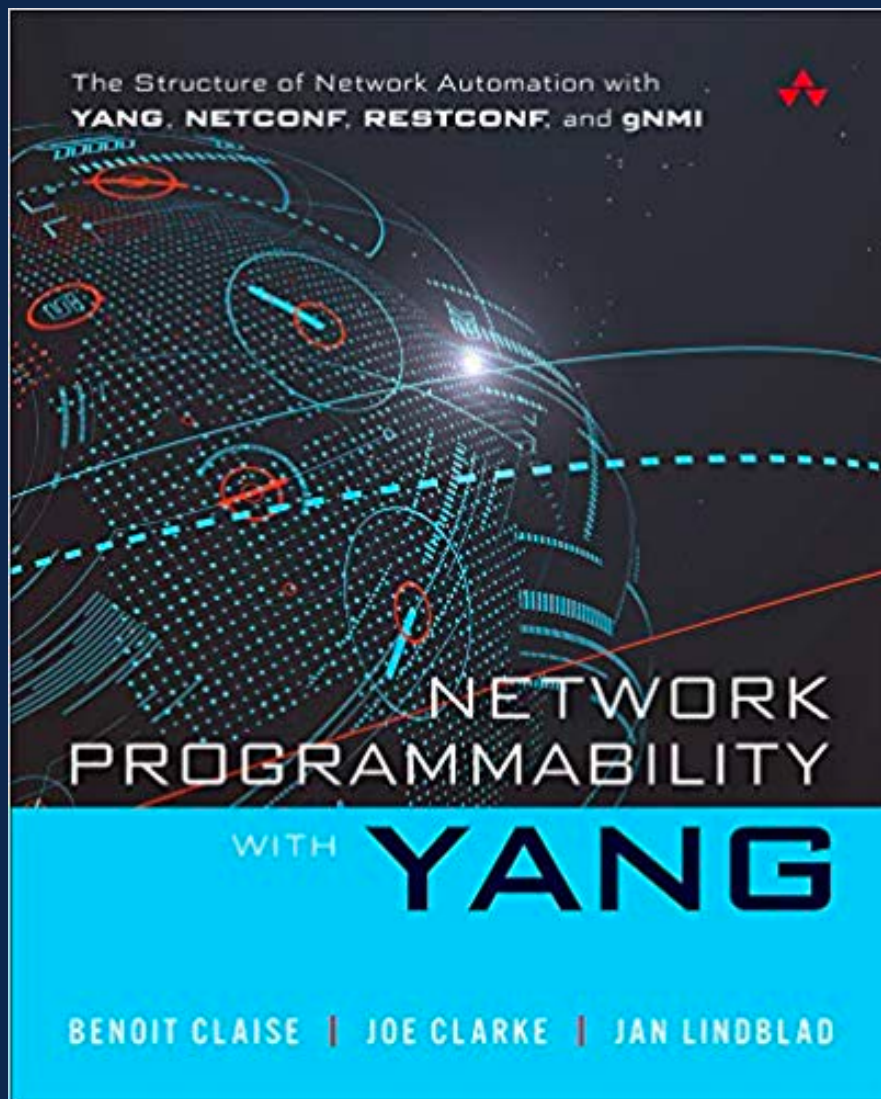




Network Programmability With YANG

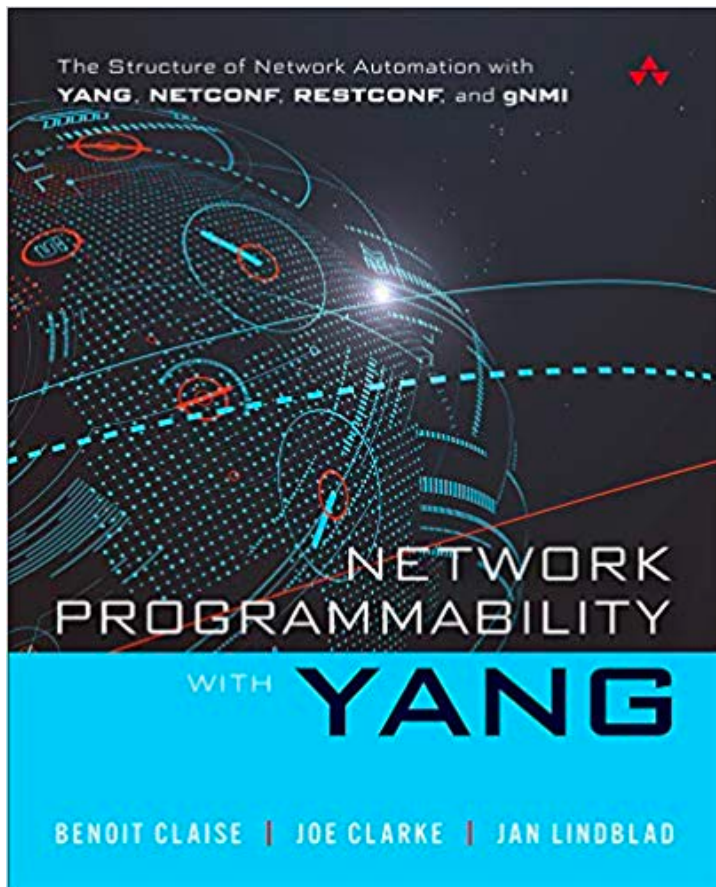
Jan Lindblad
Cisco NSO Engineering
June, 2019



DeveloperDays
Network Services Orchestrator

Network Programmability With YANG

The Structure of Network Automation with YANG, NETCONF, RESTCONF, and gNMI



1. The Network Management World Must Change: Why Should You Care?
2. Data Model-Driven Management
3. YANG Explained
4. NETCONF, RESTCONF, and gNMI Explained
5. Telemetry Explained
6. YANG Data Modeling Developments in the Industry
7. Automation...: For the Network Architect and Operator
8. Automation...: For the Module Author
9. Automation...: For the Application Developer
10. Using NETCONF and YANG
11. YANG Model Design
- Index

Automation Is as Good as the Data Models, Their Related Metadata, and the Tools:

Available as hardcover or epub/kindle

The Author Trio



Benoît Claise

Cisco Fellow, CCIE No. 2686
Benoit was IETF Operations and Management Area (OPS) co-director from 2012 to 2018
Benoît is a contributor to the IETF, with 35 RFCs.



Joe Clarke

Cisco Customer Experience Engineer, CCIE No. 5384
Co-chair of the Ops Area Working Group at the IETF.



Jan Lindblad

Cisco Business Development Manager. IETF YANG Doctor, NETCONF and YANG teacher.
