UML-link Diagram Description

UML-like Diagrams description and guidelines

- The UML-like diagrams in IEEE 802.1 standards provide a representation graphical of the management model using a subset of the UML class diagram notation. (details below)
- The UML-like diagram is not meant to be a duplicate of the YANG tree for the model.
- The UML-Like diagrams available in IEEE 802.1 standards containing YANG provides:
    - The list of objects from the YANG
    - The datatype provide in the UML-like diagram, however, represents the semantics of the property not specifically the datatype used in the encoding of either the SNMP MIB or NETCONF YANG.
    - The accessibility of the object is listed
    - A clause/sub-clause can be listed

- Explanatory text should be added as introduction to the clause containing the UML-like diagrams to explain the intent of the diagrams. It should be highlighted that the types used in the diagrams are meant to express the semantics of the objects and are not meant to provide the datatype used in the encoding of either MIB or YANG.

Suggested text: (augmenting text that already exists to describe the UML diagrams)

> A UML representation of the management model is provided in the following subclauses. The structure of the UML representation shows the name of the object followed by a list of properties for the object. The properties indicate their type and accessibility. It should be noted that the type in the UML representation is meant to express the semantics of the property and is not meant to provide the datatype used in the encoding of either MIB or YANG. In the UML representation, a box with a white background represents information that comes from sources outside of the IEEE. A box with a gray background represents objects that are defined by an IEEE Standard.

UML-like Diagrams example

From .1Q, here is an example from bridge-port. In the figure below, the non-IEEE structure has a white background and the IEEE specific work defined by the .1Q document is in grey. The format is:

Module name: stands alone in a box.

Attributes: have a type description, followed by the name of the attribute followed by a comment that has accessibility and alternatively a clause/sub-clause reference)

