RESTful Actuation

Interaction Model for Actuation using the REST Design Style

Actuation and Actions

- Actuation is the act of changing the state of something in the physical world
- Coupling to the physical world introduces some new requirements, and suggests a new general class of Actions
- Actions may be parameterized, for example delay time or transition time/effect
- Conveniently maps to commands and methods

Examples of Action Types

- setBrightness(targetValue, transitionTime)
- changeValvePosition(position, time)
- changeState(targetState, delayTime)

RESTful Actuation

- Map Actions onto the REST design style
- Need a way to invoke an Action and supply parameters
- Actions could be long-running
- Invoked Actions should be trackable and modifiable
- CREATE instances of Actions as first class REST resources that represent the current state of the invoked Action

ActionInstance

- ActionInstance is a resource type that represents the state of an invoked Action
- The Action is invoked by CREATE of an ActionInstance resource with a payload representing the parameters
- The CREATE operation returns a location value pointing to the created actionInstance
- This resource represents the current state of the invoked action, through completion or failure
- ActionInstances may be observed for state changes and progress reports

ActionInstance Collections

- ActionInstances may be created as subresources of a collection
- The collection resource may be associated with a particular capability, e.g. brightness
- There may be pending actionInstances, running actionInstances, and completed actionInstances in a collection
- New actionInstances may or may not pre-empt currently running actions

Example Action Invocation

```
Request:
POST /brightness/actions/
{
  "brightness": 75,
  "transitiontime": 10
}
-------------------
Response:
2.01 Created
Location: 10775f7e
-------------------
Request:
GET /brightness/actions/10775f7e
-------------------
Response:
{
  "brightness": 75,
  "transitiontime": 10,
  "actionstate": "active"
  "remainingtime": 5.3
}
```

Hypermedia Framework

- Action collections may be annotated with application semantics that describe their function in terms of the associated capability
- For example, a brightness attribute for a light may have an actionInstance collection for changing the light brightness in various ways
- Hypermedia forms may be used to describe the different brightness control operations available, e.g. move-to, step, pulse

Example Form


```
{
  "anchor": "/brightness/",
  "rel": "action",
  "type": "move-to",
  "method": "post",
  "href": "actions",
  "schema": [
    {
    "id": "targetValue",
    "type": "number",
    "minvalue": 0,
    "maxvalue": 100
    },
    {
    "id": "transitiontime",
    "type": "number",
    "minvalue": 0,
    "maxvalue": 6553
    }
  ]
}
```