Generating Skeleton Code for NETCONF Modules from SMI MIB Module Definitions

Torsten Klie
L3S Research Center, Hanover, Germany

International Conference on WWW/Internet
2005, Lisbon, Portugal

Outline

1. Simple Network Management Protocol (SNMP)
2. NETCONF
    1. NETCONF protocol
    2. Yenca NETCONF agent
3. Transforming MIBs to NETCONF modules
    1. Separation between state, config, and control data
    2. Data representation
    3. Code generation
4. Conclusions & outlook

SNMP Framework

Simple Network Management Protocol (SNMP)

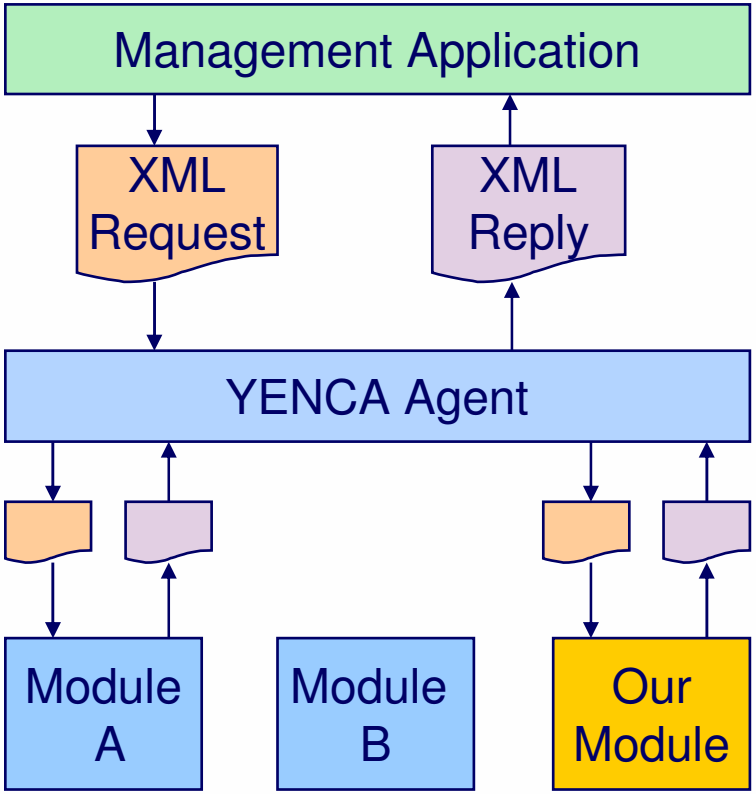
- Simple: small footprint (even in the late ´80s)
- Data Models
  - Structure of Management Information (SMI)
  - > 200 standard MIBs, >600 vendor MIBs
- Strength: Monitoring
- Weakness:
  - Low-level technology
  - Configuration

NETCONF Protocol

- Several evolutionary approaches failed
- New IETF protocol for configuration
  - XML-based RPC
  - Application protocol independent
- Existing mappings for application protocols
  - SSH
  - BEEP
  - SOAP
- Current status: Internet Draft

Yenca C NETCONF Agent

- NETCONF agent prototype
- From MADYNES Team (INRIA, France)
- Modular concept
  - Plug-in new modules
  - Loose coupling
  - Generate complete replys to incoming requests



Transforming MIBs to Yenca Modules

- Why?
  - No NETCONF data models available
  - Large number of MIBs already there
- What?
  - XML representation of data
- How?
  - Separate state, control, and configuration data
  - Generate C code

XML Data Representation

- Flattened element hierarchy
  - No deeply nested elements
  - Not more than 5 levels (except for „tables in tables“)

```
<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
 <config>
  <snmpv2>
   <system>
    <sysContact>admin@l3s.de</sysContact>
    <sysName>aligator</sysName>
    <sysLocation>L3S, Hanover, Room 104</sysLocation>
   </system>
  </snmpv2>
 </config>
</rpc-reply>
```

Separation of Config, Control, and State Data

- Data types
  - Counters always state data
  - `RowStatus` objects can be ignored
- Access level
  - Read-only data must be state data
  - Writable objects: config or control data
    - No automatic distinction possible
    - Annotations in SMI needed (e.g. special keywords in the `DESCRIPTION` clauses or in a separate file)
    - Interactive script: generate annotations

Code Generation

- Module for smidump (MIB compiler that is part of libsmi).
- Generate module header file
  - Define a data structure for module specific data
  - Declare functions for
    - registration
    - saving and restoring configurations
    - data retrieval
    - agent notification
- Generate module source file
    - Generate Code for declared functions (automatically)
    - Generate skeleton functions for setting/retrieval

Conclusion & Outlook

DONE:

- Skeleton code for Yenca NETCONF modules can be generated from SMI MIB modules almost automatically
- XML data representation
- Providing good starting point for developers

TODO:

- Optimization of the generated code
- Implement script for SMI annotations
- In the long run: define new data models

THANKS!

Questions?