Data Modeling

Glenn Waters

IETF 58 NETCONF Meeting
November 12, 2003

Discussion Areas

- Modeling and the protocol
- Modeling requirements
- Modeling conventions
- Modeling language
- Data models
- Next steps

Modeling and the Protocol

- How should documents be named
  - Naming is required so that a specific instance of a document can be retrieved
  - Naming implies that a document has one or more keys that identify the document
  - XPATH, XMLQuery are some choices
- Should protocol operations be defined to “select” documents using their key
  - Keys would need to be identified in the model
  - What granularity of a document may be selected
- Should protocol operations be defined to list documents (that match specified criteria)
- Should the actual implemented model be available through the protocol
  - Possibly just a URI but maybe we want actual XSD transferred in the protocol

Modeling Requirements

- Add new data types
  - e.g. InetAddr, etc.
  - Other data types to ease transition and interoperability with the SMI
- Need way to specify document key(s)
- Change control rules
  - How do a new revision of a model get issued
  - How is a extension from a standard model specified
    - Extensions can be standards based and enterprise specified
- Need to identify notifications

Modeling Conventions

- XSD affords a lot of power
  - Many ways to do one thing
  - Many opinions on the “right” way
  - Should we write guidelines to suggest ways to use XSD
- Do we want to restrict the data types that can be used?
- XSD annotation clause
  - Standardize use of appInfo for NETCONF required keywords; e.g.: MIN-ACCESS (read-only v. read-write)
  - Use documentation element for documenting XML tags
- Guidelines on how namespaces are used
  - Want to have consistency

Modeling Language

- XSD of course
  - Lots of tools
  - The winning standard – probably
- But…
  - Can’t combine read-only and read-write in a single XSD
    - Requires that configuration and state models are separate (assuming that XSD is used to validate documents)
  - appInfo is the only standard way to extend XSD: tools will not understand the NETCONF appInfo tags
  - XSD is not human friendly
- How do we solve the issues?

Data Models

- Standard models is one of the next steps
- Need models to manage the NETCONF "subsystem"
  - Do we need models for the protocols itself
  - e.g. equivalent to the notification MIB
    - Other examples can easily be envisioned

Next Steps

- Is some of this work required to make the protocol “whole”?
  - Some of this work is clearly not in charter
  - Some of this work can be conceived as being part of charter
- Is there work to do?
  - Anyone want to help work on this?