Joint IEEE 802 and ITU-T SG15 Workshop
“Building Tomorrow’s Networks”

ITU-T Information & Data Modeling & Status

Kam LAM (FiberHome) Q14/15 Rapporteur
Scott MANSFIELD (Ericsson) Q14/15 Associate Rapporteur
Objectives

• Objectives of this joint workshop session #4
  – Information sharing
    • IEEE
      – IEEE 802.1Qcp (Bridge YANG DM)
      – IEEE 802.1Qcx (CFM YANG DM)
      – IEEE 802.3 Ethernet YANG DM definition
      – IEEE 1588 YANG DM
    • ITU-T
      – ITU-T Information and Data modeling
      – G.8052.1 Transport Ethernet OAM Information Model and Data Model
  – Identify input for the Q14/15 Interim meeting (Tomorrow 28 Jan.)
Objectives

• Objectives of the Q14/15 Interim meeting (28 Jan.)
  ❖ From Q14/15 perspective; could be updated
  – Ensure alignment of the IEEE YANG works with ITU-T G.8052.1
    • Advancing G.8052.1 will entail the creation of an UML OAM model
      (pruned/refactored from G.8052 and supporting the G.8013/Y.1731-defined
      OAM) and its translation as a YANG module
    • Alignment of ITU-T YANG with IEEE 802.1Qcp and 802.1Qcx
  – Identify the anchor point (touch point) of the IEEE 802.1 YANG
    modules for augmentation by the G.8052.1 YANG module
  – Identify the G.8052 UML artifacts (object classes, attributes and
    operations) that should be pruned (e.g., due to out of scope, or
    available in the 802.1 YANG), kept, or refactored
ITU-T Q14/15 Modeling Work

- Mandate
- Scope
- Modeling Methodology
- UML to YANG Generation
- Status
- Alignment
  - Augmentation
  - Pruning/Refactoring
Q14/15 Mandate & Scope

Management and control of transport systems and equipment

- Technology architecture & function → Management/Control
  - requirement, information model, data model

<table>
<thead>
<tr>
<th></th>
<th>Generic</th>
<th>SDH</th>
<th>OTN (L1)</th>
<th>Carrier Ethernet</th>
<th>MPLS-TP</th>
<th>Media (L0)</th>
<th>Sync</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport Architecture</td>
<td>G.800</td>
<td>G.803</td>
<td>G.872</td>
<td>G.8010</td>
<td>G.8110.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>G.805</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G.media</td>
<td></td>
</tr>
<tr>
<td>Equipment Function</td>
<td>G.806</td>
<td>G.783</td>
<td>G.798</td>
<td>G.8021</td>
<td>G.8121.x</td>
<td>G.8265.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G.8275.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>G.8275.2</td>
<td></td>
</tr>
<tr>
<td>Mgmt/Control Requirement</td>
<td>G.7710</td>
<td>G.784</td>
<td>G.874</td>
<td>G.8051</td>
<td>G.8151</td>
<td>G.media-mgmt</td>
<td>G.sync-mgmt</td>
</tr>
<tr>
<td>Information Model</td>
<td>G.7711</td>
<td>G.774.1-10 (CMISE)</td>
<td>G.874.1</td>
<td>G.8052</td>
<td>G.8152</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Model</td>
<td></td>
<td>G.874.x</td>
<td>G.8052.x (.1 YANG)</td>
<td>G.8152.x</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Model Development Methodology

Common Information Model
- Core Model G.7711 / ONF TR-512
  - Forwarding & Termination
  - Foundation
  - Topology
  - Resilience
  - Physical
  - ....
- Technology specific models, e.g.,
  - G.874.1 OTN
  - G.8052 Transport ETH
- Application specific models (e.g. storage)
- xxx model

Interface Creation Process

Guidelines
- Model structure
- UML Common Guidelines
- Papyrus tool and GitHub
- Common process
- Interface specific
- Interface specific
- Interface specific
UML Modeling Guidelines evolution

- ITU-T SG15 Modelling Guidelines
- NGCOR Modelling Requirements
- 3GPP Model Repertoire

Other candidates:
- ONAP
- OIF
- ...
Pruning/Refactoring example

ONF OIMT

Core Information Model (TR-512/G.7711)

ONF Technology Specification Models

OTN (G.874.1)
ETH (G.8052)
MPLS-TP (G.8152)
SYNC (G.sync-mgmt)
Media (G.media-mgmt)

ITU-T SG15 Technology IMs

Pruning & Refactoring (P&R)

ONF OTCC TAPI

TAPI UML Information Model

How: Spec model approach (TR-512.7/G.7711 Annex G)
What: Technology IMs
UML to YANG Mapping Tool


  - **Background:**
    - Initially developed by the ONF open source project “EAGLE”
    - Now run by the Informal Inter SDO Open Model Initiative (IISOMI), including active participants from ONF, ITU-T SG15, MEF, TMF, ...

