

Netmod Working Group
Internet-Draft
Intended status: Informational
Expires: March 8, 2016

M. Betts, Ed.
ZTE
N. Davis, Ed.
Ciena
K. Lam, Ed.
E. Varma, Ed.
Alcatel-Lucent
B. Zeuner, Ed.
Deutsche Telekom
S. Mansfield, Ed.
Ericsson
P. Doolan, Ed.
Coriant
September 5, 2015

Framework for Deriving Interface Data Schema from UML Information Models
draft-betts-netmod-framework-data-schema-uml-02

Abstract

This draft describes a framework for how purpose and protocol specific interfaces can be systematically derived from an underlying common information model, focusing upon the networking and forwarding domain. The benefit of using such an approach in interface specification development is to promote convergence, interoperability, and efficiency.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on March 8, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Requirements Language	3
2.	Basic Concepts	3
3.	Information Modeling	4
3.1.	Unified Modeling Language	5
3.2.	Standard UML Information Model	5
4.	From UML IM to Data Schema Definition	6
4.1.	Methodology Overview	6
4.2.	Common Information Model	8
4.2.1.	Core Model fragment	8
4.2.2.	Forwarding plane technology specific or application specific model Fragments	9
4.3.	Common Information Model View for a Specific Purpose	9
4.4.	Data Schema	10
4.5.	Interface encoding	11
5.	Translation from UML	11
6.	Summary	12
7.	Acknowledgements	12
8.	Contributors	12
9.	IANA Considerations	12
10.	Security Considerations	12
11.	References	12
11.1.	Normative References	13
11.2.	Informative References	13
Appendix A.	Additional Stuff	15
Authors' Addresses	15

1. Introduction

Interface specifications are often generated as point solutions where the designer codes a particular interface from domain (problem space) concepts that may not be explicitly captured, may be defined using localized terminology that is subject to ambiguity in interpretation, and is highly focused on a particular use-case/application. The designer typically provides a representation of the interface schema in the form of a data schema [RFC3444](i.e., data structures conveyed over the interface), which only exposes the view of the domain relevant at that specific interface. As this data schema is a simple statement of the particular interface, it solely describes relationships relevant to the specific realization, having no inherent relationship to other interfaces in the system.

Approaching the development of interface specifications on a per use-case/application basis tends to promote unnecessary variety through a proliferation of similar interfaces, resulting in unnecessary divergences that limit interoperability. It also risks confusion of representational artifacts with fundamental characteristics of the information to be conveyed across the interface. There is also a risk that conflicting representations of the same information may be generated. Finally, as each such interface appears to stand alone, it thereby fails to capture relationships with other aspects of the same (or different) domains that are not explicitly needed for the interface.

This draft describes a framework for how a protocol specific data schema and the encoding used for the interface can be systematically derived from an underlying common information model, focusing upon the networking and forwarding domain. The benefit of using such an approach in the development of interface specifications is to promote convergence, interoperability, and efficiency.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Basic Concepts

An information model condenses domain knowledge and insights to provide a representation of its essential concepts, structures, and inter-relationships. In capturing domain understanding, such a model offers a coherent and consistent terminology and structure, expresses the semantics of the domain, and interrelates all relevant aspects of the domain. It enables a consistent expression of information that

improves interoperability between software components at interfaces derived from it. A "good" information model should capture domain best practices, and be designed to support domain variety as well as extensibility and evolution. Examples of domains include networking and forwarding, storage, etc. A common industry information model is the assembly of all domain information models, which inter-relate at "touch points". Note that a common industry information model should not be interpreted as being a monolithic entity; in particular, a modular structure is essential to allow for extensibility.

There may be several relevant views of any particular domain, depending upon the perspective of the viewer, all of which are interrelated and involve subsets of the information model, and none of which contradict each other. (It should be noted that one view provides the information model representation of the overall domain.) To form a particular (purpose-specific) view, some elements of the model may be pruned. Additionally, for efficiency, some systematic refactoring of the information model may also occur.

In this draft, the term data schema is used in the context of either: (i) a specific protocol that is used to implement a purpose specific interface, or (ii) a programming language that is used to invoke a purpose specific API. Note that it is possible to map directly from the purpose specific information model to interface encoding.

While a purpose specific interface/API is not a simple direct encoding of the information model of the overall domain, it is by its nature based on a relevant view of the information model of the domain (i.e., a purpose specific information model view). It must be completely and consistently traceable to this view and should use the associated domain terminology. Depending on its application, a particular view may lead to a number of encoded forms at various types of interfaces/APIs. The information model does not dictate the encoded form, which will depend upon such factors as necessary capability, interaction style, and programming language.

3. Information Modeling

This section introduces the Unified Modeling Language (UML), which has been used to model application structure, behavior, and architecture (in addition to business process and data structure). It also provides references to existing and ongoing work on standard information models based on UML.

