

# Translation of SMIv2 MIB Modules to YANG Modules

Jürgen Schönwälder



JACOBS  
UNIVERSITY

75. IETF 2009, Stockholm, 2009-07-27

Support: EU IST-EMANICS Network of Excellence (#26854)

# Motivation and Background

- 1 Motivation and Background
- 2 Example: IF-MIB Translation
- 3 Open Issues
- 4 References

# Motivation

## Goal

- Access to existing SNMP instrumentation via NETCONF
- Direct translation without bells and whistles

## Non-Goals

- Generation of “nice” YANG configuration models out of SMIv2 data models (no attempts to “beautify” MIBs)
- Translation of YANG to SMIv2

## Side Effects

- SMIv2  $\longrightarrow$  YANG  $\longrightarrow$  XSD
- SMIv2  $\longrightarrow$  YANG  $\longrightarrow$  RNG

# Background

## libsmi

- The libsmi MIB compiler smidump can already translate to numerous formats
- Translation to YANG is a straight-forward addition of another backend to the compiler
- Core implementation written during the Stockholm YANG Design Team meeting
- Open source: google, download, send patches

## Future...

- YANG parser frontend integration into libsmi
- Rewrite of the translation backend to generate an in memory YANG representation of an SMIv2 module

# Example: IF-MIB Translation

- 1 Motivation and Background
- 2 Example: IF-MIB Translation**
- 3 Open Issues
- 4 References

# Example: IF-MIB Translation

```
module IF-MIB {  
  
    namespace "urn:ietf:params:xml:ns:yang:smiv2:IF-MIB";  
    prefix "if-mib";  
  
    import IANAifType-MIB { prefix "ianaiftype-mib"; }  
    import SNMPv2-TC      { prefix "smiv2"; }  
    import yang-types     { prefix "yang"; }  
  
    // definitions omitted  
}
```

- YANG module name = SMIv2 module name
- Generation of a namespace from the SMIv2 module name
- Short unique prefix calculation algorithm (see draft)

# Example: IF-MIB Module Information

```
organization
  "IETF Interfaces MIB Working Group";

contact
  "Keith McCloghrie";

description
  "The MIB module to describe generic objects for network
  interface sub-layers. This MIB is an updated version of
  MIB-II's ifTable, and incorporates the extensions defined in
  RFC 1229.";

revision "2000-06-14" {
  description
    "Clarifications agreed upon by the Interfaces MIB WG, and
    published as RFC 2863.";
}
revision "1996-02-28" {
  description
    "Revisions made by the Interfaces MIB WG, and published in
    RFC 2233.";
}
revision "1993-11-08" {
  description
    "Initial revision, published as part of RFC 1573.";
}
```

# Example: IF-MIB Textual Conventions

```
typedef OwnerString {  
    type string {  
        length "0..255";  
        pattern "\p{IsBasicLatin}{0,255}";  
    }  
    status deprecated;  
    description  
        "This data type is used to model an administratively  
        assigned name of the owner of a resource. This information  
        is taken from the NVT ASCII character set. It is suggested  
        that this name contain one or more of the following: ASCII  
        form of the manager station's transport address, management  
        station name (e.g., domain name), network management  
        personnel's name, location, or phone number. In some cases  
        the agent itself will be the owner of an entry. In these  
        cases, this string shall be set to a string starting with  
        'agent'. ";  
}
```



# Example: IF-MIB Textual Conventions

```
typedef InterfaceIndex {  
    type int32 {  
        range "1..2147483647";  
    }  
    description  
        "A unique value, greater than zero, for each interface or  
        interface sub-layer in the managed system. It is  
        recommended that values are assigned contiguously starting  
        from 1. The value for each interface sub-layer must remain  
        constant at least from one re-initialization of the entity's  
        network management system to the next re-initialization."  
}
```

- The translation of INTEGER and OCTET STRING types depends on the presence of DISPLAY-HINTs or enumerated values

# Example: IF-MIB container, leafs, lists

```
container interfaces {  
  
    leaf ifNumber {  
        type int32;  
        config false;  
        description  
            "The number of network interfaces (regardless of their  
            current state) present on this system.";  
    }  
  
    list ifEntry {  
        key "ifIndex";  
        description  
            "An entry containing management information applicable to a  
            particular interface.";  
  
        // list member definition omitted  
  
    }  
}
```

