



OPEN NETWORKING  
FOUNDATION

# Common Information Model Overview

Version 1.0  
March 12, 2015

ONF TR-513



ONF Technical Recommendation:  
Common Information Model Overview

## **Disclaimer**

THIS SPECIFICATION IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

Any marks and brands contained herein are the property of their respective owners.

Open Networking Foundation  
2275 E. Bayshore Road, Suite 103, Palo Alto, CA 94303  
[www.opennetworking.org](http://www.opennetworking.org)

©2015 Open Networking Foundation. All rights reserved.

Open Networking Foundation, the ONF symbol, and OpenFlow are registered trademarks of the Open Networking Foundation, in the United States and/or in other countries. All other brands, products, or service names are or may be trademarks or service marks of, and are used to identify, products or services of their respective owners.

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
<b>2</b>	<b>References .....</b>	<b>4</b>
<b>3</b>	<b>Abbreviations .....</b>	<b>4</b>
<b>4</b>	<b>Common Information Model.....</b>	<b>5</b>
	4.1 Core model fragment .....	6
	4.2 Forwarding Technology or application specific model fragments.....	6
<b>5</b>	<b>Common Information Model view for a specific purpose .....</b>	<b>7</b>
	5.1 Pruning and refactoring.....	7
	5.2 Definition of access rights.....	8
<b>6</b>	<b>Data Schema .....</b>	<b>8</b>
<b>7</b>	<b>Interface encoding .....</b>	<b>9</b>
<b>8</b>	<b>Common information model team .....</b>	<b>9</b>

## List of Figures

Figure 1.1: Overview of the ONF common information model (CIM).....	5
---	---

# 1 Introduction

This document describes the overall structure of the common information model (CIM), provides guidelines on the process used to develop the CIM and derive interfaces. Other, more detailed guidelines, as described below, are also provided. These guidelines are intended to capture the best practices and will be updated to reflect the experience that is gained as the CIM evolves.

The purpose of the CIM is to provide a common repository for any information models that are developed in ONF. Figure 1 below provides an overview of the ONF CIM and shows how the purpose and protocol specific interfaces may be derived from the CIM.

The first step in the process is to prune and refactor the CIM into a purpose specific IM view and then map this to a protocol specific data schema<sup>1</sup> and the encoding used for the interface. The process of mapping to a protocol specific data schema includes the mapping of the common operation pattern to the operations supported by that protocol. The term Data Schema (DS) in this document is used in the context of 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. The ONF Common Information Model is stored in an ONF-specific area of GitHub. Guidelines for the use of UML, the Papyrus tool and GitHub in the Common IM are provided in [1] and [2]. Guidelines for pruning and refactoring the CIM to provide a purpose specific view, and ultimately mapping to a data schema and mapping to a protocol or programming language, are being developed.

## 2 References

[1] onf2015.018: UML Modeling Guidelines

[2] onf2015.017: Papyrus Guidelines

## 3 Abbreviations

API Application Programming Interface

CIM ONF Common Information Model

DS Data Schema

IM Information Model

IMP ONF Information Modeling Project

UML Unified Modeling Language

---

<sup>1</sup> The term data schema is used instead of data model since the term data model is also used in a wider context and is sometimes used to refer to an information model

