

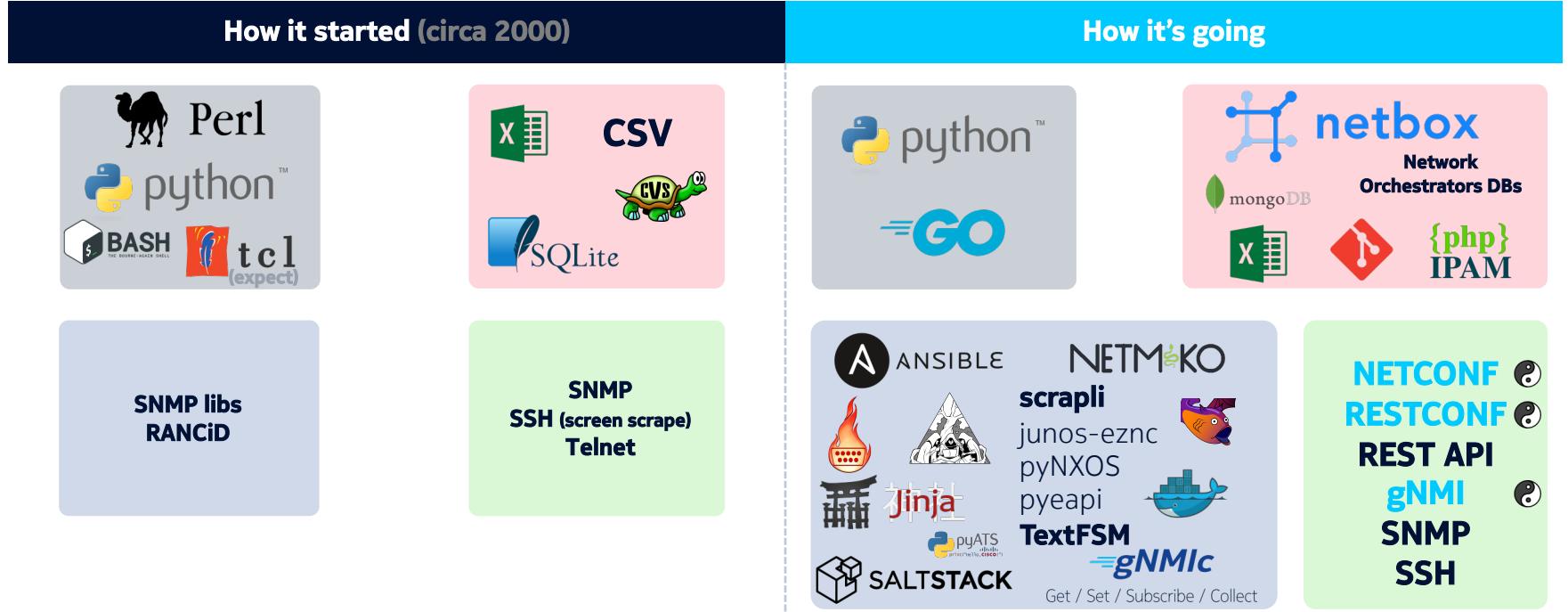
Navigating the YANGscape of Network Automation

Roman Dodin

Consulting Engineer, Nokia

10-12-2020

Network automation landscape development



Gen purpose langs

Source of truth

Frameworks & Libs

Interfaces

Public



[awesome-network-automation](https://awesome-network-automation.com)

Management interfaces

Structured vs unstructured

SSH/CLI



```
A:admin@R3_20.10.R1# show version
TiMOS-B-20.10.R1 both/x86_64 Nokia 7750 SR Copyright (c) 2000-2020 Nokia.
All rights reserved. All use subject to applicable license agreements.
Built on Wed Nov 4 09:18:17 PST 2020 by builder in /builds/c/2010B/R1/panos/main/sros
```