# Example: IF-MIB list members

```
leaf ifIndex {  
    type if-mib:InterfaceIndex;  
    config false;  
    description  
        "A unique value, greater than zero, for each interface. It  
        is recommended that values are assigned contiguously  
        starting from 1. The value for each interface sub-layer  
        must remain constant at least from one re-initialization of  
        the entity's network management system to the next re-  
        initialization.";  
}
```

- Read-only and not-accessible objects are config false
- Read-write or read-create objects are config true

# Example: IF-MIB list members

```
leaf ifAdminStatus {  
    type enumeration {  
        enum up      { value 1; }  
        enum down    { value 2; }  
        enum testing { value 3; }  
    }  
    config true;  
    description  
        "The desired state of the interface. The testing(3) state  
        indicates that no operational packets can be passed. When a  
        managed system initializes, all interfaces start with  
        ifAdminStatus in the down(2) state. As a result of either  
        explicit management action or per configuration information  
        retained by the managed system, ifAdminStatus is then  
        changed to either the up(1) or testing(3) states (or remains  
        in the down(2) state).";  
}
```

# Example: IF-MIB list augmentations

```
augment "/if-mib:interfaces/if-mib:ifEntry" {
  description
    "An entry containing additional management information
    applicable to a particular interface.";

  leaf ifName {
    type smiv2:DisplayString;
    config false;
    description
      "The textual name of the interface. The value of this
      object should be the name of the interface as assigned by
      the local device and should be suitable for use in commands
      entered at the device's 'console'. This might be a text
      name, such as 'le0' or a simple port number, such as '1',
      depending on the interface naming syntax of the device. If
      several entries in the ifTable together represent a single
      interface as named by the device, then each will have the
      same value of ifName. Note that for an agent which responds
      to SNMP queries concerning an interface on some other
      (proxied) device, then the value of ifName for such an
      interface is the proxied device's local name for it."
  }
}
```

# Example: IF-MIB notifications

```
notification linkDown {
  container linkDown-ifIndex {
    leaf ifIndex {
      type leafref {
        path "/if-mib:interfaces/if-mib:ifEntry/if-mib:ifIndex";
      }
    }
  }
  container linkDown-ifAdminStatus {
    leaf ifIndex { /* ... */ }
    leaf ifAdminStatus { /* ... */ }
  }
  container linkDown-ifOperStatus {
    leaf ifIndex { /* ... */ }
    leaf ifOperStatus { /* ... */ }
  }
}
```

- Each SMIv2 notification object becomes a container with a leaf for the value and leafrefs for INDEX elements

# Open Issues

- 1 Motivation and Background
- 2 Example: IF-MIB Translation
- 3 Open Issues**
- 4 References

# Translation of SMIv2 Conformance Statements

## Question #1

Is it possible and does it make sense to translate conformance definitions into YANG features? If not, is it an issue to lose this information?



# Regular Expression Generation Algorithm

## Question #2

How to translate DISPLAY-HINTs into “nice” YANG pattern?  
The current algorithm sometimes produces very ugly pattern  
and occasionally wrong pattern.

SMLv2  $\longrightarrow$  YANG  $\longrightarrow$  XSD

Question #3

Does the result of SMLv2  $\longrightarrow$  YANG  $\longrightarrow$  XSD satisfy the requirements of those who like to have SMLv2  $\longrightarrow$  XSD?

- 1 Motivation and Background
- 2 Example: IF-MIB Translation
- 3 Open Issues
- 4 References**

# References



M. Björklund.

YANG - A data modeling language for NETCONF.

Internet Draft (work in progress) <draft-ietf-netmod-yang-07.txt>, Tail-f Systems, July 2009.



J. Schönwälder.

Common YANG Data Types.

Internet Draft (work in progress) <draft-ietf-netmod-yang-types-03.txt>, Jacobs University, May 2009.



J. Schönwälder.

Translation of SMIv2 MIB Modules to YANG Modules.

Internet Draft (work in progress) <draft-schoenw-netmod-smi-yang-00>, Jacobs University, January 2009.



K. McCloghrie, D. Perkins, and J. Schönwälder.

Structure of Management Information Version 2 (SMIv2).

RFC 2578, Cisco Systems, SNMPinfo, TU Braunschweig, April 1999.



K. McCloghrie, D. Perkins, and J. Schönwälder.

Textual Conventions for SMIv2.

RFC 2579, Cisco Systems, SNMPinfo, TU Braunschweig, April 1999.



K. McCloghrie, D. Perkins, and J. Schönwälder.

Conformance Statements for SMIv2.

RFC 2580, Cisco Systems, SNMPinfo, TU Braunschweig, April 1999.