Key participant at EANTC NETCONF/YANG interop events.

Interviews with the Experts



1

Victor Kuarsingh
Oracle



1

Russ White
Juniper



2

Jürgen Schönwälder
Jacobs University



3

Martin Björklund
Cisco



4

Kent Watsen
Watsen Networks



5

Alexander Clemm
Huawei



6

Carl Moberg
Cisco



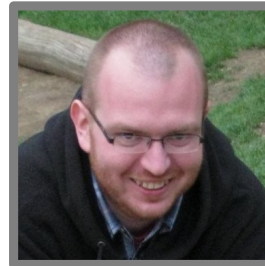
7

Einar Nilsen-Nygaard
Cisco



8

William Lupton
Honu



9

Radek Krejčí
Cesnet



10

Kristian Larsson
Deutsche Telekom



11

Andy Bierman
YumaWorks

GitHub

There is a companion project on GitHub

- Clone bookzone and follow/play with the examples in the book
- Folders 1-6 match chapter 3 sections, folder 7 matches chapter 10 project
- Examples in chapters 4 and 8 use this too

The screenshot shows the GitHub interface for the repository 'janlindblad / bookzone'. The browser address bar shows 'https://github.com/janlindblad/bookzone'. The repository page includes a search bar, navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore', and a secondary navigation bar with 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Security', and 'Insights'. The repository statistics show 20 commits, 6 branches, 0 releases, and 1 contributor. A commit history table is visible, listing recent updates to various folders and files.