NETCONF



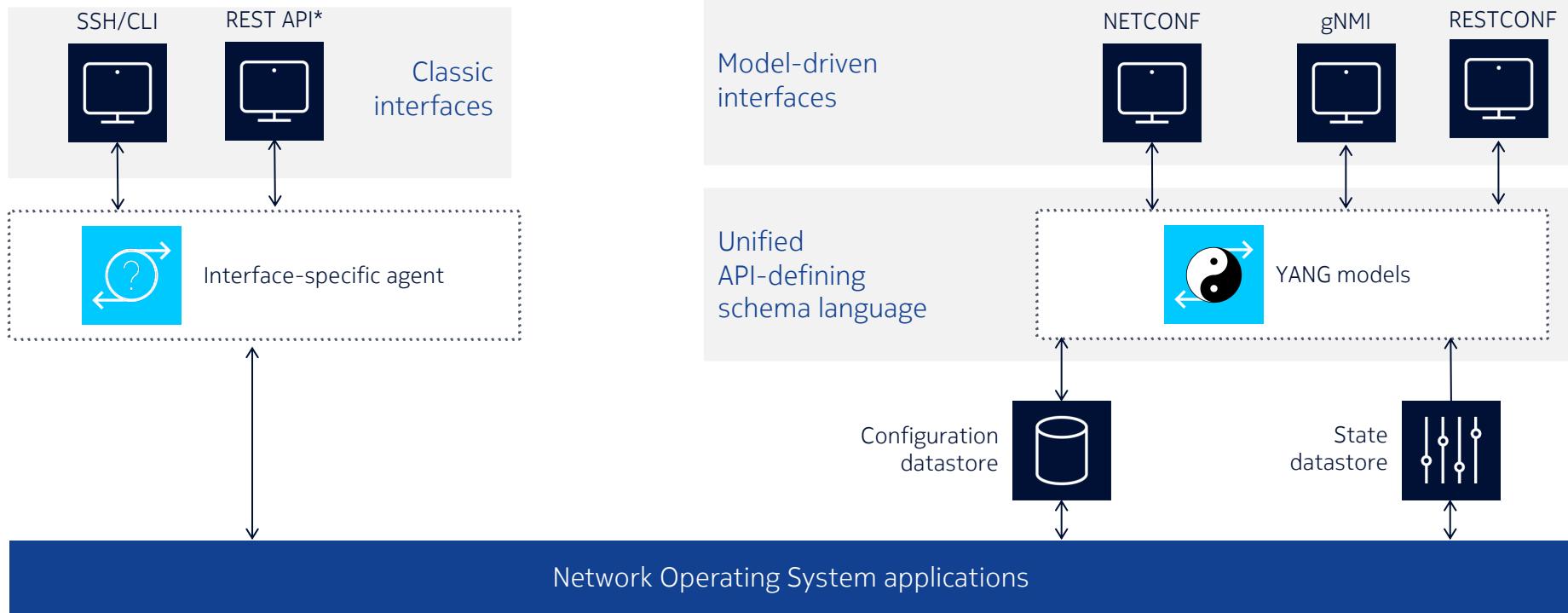
```
<state xmlns="urn:nokia.com:sros:ns:yang:sr:state">
  <system>
    <version>
      <version-number>B-20.10.R1</version-number>
      <version-string>TiMOS-B-20.10.R1 both/x86_64 Nokia 7750 SR Copyright (c) 2000-2020 Nokia.
All rights reserved. All use subject to applicable license agreements.
Built on Wed Nov 4 09:18:17 PST 2020 by builder in /builds/c/2010B/R1/panos/main/sros</version-string>
    </version>
  </system>
</state>
```

RESTCONF



```
{
  "nokia-state:version": {
    "version-string": "TiMOS-B-20.10.R1 both/x86_64 Nokia 7750 SR Copyright (c) 2000-
2020 Nokia.\nAll rights reserved. All use subject to applicable license agreements.\nBuilt on Wed Nov 4 09:18:17 PST 2020 by builder
in /builds/c/2010B/R1/panos/main/sros",
    "version-number": "B-20.10.R1"
  }
}
```