3.1. Unified Modeling Language

The information model is expressed in terms of the Unified Modeling Language (UML) [OMG_UML], which was developed by the Object Management Group. It is a general-purpose modeling language in the field of software engineering. In 2000 the Unified Modeling Language was also accepted by the International Organization for Standardization (ISO) as an approved ISO standard [ISO_IEC_UML]. UML may be used in four ways:

- o To define a set of objects (instantiated classes that, if organized, describe a data model)
- o As an information model
- o As a metamodel (used to create an information model)
- o As a meta-metamodel

UML defines a number of basic model elements (UML artifacts), such as object classes, attributes, associations, interfaces, operations, operation parameters, data types, etc. In order to assure a consistent and harmonized modelling approach, and to ensure uniformity in the application of UML to a problem domain, a subset of the basic model artifacts should be selected according to guidelines for creating an information model expressed in UML [ONF_TR-514]. The guidelines are generic; i.e., they are not specific to any particular domain that the information model is addressing, nor are they restricted to any particular protocol interface data schema. A UML information model may be created using Open Source UML tools; guidelines to be taken into account during the creation of a UML information model for the Open Source tool Papyrus have been developed in [ONF_TR-515].

3.2. Standard UML Information Model

Information models expressed in UML, primarily focused upon the networking and forwarding domain, have been, and are in the process of being, developed in ITU-T, TM Forum, NGMN, 3GPP, MEF, ONF, and others.

ITU-T Recommendations are focused on understanding the telecommunications problem space and developing information models addressing network and network element considerations. Some examples of available standard ITU-T information models relevant to the networking and forwarding domain include:

- o ITU-T G.874.1 (2012), Optical transport network: Protocol-neutral management information model for the network element view [ITU-T_G.874.1]
- o ITU-T G.8052/Y.1346 (2013), Protocol-neutral management information model for the Ethernet Transport capable network element [ITU-T_G.8052]
- o ITU-T G.8152/Y.1375, Protocol-neutral management information model for the MPLS-TP network element [ITU-T_G.8152]
- o ITU-T G.7711, Generic protocol-neutral management Information Model for transport resources [ITU-T_G.7711]

The above information models are developed from ITU-T Recommendations that define the respective transport technology functional models and management requirements.

The TM Forum community has likewise developed extensive models of the same space from the network level management perspective [TMF_MTNM] [TMF_MTOSI] [TMF_TR225]. The basis for all functions made available to the network level management is defined in the protocol-neutral network element level management work done in ITU-T. Its models thus complement the ITU-T information models. In further collaboration with 3GPP, considerable joint effort has been devoted to develop a consistent and coherent approach to that space.

The NGMN has published a document called Next Generation Converged Operations Requirements (NGCOR) [NGMN_NGCOR], with the expressed purpose of taking these requirements into account when converged management interfaces for mobile and fixed networks are being standardized in the SDOs. An ongoing collaboration called the Multi-SDO Project on Converged Management is taking care that the requirements are considered during the specification of new interfaces. It includes participants from ETSI, NGMN, TMF, 3GPP, and other SDOs, equipment vendors, OS vendors and service providers.

4. From UML IM to Data Schema Definition

This section outlines the overall structure of a modular and evolvable common information model and how purpose specific IM views and data schema may be derived from it [ONF_TR-513].

4.1. Methodology Overview

As illustrated in Figure 1 below, the common information model is comprised of a library of model artifacts (objects, attributes, and associations) organized into a number of information model fragments

technology of the network. This module currently encompasses Topology, Termination, and Forwarding subsets of the core network module.

- o The core foundation module defines the artifacts for referencing entities, so that communications about an entity can take place.

4.2.2. Forwarding plane technology specific or application specific model Fragments

These fragments contain the artifacts (objects, attributes and associations) that relate solely the specific technology or application. In some cases an application or forwarding technology addition will also require enhancement of the core model fragment.

4.3. Common Information Model View for a Specific Purpose

The next step is the development of a purpose specific information model view, which is a true subset of the common information model. To provide maximal reuse, the purpose specific view is developed in two steps: (1) pruning and refactoring to provide a purpose specific information model of the network to be managed, where only those artefacts that represent the capabilities that are both in scope and supported are included, and (2) defining the access rights for the various groups of users that will manage that network. Pruning and refactoring provides a purpose specific information model that represents the capabilities of the network of interest. The definition of access rights provides the ability to limit the actions that can be taken by the various user groups that will use that information model.