  - **Mapping Guidelines (rules):** IISOMI-531 v1.0

  - **Mapping Tool:** [xmi2yang](https://github.com/OpenNetworkingFoundation/EAGLE-Open-Model-Profile-and-Tools/tree/ToolChain/UmlYangTools)
    - Programming language: JavaScript
    - Running environment: node.js (downloadable from: [https://nodejs.org/en/](https://nodejs.org/en/))
    - **How to run:**
      1. Copy the UML file of the UML model into the “project” sub-folder of the xmi2yang directory
      2. Run the command “node main.js” at the xmi2yang directory
      3. The YANG file will be generated in the “project” sub-folder
UML to YANG Mapping Tool - Demo

The model

The tool

Run!

The output
Status of ITU-T Technology-specific and Generic Information Models

- **G.874.1** – OTN (L0, L1)
  - V1 (01/2002); UML using Rational Rose
  - V2 (10/2012); UML using IBM RSA; Same modeling guidelines and profile with G.8052
  - V3 (11/2016); Using **Papyrus**; Same UML Modeling Guidelines and Open Model Profile with IISOMI
  - Key object classes: OCh/OTU/ODU TTP, CTP, TCM (MEP), SN (FD), SNC (FC), PG (FcSwitch)

- **G.8052** – Carrier Ethernet
  - V1 (08/2013); UML using IBM RSA; Same modeling guidelines and profile with G.874.1
  - V2 (11/2016); Using **Papyrus**; Same UML Modeling Guidelines and Open Model Profile with IISOMI
  - Key object classes: ETH/ETY TTP, CTP, MEP, MIP, Proactive/On-Demand OAM & PM Control

- **G.8152** – MPLS-TP
  - V1 (12/2016); Using **Papyrus**; Same UML Modeling Guidelines and Open Model Profile with IISOMI

- **G.7711** – Generic, Nodal and Network view
  - V1 (08/2015); Using **Papyrus**; Same UML Modeling Guidelines and Open Model Profile with IISOMI
  - V2 (12/2016); Using **Papyrus**; Same UML Modeling Guidelines and Open Model Profile with IISOMI
  - V2.02 (1/2018); Align with ONF TR-512 v1.3.1-info; For consent at 2/2018 SG15 plenary meeting
  - Same Core Model as ONF TR-512
  - Key object classes: LTP, LP, Link, LinkPort, FD, FC, FcPort, FcSwitch, FcRoute, ...

- **G.sync-mgmt** – Synchronization management
  - V0.04 (1/2018); Using **Papyrus**; Same UML Modeling Guidelines and Open Model Profile with IISOMI

- **G.media-mgmt** – Media layer management
See the backup slides for the analysis of G.8052.1 (draft) and IEEE CFM (draft ieee802-dot1q-cfm) YANG for Alignment
THANK YOU
Backup material
Transport Ethernet object class to atomic function mapping (Figure 6-1/G.8052)

[Diagram showing Transport Ethernet object class to atomic function mapping]

(Note): ETH_TFP interface of adaptation functions towards the ETH_FT functions connects to logical link control. See [ITU-T G.8010] and function definition for details.

Grey marked functions are for further study.