Classic vs Model driven interfaces



* May or may not have a model

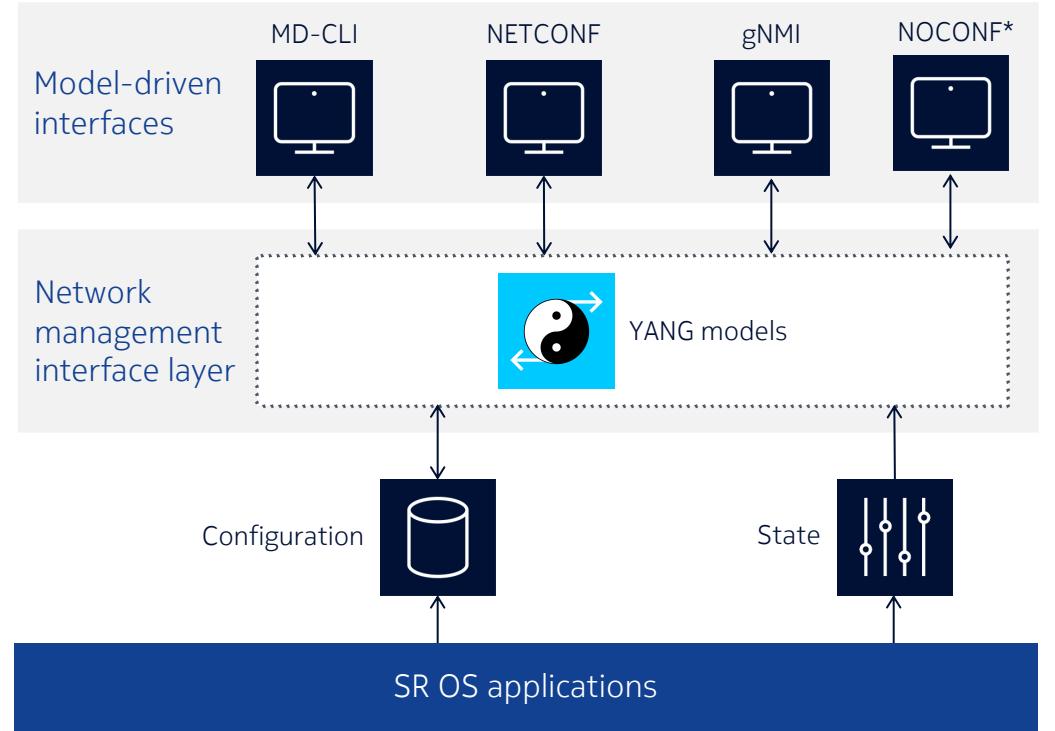
YANG modelling language

Role & purpose



[RFC6020](#) - YANG
[RFC6021](#) - YANG types
[RFC7950](#) - YANG 1.1

YANG is a data modelling language used to model configuration data, state data, Remote Procedure Calls, and notifications for network management protocols.



* Some new management interface

YANG

Inside the YANG module

- YANG models the hierarchical organization of data as a tree in which
 - each node has a name,
 - and either a value or a set of child nodes.
- YANG provides clear and concise descriptions of the nodes, as well as the interaction between those nodes.

ietf-interfaces.yang - tree

```
++-rw interfaces
| +-rw interface* [name]
| | +-rw name
| | | string
| | +-rw description?
| | | string
| | +-rw type
| | | identityref
| | +-rw enabled?
| | | boolean
```

ietf-interfaces.yang

```
module ietf-interfaces {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-interfaces";

    import ietf-yang-types {
        prefix yang;
    }

    revision 2018-02-20;

    container interfaces {
        description
            "Interface parameters.";

        list interface {
            key "name";
            leaf name {
                type string;
            }
            leaf description {
                type string;
            }
            leaf enabled {
                type boolean;
                default "true";
            }
        }
    }
}
```

Model-driven interfaces & YANG

How do Model Driven interfaces leverage YANG?

NETCONF

Uses YANG

edit-config operation to add the user

```
<rpc message-id="104" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target><candidate/></target>
    <config>
      <configure xmlns="urn:nokia.com:sros:ns:yang:sr:conf">
        <system>
          <security>
            <user-params>
              <local-user>
                <user>
                  <user-name>john</user-name>
                  <password>test123</password>
                  <access>
                    <console>true</console>
                  </access>
                  <console>
                    <member>default</member>
                  </console>
                </user>
              </local-user>
            </user-params>
          </security>
        </system>
      </configure>
    </config>
  </edit-config>
</rpc>
```

