

Review of <http://open-services.net/wiki/core/Exposing-arbitrary-actions-on-RDF-resources/>

Review v 0.1 by ian green, as at 17th july 2014.

## Part I

# General comments/questions

1. I find it easier to read and review these documents when sections are numbered. Is this possible?
2. Is there any guidance on how an automation provider would offer automation around resources of another provider?
3. It seems that having AutomationRequests (and Delegated UI dialog for later execution and Automation Creation Factory) defined in the Actions spec creates a pair of dependencies which gives the impression that the two specs are mutually dependent. Is there a reason for this arrangement?
4. All the other OSLC 2.0 specs REQUIRE support for application/rdf+xml, but my reading of section “Resources” doesn’t seem to require that. Does the requirement for rdf+xml come from some OSLC Core dependency? To be clear, I think rdf+xml support has to be mandatory.
5. This spec uses http:Request as a way of describing a “class” of acceptable requests - for example, there can be more than one http:body and more than one http:httpVersion value. The intention is that the consumer will cherry-pick these properties and build an actual http request. Is the semantics of this cherry-picking defined? Would it not be simpler, and easier to specify, to require that each osc:binding of type http:Request specifies a single http:Request rather than a “class” of such requests? This would avoid the need for Action-specific interpretation of http:Request resources.

## Part II

# Typos, Clarifications

## 1 Section “Terminology”

Suggest “is authority of” rather than “creates” as definition of “Provider”.

## 2 Section “Domain-specific consumers”

1. Do actions get “consumed” – I was expecting “executed”?

2. Suggested rephrase: “For each interaction pattern that is supported by the consumer (the interaction patterns supported by the consumer, along with any restrictions, were if any, defined by the profile(s) they were implemented against):”
3. Typo: “If more than one binding is compatible with the consumer, choose one using whatever criteria the consumer prefers. e.g. it can have an order of preference (decided at implementation time) based on the interaction patterns’ cost of execution (e.g. time, resources), or the desired interaction with the user (e.g. dialogs, or ability to execute at a later time).”

### 3 Section “Description”

1. Is there a better name for this section?
2. “The working groups defining them **MUST** consult with OSLC Core before defining new profiles, to determine if they are better added to Core or kept separate.” This text seems too much like OSLC workgroup organization rather than being part of the spec. Suggest it be removed?
3. Suggest rephrase: “Domain specifications re-using this specification **MAY** define new predicates beyond the ones here that link to actions not ~~currently~~ **available** defined in this specification. For example, as OSLC Automation 2.1 does.”
4. Not sure I see the need to point out that hash uris are permitted. this is basic rdf. “Hash URI” is not a defined term in the RFCs, is it?

### 4 Section “Types of actions”

1. More recommendation on how OSLC workgroups should operate and I think this is inappropriate material for a spec.
2. Why is subclassof recommended given that clients may not do inferencing? Providers of course can make such assertions if they choose but this spec. should **REQUIRE** that the rdf:type be directly asserted on the resource.

### 5 Section “Instructions for executing...”

1. In “Each currently available action contains one or more oslc:binding predicates, each of which links to a resource containing instructions for executing the action, such as the HTTP request resource (http:Request) shown below; the resource shape allows zero bindings in other cases.” what does the final clause after the semi-colon refer to? Why is resource shape being mentioned here?

## 6 Section “Interaction patterns”

I'm not sure why there is a switch to all caps section titles. Am i missing something?

1. “Interaction patterns’ final status” should be “Final Status”, following the style of peer sections.
2. “These are the URIs that this specification defines as values for the `oslc:finalStatusLocation` predicate, ...” suggest rephrasing to “This specification defines the following values for `oslc:finalStatusLocation`, ...”.
3. “and is included in the patterns’ recognition rules” should be “and is included in the pattern’s recognition rules” apostrophe in the wrong place.
4. `oslc:ActionDialog` is not a class - why does it have an initial capital?
5. Suggestion: “The consumer constructs the HTTP body by serializing the resource that is the ~~target~~ object of the `rdf:value` property of the...”
6. “`oslc-automation: canceled` means that the dialog was cancelled, whether or not it was attempted” to make sense of this, I understand a provider can start “it” – i.e., some automation, but allow the user to cancel that execution before completion. Is that right? Perhaps a non-normative note to help the reader?

## 7 Section “Resources”

1. Is there guidance on giving a title to a binding? I can see a confusion around the title of the action and the title of the binding, e.g., when the consumer needs to show these to a user for them to select.
2. “Before defining new subtypes, please coordinate with the community.” ditto above. Such comment doesn’t belong in this spec.
3. What is the status of the red text in the section on Request resource? The cited spec is a working draft, and it it contradicts the description this spec gives to `requestURI`. I don’t see the value in departing from the working draft on this point.
4. The description of `oslc:label` on the results resource refers to “action dialog” which I think is not wanted.
5. `ParameterInstance` `rdf:value` property may be a literal, let the rules for forming the `httpRequest` do not cater for content-type in this case. The diagram indicates that the `rdf:value` will be a resource (intending to exclude the RDF literal case, I suspect). There seems to be some mismatch here.

## 8 Section “Specification Profiles”

1. The wording “To meet the constraints of this profile, Action bindings MUST use one of the following Core Actions’ interaction patterns:” has a reading that all action bindings of a provider must use one of the specified interaction patterns. But I think the intention is that a provider supports a profile, and this means only that *some* of the actions offered by the provider need to use one of the specified interaction patterns. A provider can support multiple profiles simultaneously by offering distinct actions which are compatible with each supported profile. Did I get this right?
2. I think the naming/identification of profiles needs to be less verbose. I suggest an identifier - even if it is not to be used programmatically.

## 9 Section “Appendix A”

1. The rules for what a consumer MUST do to make an HTTP request use “... the...” even though the properties in question are not necessarily unique. The rules need to account, for example, for what to do when both 1.0 and 1.1 httpVersion is specified. See my question in Part I above.
2. There is no mention of what content-type to choose (although this is mentioned as part of the HTTP request with fixed body pattern), for example, what content-type to specify when the rdf:value is an RDF Literal (see q above about rdf:value on a ParameterInstance resource)
3. “specifies the http:requestURI value as a URI, NOT a literal” suggests that a literal value for requestURI is permitted by this spec, yet that is excluded by the description of the usage of this property in the section “Resource: Request”.
4. I think the “Standard restrictions” should be given a precise identifier, even if only for humans to use. Additionally, suggest “specifies “1.1” as the one and only value of the http:httpVersion property”. The wording “Consumers supporting this profile MAY decide not to implement...” refers to “this profile” yet a profile is not being defined.