
Mapping of YANG to DSDL

draft-lhotka-yang-dsdl-map

Ladislav Lhotka

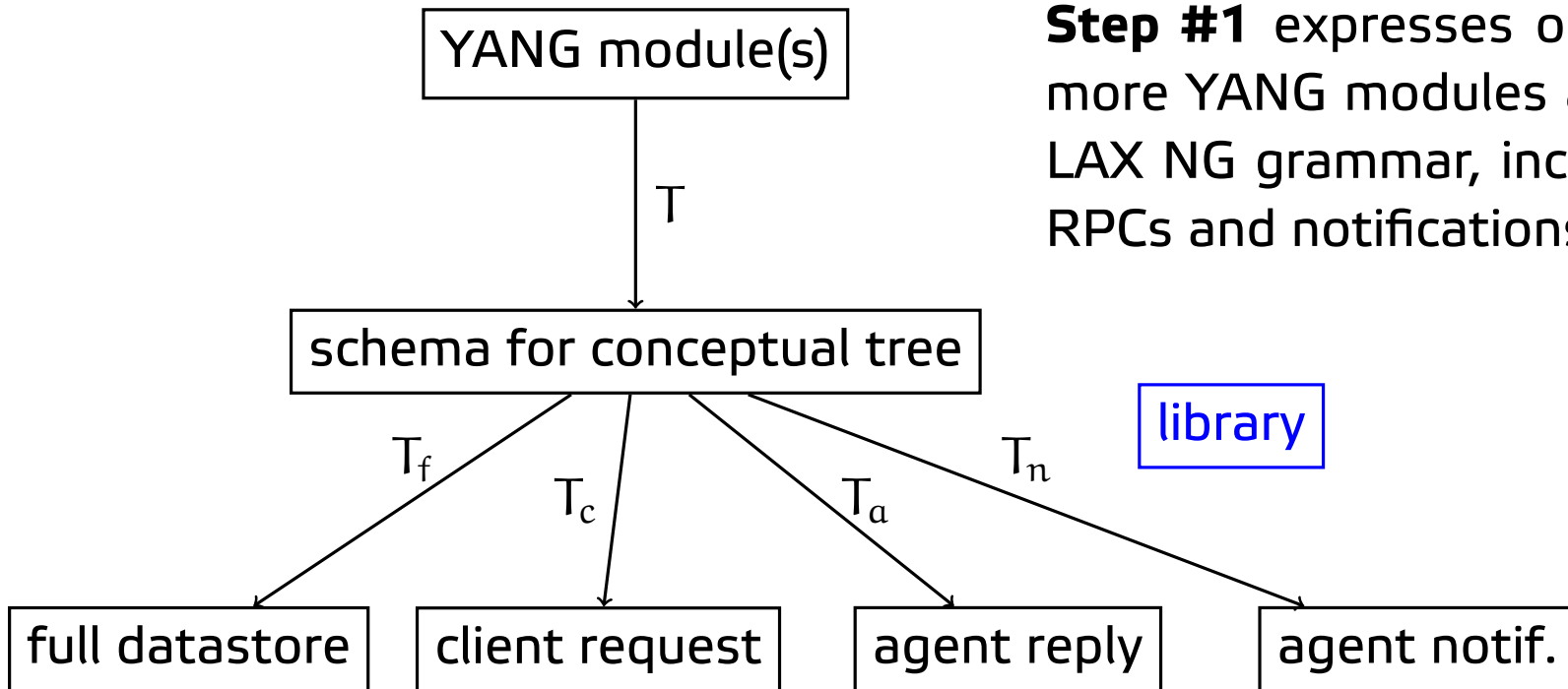
<lhotka@cesnet.cz>

Sharon Chisholm

<schishol@nortel.com>

21 November 2008

DSDL mapping in two steps



Step #1 expresses one or more YANG modules as RELAX NG grammar, including RPCs and notifications.

Step #2 extracts, e.g., via XSLT, validation schemas for particular entities and contexts. This step will use a *schema-independent library* of common definitions and patterns.

Step #1

More than one input YANG module may be used for devices that support multiple data models. The conceptual tree schema will cover their combination.

The result is an annotated RELAX NG grammar expressing the same information as the input YANG modules.