nokia-conf-combined.yang

```
module nokia-conf {
  yang-version 1.1;
  namespace "urn:nokia.com:sros:ns:yang:sr:conf";
  prefix "conf";
```

```
module: nokia-conf
  +-rw configure
    +-rw system
      +-rw security
        |+-rw user-params
          +-rw local-user
            +-rw user* [user-name]
              +-rw user-name      types-sros:named-item
              +-rw password       types-sros:hashed-leaf
              +-rw access
                +-rw console?    boolean
                +-rw ftp?        boolean
                +-rw snmp?       boolean
                +-rw netconf?    boolean
                +-rw grpc?       boolean
                |+-rw li?         boolean
                +-rw console
                  +-rw member*   ->.../aaa/local-profiles...
```

gNMI

Uses YANG

*The data tree supported by the target is expected to be defined by a set of schemas.
The definition and format of these models is out of scope of this specification (YANG-modeled data is one example).*

gNMI specification ch. 2.6

```
$ gnmic --address 10.1.0.11:57400 -u admin -p admin --insecure subscribe --path state/port[port-id=1/1/1]/statistics/in-packets
```

```
{  
  "timestamp": 1588923058095558213,  
  "prefix": "state/port[port-id=A/1]/statistics",  
  "updates": [  
    {  
      "Path": "in-packets",  
      "values": {  
        "in-packets": "423"  
      }  
    }  
  ]  
}
```

```
module: nokia-state  
++-ro state  
| << SNIPPED ... >>  
++-ro port* [port-id]  
| +-ro port-id  
| +-ro statistics  
| << SNIPPED ... >>  
| +-ro in-octets?  
| +-ro in-packets?  
| +-ro out-packets?  
| +-ro out-broadcast-packets?  
| +-ro out-multicast-packets?  
| +-ro out-unicast-packets?  
types-sros:port  
yang:counter64  
yang:counter64  
yang:counter64  
yang:counter64  
yang:counter64  
yang:counter64
```

RESTCONF Uses YANG



* Executed via NSP MDC

[GET] [https://nsp:8545/restconf/data/network-device-mgr:network-devices/
network-device=192.168.1.12/root/nokia-conf:configure/policy-options/policy-statement=policy1](https://nsp:8545/restconf/data/network-device-mgr:network-devices/network-device=192.168.1.12/root/nokia-conf:configure/policy-options/policy-statement=policy1)

```
"nokia-conf:policy-statement": [  
  {  
    "default-action": {  
      "action-type": "reject"  
    },  
    "entry": [  
      {  
        "entry-id": 10,  
        "action": {  
          "action-type": "next-policy"  
        },  
        "from": {  
          "protocol": {  
            "name": [  
              "bgp-vpn"  
            ]  
          },  
          "community": {  
            "name": "rt-com"  
          }  
        },  
        "to": {}  
      }  
    ]  
  }]
```

```
module: nokia-conf  
++-rw configure  
|  
+-rw policy-options  
| +-rw policy-statement* [name]  
| | +-rw name          types-sros:named-item-64  
| | +-rw entry* [entry-id]  
| | | +-rw entry-id      uint32  
| | | +-rw from!  
| | | | +-rw level?      int32  
| | | | +-rw community  
| | | | | +-rw (community)?  
| | | | | | +-:(name)  
| | | | | | | +-rw name?    string  
| | | | | | | +-:(expression)  
| | | | | | | | +-rw expression?  string  
| | +-rw protocol  
| | | +-rw name*        enumeration  
| | | +-rw instance?    union  
| +-rw to!  
| | +-rw level?        Int32  
| +-rw action!  
| | +-rw action-type    enumeration
```