Green marked functions are not defined in G.8021.
G.8052 OAM Classes for G.8052.1
Current Re-engineered UML from IEEE CFM (draft ieee802-dot1q-cfm) YANG
Current Re-engineered UML from IEEE CFM (draft ieee802-dot1q-cfm) YANG – detailed
Since the IEEE CFM YANG model is still under development, we will just use the current available re-engineered UML (in particular the lists of attributes of the Mep, MepDb, and LinktraceReply) as the base for pruning the G.8052 model for G.8052.1.
Since the IEEE CFM YANG model is still unstable, we decide to put on hold the re-engineering around the unstable CFM YANG draft model. We will just use the current available re-engineered UML (in particular the lists of attributes of the MEP, MepDb, and LtrEntry) as the base for pruning the G.8052 model for G.8052.1.
Spec approach:

- The leeeMep and leeeComponent instances will be instantiated with the G.8052.1 specified decorrion.
- This approach is for augmenting the IEEE YANG with the G.8052.1 OAM YANG (which supports the G.8013/Y.1731 OAM features).
Touch Point Model Sketch – 2: Subclassing approach

Subclassing approach: The G.8052.1 MEP classes are subclasses of IEEE MEP, and G.8052.1 ETI TP are subclasses of IEEE Component.

* The G.8052.1 MEP and TP instances will be instantiated with the inherited properties from the IEEE and IETF superclasses.

* This gives same effect as augmenting the G.8052.1 MEP and TP YANG with the IEEE and IETF properties.
Referencing approach: The IEEE and G.8052.1 instances are instantiated and point to each other for the needed properties.
P&R Model Sketch – 1: G.8052 MEP Pruning & Refactoring

```
<OpenModelClass>
  Mep
  + adminState: AdminState [1]
  + mepMac: MacAddress [1]
  + m: Integer [1]
  + clientM: Integer [1]
  + mgIdentifer: String [1]
  + isCcEnabled: Boolean [1]
  + ccPeriod: OamPeriod [1]
  + ccPriority: Integer [1]
  + lkPeriod: OamPeriod [1]
  + lkPriority: Integer [1]
  + localId: String [1]
  + name: NameAndValue [1..*]
  + label: NameAndValue [*]
  + extension: NameAndValue [*]
  + operationalState: OperationalState [0..1]
  + administrativeControl: AdministrativeControl [1]
  + administrativeState: AdministrativeState [1]
  + lifecycleState: LifecycleState [1]
```

Color code:
* Red means remove
* Blue means keep
* Black means more thought required
* Purple means refactor
* Brown means aim to augment IEEE YANG list
P&R Model Sketch – 1: G.8052 MEP Sink Pruning & Refactoring

IEEE 802

The operations Establish/Enable/Disable/Terminate/Abort OnDemand/ProActive Single/Dual Ended MeasurementJob ControlSink/Source will be normalized to the creation/deletion/setting of instance of MeasurementJobControl Sink/Source (and the creation/deletion of Current/History Data in case of ProActive measurement job). The operations getCurrent/HistoryDataValues operations will be normalized into retrieving the attributes of the Current/HistoryData objects.

Question: Does IEEE YANG have a place where we can augment the defects (i.e., the ones in Table 2 of WD14-29, 12/2017, London, such as cDEO, cLOC, cUNL, cMMG, cUNM, cJNP, cUNPR, cRD0)
P&R Model Sketch – 1: G.8052 MEP Source Pruning & Refactoring

The operations testInitStart and testInitTerminate (at the Source function of the MEP) will be refactored into an attribute *testInit* with complex data type with fields:
- active (Boolean, true means start, false means terminate)
- destination macAddress
- dropAbility: Boolean
- priority: Integer
- dataTVLength: Integer
- period: OamPeriod
- testPattern: Enum
- numberOfTestFramesSent: Integer. (This field is reset to zero when active, and will be replaced with the number of test frames that has been sent when de-activated. There should be an autonomous notification to report the value of the numberOfTestFramesSent.)

The operations loopbackTest, loopbackTestTerminate, loopbackTestSeries, and loopbackDiscover will be normalized into 3 attributes, each of which will have complex data type to represent the input and output parameters of the original operations.

The IEEE Yang module’s Entity is the linkTrace. It has been re-engineered into Bernd’s UML.
P&R Model Sketch – 1: G.8052 TP Pruning & Refactoring

- OpenModelClass
  - ETH_TrailTerminationPoint

  - OpenModelAttribute
    - localId: String
    - name: NameAndValue
    - label: NameAndValue
    - extension: NameAndValue
    - Preliminary: OperationalState
    - Experimental: AdministrativeControl
    - Lifecycle: LifecycleState
    - localIdList: NameAndValue
    - uuid: String
    - layerProtocolName: LayerProtocolName
    - IPvSpec: IpSpec
    - configuredClientCapacity: Undefined
    - direction: TerminationDirection
    - terminationState: Undefined

- OpenModelAttribute
  - localId: NameAndValue
  - experimental: Address
  - crossConnectionObjectPointer: Undefined
  - currentProblemList: Undefined
  - alarmSeverityAssignmentProfilePointer: Undefined
  - containingPhysicalSubsystem: PhysicalSubsystem

- OpenModelOperation
  - createMep()
  - modifyMep()
  - deleteMep()
  - getAllContainedMeps()

---

Need to check if the auxiliaryFunctionPositionSequence should be kept or not.