Commit	Description
Jan Lindblad	Updated 3-action-notif/README.md and .gitignore
1-intro	Added support for C builds in all Makefiles
2-config	Added support for C builds in all Makefiles
3-action-notif	Updated 3-action-notif/README.md and .gitignore
4-oper	Added support for C builds in all Makefiles
5-precision	Added support for C builds in all Makefiles
6-augment	Added support for C builds in all Makefiles
7-service	Added support for C builds in all Makefiles

Book Purpose

“ The entire networking industry is being pressured to automate in order to scale and move faster; this book explains how to unlock the power of network automation using YANG.

There are many barriers to automation. The strongest one is a need for a common understanding between network operators and software and hardware providers. To build this understanding, three things are sorely needed. ”

- First, the participants need to have a common language. This book provides common terminology, models, and awareness of use cases and tools so that effective communication is possible.
- Second, automation is not something that happens because you make a computer run in a loop. Network automation is a very hard problem. It is a distributed, parallel, real-time, highly available, performance-sensitive, security-sensitive control problem at the heart of society. This requires a system architecture.

This system architecture already exists but is not well known. Even many seasoned professionals are missing key pieces of the overall picture. This book paints the landscape so all parties understand where their pieces fit and how to achieve common goals.

- Third, as you may have heard, “He who knows how will be hired, she who knows why will be his manager.” Understanding the reasons for an architectural choice is a core aspect of knowing the architecture itself. This book provides this background.

By giving rich backgrounds, using examples, explaining “why,” and providing ample opportunities for hands-on work, we hope this book will be useful to you as a networking professional, as well as for the advancement of the industry as a whole. ”

1

The Network Management World Must Change Why Should You Care?

Chapter 1 discusses what network management is, and some of the bigger trends and concepts, such as

- FCAPS, DevOps, Software-Defined Networking, Network Function Virtualization, Elastic Cloud, Data-Model Driven Management, Telemetry, Intent-Based Networking, Information Models vs. Data Models

It then gives a basic introduction to some popular management protocols

- CLI, SNMP, NetFlow, IPFIX, Syslog

2 Data Model-Driven Management

This chapter introduces the future:
Model-Driven Management

- Network management requirements, including the famous RFC 3535
- Different types of YANG modules: Device and Service modules
- Encodings: XML, JSON, Google Protobufs, CBOR
- Protocols: NETCONF, RESTCONF, gNMI, CoMI

“ In order to add an L3VPN leg to the network, the L3VPN application running in the NMS must touch at least the CE device on the new site, the PE device to which the CE device is connected, the monitoring system, and probably a few devices related to security. It could happen that the CE is a virtual device, in which case the NMS may have to speak to some container manager or virtual infrastructure manager (VIM) to spin up the virtual machine (VM). Sometimes 20 devices or so must be touched in order to spin up a single L3VPN leg. All of them are required for the leg to be functional. All firewalls and routers with access control lists (ACLs) need to get their updates, or traffic does not flow. Encryption needs to be set up properly at both ends, or traffic is not safe. Monitoring needs to be set up, or loss of service is not detected. ”

3 YANG Explained

Incrementally builds a real YANG module for BookZone

- Each of the six steps on the way is available in GitHub
- All the way from leaf, list, key to identity, extension, deviation, ...
- Explains linked leafrefs, XPath expressions, reasons for NMDA

“ One of the merits of using groupings like this is that the number of lines in the YANG module gets smaller, so you have less model text to maintain. Another benefit is that if you improve the grouping, the improvement immediately applies everywhere. Just make sure that is what you want! A downside with using too many groupings is that it sometimes gets hard to follow multiple levels of uses statements crisscrossing the model. In other words, do not go overboard with groupings.

Actually, the same references to *title* and *format* are also found in the *purchase* action. Update it to use your new grouping to be consistent (and save a few lines), as shown in Example 3-21.

EXAMPLE 3-21 Action purchase That Uses the Grouping title-format-ref

```
container users {  
  list user {  
    key user-id;
```

”

4 NETCONF, RESTCONF, and gNMI Explained

Uses the BookZone model to show protocol operations.

- Lot's of pasted messages.
- Shows in great detail why transactions are useful and how they work.
- Compares the strengths and weaknesses of each protocol and discusses use cases where they are suitable.