The schema of the conceptual tree will not be directly usable for any validation but rather intended (i) for human readers and (ii) as a common starting point for all transformations in step #2.

Purpose (i) mandates that the schema be reasonably readable.

Proposal: Step #1 will generate a single annotated RELAX NG schema. YANG concepts without counterparts in RELAX NG (default, must, key, unique, when) will be mapped to very simple NETMOD-specific annotations.

Step #2

The result is a coordinated set of DSDL schemas tailored for a specific validation task. In particular, all NETMOD-specific annotations used in the conceptual tree schema will be translated to Schematron rules, `<dsr1:default-content>` etc.

The schema-independent library will contain definition, Schematron abstract patterns etc. that represent common concepts (e.g., parametrized rule for keys) that can be used by the generated schemas.

Validation is NETMOD-wide issue!

How to approach validation is an issue for YANG as well and should therefore be addressed by the WG. Some questions to answer:

1. What is the information model of the datastore content?
2. Granularity of the PDU-validating schemas? The simplest choice is to handle separately manager requests, agent replies and agent notifications.
3. Validation phases? For example: full validation, validation without reference checking and grammatic validation.
4. Other input information? For example: active features, capabilities.

Example conceptual tree instance

```
<nmt:netmod-tree yang-module="dhcp"  
  xmlns:nmt="urn:ietf:params:xml:ns:netmod:tree:1"  
  xmlns="http://example.com/ns/dhcp">  
  <nmt:main>  
    <dhcp>  
      <max-lease-time>7200</max-lease-time>  
      ...  
    </dhcp>  
  </nmt:main>
```

(continued)

```
<nmt:rpc-methods>
  <nmt:rpc-method name="...">
    <nmt:input>
      ...
    </nmt:input>
    <nmt:output>
      ...
    </nmt:output>
  </nmt:rpc-method>
  ...
</nmt:rpc-methods>
<nmt:notifications>
  <nmt:notification name="...">
    ...
  </nmt:notification>
  ...
</nmt:notifications>
</nmt:netmod-tree>
```

Positioning issue 1

DSDL is to be used as (A) data modeling language, or (B) ad hoc DSDL schemas for specific, mostly short-term, purposes such as PDU validation.

(A) doesn't seem to be supported in the NETMOD charter.

Positioning issue 2

DSDL mapping will be developed as (A) mere convenience for those who don't understand YANG, or (B) interim validation method before native YANG tools are written, or (C) substantial component of the NETMOD toolbox.

Proposal: All of the above, (C) implies long-term utility of the DSDL work.

DSDL can be the glue between data models and NETCONF RPC layer.
(Perhaps RFC 4741bis should use DSDL instead of XSD?)

YANG draft specifies the XML encoding rules and future tools may be able to perform PDU validation directly from YANG, but having a formal schema is still useful.

Technical issue 1

Given the structure of the mapping (step #1, multiple branches of step #2 and the schema-independent library), identify the parts that can be done quickly.

Proposal: The first version of the WG DSDL draft should include step #1, schema-independent library for step #2 and one branch of step #2 as the proof of concept – most likely the validation of full datastore.

The WG should start discussing the validation issues mentioned above.

Technical issue 2

RELAX NG syntax: XML versus compact

Compact syntax is really nice only in the absence of annotations. All annotations look essentially the same in the XML syntax (elements or attributes in foreign namespaces) whereas in the compact syntax they have four different syntactic forms.

From the implementation point of view, it is easier to construct an XML tree using one of the common libraries.

Proposal: In the mapping specification, use exclusively the XML syntax.

This doesn't preclude use of compact syntax obtained by automatic translation (e.g., via trang).

Technical issue 3

Annotated RELAX NG (everything in one schema document) versus several stand-alone DSDL schemas.

Proposal: Single schema in step #1, separate schemas in step #2.

Technical issue 4

Pending changes to YANG syntax and semantics (augments, when, ...) may affect some aspects of the mapping significantly.

Let's resolve them soon.

Technical issue 4

Should all YANG module metadata (especially contacts) be simply copied to the DSDL schemas?

The author of the YANG module may not wish to become responsible for the DSDL stuff.

Proposal: Copy just the description and reference strings and refer to the source YANG modules (with a note that the schema is a result of an automatic translation).