ASN.1 for SMIng

Design goals:

- Specify a new language that meets as many as possible of the objectives for the next-generation SMI
- Enable MIB writers to define arbitrarily-complex data structures, by combining arrays, structures, unions and atomic data items in arbitrary ways
- Separate the definition of reusable data classes from the declaration of the variables (entity instances) that become part of the MIB
- Define a flexible instance naming scheme that works both for traditional SMI flat tables and for any more-complex data structures
- Provide a data model compatible with both SNMP and COPS-PR

Example - Start of module

```
RMON2-MIB { mib-2 16 } DEFINITIONS ::= BEGIN

   IMPORTS ....;

   module-info MODULE-INFO ::= {
      ORGANIZATION "ACME MIB Writers Ltd"
      CONTACT "Last Name First Name
         My Street
         My City
         My State"
      REVISIONS {
         { DATE "200111301240Z"
           DESCRIPTION "Latest Revision - Aaa bb c ddddd" } |
         { DATE "200007030917Z"
           DESCRIPTION "First Version" } }
      DESCRIPTION "This is an example MIB module" }

   rmon            OBJECT IDENTIFIER ::= { mib-2 16 }
   protocolDir     OBJECT IDENTIFIER ::= { rmon 11 }
   protocolDist    OBJECT IDENTIFIER ::= { rmon 12 }
   addressMap      OBJECT IDENTIFIER ::= { rmon 13 }

   protocolDirExamples      OBJECT IDENTIFIER ::= { protocolDir 2 }
.....
```

Example - Atomic entity class

```
protocolDirLastChange ENTITY-CLASS ::= {
   SYNTAX TimeStamp
   STATUS current
   DESCRIPTION
      "An entity class for the values of sysUpTime
      at the time the protocol directory
      was last modified, either through insertions or deletions,
      or through modifications of either
      the protocolDirAddressMapConfig, protocolDirHostConfig, or
      protocolDirMatrixConfig." }
```

Example - Union class

```
inetAddressClass ENTITY-CLASS ::= {
   SYNTAX
      CHOICE {
         ipUnknown OCTET STRING (SIZE(0..65535)),
         ipv4Addr InetAddressIPv4,
         ipv6Addr InetAddressIPv6,
         ipDnsAddr InetAddressDNS }

   ATTRIBUTE-INFO {
      { ATTRIBUTE {1}   MAX-ACCESS readWrite   STATUS current
        DESCRIPTION "Represents an address using an ext'ly def'd format." } |
      { ATTRIBUTE {2}   MAX-ACCESS readWrite   STATUS current
        DESCRIPTION "Represents an IPv4 address" } |
      { ATTRIBUTE {3}   MAX-ACCESS readWrite   STATUS current
        DESCRIPTION "Represents an IPv6 address" } |
      { ATTRIBUTE {4}   MAX-ACCESS readWrite   STATUS current
        DESCRIPTION "Represents a DNS domain name" } }

   STATUS current
   DESCRIPTION "Internet address in 4 different representations." }
```

Example - Structure class

```
protocolDirEntry ENTITY-CLASS ::= {
   SYNTAX
      SEQUENCE {
         protocolDirID            OCTET STRING,
         protocolDirParameters    OCTET STRING,
         protocolDirLocalIndex    Integer32(1..2147483647),
         protocolDirDescr         DisplayString(SIZE(1..64)),
         protocolDirType          BIT STRING
            { extensible(0), addressRecognitionCapable(1) },
         protocolDirAddressMapConfig    ENUMERATED
            { notSupported(1), supportedOff(2), supportedOn(3) },
         protocolDirHostConfig    ENUMERATED
            { notSupported(1), supportedOff(2), supportedOn(3) },
         protocolDirMatrixConfig  ENUMERATED
            { notSupported(1), supportedOff(2), supportedOn(3) },
         protocolDirOwner         OwnerString,
         protocolDirStatus        RowStatus }

   ASSIGNED-ATTRIBUTE-IDS          --this is optional
      { {1}, {2}, {3}, {4}, {5}, {6}, {7}, {8}, {9}, {10} }
```

(cont.)

Example - Structure class

(cont.)

```
      ATTRIBUTE-INFO {
         { ATTRIBUTE {1}
           MAX-ACCESS notAccessible
           STATUS current
           DESCRIPTION
            "A unique identifier for a particular..." } |
         { ATTRIBUTE {2}
           MAX-ACCESS notAccessible
           STATUS current
           DESCRIPTION
            "A set of parameters for the associated..." } |
         { ATTRIBUTE {3}
           MAX-ACCESS notAccessible
           STATUS current
           DESCRIPTION
            "The locally arbitrary..." } |
....
      STATUS current
      DESCRIPTION
         "A structure class..." }
```

Example - Array class

```
protocolDirTable ENTITY-CLASS ::= {
   SYNTAX SEQUENCE OF protocolDirEntry.&Syntax
   INDEXING-ATTRIBUTES { {1}, {2} }
   STATUS current
   DESCRIPTION
      "An entity class for a flat table that lists
      the protocols that this..." }
```