Pruned out the alarm related attributes of TP (i.e., alarmStatus, currentProblemList, and alarmSeverityAssignmentProfilePointer). GAl related alarm is covered by the fault causes at the MEP, such as cUPnP, cMMG, etc.

Normalize the create/modify/delete/getAllContainedMep operation into management of MEP instance.
P&R Model Sketch – 1: G.8052 TP Sink Pruning & Refactoring

**ETH_TrafTerminationPointSink**

- **<OpenModelClass>**
  - **ETH_TrafTerminationPointSink**
  - **<OpenModelAttribute>**
    - multiplexing: EthServerTpSink_Pac [0..1]
    - _ethTpSink: EthTpSink [0..1]
    - _ethCtpSink: ETH_ConnectionTerminationPointSink [*]
    - _mapSink: MepSink [*]
  - **<OpenModelAttribute>**
    - _ieeComponent: IeeeComponent [0..1]
  - **<OpenModelAttribute>**
    - label: NameAndValue [*]
  - **<OpenModelAttribute>**
    - extension: NameAndValue [*]
  - **<OpenModelAttribute>**
    - preliminary: operationalState: OperationalState [0..1]
  - **<OpenModelAttribute>**
    - experimental: administrativeControl: AdministrativeControl [1]
  - **<OpenModelAttribute>**
    - lifecycleState: LifecycleState [1]
  - **<OpenModelAttribute>**
    - localId: NameAndValue [*]
  - **<OpenModelAttribute>**
    - usiid: String [1]
  - **<OpenModelAttribute>**
    - layerProtocolName: LayerProtocolName [1]
  - **<OpenModelAttribute>**
    - _ipSpec: Ipv4Spec [*]
  - **<OpenModelAttribute>**
    - configuredClientCapacity: <Undefined> [0..1]
  - **<OpenModelAttribute>**
    - terminationState: TerminationState [1]
  - **<OpenModelAttribute>**
    - alarmStatus: Address [Address] [*]
  - **<OpenModelAttribute>**
    - crossConnectionObjectPointer: <Undefined> [1]
  - **<OpenModelAttribute>**
    - currentProblemList: <Undefined> [1]
  - **<OpenModelAttribute>**
    - containingPhysicalSubSystem: PhysicalSubSystem [1]

**<OpenModelOperation>**

- createTrafficConditioning()
- deleteTrafficConditioning()
- modifyTrafficConditioning()
P&R Model Sketch – 1: G.8052 TP Bidirectional Pruning & Refactoring
P&R Model Sketch – 1: G.8052 TP Bidirectional Pruning & Refactoring
Links to ITU-T Recommendations

- G.8152 “Protocol-neutral management information model for the MPLS-TP network element” (Draft in progress)
Functional Architecture of Transport Networks

- G.800 / G.805 functional model
  - Adaptation, Termination, Link, Subnetwork, Layer network, Recursion, Partitioning
G.806 atomic functions and MCC input/output
Transport Functional Model (FM) to Information Model (IM) modeling

Layer examples
LR x = OTU
LR y = ODU (HO)
LR z = ODU (LO)

ITU-T FM

- Expanded G.805 Representation
- Rationalized Representation (G.805 / G.800 terms)

ITU-T IM

- G.874.1, G.8052, G.8152
- TTP / CTP
Derivation of LTP & LP from TTP & CTP

**ITU-T**
- Expanded G.805 Representation
- Rationalized Representation (G.805 / G.800 terms)
- Semi-flexible SNC / FR

**TMF**
- MTNM/MTOSI/SID
- Layered parameter list used to capture per-layer detail

**TMF**
- TR-225/215
- Per-layer detail captured in LT entities

**ONF**
- LP
- LTP

LT = LayerTermination
TPE = Termination Point Encapsulation
THANK YOU