RESTCONF Versus NETCONF

Many of the REST principles are very similar to how NETCONF works. The client-server model, the layered system principle, and the first two uniform interface principles are exactly the same. A key differentiator is the stateless server principle. NETCONF is based on clients establishing a session to the server, and that is certainly not stateless. In NETCONF, clients often connect and then manipulate the candidate datastore with a number of edit-config operations. At a certain time, the client may issue a validation call. If that succeeds on all servers, a commit operation may follow.

This sort of behavior is not possible in RESTCONF or any system that strictly follows the REST principles. It requires the server to keep some client state. If that was added to RESTCONF, the protocol no longer fulfills the design principles of REST. Neither could this be added in a future version of RESTCONF, for the same reason. Strictly speaking, it wouldn't be “REST” anymore.



5 Telemetry Explained

Telemetry Architecture, History and Implementations

- Telemetry concepts:
On-change, Periodic, Dial-In, Dial-Out
- Telemetry mechanisms:
 - NETCONF Event Notifications
 - YANG Push
 - OpenConfig Streaming Telemetry

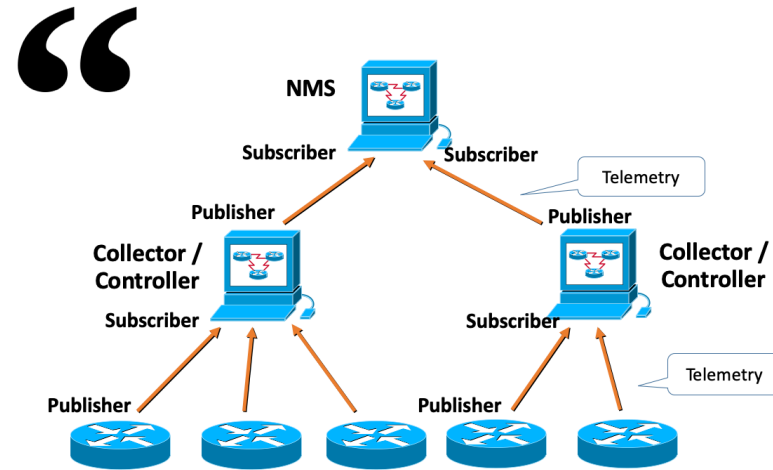


FIGURE 5-2 Telemetry Architecture

6 YANG Data Modeling Developments in the Industry

Who is adopting YANG, how fast, and what's going on at different places?

- OpenDaylight, BBF, MEF, IEEE, DMTF, Sysrepo, ITU-T, OpenROADM, FD.io, O-RAN, 3GPP, ETSI, OpenConfig, YangCatalog, EANTC

“

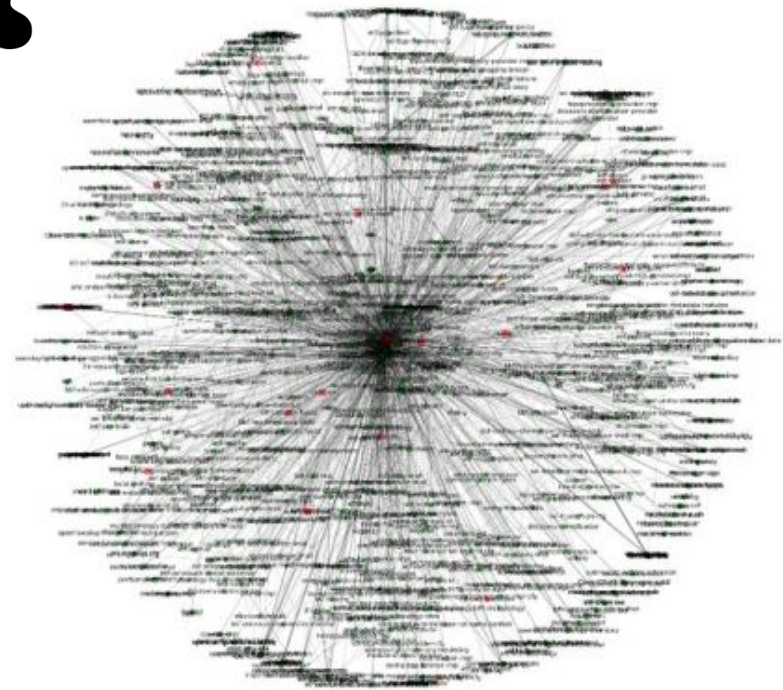


FIGURE 6-8 All YANG Modules Dependency Graph, YANG Catalog Symbol

”