Example - Entity instances

An instance of an atomic entity class:

```
protocolDirLastChangeObj ENTITY-INSTANCE ::= {
   OID { protocolDir 1 }
   ENTITY-CLASS protocolDirLastChange
   MAX-ACCESS readOnly }
```

An instance of an array entity class:

```
protocolDirTableObj ENTITY-INSTANCE ::= {
   OID { protocolDir 2 }
   ENTITY-CLASS protocolDirTable
   MAX-OPERATION createDelete }
```

Features

Separation between entity classes and entity instances:

- Reusability of entity classes:
  - Classes of various kinds (atomic types, structures, arrays, unions) can be re-used in the definition of:
    - multiple objects of the same kind (e.g.: multiple objects of the same constrained or unconstrained atomic type, multiple tables with the same structure, etc.)
    - more-complex data structures (higher-level structures, arrays and unions where the class appears as a component)

(cont.)

Features

Separation between entity classes and entity instances (cont.)

- Entity classes completely specify the naming of every subordinate component at any level of nesting, by way of "relative" object identifiers.
- Only entity instances are assigned an object identifier.
  - For entity instances that are complex data structures, each component is implicitly assigned an object identifier by concatenating the outer OID of the entity instance with its own relative OID, all the way down to the leaf-level components.

Features

Protocol independence:

- SNMP will have visibility only of atomic data items, that are often the leaf-level components of some more-complex data structure.
- COPS-PR will have visibility only of the structures consisting entirely of atomic data items; these structures may belong to more-complex data structures.

Example - OID assignments

Definition of a structure entity class used as a row class in these examples:

```
protocolDirEntry ENTITY-CLASS ::= {      /* Simplified definition */
   SYNTAX
      SEQUENCE {
         protocolDirID            OCTET STRING,
         protocolDirParameters    OCTET STRING,
         protocolDirLocalIndex    Integer32(1..2147483647),
         protocolDirDescr         DisplayString(SIZE(1..64)),
         protocolDirStatus        RowStatus }

   ATTRIBUTE-NAMING compact   --default
   ASSIGNED-ATTRIBUTE-IDS   { {1}, {2}, {3}, {4}, {5} }

   ATTRIBUTE-INFO {
      { ATTRIBUTE {1}    MAX-ACCESS notAccessible
        DESCRIPTION "A unique identifier for a particular..." } |
      { ATTRIBUTE {2}    MAX-ACCESS notAccessible
        DESCRIPTION "A set of parameters for the associated..." } |
      { ATTRIBUTE {3}    MAX-ACCESS readOnly
        DESCRIPTION "The locally arbitrary..." } |
      { ATTRIBUTE {4}    MAX-ACCESS readWrite
        DESCRIPTION "..." } |
      { ATTRIBUTE {5}    MAX-ACCESS readWrite
        DESCRIPTION "..." } }

   DESCRIPTION "A structure class..." }
```

Example - OID assignments

Definition of an array entity class based on the row class defined above:

```
protocolDirArrayClassA ENTITY-CLASS ::= {
   SYNTAX SEQUENCE OF protocolDirEntry.&Syntax
   INDEXING-ATTRIBUTES { {1}, {2} }
   NAMING-SCHEME tabular
   STATUS current
   DESCRIPTION
      "An entity class for a flat table that lists
       the protocols that this..." }
```

Example - OID assignments

An instance of a traditional flat table (tabular naming scheme):

```
protocolDirTableA ENTITY-INSTANCE ::= {
   OID { protocolDirExamples 1 }
   ENTITY-CLASS protocolDirArrayClassA
   MAX-OPERATION createDelete }
```

If protocolDirExamples = 1.3.6.1.4.1.9626.135.18.2 then the OID assignments for all the data objects in the table will be as follows:

protocolDirTableA

1.3.6.1.4.1.9626.135.18.2.1

the conceptual row (of class **protocolDirEntry**) of **protocolDirTableA**

1.3.6.1.4.1.9626.135.18.2.1 **.1**

the columnar object of class **protocolDirDescr** of **protocolDirTableA**

1.3.6.1.4.1.9626.135.18.2.1 **.1 .4**

one particular instance of **protocolDirDescr** in **protocolDirTableA**

1.3.6.1.4.1.9626.135.18.2.1 **.1 .4 .4.41.15.128 .3.71.89.51**

Example - OID assignments

Definition of another array entity class based on the same row class defined above:

```
protocolDirArrayClassB ENTITY-CLASS ::= {
   SYNTAX SEQUENCE OF protocolDirEntry.&Syntax
   INDEXING-ATTRIBUTES { {1}, {2} }
   NAMING-SCHEME hierarchical
   STATUS current
   DESCRIPTION
      "An entity class for a table that lists
       the protocols that this..." }
```

Note that we are able to re-use the same structure class to define two independent array classes that have a different naming scheme and may also have different sets of indexing attributes.