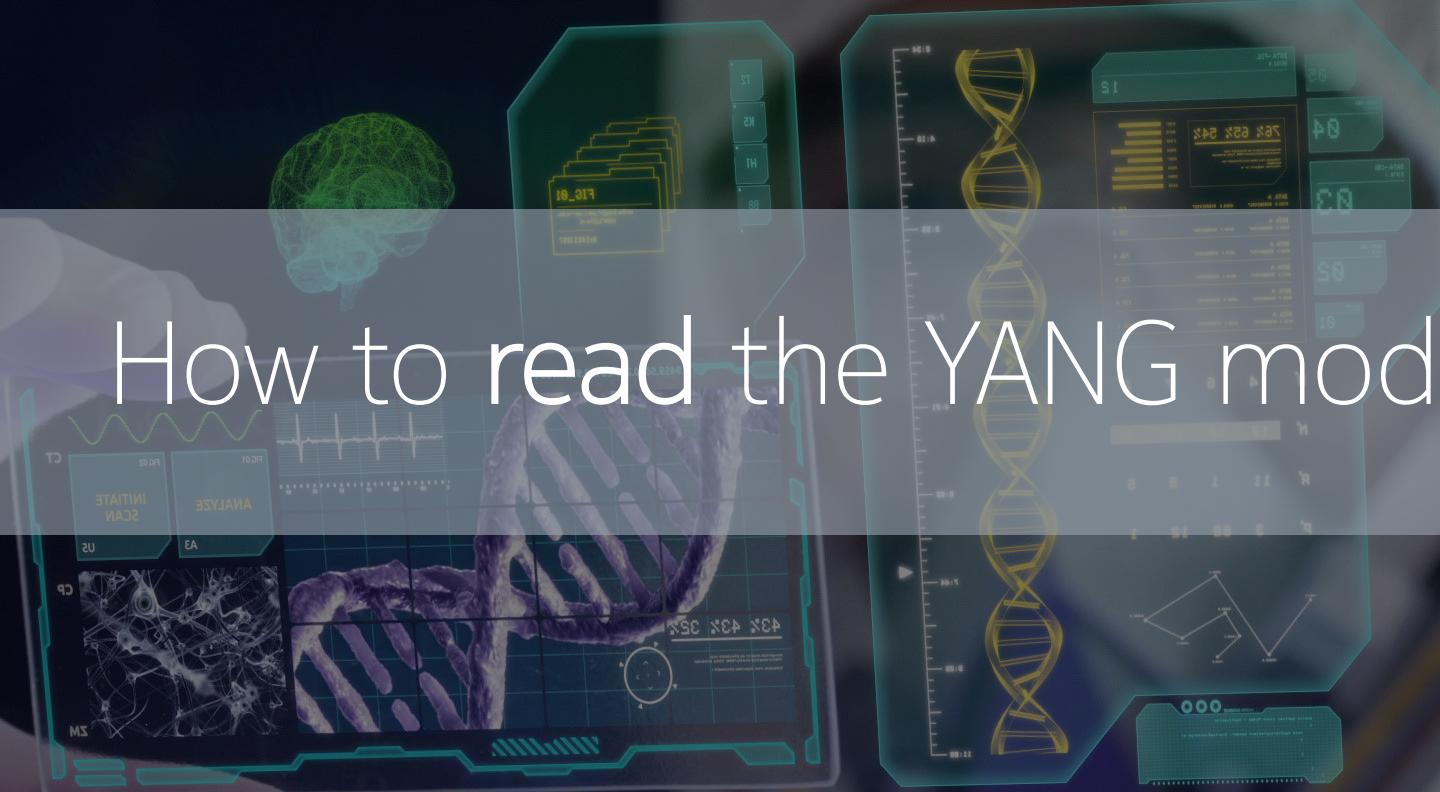
MD-CLI

And again, YANG is your source of truth

```
module: nokia-conf
++-rw configure
+-+rw system
| +-+rw security
| | +-+rw user-params
| | | +-+rw local-user
| | | | +-+rw user* [user-name]
| | | | | +-+rw user-name
| | | | | | +-+rw password
| | | | | | | +-+rw access
| | | | | | | | +-+rw console? boolean
| | | | | | | | +-+rw ftp? boolean
| | | | | | | | +-+rw snmp? boolean
| | | | | | | | +-+rw netconf? boolean
| | | | | | | | +-+rw grpc? boolean
| | | | | | | | +-+rw li? boolean
| | | | | | | | +-+rw console
| | | | | | | | | +-+rw member* ->../aaa/local-profiles...
```

```
(ro)[configure system security user-params]
A:admin@vsim10# info
  local-user {
    user "admin" {
      password "$2y$10$TQrZlpBDra86"
      access {
        console true
        netconf true
      }
      console {
        member ["administrative"]
      }
    }
    user "grpc" {
      password "$2y$10$Bxvc0xLu5rsxcfdttf66"
      access {
        grpc true
      }
      console {
        member ["administrative"]
      }
    }
  }
```

How to read the YANG models?



