OSLC Configuration Management

> *DRAFT – This is not yet an official OSLC workgroup - To be discussed with workgroup and Steering Committee*

Terminology

Configuration: a resource that identifies a set of versions of some other resources.

Configuration management: the practices of managing configurations, their contents, their lifecycles - in particular, identifying and controlling changes to configurations.

Charter

Produce an RDF vocabulary and associated semantics for configuration management of linked data, capable of addressing the scenarios described below.

In order to keep the specification easy to adopt yet of broad value and appeal, the workgroup is encouraged to avoid constraining the nature of the resources in a configuration, but rather embrace the open principles of linked data, make appropriate use of existing or developing standards such as the Linked Data Basic Profile, refer to other existing vocabularies if appropriate, and be sparing in the number of resource types and REST services required.

Capabilities

The workgroup shall determine the capabilities required to execute the scenarios described below. It is expected, but not absolutely required, that those capabilities include:

- Create and update configurations
- Identify the version of a given resource within the context of a configuration
- Establish a baseline (snapshot) of a configuration
- Describe the contents of a configuration in terms of sets of changes with respect to some other configuration

Other capabilities might be added to this list, if the workgroup decides they are required for full execution of the elaborated scenarios. For example, a client might need to determine the history of the versions of a resource.

Background

The OSLC SCM workgroup produced a draft specification for Software Configuration Management, addressing some code browsing scenarios with read-only access. That specification had very limited adoption, partly because the specification was quite complex, involving 16 resource types and several extensions to OSLC core

standards, and partly because the specification was seen as addressing only a very narrow range of use cases and problem domains.

During the development of the OSLC SCM draft specification, the OSLC Core workgroup held several discussions about baselines, but did not complete a baseline extension or separate specification. This Configuration Management workgroup shall address these previously discussed scenarios. It is the intent that the new specification will be less complex than OSLC SCM 1.0, while at the same time enlarged in applicability, hence increasing the perceived benefit.

Compatibility

Compatibility with OSLC SCM 1.0 is desired, but not required. An addendum to the SCM 1.0 specification shall be produced describing how to migrate usage of that specification to this Configuration Management specification.

Interactions with other workgroups

The Configuration Management workgroup must collaborate with the PLM-ALM workgroup, producing a specification that satisfies, partially satisfies, or is at least compatible with, the versioning requirements expressed by that workgroup. In order to achieve this, the Configuration Management workgroup must work with the PLM-ALM workgroup to agree on the requirements, scenarios, and the resulting vocabulary.

Scenarios and Requirements

This section is an outline of the likely first scenario, provided here to give more context to the workgroup charter; the workgroup shall elaborate the scenario as part of its initial work.

The primary scenario to be developed and addressed is that of a cross-tool and cross-domain baseline, by which we mean a snapshot in time of the state of a set of resources, including the links from those resources.

The specification must also provide low-cost means of integrating tools that already have an existing versioning system, possibly including configurations and baselines, without requiring such tools to re-implement those capabilities.

As noted above, the resources to be managed by this cross-tool, cross-domain service may include configurations or baselines defined by existing tools; as such, the service provides composite or aggregate configurations and baselines.

Use cases shall include, but not necessarily be limited to, the following:

- As a user of a set of OSLC providers, I want to establish a consistent snapshot of the state of resources (a *baseline)* across all those providers, so that I can record this state for future review and audit.
- As a user of other OSLC services such as Requirements Management, Change Management, and Quality Management, I want to have traceability of requirements, change requests, and test suites or test cases to the corresponding set of related or dependent changes to other resources (a *change set*).
- As a consumer of a set of OSLC providers and their resources, I want to determine if a given change set is or is not included in a given baseline.

- As a quality engineer, or an auditor, I want to compare two baselines so I can see the differences between the two sets of resources and their properties, and hence assess the test needs of, and risks in, the new baseline.

As an example, a composite baseline of OSLC resources might include:

- An oslc_rm:RequirementCollection and the set of oslc_rm:Requirements in that collection, describing the requirements for a particular release of some system
- A set of oslc_am:Resources, describing the use cases and elaborating the designs for the same release of that system
- A set of oslc_qm:TestPlans, oslc_qm:TestCases, oslc_qm:TestResults, etc., describing the tests for the same release of the system

Deliverables

- Elaborated use cases
- An OSLC Configuration Management specification
- RDF vocabulary documents
- A primer with examples
- Test cases
- An Addendum to OSLC SCM 1.0 describing migration strategies

Schedule

1. Use cases and requirements defined by the end of October 2012
2. First draft specification written by the end of December 2012
3. Draft OSLC SCM migration strategy published by February 2013
4. Specification in convergence by the end of March 2013
5. Finalization in June 2013