Mapping of YANG to DSDL draft-ietf-netmod-dsdl-map-04

Ladislav Lhotka
⟨*lhotka@cesnet.cz*⟩

Rohan Mahy
⟨*rohan@ekabal.com*⟩

Sharon Chisholm
⟨*schishol@nortel.com*⟩

10 November, 2009

Main changes between -03 and -04

❶ Adjustments in element order;

❷ Placement of “orphaned” typedef defaults;

❸ YANG documentation statements MAY be ignored;

❹ NETMOD-specific annotations defined using NVDL;

❺ Editorial changes (required sections added etc.);

❻ New namespace URL for XPath extensions.

Element order

- Children of a container – except in RPC input and output parameters – are enclosed in `<interleave>`, which means arbitrary order. [-yang-08, Sec. 7.5.7]
- Children of a case are enclosed in `<group>` in RPC input and output parameters and in `<interleave>` otherwise. [Not specified in -yang-08, probably should be?]
- Children of a list: list keys come first in the order specified in the 'key' statement, followed by other children enclosed in `<interleave>` - again except RPC parameters. [-yang-08, Sec. 7.8.5]

Placement of typedef defaults

```
typedef foo {
    type uint8;
    default 42;
}
leaf bar {
    type foo;
}
```

```
<optional>
  <element name="tt:bar">
    <ref name="test__foo"/>
  </element>
</optional>
...
<define name="test__foo"
        nma:default="42">
  <data type="unsignedByte"/>
</define>
```

```
typedef foo {
    type uint8;
    default 42;
}
leaf bar {
    type foo {
        range 0..50;
    }
}
```

```
<optional>
  <element name="tt:bar" nma:default="42">
    <data type="unsignedByte">
      <param name="minInclusive">0</param>
      <param name="maxInclusive">50</param>
    </data>
  </element>
</optional>
```

Documentation statements

'description', 'reference', 'status' MAY be ignored.

Reasoning:

- Documentation strings are best interpreted in the context of the YANG module.
- They have no impact on validation with DSDL schemas and other uses of existing XML tools.
- Their presence in the conceptual tree schema may be confusing if the documentation target is not directly mapped (expanded typedefs and groupings).

NETMOD-Specific Annotations

Normative Appendix A now uses Namespace-Based Validation Dispatching Language (NVDL, Part 4 for DSDL) for defining NETMOD-specific annotations as a set of patches to the RELAX NG schema for RELAX NG.

This allows for stating that, e.g., RELAX NG `<define>` pattern may be annotated with `nma:default`, `nma:status` and `nma:units` attributes and nothing else.

Editorial Changes

- Sections “Terminology and Notation”, “Security Considerations” and “Acknowledgements” were added.
- References were split into normative and non-normative.
- CODE BEGINS/ENDS markers were added to delimit *normative* schemas.

New namespace URI

A new namespace URI is registered in “IANA Considerations” for XPath extension functions:

`urn:ietf:params:xml:ns:netmod:xpath-extensions:1`

This namespace URI could also be used in the future for other XPath extension functions that have been proposed for YANG.

Remaining issue

The way how “identityref” values are checked won’t work in general – instance documents needn’t use the standard namespace prefix.

```
module crypto-base {
    namespace "http://example.com/crypto-base";
    prefix "crypto";
    identity crypto-alg;
}

module des {
    namespace "http://example.com/des";
    prefix "des";
    import "crypto-base" { prefix "crypto"; }
    identity des {
      base "crypto:crypto-alg";
    }
}
```

```
module my-crypto {
    namespace "http://example.com/my-crypto";
    prefix mc;
    import "crypto-base" { prefix "crypto"; }
    leaf crypto {
        type identityref {
            base "crypto:crypto-alg";
        }
    }
}
```

The following two encodings of the leaf “crypto” are equivalent but only the former will work with the mapping defined in -04:

```
<crypto
  xmlns:des="http://example.com/des">des:des</crypto>
```

```
<crypto
  xmlns:x="http://example.com/des">x:des</crypto>
```

Proposed solution: new annotation

```
<element name="mc:crypto">
  <data type="QName"/>
  <nma:identities>
    <nma:identity>
     <nma:uri>http://example.com/crypto-base</nma:uri>
     <nma:name>crypto-alg</nma:name>
    </nma:identity>
    <nma:identity>
     <nma:uri>http://example.com/des</nma:uri>
     <nma:name>des</nma:name>
    </nma:identity>
  </nma:identities>
</element>
```

All identities derived from the base identity *in all input YANG modules* MUST be listed.

The “identityref” value is then validated by the DSDL schemas generated in the second mapping step:

1. “QName” datatype is verified by RELAX NG
2. A Schematron rule then checks that the value belongs to the permitted set of identities including full namespace URI.

Conclusion

The draft should be ready for WGLC.

The mapping is able to generate DSDL schemas performing quite strict validation of datastore content, unfiltered get-config reply PDUs and also notifications and RPC requests/replies defined in the input YANG modules.

Validation of other PDUs is a more complicated issue, but it is not specific to DSDL.

Implementation (plugin for the *pyang* tool) is essentially complete except:

- updates are needed to get in sync with -04,
- mapping of 'identityref' (see above),
- translation from conceptual tree schema to DSRL.