Augmented-by Addition into the ietf-yang-library

draft-lincla-netconf-yang-library-augmentation-01

NETCONF
19 March 2024
Brisbane, Australia
Authors: Zhuoyao Lin, Benoît Claise, Ignacio Dominguez Martinez-Casanueva
Presenter: Zhuoyao Lin
The YANG library [RFC8525] specifies a YANG module that provides the information about the YANG models and datastores to facilitate a client application to fully utilize and understand the YANG data modelling language.

With the requirement of automated network management arises, more and more scenarios appear where the **real-time knowledge of YANG dependencies** are needed.

While the deviation is used to understand the API contract of server and client, the augmentation is required to directly understand YANG dependency from ietf-yang-library [RFC8525], and obviously it is **missing** for the moment.

**Import:** Parse from module source code

**Include:** Parse from module source code/<get> request to ietf-yang-library

**Deviation:** <get> request to ietf-yang-library

**Augmentation:** missing
Use Cases where full YANG dependencies are needed

**YANG & Kafka Data Mesh Architecture**

With the schema registry introduce in this architecture to maintain YANG semantics in the whole pipeline, there is a need for it to know the dependencies of each modules, in order to provide the correct and complete module set.

**Data Catalog**

The idea of data catalog is proposed to find the YANG models implemented by a network device, which is paramount for configuring and monitoring the status of a network.

It provides a registry of the datasets exposed by remote data sources for consumers to discover data of interest.

The current ietf-yang-library can tell if the module of interest in installed or not, but cannot tell if there is an augment relation between them.

Libyangpush: https://github.com/network-analytics/libyangpush
Current solution & Limitation

Get-all-schemas

What it is: Fetch all YANG models and parse them to fully understand their relationship.

Limitation: Time-consuming. Hard to reflect module changes. Require extra procession of all modules.

Performance:

<table>
<thead>
<tr>
<th>Device 1</th>
<th>Device 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Around 5 min</td>
<td>Around 3 min</td>
</tr>
</tbody>
</table>

Evaluation: Not ideal in the data real-time ingestion background.

Real-time parse message

What it is: Parse prefix in the message to know the augmentation.

Principle: In the received YANG-push message, fields name with prefix (or element with namespace) are from the augment module.

Evaluation: Dependency missing are highly possible, since not all augments will appear in the first message but it may do in the following.

libyangpush: https://github.com/network-analytics/libyangpush
Values

Ease to implement: with deviation in the ietf-yang-library, the augmented-by will have similar way of working.

Provide full YANG relationship: The user can get full YANG dependency(including the reverse) through a <get> request. Facilitate the use case “Data-catalog”.

Meet with real-time requirement: One NETCONF <get> request is fast and easy to proceed.

Improve the Data-mesh solution: Compared with get-all-schemas, it enables collector to fetch up-to-date data since the queries are sent at run time, while get-all-schemas forces collector to always work with not updated data. On another hand, user do not bother waiting for ten minutes every time when starting collector to either get data updated or to obtain YANG relationship.
Questions

**Question 1: If the deprecated container “modules-state” should be augmented too?**

In the appendix B there is the module for augmenting the ietf-yang-library in RFC7895. The container “modules-state” defined in ietf-yang-library[RFC7895], in ietf-yang-library[RFC8525] is marked as deprecated. Whether the “moduels-state” should be augmented to contain the augment-by list?

**Question 2: Should the module-set name also be modeled as a key?**

Some operator define different module-sets to separate modules. Making the module-set name also as a key can allow the module in one module-set to augment or be augmented by module in another module-set. It could be useful, in condition that there is such need. Otherwise it will be modeled the same way as the deviation, in which the following use case is out-of-scope.

(for Q2, yanglint validation example is provided in Appendix A)
YANG tree of **ietf-yang-library** Plus the augmentation

**ietf-yang-library in RFC7895**

```
module: ietf-yang-library
  +--ro yang-library
    |    +--ro module-set* [name]
    |    |    +--ro name string
    |    |    +--ro module* [name]
    |    |    |    +--ro name yang:yang-identifier
    |    |    |    +--ro revision? revision-identifier
    |    |    |    +--ro namespace inet:uri
    |    |    |    +--ro location* inet:uri
    |    |    |    +--ro submodule* [name]
    |    |    |    |    +--ro name yang:yang-identifier
    |    |    |    |    +--ro revision? revision-identifier
    |    |    |    |    +--ro location* inet:uri
    |    |    |    |    +--ro feature* yang:yang-identifier
    |    |    |    |    +--ro deviation* -> ../../module/name
    |    |    |    +--ro yanglib-aug:augmented-by* -> ../../yanglib:module/name
```

**ietf-yang-library in RFC7895**

```
module: ietf-yang-library
  +--ro modules-state
    +--ro module-set-id string
    +--ro module* [name revision]
      +--ro name yang:yang-identifier
      +--ro revision union
      +--ro schema? inet:uri
      +--ro namespace inet:uri
      +--ro feature* yang:yang-identifier
      +--ro deviation* [name revision]
      |    +--ro name yang:yang-identifier
      |    +--ro revision union
      |    +--ro conformance-type enumeration
      +--ro submodule* [name revision]
      |    +--ro name yang:yang-identifier
      |    +--ro revision union
      |    +--ro schema? inet:uri
    x--ro yanglib-aug:augmented-by* [name revision]
      +--ro yanglib-aug:name -> /yanglib:modules-state/module/name
      +--ro yanglib-aug:revision -> /yanglib:modules-state/module/revision
```
Hackathon Demo  Based on libyang&sysrepo
Run on netopeer2

```
<module>
  <name>ietf-yang-library</name>
  <revision>2019-01-04</revision>
  <augmented-by xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library-augmentedby">
    <name>ietf-yang-library-augmentedby</name>
    <revision>2023-10-27</revision>
  </augmented-by>
</module>

<module>
  <name>ietf-interfaces</name>
  <revision>2018-02-20</revision>
  <augmented-by xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library-augmentedby">
    <name>ietf-ip</name>
    <revision>2018-02-22</revision>
  </augmented-by>
  <augmented-by xmlns="urn:ietf:params:xml:ns:yang:ietf-yang-library-augmentedby">
    <name>ietf-network-instance</name>
    <revision>2019-01-21</revision>
  </augmented-by>
</module>
```
Wrap Up

Team members:
Zhuoyao Lin
<zephyre888@gmail.com>
Benoit Claise
<benoit.claise@huawei.com>
IGNACIO DOMINGUEZ MARTINEZ-CASANUEVA
<ignacio.dominguezmartinez@telefonica.com>

https://github.com/Zephyre777/draft-lincla-netconf-yang-library-augmentation.git

https://datatracker.ietf.org/doc/draft-lincla-netconf-yang-library-augmentation/