Mapping of YANG to DSDL
draft-ietf-netmod-dsdl-map-03

Ladislav Lhotka
⟨*lhotka@cesnet.cz*⟩

Rohan Mahy
⟨*rohan@ekabal.com*⟩

Sharon Chisholm
⟨*schishol@nortel.com*⟩

27 July, 2009

Main changes between -01 and -03 (since IETF74)

❶ Detailed specification of DSRL mapping (default values)

- new annotation attribute `nma:implicit`
- full DSRL schema for the DHCP module

❷ Tracking changes in YANG:

- Removed *float32* and *float64* types, added *decimal64*
- No more **require-instance** for *leafref* type

❸ Section on (avoiding) handling of **deviation** and **if-feature**

❹ Proposed solution for *instance-identifier* type

DSRL mapping

In a DSRL schema the default content of every element (container of leaf) is specified separately in this form:

```
<dsrl:element-map>
  <dsrl:name>ELEMENT-QNAME</dsrl:name>
  <dsrl:parent>
    ... XSLT pattern specifying the parent element ...
  </dsrl:parent>
  <dsrl:default-content>
    ... default content ...
  </dsrl:default-content>
</dsrl:element-map>
```

If the element ELEMENT-QNAME is missing or empty, the default content is applied.

```
...
prefix ex;
...
container outer {
  presence "...";
  container foo {
    leaf bar {
      type uint8;
      default 42;
    }
  }
}
```

```
<dsrl:element-map>
  <dsrl:name>ex:foo</dsrl:name>
  <dsrl:parent>/ex:outer</dsrl:parent>
  <dsrl:default-content>
    <ex:bar>42</ex:bar>
  </dsrl:default-content>
</dsrl:element-map>
<dsrl:element-map>
  <dsrl:name>ex:bar</dsrl:name>
  <dsrl:parent>
    /ex:outer/ex:foo
  </dsrl:parent>
  <dsrl:default-content>
    42
  </dsrl:default-content>
</dsrl:element-map>
```

Implicit containers

A container node is *implicit* if it is added in the process of filling in the defaults.

All implicit containers are marked with `nma:implicit="true"` in the conceptual tree schema.

```
<rng:element name="ex:foo" nma:implicit="true">
  <rng:element name="ex:bar" nma:default="42">
    <rng:data type="unsignedByte"/>
  </rng:element>
</rng:element>
```

Type *decimal64*

```
type decimal64 {
  fraction-digits 2;
}
```

is mapped to

```
<rng:data type="decimal">
  <rng:param name="totalDigits">19</rng:param>
  <rng:param name="fractionDigits">2</rng:param>
</rng:data>
```

deviation, if-feature

As we agreed in San Francisco, the modules have to be pre-processed first to reflect the active features and deviations so that DSDL mapping needn’t deal with them.

Type *instance-identifier*

To validate this type, we need to evaluate (simplified) XPath expressions at runtime. This is impossible in standard XSLT, requires an extension function, e.g. `saxon:evaluate()` in Saxon.

Proposal: Define extension function `nmf:evaluate()` (in the namespace of "NETMOD functions"). Schematron mapping of *instance-identifier* type with `require-instance true;` would then be

```
<sch:assert test="nmf:evaluate(XPATH)">
 Node pointed to by an instance-identifier must exist.
</sch:assert>
```

Status

Draft: Minor adjustments (sync with YANG -07) needed:

- sync with YANG -07
- add required sections (definitions, security considerations)

The document then should be ready for WGLC.

Implementation TODO:

- mapping from conceptual schema tree to DSRL
- transformation of XPath expressions (adding prefixes)
- handling of *instance-identifier* type