7

Automation Is as Good as the Data Models, Their Related Metadata, and the Tools: For the Network Architect and Operator

Tools for finding YANG modules and interacting with network elements

- YANG Tools: YANG Catalog, pyang
- NETCONF/RESTCONF Tools: YANG Suite, netconf-console, ncclient, Ansible, cURL, HTTPie, Postman
- Telemetry Tools: Pipeline, ANX
- Commercial Products

“

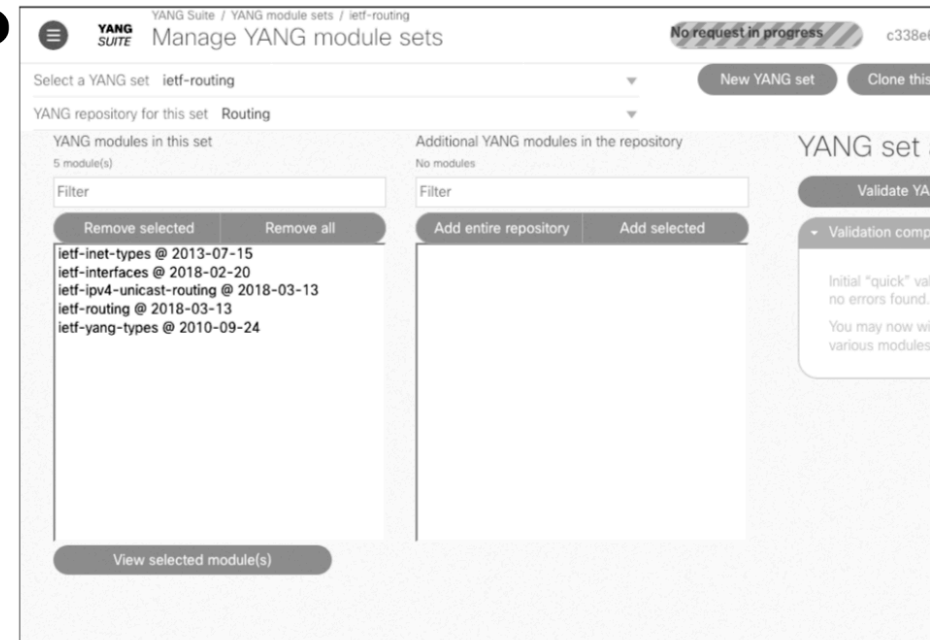


FIGURE 7-7 YANG Set in YANG Suite

”

8

Automation Is as Good as the Data Models, Their Related Metadata, and the Tools: For the Module Author

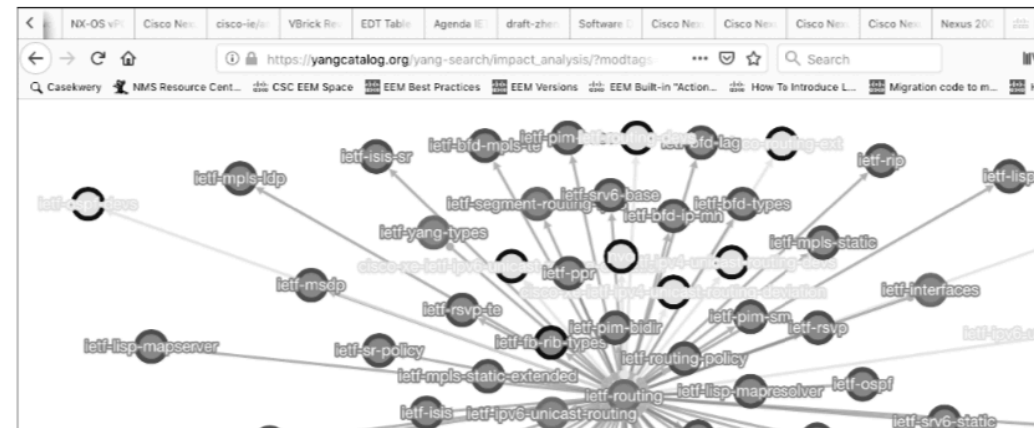
Tools for finding YANG modules and their meta-data. Tools for extracting and validating YANG modules.