- o Pruning to remove the objects/packages/attributes that are not required.
 - Selecting the required object classes from the common IM (all mandatory attributes and packages must be included)
 - Selecting the required conditional packages and optional attributes (note that, where appropriate, conditional packages and optional attributes may be declared mandatory in the purpose specific IM)
 - Removing any optional associations that are not required
- o Refactoring to reduce association flexibility, such as:
 - Reducing multiplicity (e.g., from [1..*] to [1]). When this results in a composition association of multiplicity [1] between

a subordinate and superior object class, they can be combined into a single object class by moving the attributes of the superior class into the subordinate class.

- Where possible, reducing the depth of the inheritance (i.e., combining object classes by moving the attributes of the super class into the subclass).
- Adding reverse navigation, if useful for the client. The common IM only supports navigation from a subordinate object class to a superior object class. This allows new subordinate object classes to be added without any impact on the superior object class. In a network specific implementation it is frequently useful to be able to navigate the relationship between superior and subordinate object classes in both directions.
- Constraining attribute definitions. This can be done by reducing legal value ranges, defining which (if any) attributes should be read only (for all users), and/or defining constraints between attributes.

- o Definition of access rights

If only one group will use the network specific IM then this step is not required. If more than one group will use the network specific IM this optional step provides a profile for each user group to:

- Convert some attributes defined as read/write in the network specific IM to read only
- Remove the right to create/delete some or all object instances

4.4. Data Schema

A data schema (DS) is constructed by mapping of the purpose specific information model view into the DS together with the operations patterns from the common information model to provide the interface protocol specific operations and notifications. The operations should include data structures taken directly from the purpose specific information model view with no further adjustment. (Note that it is possible to map directly from the purpose specific information model to interface encoding).

The development of the data schema should consider the following:

- o The operations should act on the information in a way consistent with the modeled object lifecycle interdependency rules.

- Instance lifecycle dependencies to ensure sensible interface operation structuring and interface flow rules
- Usage of transaction approach style of interface to account for lifecycle dependencies of the model
- o The operations should abide the attribute properties. Read only attributes (except those which are defined as setByCreate) should not be included in data related to creation of an object (e.g., not in createData) or in a specification of a desired object structure outcome.
- o Usage of attribute value ranges, etc. to allow "effort" statement, optionality and negotiation to be supported by the interface.

4.5. Interface encoding

This step encodes the purpose specific data schema or purpose specific information model into either a specific protocol that is used to implement a purpose specific interface or; a programming language that is used to invoke a purpose specific API. If the interface is encoded directly from the purpose specific information model then the interface operations must be added as described above.

5. Translation from UML

Applying the methodology outlined in Section 4, protocol-specific interface data schema/encodings may be derived from existing, and emerging, standard UML information models addressing the forwarding and networking domains (e.g., [ITU-T_G.7711], refactored G.874.1 [ITU-T_G.874.1]).

In order to assure a consistent and valid data modelling language representation that enables maximum interoperability, translation guidelines from UML information models to data schema/interface encodings are required. A set of translation rules also assists in development of automated tooling.

Guidelines are currently under development for translation of data modeled with UML to YANG including mapping of object classes, attributes, data types, associations, interfaces, operations and operation parameters, notifications, and lifecycle [I-D.mansfield-uml].

It should be noted that the concept of deriving protocol-specific modules from UML information models is not new (e.g., MEF 38 [MEF_38] and MEF 39 [MEF_39] provide YANG modules derived from UML information models G.8052 [ITU-T_G.8052] and MEF 7.1 [MEF_7.1] for Service OAM

Fault and Performance Monitoring, respectively.). What is new is the concept of an open, modular, evolvable common information model, coupled with an associated suite of essential guidelines (e.g., UML, Open Source tooling, translation, etc.), for realizing a coherent set of solution modules.

6. Summary

This draft describes a modular and scalable approach for systematically deriving purpose and protocol specific interfaces from an underlying common information model, focusing upon the networking and forwarding domain. Building upon an underlying common information modeling description of network resources (functionality, capabilities, flexibility) is a key enabler to convergence and interoperability. It is also future proof in the sense that the emergence of new protocols becomes solely a non-disruptive mapping issue. It should be noted that not all domains require development of information model prior to solutions development; the domains where this is of greatest benefit involve networking domains requiring support for an enhanced level of control and network programmability.

7. Acknowledgements

8. Contributors

Dave Hood
Ericsson
USA
email dave.hood@ericsson.com

9. IANA Considerations

This memo includes no request to IANA.

10. Security Considerations

This informational document only describes a framework for deriving interface data schema from UML Information Models. As such, security concerns are out of the scope of this document.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, DOI 10.17487/RFC3444, January 2003, <<http://www.rfc-editor.org/info/rfc3444>>.

