC base service providers **SHOULD** support [Creation Factories](https://www.oslc.org) for creating versions and list them in the Service Provider Resource as defined by OSLC Core. OSLC base resource service providers **SHOULD** support [Resource Shapes for Creation Factories](https://www.oslc.org) as defined in [OSLC Core Specification](https://www.oslc.org).

### Related References

- [OSLC SCM draft v1.0 Specification](https://www.oslc.org)
- [OSLC PLM draft v1.0 Specification](https://www.oslc.org)

### Appendix A - Sample versioned OSLC Resources

<table>
<thead>
<tr>
<th>Resource type</th>
<th>OSLC Ref model example</th>
<th>Scenario version</th>
<th>Usage example</th>
<th>Relationships</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM</td>
<td>AMG60104 POWERSUBSYSTEM</td>
<td>Base</td>
<td>Base AM Implementation Resource</td>
<td>Optional hasVersion UR2 (Non-preferred property)</td>
<td>AMG60104</td>
</tr>
<tr>
<td>AM</td>
<td>AMG60104/004-POWERSUBSYSTEM</td>
<td>004</td>
<td>AM Implementation Version Resource</td>
<td>isVersionOf URI1</td>
<td>AMG60104_004-POWERSUBSYSTEM</td>
</tr>
<tr>
<td>CM</td>
<td>ECR-000031</td>
<td>Base</td>
<td>Base Change Request Resource</td>
<td>Optional hasVersion UR2 (Non-preferred property)</td>
<td>ECR-000031</td>
</tr>
<tr>
<td>RM</td>
<td>REQ-20188</td>
<td>Base</td>
<td>Requirement base resource</td>
<td>Optional hasVersion (Non-preferred property)</td>
<td>REQ -20188</td>
</tr>
<tr>
<td>RM</td>
<td>REQ-20188 - A</td>
<td>A</td>
<td>Requirement version resource</td>
<td>isVersionOf REQ20188</td>
<td>REQ -20188 - A</td>
</tr>
<tr>
<td>RM</td>
<td>REQ-20188 - B</td>
<td>B</td>
<td>Requirement version resource</td>
<td>isVersionOf REQ-201188 replacesREQ-20188-A</td>
<td>REQ -20188 - B</td>
</tr>
</tbody>
</table>