Getting YANG models

IETF



[YangModels/yang](#)

OpenConfig



[openconfig/public](#)

Nokia



[Nokia/7x50_YangModels](#)

Cisco



[YangModels/yang](#)

Arista



[aristanetworks/yang](#)

Juniper



[Juniper/yang](#)

Getting YANG models

Nokia SR OS YANG repository

1. Open the [nokia/7x50_YangModels](#) public github repository
2. Clone it

The screenshot shows the GitHub repository page for 'nokia / 7x50_YangModels'. The repository has 11 stars, 13 forks, and 4 open issues. It contains 9 commits, 6 branches, 0 packages, 12 releases, and 0 contributors. The 'Code' tab is selected. A message states 'No description, website, or topics provided.' Below the stats, there's a dropdown for the 'master' branch and a 'New pull request' button. At the bottom, there's a 'Clone or download' button. The commit history lists several automated updates from 'SRBuilds' and manual commits for '.gitignore', 'LICENSE.md', and 'README.md'.

Commit	Message	Time
SRBuilds	Automated updating of releases. Release sros_20.5	Latest commit d007116 6 days ago
latest_sros_19.10	Automated updating of releases. Release sros_19.10	last month
latest_sros_19.5	Automated updating of releases.	6 months ago
latest_sros_19.7	Automated updating of releases.	6 months ago
latest_sros_20.2	Automated updating of releases. Release sros_20.2	2 months ago
latest_sros_20.5	Automated updating of releases. Release sros_20.5	6 days ago
.gitignore	Initial commit. Nokia YANG Data model to configure authentication, au...	6 months ago
LICENSE.md	Initial commit. Nokia YANG Data model to configure authentication, au...	6 months ago
README.md	README is updated	6 months ago

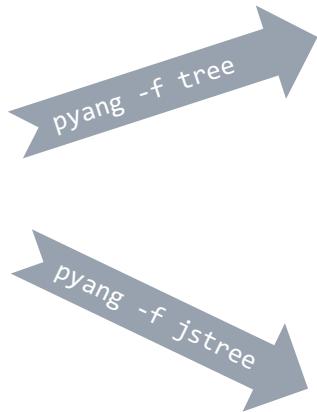
YANG models

Easy for machines, what about humans?

```
ietf-interfaces.yang
module ietf-interfaces {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-interfaces";
    import ietf-yang-types {
        prefix yang;
    }
    revision 2018-02-20;

    container interfaces {
        description
            "Interface parameters.";

        list interface {
            key "name";
            leaf name {
                type string;
            }
            leaf description {
                type string;
            }
            leaf enabled {
                type boolean;
                default "true";
            }
        }
    }
}
```



ietf-interfaces.yang - tree

```
+--rw interfaces
|   +-+rw interface* [name]
|       +-+rw name
|       +-+rw description?
|       +-+rw type
|       +-+rw enabled
```

ietf-interfaces.yang - HTML

Element [+][Expand all](#) [-][Collapse all](#)

Element	Schema	Type
ietf-interfaces	module	
interfaces	container	
interface	list	
name	leaf	string
description	leaf	string
type	leaf	identityref
enabled	leaf	boolean
link-up-down-trap-enable	leaf	enumeration
admin-status	leaf	enumeration
oper-status	leaf	enumeration
last-change	leaf	yang:date-and-time
if-index	leaf	int32
phys-address	leaf	yang:phys-address
higher-layer-if	leaf-list	interface-ref
lower-layer-if	leaf-list	interface-ref
speed	leaf	yang:gauge64
statistics	container	



YANG tree repository

For Nokia models

YANG Tree Browser

Location: [hellt/nokia-yangtree](https://github.com/hellt/nokia-yangtree)

Purpose: easily navigate through the YANG tree

The screenshot shows the GitHub repository page for `hellt/nokia-yangtree`. The repository has 27 commits, 1 branch, 0 packages, 9 releases, and 1 contributor. The latest commit was made 5 days ago. The repository description is "HTML tree and Path Browser for Nokia 7x50 YANG models". The code tab is selected, showing a list of files: .gitignore, README.md, sros_20.5.r1-nokia-conf-combined-paths.html, sros_20.5.r1-nokia-conf-combined-paths.txt, sros_20.5.r1-nokia-conf-combined.html, sros_20.5.r1-nokia-conf-combined.txt, sros_20.5.r1-nokia-conf-combined.xml, sros_20.5.r1-nokia-state-combined-paths.html, sros_20.5.r1-nokia-state-combined-paths.txt, sros_20.5.r1-nokia-state-combined.html, sros_20.5.r1-nokia-state-combined.txt, and sros_20.5.r1-nokia-state-combined.xml.

