A YANG Data Model for Network Hardware Inventory

CCAMP WG, IETF114

draft-yg3bp-ccamp-network-inventory-yang-01

Authors:

Chaode Yu (yuchaode@huawei.com)

Italo Busi (Italo.Busi@huawei.com)

Aihua Guo (aihuaguo.ietf@gmail.com)

Sergio Belotti(sergio.belotti@nokia.com)

Jean-Francois Bouquier(jeff.bouquier@vodafone.com)

Fabio Peruzzini(fabio.peruzzini@telecomitalia.it)

Oscar Gonzalez de Dios(oscar.gonzalezdedios@telefonica.com)

Victor Lopez(victor.lopez@nokia.com)

Chenfang Zhang(zhangcf80@chinaunicom.com)

Major Updates Since IETF 113

➢Enriched more attributes in data model

- Found some common attributes for all inventory objects (uuid, name, alias, description, location)
- **Refined some attributes from RFC8348**
- Enriched more attributes for rack (height, width, depth)
- Enriched more attributes for NE (based on RFC8348)

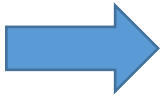
➢Had more discussion on the scalability issue

Refinement of attributes from RFC8348 (1)

A new design of parent identifiers' reference

```
module: ietf-hardware
  +--rw hardware
     +--ro last-change?   yang:date-and-time
     +--rw component* [name]
        +--rw name               string
        +--rw class              identityref
        +--ro physical-index?    int32 {entity-mib}?
        +--ro description?       string
        +--rw parent?            -> ../../component/name
        +--rw parent-rel-pos?    int32
        +--ro contains-child*    -> ../../component/name
```

Definition in RFC8348



```
+--ro components
   +--ro component* [uuid]
      .................................
      +--ro parent-references
      |  +--ro equipment-room-uuid?    leafref
      |  +--ro ne-uuid?                leafref
      |  +--ro rack-uuid?              leafref
      |  +--ro component-references
      |     +--ro component-reference* [index]
      |        +--ro index    uint8
      |        +--ro class?   leafref
      |        +--ro uuid?    leafref
      .................................
```

Refinement definition

- If the client wants to retrieve a component's grandparent or higher hierarchical component, the client needs to do the retrieval step by step which is time consuming;
- To help this process the model provides direct list of hierarchical parent component identifiers.
- The refined solution could support more flexible retrieval function;

Refinement of attributes from RFC8348 (2)

A new extension structure for component-specific attributes

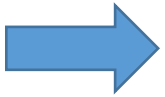
```
+--ro components
   +--ro component* [uuid]
      ...................................
      +--ro (component-class)?
         +--:(chassis)
         |  +--ro chassis-specific-info
         +--:(container)
         |  +--ro slot-specific-info
         +--:(module)
         |  +--ro board-specific-info
         +--:(port)
            +--ro port-specific-info
      ...................................
```

Considering the attributes defined in RFC8348 could not cover all the requirements from operators, and some of the missing attributes could be component-specific, we provide this choice-case structure for component-specific attributes extension. The details in each of these *-specific-info attributes are still under discussion.

Refinement of attributes from RFC8348 (3)

Rename “model-name” to “part-number”

```
module: ietf-hardware
  +--rw hardware
     +--ro last-change?   yang:date-and-time
     +--rw component* [name]
        +--rw name               string
        +--rw class              identityref
        +--ro physical-index?    int32 {entity-mib}?
        +--ro description?       string
        +--rw parent?            -> ../../component/name
        +--rw parent-rel-pos?    int32
        +--ro contains-child*    -> ../../component/name
        +--ro hardware-rev?      string
        +--ro firmware-rev?      string
        +--ro software-rev?      string
        +--ro serial-num?        string
        +--ro mfg-name?          string
        +--ro model-name?        string
        +--rw alias?             string
```



```
+--ro components
   +--ro component* [uuid]
      ....................................
      +--ro part-number?            string
      ....................................
```

Refinement definition

```
leaf model-name {
  type string;
  config false;
  description
    "The vendor-specific model name identifier string
     associated with this physical component.  The preferred
     value is the customer-visible part number, which may be
     printed on the component itself.
```

Definition in RFC8348

Part-number is more recognized term than model-name.

Some open issue to discuss

The design of location:

```
module: ietf-hardware
  +--rw hardware
     +--ro last-change?   yang:date-and-time
     +--rw component* [name]
        +--rw name              string
        +--rw class             identityref
        +--ro physical-index?   int32 {entity-mib}?
        +--ro description?      string
        +--rw parent?           -> ../../component/name
        +--rw parent-rel-pos?   int32
        +--ro contains-child*   -> ../../component/name
        +--ro hardware-rev?     string
        +--ro firmware-rev?     string
```

Definition in RFC8348

```
+--ro components
   +--ro component* [uuid]
      ...................................
      +--ro location
      |  +--ro equipment-room-name?   leafref
      |  +--ro ne-name?               leafref
      |  +--ro rack-number?           leafref
      |  +--ro chassis-index?         uint16
      |  +--ro slot-index?            uint16
      |  +--ro port-index?            uint16
      ...................................
```

Option 1

```
+--ro components
   +--ro component* [uuid]
      ...................................
      +--ro location?            string
      ...................................
```

String pattern:
"/ne=<nw-ne-name>[/r=<r_index>][/sh=<sh_index>[/s_sh=<s_sh_index> ...]][[/sl=<sl_index>[/s_sl=<s_sl_index> ...]][/p=<p_index> ...]]"

Option 2

- Definition in RFC8348: It has only provided a relative position info instead of whole position. The whole position info needs to be retrieved one by one.
- Option1: It provides a hierarchical structure which could support flexible retrieval. But it seems applicable to Optical NE only.
- Option2: It is generic and has more flexibility.
  - Need to get confirmation if using the string pattern applies only to Optical NE or can cover all types of equipment.
  - This option is aligned with Openconfig approach.

Discussion on the scalability issue

Conclusion of the discussion:

- When the networks get big, there would be scalability issue regardless what kind of data model is adopted;
- The same issue could also be found in other integration scenarios, such as topology discovery;
- The root cause of this scalability issue comes from the limited transfer capabilities of HTTP protocol. Maybe other export mechanism could solve this issue, but it is out of scope for this draft;
- The full synchronization scenario usually occurs at Day 1 of the first integration, the APIs mostly used in daily maintenance are NE-based, this scalability issue may be less impacting as originally;
- Though a hierarchical model is much easier to understand, considering the extension possibility in the future, **we plan to continue to work with the current model structure**;

Note: The whole discussion record can be found on GitHub. (issue#32)

Summary & Next Step

Summary

- Introduced the attributes discussion progress
- Introduced the discussion status of efficiency issue raised in last version

Next Step:

- Request CCAMP WG adoption
- Get more input from IP and microwave technologies
- Close some more open issues

Welcome to join our weekly discussion

- Meeting slot: Wednesday 3-4pm CET (9-10am EST)
- Github: https://github.com/italobusi/ietf-network-inventory

Thank You