11.2. Informative References

- [I-D.lam-topology]
Lam, K., Varma, E., Doolan, P., Davis, N., Zeuner, B., Betts, M., Busi, I., and S. Mansfield, "Usage of IM for network topology to support TE Topology YANG Module Development", 2015, <<https://datatracker.ietf.org/doc/draft-lam-teas-usage-info-model-net-topology/>>.
- [I-D.mansfield-uml]
Mansfield, S., Zeuner, B., Davis, N., Yun, Y., Tochio, Y., Lam, K., and E. Varma, "Guidelines for Translation of UML Information Model to YANG Data Model", 2015, <<http://datatracker.ietf.org/doc/draft-mansfield-netmod-uml-to-yang/>>.
- [ISO_IEC_UML]
ISO/IEC, "ISO/IEC 19505-1:2012 - Information technology - Object Management Group Unified Modeling Language (OMG UML) - Part 1: Infrastructure. Iso.org.2012-04-20. Retrieved 2014-04-10", 2012.
- [ITU-T_G.7711]
ITU-T, "Generic protocol-neutral management Information Model for transport network resources", 2015.
- [ITU-T_G.8052]
ITU-T, "ITU-T G.8052/Y.1346 (2013), Protocol-neutral management information model for the Ethernet Transport capable network element", 2013, <<http://www.itu.int/rec/T-REC-G.8052/en>>.
- [ITU-T_G.8152]
ITU-T, "ITU-T G.8152/Y.1375 (draft in progress), Protocol-neutral management information model for the MPLS-TP network element", 201x.

- [ITU-T_G.874.1] ITU-T, "ITU-T G.874.1 (2012), Optical transport network: Protocol-neutral management information model for the network element view", 2012, <<http://www.itu.int/rec/T-REC-G.874.1/en>>.
- [MEF_38] MEF, "Service OAM Fault Management YANG Modules", 2012, <http://www.metroethernetforum.org/Assets/Technical_Specifications/PDF/MEF_38.pdf>.
- [MEF_39] MEF, "Service OAM Performance Monitoring YANG Module", 2012, <http://www.metroethernetforum.org/Assets/Technical_Specifications/PDF/MEF_39.pdf>.
- [MEF_7.1] MEF, "Carrier Ethernet Management Information Model [superseded by MEF 7.2, which supports additional sets of service attributes defined in recent MEF specifications]", 2009, <http://www.metroethernetforum.org/Assets/Technical_Specifications/PDF/MEF7.1.pdf>.
- [NGMN_NGCOR] NGMN Alliance, "Next Generation Converged Operations Requirements (NGCOR)", 2013, <http://www.ngmn.org/uploads/media/NGMN_Next_Generation_Converged_Operations_Requirements_-_Final_Deliverable.pdf>.
- [OMG_UML] OMG, "OMG Unified Modelling Language (UML), Infrastructure, Version 2.4.1", 2011.
- [ONF_TR-513] ONF, "TR-513 Common Information Model Overview 1.0", 2015, <https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/Common_Information_Model_V1.0.pdf>.
- [ONF_TR-514] ONF, "TR-514 UML Modeling Guidelines 1.0", 2015, <https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/UML_Modeling_Guidelines_V1.0.pdf>.
- [ONF_TR-515] ONF, "TR-515 Papyrus Guidelines 1.0", 2015, <https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/Papyrus_Guidelines_V1.0.pdf>.

[TMF_MTNM]

TM Forum, "TM Forum Multi Technology Network Management, Release 3.5", 2009,
<<http://www.tmforum.org/MTNM/1689/www.tmforum.org/DownloadCenter/7549/home.html#mtnm>>.

[TMF_MTOSI]

TM Forum, "TM Forum Multi Technology OS Interface, Release 3.0", 2012,
<<http://www.tmforum.org/MTNM/1689/www.tmforum.org/DownloadCenter/7549/home.html#mtosi>>.

[TMF_TR225]

TM Forum, "TM Forum TR225, Logical Resource: Network Function Model", 2014.

Appendix A. Additional Stuff

TBD

Authors' Addresses

Malcolm Betts (editor)
ZTE
Canada

Phone: +1 678 534 2542
Email: malcolm.betts@zte.com.cn

Nigel Davis (editor)
Ciena
UK

Email: ndavis@ciena.com

Kam Lam (editor)
Alcatel-Lucent
USA

Phone: +1 732 331 3476
Email: kam.lam@alcatel-lucent.com

Eve Varma (editor)
Alcatel-Lucent
USA

Email: eve.varma@alcatel-lucent.com

Bernd Zeuner (editor)
Deutsche Telekom
Germany

Phone: +49 6151 58 12086
Email: b.zeuner@telekom.de

Scott Mansfield (editor)
Ericsson
USA

Phone: +1 724 931 9316
Email: scott.mansfield@ericsson.com

Paul Doolan (editor)
Coriant
Germany

Phone: +1 972 357 5822
Email: paul.doolan@coriant.com