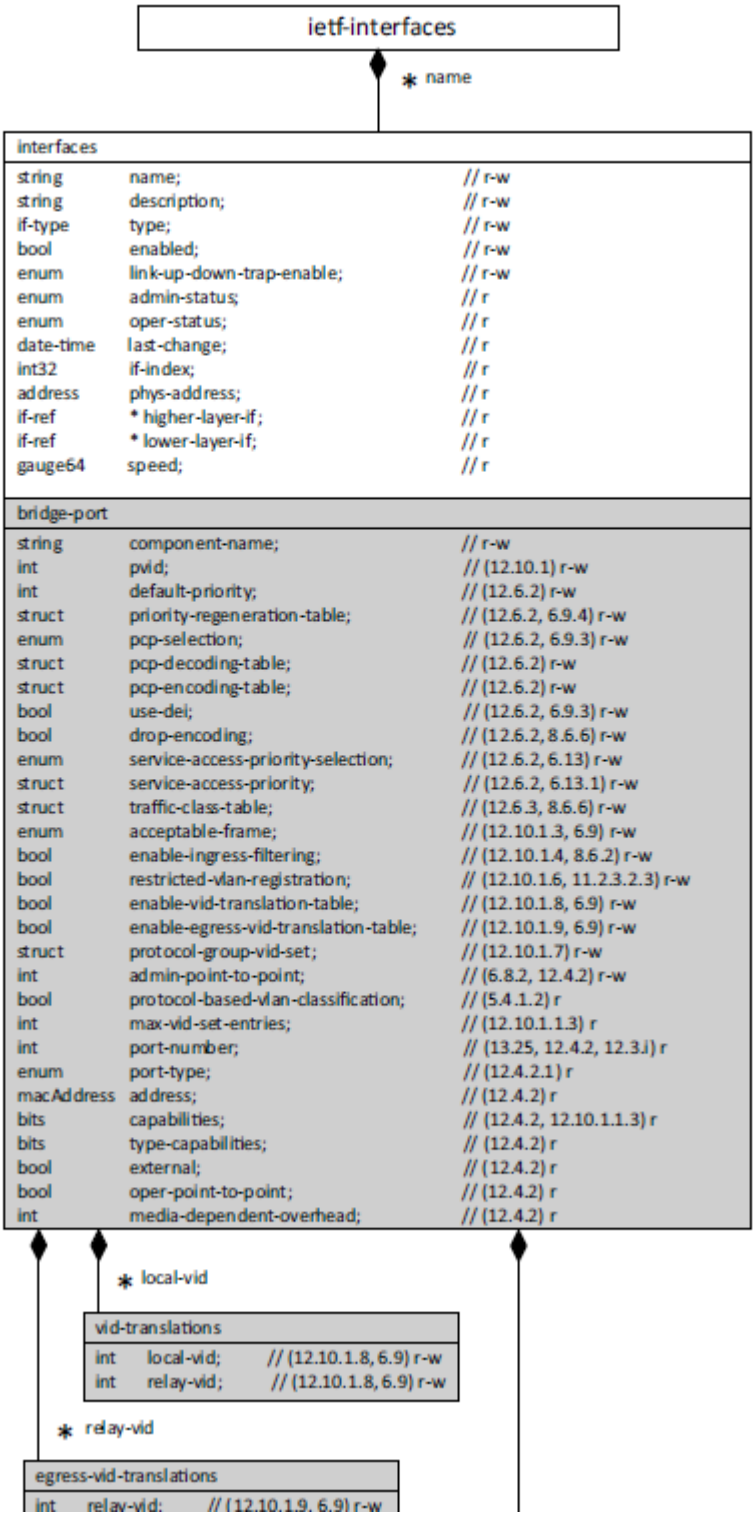

Figure 1 bridge-port example

Comparing this to a portion of the YANG tree

```
|      +--rw dot1q:bridge-port
|         +--rw dot1q:component-name?                          string
|         +--rw dot1q:port-type?                               identityref
|         +--rw dot1q:pvid?                                    dot1qtypes:vlan-index-type
|         +--rw dot1q:default-priority?                        dot1qtypes:priority-type
|         +--rw dot1q:priority-regeneration
|         |  +--rw dot1q:priority0?   priority-type
|         |  +--rw dot1q:priority1?   priority-type
|         |  +--rw dot1q:priority2?   priority-type
|         |  +--rw dot1q:priority3?   priority-type
|         |  +--rw dot1q:priority4?   priority-type
|         |  +--rw dot1q:priority5?   priority-type
|         |  +--rw dot1q:priority6?   priority-type
|         |  +--rw dot1q:priority7?   priority-type
|         +--rw dot1q:pcp-selection?                           dot1qtypes:pcp-selection-type
|         +--rw dot1q:pcp-decoding-table
|         |  +--rw dot1q:pcp-decoding-map* [pcp]
|         |     +--rw dot1q:pcp             pcp-selection-type
|         |     +--rw dot1q:priority-map* [priority-code-point]
|         |        +--rw dot1q:priority-code-point    priority-type
|         |        +--rw dot1q:priority?              priority-type
|         |        +--rw dot1q:drop-eligible?         boolean
|         +--rw dot1q:pcp-encoding-table
|         |  +--rw dot1q:pcp-encoding-map* [pcp]
```

Figure 2 bridge-port yang tree

Not all the types used in the UML-like diagram are identical to the types in the YANG. The UML-like diagram simplifies to provide the semantics without the complexity.

Comparing this to the pyang generated diagram

The pyang tool (along with PlantUML) can produce UML-like diagrams, but it is a representation of the YANG with all the complexity of the model included. Resulting in a large and unwieldy diagram that is harder to use than the IEEE UML-like diagram.

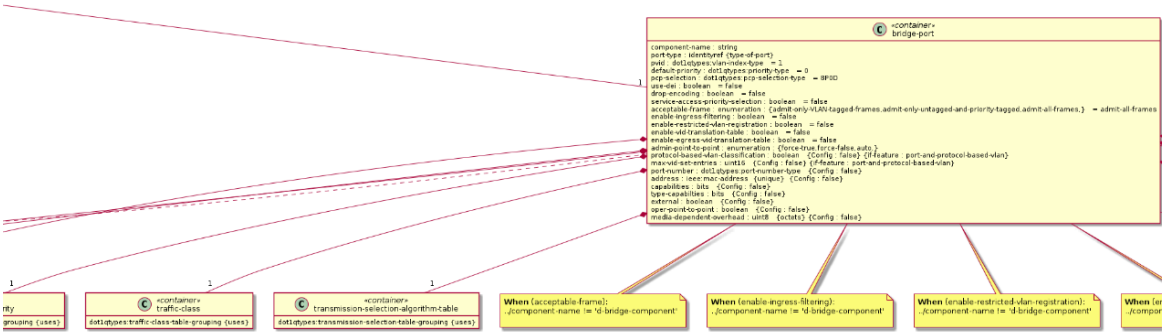

Figure 3 pyang generated uml-like diagram

Bottom Line:

The IEEE UML-like diagram provides an easy to use representation of the model that helps explain the model in understandable terms even for that do not understand UML modeling.