

# **SMLv2 Translation to YANG**

draft-schoenw-netmod-smi-yang-02

Jürgen Schönwälder  
Jacobs University

# Changes since draft-schoenw-netmod-smi-yang-01

- Inlined examples again (avoids page flipping)
- Preserving the SMIv2 nesting instead of flattening things (see discussion in Beijing)
- Wrote some security considerations text
- Clarifications and editorial improvements

# Issue #1: DISPLAY-HINTs

- [...] the DISPLAY-HINT value MAY be used to generate a regular expression for the YANG pattern statement within the type statement.
- Options:
  - Declare this text to be sufficient
  - Elaborate how certain *simple* DISPLAY-HINTs can be translated, leaving the translation of more complex ones an implementation choice
  - Define a complete translation algorithm (hard, output usually not readable)
  - Declare that DISPLAY-HINTs are not translated at all

# Issue #2: Name Ambiguities

```
foo OBJECT IDENTIFIER ::= { baz 1 }  
bar OBJECT IDENTIFIER ::= { baz 1 }
```

- Options:
  - First definition encountered in a translation unit wins and declare certain valid SMIv2 updates to be avoided
  - Drop the idea to generate a hierarchy of nested containers (again) and instead merge all scalars into one container (name from MODULE-IDENTITY) followed by the tables
  - ?

# Issue #3: Enum/Bits Ambiguities

- SMIv2 allows the descriptors of enumerated integers and named bits to change during revisions
- YANG does not allow enums to be changed
- Options:
  - declare certain valid SMIv2 updates to be avoided
  - ??

# Issue #4: OBJECT-IDENTITY

```
container usmNoAuthProtocol {
  description
    "No Authentication Protocol.";
}

identity usmNoAuthProtocol {
  base "smiv2:object-identity";
  description
    "No Authentication Protocol.";
}
```

- Options:
  - Translate OBJECT-IDENTITIES into YANG containers?
  - Translate OBJECT-IDENTITIES into YANG identities?
  - Do both or do nothing?

# Issue #5: string vs binary

- Presence of a DISPLAY-HINT is used to decide whether an OCTET STRING translates to a string or binary type
- SMIv2 allows to add DISPLAY-HINTs during revisions, hence this is not safe
- Options:
  - Use a union type to always allow textual strings and binary values; requires to “quote” string values in some way to avoid ambiguities
  - Always use the binary type (encoded in base64)
  - declare certain valid SMIv2 updates to be avoided
  - ??

# Issue #6: smiv2 yang extensions

smiv2:max-access, smiv2:defval) are defined to keep SMlv2 information not directly translatable in a machine readable format

- Options:
  - Implementations **MUST** generate these statements
  - Implementations **SHOULD** generate these statements