File	Description	Age
.gitignore	ignore pub script	last month
README.md	updated animation in readme	5 days ago
sros_20.5.r1-nokia-conf-combined-paths.html	regenerated nokia-conf combined paths	5 days ago
sros_20.5.r1-nokia-conf-combined-paths.txt	regenerated nokia-conf combined paths	5 days ago
sros_20.5.r1-nokia-conf-combined.html	models for sros_20.5.r1	5 days ago
sros_20.5.r1-nokia-conf-combined.txt	models for sros_20.5.r1	5 days ago
sros_20.5.r1-nokia-conf-combined.xml	models for sros_20.5.r1	5 days ago
sros_20.5.r1-nokia-state-combined-paths.html	models for sros_20.5.r1	5 days ago
sros_20.5.r1-nokia-state-combined-paths.txt	models for sros_20.5.r1	5 days ago
sros_20.5.r1-nokia-state-combined.html	models for sros_20.5.r1	5 days ago
sros_20.5.r1-nokia-state-combined.txt	models for sros_20.5.r1	5 days ago
sros_20.5.r1-nokia-state-combined.xml	models for sros_20.5.r1	5 days ago

DEMO

Leveraging pyang tree outputs

YANG schema paths

Telemetry subscriptions bread and butter

NETCONF (w XPATH) & gNMI

/state/port[port-id=1/1/c1/1]/statistic/in-octets

```
module: nokia-state
++-ro state◆
| << SNIPPED >>
++-ro port* [port-id]◆
| ++-ro port-id◆
++-ro statistics◆
| << SNIPPED >>
| ++-ro in-octets?◆
| +-ro in-packets?
| +-ro out-packets?
| +-ro out-broadcast-packets?
| +-ro out-multicast-packets?
| +-ro out-unicast-packets?
```

RESTCONF

/nokia-state:state/port=1%2F1%2Fc1%2F1/statistics/in-octets

Exporting Paths from YANG



YANG Path Exporter

yangpath.netdevops.me



oc-pyang

github.com/openconfig/oc-pyang

D E M O

Exporting paths from YANG

YANG schema paths

Nokia path browser

YANG Path Browser

Location: [hellt/nokia-yangtree](https://github.com/hellt/nokia-yangtree)

Purpose: effectively search through the XPATH paths

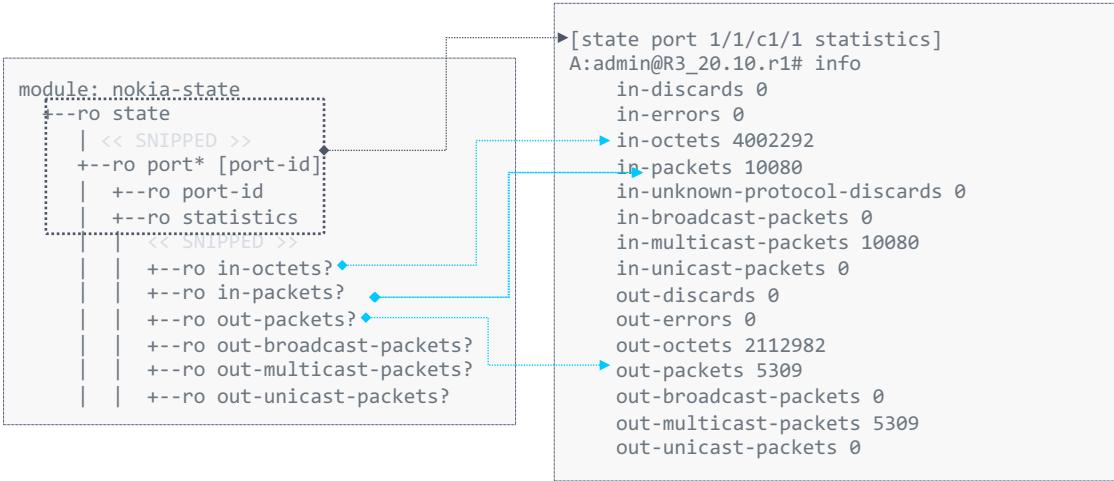
The screenshot shows the GitHub repository page for `hellt/nokia-yangtree`. The repository has 27 commits, 1 branch, 0 packages, 9 releases, and 1 contributor. The latest commit was `ff159ee` 5 days ago. The commits are listed below, all related to regenerating combined paths for Nokia models.

