



YangForge Data Models for Everyone

Model-driven Multi-layer abstractions for flexible cross-platform integration

10 November 2015

Peter Lee

ClearPath Networks, CTO

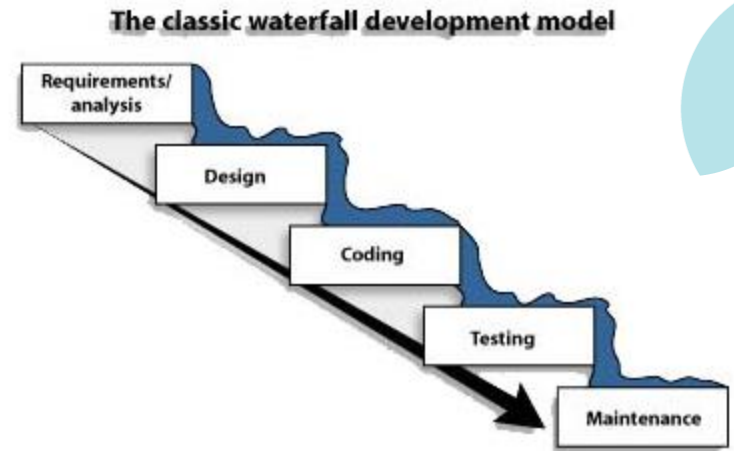
Content Overview

- Why do we need this?
- What does it do?
- Let's see it in action!
- How is being used today?
- Where is it going?
- Q&A



Why do we need this?

- It takes **way too long** to standardize
 - **Information Models** created by SDOs takes a long time before it gets realized into working implementations
- Even with standards, things still **do not work well together**
 - **Data Models** created by industry implementations only work for specific implementation needs, often only within a given vendor's solution
- After we finally get things to work, it then becomes almost **impossible to make changes**
 - **Implementations** are costly to modify, prone to errors and well... how do you get everyone to change?



How does YANG help?

- It **standardizes** how to standardize
 - **Information Models** created by SDOs can be expressed as **YANG Data Models** which then become *immediately* consumable for implementation
- It is **agnostic** to implementation-specific Data Models
 - **Data Models** created by industry implementations can be expressed as **YANG Data Models** which then become *immediately* consumable for integration
- It is **modular** and **extensible** for sharing, augmenting and deviating existing Data Models
 - **Implementations** can dynamically adapt to changing YANG Data Models with minimal changes

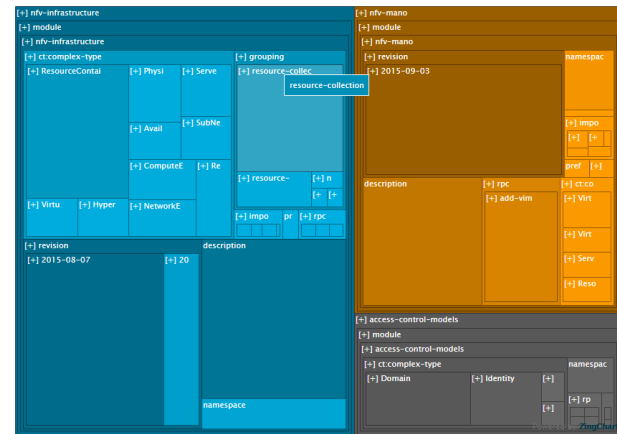


Disclaimer: This slide contains "**forward-looking statements**" – that is, statements related to future, not past, events

What does YangForge do exactly?

- It brings YANG Data Models **to life**
 - **One** command *transforms* YANG input into a runtime instance with dynamically generated interfaces
 - CLI
 - REST/JSON
 - SOCKET.IO
 - BROWSER
 - **Visualize** and **Interact** with the Data Models with rich web-client application
- It **empowers** YANG Data Models with custom behaviors and control logic
 - YAML control logic definitions
 - JSON configuration data

```
~/hack/yangforge
[master]~$ yfc run examples/jukebox.yang
executing rpc run...
executing rpc run...
forgery firing up...
running in development mode
express: listening on 5000
generating REST/JSON interface...
restjson: binding forgery to /restjson
websocket: binding forgery to /socket.io
browser: yangforge browserification started (will take a few seconds)
browser: binding to express /browser
browser: binding to websocket /browser
executing rpc infuse...
running with: yangforge,example-jukebox
```



Agile is more than a process... it requires the *proper* enablement tools!



How does it compare to other YANG tools?

- It is written in **CoffeeScript** which transpiles to **JavaScript**
 - **YangForge** runs on **Node.js** and all modern web browsers
- It is **NOT** a code stub generator
 - **YangForge** completely separates **Data Models** from **Control Logic**
- It does everything **on-the-fly** in runtime
 - **YangForge** can be re-programmed while already running
- It is **100% open-source** software

YangForge is designed for *fostering collaboration* between standards organizations and the development community



How is it being used today?



- **OPNFV Promise – Resource Reservation**
 - Resource Abstraction Layer being expressed using YANG
 - Come and see the demo at OPNFV Project Theater
 - **Thursday 2:25 - 2:45pm**
 - Promise - Planning the Future
- **ON.LAB – XOS/ONOS**
 - Service Abstraction Layer being expressed using YANG
 - Streamline Service Controller App management
- **ClearPath Networks – VNF Catalog**
 - Function Abstraction Layer being expressed using YANG
 - Streamline VNF on-boarding for applications and libraries

Where is it going?



- **Live Editor**
 - Enable users to make changes to the data model using the web browser
- **Additional Interfaces**
 - Storage Adaptors
 - XMPP
 - Protocol Buffers
- **More Visualization**
 - Interactive Data Model Network Graph
 - Performance Charts and Graphs

Closing Remarks and Summary



- We want your feedback – give **YangForge** a try and let us know your experience!
 - <http://npmjs.org/package/yangforge>
 - <http://github.com/saintkepha/yangforge>
- We need your help – help make **YangForge** better!
 - Contact plee@clearpathnet.com
 - Also on [#yangforge](https://opnfv.slack.com)
- Help spread the word – let's build a community!

Q&A