- xym
- YANG Catalog
- pyang, confdc, yanglint

“ Understanding Your Module’s Impact

Another tool provided by the YANG Catalog is called “Impact Analysis.” This tool provides you with a color-coded graph showing how one or more modules interrelate in terms of their dependencies and dependents. That is, Impact Analysis shows what modules depend on a given module, as well as what modules a given module depends on. Figure 8-2 provides an Impact Analysis graph showing both dependencies and dependents for ietf-routing.

”



9

Automation Is as Good as the Data Models, Their Related Metadata, and the Tools: For the Application Developer

Tools for parsing YANG modules, interacting with devices over NETCONF, RESTCONF, generating code.

- YANG Catalog, pyang plugins, libyang, ncclient, libnetconf2, requests, curl, rest-client, REST::Client, Sling, YDK, Pyangbind

“ This creates an oc_if.py file that contains all the binding definitions for openconfig-interfaces.yang and openconfig-if-ip.yang. This module can then be used to build the payload for various NETCONF or RESTCONF operations. Those payloads can be used by ncclient or the requests module for sending the data directly to the device. Example 9-20 shows how to build an RPC payload using pyangbind and then using ncclient to add an IP address to an interface.

EXAMPLE 9-20 Building and Executing an RPC with Pyangbind and Ncclient

```
1      #!/usr/bin/env python
2
3      from oc_if import openconfig_interfaces
4      from pyangbind.lib.serialise import
pybindIETFXMLEncoder
5      from ncclient import manager
6
7      def send_to_device(**kwargs):
8          rpc_body = '<config>' +
```

”

10 Using NETCONF and YANG

“ The purpose of this chapter is to show a full story, starting with a business need and taking it all the way to verifying that it works correctly in the network. This chapter builds a somewhat real software-defined wide area network (SDWAN) service using the Network Services Orchestrator (NSO) product as the orchestrator. The first step is to design a YANG service model, map it to the device models, write a little bit of code to assist in that mapping, and then plug in some devices, configure them for NETCONF management, and try out the service with some service creation, modifications, and rollback. This chapter also examines how the service gets deployed from an orchestrator point of view as well as the NETCONF messages flying around between the orchestrator and devices. The project is available as a hands-on project you can clone from GitHub and then build, run, and play with as you please. Instructions for how to obtain the necessary free tools are found within the project README file at <https://github.com/janlindblad/bookzone>. ”

BookZone project #7 builds an NSO Service Application

“ Once the key is created, log in to the device, install the private key on the device, and enable NETCONF, as shown in Example 10-17.

EXAMPLE 10-17 Installing the Crypto Key Pair and Enabling NETCONF on JunOS

```
edit system login user username
authentication
set load-key-file sftp://.../mykey
commit
edit system services
set netconf ssh
commit
```

On a Cisco IOS-XE device, the procedure is similar. First, a crypto key pair needs to be generated, ”

11 YANG Model Design

- Modeling Strategy
- YANG Modeling Tips
 - Naming a Module, Using Leafrefs, Minding That XPath Feature, Enumerating and More, Choosing Keys, Types Empty and Boolean, Deviating...
- Common Mistakes
- Backwards Compatibility

“ Therefore, there is a YANG statement called deviate that allows implementers to officially declare in what ways an implementation differs from the standard. The existence of this keyword, however, does not imply in any way that clients will be able to deal with the declared deviation. In short, deviations are very poor substitutes for not implementing a standard.

A deviation is somewhat akin to an “out of order” sign hanging off a coffee machine or a conference room projector. It signals a non-ideal condition, but at least the user is made aware, doesn’t waste her time figuring out the problem, and is less likely to send reports about the issue.

”

“It is well-written book ... My group especially will mandate every new guy to go through this book. I personally spent a lot of time to find tools/understand yang etc, when I joined yang group. All this time could have been saved with this book.”

Harjinder Singh

