EOS OOPS
SMIng Issues

Wes Hardaker
<hardaker@tislabs.com>

draft-hardaker-eos-oops-02pre.txt

2002.Nov.20

Overview

- Indexing issues
- External augmentations to structures/data
- How is data expected to be accessed?
  - Search examples

Upfront Example

- Interface 1
  - Address 11, IPType11
  - Address 12, IPType12
- Interface 2
  - Address 21, IPType21
  - Address 22, IPType22

- v3ifTable index   = ifIndex
- v3ifTable.addresstable index = address

Indexing Issues

- Referencing of defined elements is easy:
  - ifArray.addressArray.iptype
  - 1.1.1
- Refercing indexes in sub-elements is hard.
  - ifArray.addressArray.address
    - (where address is an index)
  - 1.1.??

Indexing Issues

○ OOPS does this a lot:

```
CHOICE {
    index-number[0]
        IMPLICIT INTEGER,

    element-number[1]
        IMPLICIT INTEGER,

    XXX: element, sub-element
    XXX: element, sub-index
    XXX: element, sub-element, sub-index
}
```

Indexing Proposal:

- All Indexing, including external, in SMIv3 be assigned to a local number.
- Including external indexes.
- This enables all elements to be mapped to an OID.
- Something like:

```
INDEX otherIndex
FROM  otherTable
::= { base 1 }
```

External Augmentations are a pain

- External augmentations are a pain, protocol-wise
  - OID assignments
- Any chance we can fix this in SMIv3?
  - Forced local extension node? (.0.enterprise, ...)
  - other?

NOTIFICATIONs

- Can NOTIFICATIONs contain:
  - arrays?
  - structs?
  - (If so, then it would be for OOPS only obviosuly)
- Example:
  - send all addresses associated with linkUp