CoRAL

Self-Describing Interactions for Evolvable Internet of Things Systems

Klaus Hartke

2017-09-23

CoRAL

A language for the description of typed connections between resources on the Web ("links"), possible operations on such resources ("forms"), as well as simple resource metadata.

CoRAL is intended for driving automated software agents that navigate a Web application based on a standardized vocabulary of link and form relation types.

- Data and interaction model
- Compact, binary format
  — suitable for constrained environments
- Lightweight, textual format
  — easy to read and write by humans

Discovering Resources: Web Linking

HTML

```
<link rel="next" href="./chapter4">
<link rel="previous" href="./chapter2">
<link rel="icon" href="/favicon.png">
```

HTTP

```
Link: <./chapter4>; rel="next",
      <./chapter2>; rel="previous",
      </favicon.png>; rel="icon"
```

CoRAL

```
#using iana = <http://www.iana.org/assignments/relation/>

iana:next     <./chapter4>
iana:previous <./chapter2>
iana:icon     </favicon.png>
```

Resource Metadata: RFC6690 Attributes

```
// example from page 14 of RFC6690

#using iana = <http://www.iana.org/assignments/relation/>
#using attr = <http://TBD/>

iana:hosts </sensors> {
  attr:ct "40"
}

iana:hosts </sensors/temp> {
  attr:rt "temperature-c"
  attr:if "sensor"
  iana:describedby
    <http://www.example.com/sensors/t123>
  iana:alternate </t>
}

iana:hosts </sensors/light> {
  attr:rt "light-lux"
  attr:if "sensor"
}
```

Resource Metadata: RDF

- Many RDF predicates are actually quite good link relation types

```
// representation of <coap://server/somedocument>

#using iana = <http://www.iana.org/assignments/relation/>
#using foaf = <http://xmlns.com/foaf/0.1/>

foaf:maker null {
   iana:type
         <http://xmlns.com/foaf/0.1/Person>
   foaf:familyName "Hartke"
   foaf:givenName  "Klaus"
   foaf:mbox       <mailto:hartke@tzi.org>
}
```

Discovering Actions: Resource Types

- Resource type indicates a set of possible actions on a resource
- Documentation specifies how to construct request for each action
- Not extensible: can only define a new resource type for new actions

```
// representation of <coap://server/.well-known/core>

#using iana = <http://www.iana.org/assignments/relation/>
#using attr = <http://TBD/>

iana:hosts </pubsub> { attr:rt "core.ps" }
```


https://tools.ietf.org/html/draft-ietf-core-coap-pubsub-02

Discovering Actions: Forms

- Each form identifies a possible action on a resource
- The form specifies how to construct the request for the action
- Documentation specifies which forms must be supported
- Extensible: can simply add new forms for new actions at runtime

```
// representation of <coap://server/.well-known/core>

#using iana = <http://www.iana.org/assignments/relation/>
#using coral = <urn:ietf:rfc:XXXX#>

iana:hosts </sensordata> {

  coral:create -> POST </sensordata> [
    coral:accept "application/senml+cbor"
  ]

}
```

Forms

- Semantics of a form are identified by the form relation type

```
// representation of <coap://server/sensordata>

#using iana = <http://www.iana.org/assignments/relation/>
#using coral = <urn:ietf:rfc:XXXX#>

coral:create -> POST </sensordata> [
  coral:accept "application/senml+cbor"
]

iana:item </sensordata/1> {
  coral:update -> PUT </sensordata/1> [
    coral:accept "application/senml+cbor"
  ]
  coral:delete -> DELETE </sensordata/1>
}
```

Form Data

```
// representation of <coap://server/orders>

#using coral = <urn:ietf:rfc:XXXX#>
#using ex = <http://example.org/order#>

coral:create -> POST </orders> [
  coral:accept "application/order+json"
]

coral:create -> POST </orders> [
  coral:accept "application/json"
  ex:schema <http://example.org/order.cddl>
]

coral:create -> POST </orders> [
  coral:accept "application/json"
  ex:customer  "string"
  ex:total     "int32"
  ex:currency  "EUR|USD"
]
```

https://tools.ietf.org/html/draft-hartke-t2trg-coral-03