Commit	Description	Time Ago
.gitignore	ignore pub script	last month
README.md	updated animation in readme	5 days ago
sros_20.5.r1-nokia-conf-combined-paths.html	regenerated nokia-conf combined paths	5 days ago
sros_20.5.r1-nokia-conf-combined-paths.txt	regenerated nokia-conf combined paths	5 days ago
sros_20.5.r1-nokia-conf-combined.html	models for sros_20.5.r1	5 days ago
sros_20.5.r1-nokia-conf-combined.txt	models for sros_20.5.r1	5 days ago
sros_20.5.r1-nokia-conf-combined.xml	models for sros_20.5.r1	5 days ago
sros_20.5.r1-nokia-state-combined-paths.html	models for sros_20.5.r1	5 days ago
sros_20.5.r1-nokia-state-combined-paths.txt	models for sros_20.5.r1	5 days ago
sros_20.5.r1-nokia-state-combined.html	models for sros_20.5.r1	5 days ago
sros_20.5.r1-nokia-state-combined.txt	models for sros_20.5.r1	5 days ago
sros_20.5.r1-nokia-state-combined.xml	models for sros_20.5.r1	5 days ago

Not an official Nokia product

On-box config & state navigation

Navigating YANG model from CLI



Access to YANG config & state in MD-CLI

- Easily navigate and display state with **tree**, **info**, ? help and command completion like you can for configuration
- Integrate with telemetry automation with XPATH and model-path output
- Validate telemetry automation with the MD-CLI by comparing data
- Display state info in structured output like JSON or XML (roadmap) as show command replacements

DEMO

Navigating YANG datastores within Nokia MD-CLI

YANG path completions

YANG browser in your CLI



Get / Set / Subscribe / Collect

gNMic prompt mode

provides interactive YANG path auto-suggestions for gNMI **get, set, subscribe**

commands effectively making the terminal your **YANG browser**

Location: [karimra/gnmic](#)

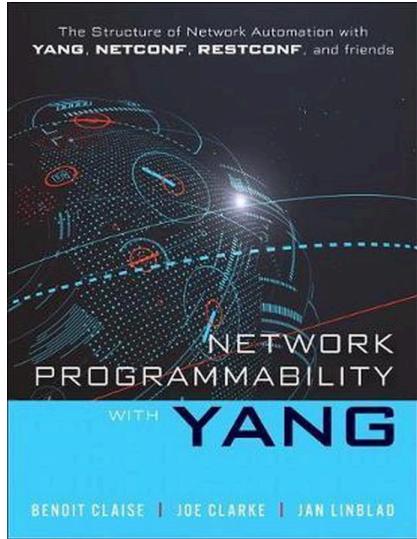
Documentation: [Prompt suggestions](#)

```
root@srl-centos7 17:02:44
~> gnmic --address 10.1.0.11 --insecure --username admin --password admin \
--file ~/7x50_YangModels/YANG/nokia-combined --dir ~/7x50_YangModels/YANG \
prompt
gnmic> get --path /state/system/■
active-cpm-slot
alarm-contact-input
alarm-contact-input[input-pin-number=**]
alarms
base-mac-address
(ro) The value indicates which CPM is...
[+] (ro) Enter the alarm-contact-input li...
[+] (ro) Enter the alarm-contact-input li...
[+] (ro) Enter the alarms context
[+] (ro) Indicates the base system ethern...
```

DEMO

Traversing YANG models within CLI

Where to go next?



Summary

1

YANG is in the DNA
of model driven
interfaces

2

Vendors and standard
bodies publish YANG
models on GitHub

3

Reading and
leveraging YANG
outputs
becomes an
essential skill

4

YANG source is repr.
is tricky for humans.
Use tree and HTML
output options

5

Nokia YANG tools
are here help to get
started with
YANG-based automation

6

The YANG shift is
happening.
Take advantage,
start early
and contribute!

Thank you