Example - OID assignments

An instance of a table with hierarchical object naming:

```
protocolDirTableB ENTITY-INSTANCE ::= {
   OID { protocolDirExamples 2 }
   ENTITY-CLASS protocolDirArrayClassB }
```

If protocolDirExamples = 1.3.6.1.4.1.9626.135.18.2 then the OID assignments for all the data objects in the table will be as follows:

protocolDirTableB

1.3.6.1.4.1.9626.135.18.2.2

one particular row instance of **protocolDirTableB**

1.3.6.1.4.1.9626.135.18.2.2 **.4.41.15.128 .3.71.89.51**

the **protocolDirDescr** object in this particular row instance of **protocolDirTableB**

1.3.6.1.4.1.9626.135.18.2.2 **.4.41.15.128 .3.71.89.51 .4**

Example - OID assignments

Definition of a structure entity class with extensible attribute naming:

```
protocolDirEntry ENTITY-CLASS ::= {      /* Simplified definition */
   SYNTAX
      SEQUENCE {
         protocolDirID            OCTET STRING,
         protocolDirParameters    OCTET STRING,
         protocolDirLocalIndex    Integer32(1..2147483647),
         protocolDirDescr         DisplayString(SIZE(1..64)),
         protocolDirStatus        RowStatus }

   ATTRIBUTE-NAMING extensible
   EXTENSION-ID {0}     --default
   ASSIGNED-ATTRIBUTE-IDS   { {1}, {2}, {3}, {4}, {5} }

   ATTRIBUTE-INFO {
      { ATTRIBUTE {1}    MAX-ACCESS notAccessible
        DESCRIPTION "A unique identifier for a particular..." } |
      { ATTRIBUTE {2}    MAX-ACCESS notAccessible
        DESCRIPTION "A set of parameters for the associated..." } |
      { ATTRIBUTE {3}    MAX-ACCESS readOnly
        DESCRIPTION "The locally arbitrary..." } |
      { ATTRIBUTE {4}    MAX-ACCESS readWrite
        DESCRIPTION "..." } |
      { ATTRIBUTE {5}    MAX-ACCESS readWrite
        DESCRIPTION "..." } }

   DESCRIPTION "A structure class..." }
```

Example - OID assignments

Definition of an array entity class based on the row class defined above:

```
protocolDirArrayClassC ENTITY-CLASS ::= {
   SYNTAX SEQUENCE OF protocolDirEntry.&Syntax
   INDEXING-ATTRIBUTES { {0 1}, {0 2} }
   NAMING-SCHEME hierarchical
   STATUS current
   DESCRIPTION
      "An entity class for a table that lists
       the protocols that this..." }
```

Example - OID assignments

A table with hierarchical naming scheme and extensible attribute naming:

```
protocolDirTableSourceToDest ENTITY-INSTANCE ::= {
   OID { protocolDirExamples 3 1 }
   ENTITY-CLASS protocolDirArrayClassC }
```

If protocolDirExamples = 1.3.6.1.4.1.9626.135.18.2 then the OID assignments for all the data objects in the table will be as follows:

protocolDirTableSourceToDest

1.3.6.1.4.1.9626.135.18.2.3.1

one particular row instance of **protocolDirTableSourceToDest**

1.3.6.1.4.1.9626.135.18.2.3.1 **.4.41.15.128 .3.71.89.51**

the **protocolDirDescr** object in this particular row instance of **protocolDirTableSourceToDest**

1.3.6.1.4.1.9626.135.18.2.3.1 **.4.41.15.128 .3.71.89.51 .0 .4**

Example - OID assignments

Another table based on the same array class as above:

```
protocolDirTableDestToSource ENTITY-INSTANCE ::= {
   OID { protocolDirExamples 3 2 }
   ENTITY-CLASS protocolDirArrayClassC }
```

If protocolDirExamples = 1.3.6.1.4.1.9626.135.18.2 then the OID assignments for all the data objects in the table will be as follows:

protocolDirTableDestToSource

1.3.6.1.4.1.9626.135.18.2.3.2

one particular row instance of **protocolDirTableDestToSource**

1.3.6.1.4.1.9626.135.18.2.3.2 **.4.41.15.128 .3.71.89.51**

the **protocolDirDescr** object in this particular row instance of **protocolDirTableDestToSource**

1.3.6.1.4.1.9626.135.18.2.3.2 **.4.41.15.128 .3.71.89.51 .0 .4**

Note that we are able to re-use the same table class to define two independent tables that have exactly the same structure.

Example - Compliance statement

```
dsmonAggControlCompliance COMPLIANCE-STATEMENT ::= {
     OID { dsmonCompliances 1 }
     MANDATORY-NODE-GROUPS { dsmonAggControlGroup.&oid }
     STATUS current
     DESCRIPTION "Example compliance for the aggregation
           control portion of the DSMON-MIB module." }

dsmonAggControlGroup NODE-GROUP ::= {
     OID { dsmonGroups 1 }
     MEMBERS { dsmonMaxAggGroups.&oid,
           dsmonAggControlLocked.&oid,
           dsmonAggControlChanges.&oid,
           dsmonAggControlLastChangeTime.&oid  --more?}
     STATUS current
     DESCRIPTION "A collection of objects used to configure
           and manage aggregation groups for DSMON collection
           purposes." }
```