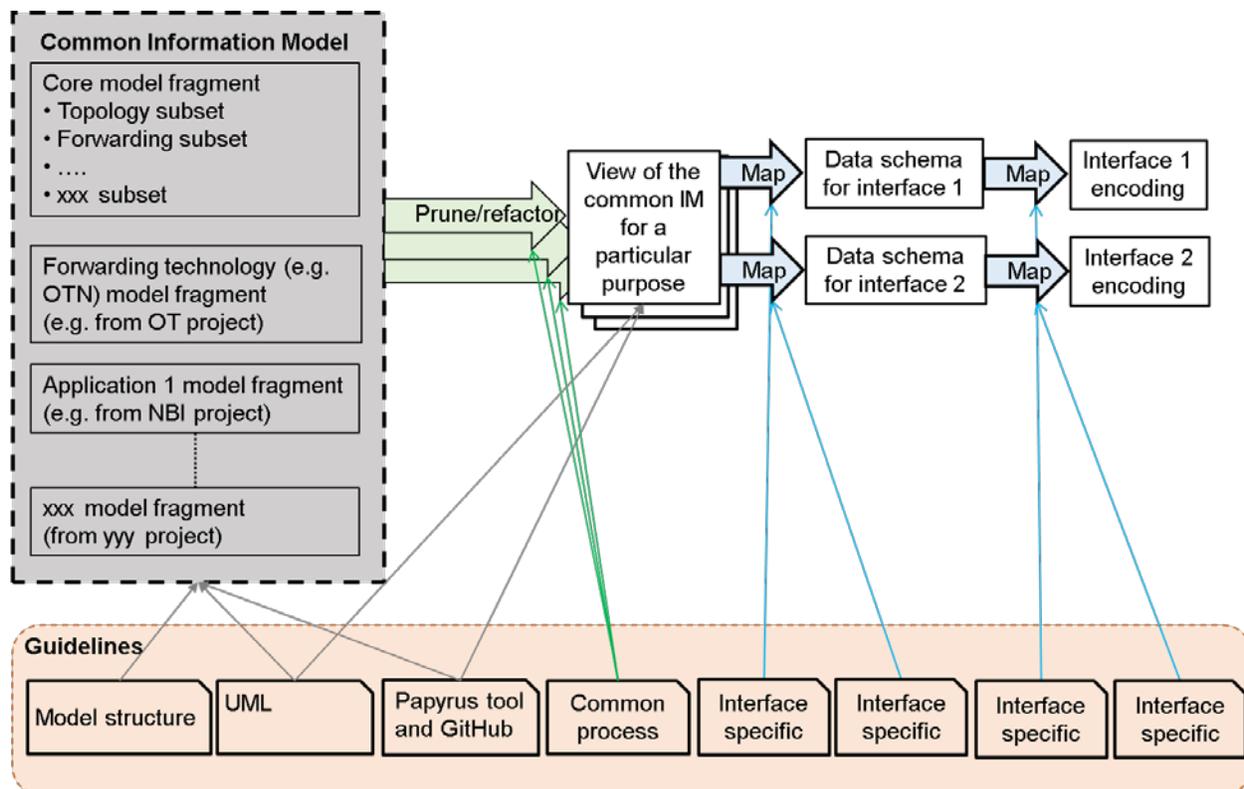


Figure 1.1: Overview of the ONF common information model (CIM)

## 4 Common Information Model

An information model describes the things in a domain in terms of objects, their properties (represented as attributes), and their relationships. The CIM should include all of the artifactartifacts (objects, attributes, relationships, etc.) that are necessary to describe the domain for the applications being developed.

It will be necessary to continually expand and refine the CIM over time as, for example to add, new applications, capabilities or forwarding technologies, or to refine the CIM as new insights are gained.

To allow these extensions to be made in a seamless manner, the CIM will be structured into a number of model fragments<sup>2</sup>. This modeling process allows the fragments that contain these extensions to be developed, by the domain experts, with as much independence as possible.

As a result of advancements in the industry it may be recognized that some parts of the CIM may need to be augmented or changed. Each IM team will ensure that any such areas are clearly identified using model lifecycle stereotypes (i.e. annotations). The older model forms will be maintained to ensure ongoing compatibility and to ease migration. The use of these stereotypes (e.g. experimental; preliminary; obsolete) is described in the UML modeling guidelines [1].

<sup>2</sup> The Papyrus UML modeling tool supports the use of fragments (sub-models)

When a version of the CIM is released all of the artifacts will be annotated with the model lifecycle stereotypes. All artifacts, including those marked for example as experimental or obsolete, will be in the CIM release.

#### 4.1 Core model fragment

It is expected that the artifacts in the core model fragment will be used by multiple project teams. The core model fragment is divided into a number of subsets each addressing a specific topic to allow for easier navigation. The Information Modeling Project (IMP) will be responsible for maintaining the core model fragment.

#### 4.2 Forwarding Technology or application specific model fragments

It is expected that the forwarding technology or application specific projects will contribute to the CIM by developing fragments which contain the artifacts (objects, attributes and associations) that relate to their area of expertise. In some cases it may be useful to document a pruned and refactored view of other artifacts in the CIM. For example the topology subset of the CIM could be pruned and refactored to contain only the information required to show reachability.

It is expected that new fragments will initially contain artifacts that have the experimental stereotype. As the model is validated, for example by developing interfaces, the artifacts can be adjusted and moved through the lifecycle states. This allows for rapid feedback between the model and an implementation. These technology or application specific fragments will be placed in the common repository where they will be available for reuse by other IM teams. The stereotypes allow all users of the CIM to understand the status of particular artifacts and make informed decisions on using them in implementations.

In some cases an application or forwarding technology addition will also require enhancement of the core model fragment. The CIM team will work with the forwarding technology/application team to ensure that these extensions are backwards compatible, can be reused by other groups and that work is not inadvertently duplicated in multiple projects. Enhancements of this sort may need to be carried out by a CIM team (of the IMP), supported by the IM teams developing the forwarding technology or application specific model fragments<sup>3</sup>.

To assure coherency, any object, attributes or associations that might be identified during the development of forwarding technology or application specific views should be included in the appropriate fragment of the CIM. Only those properties that relate to the specific encoding or style of interaction of an interface may be added outside the CIM.

---

<sup>3</sup> The core modeling team will often need to identify such cases and ensure that they work with the forwarding technology or application specific teams, as appropriate.

## 5 Common Information Model view for a specific purpose

A purpose specific information model is a true subset of the CIM and should be expressed in UML. These views should be developed by the relevant application, forwarding technology or interface specific project teams.

A purpose specific information model will typically be much smaller than the entire CIM. If additional artifacts (objects, packages, attributes or associations) are identified while establishing a specific view, these should be added to the appropriate fragment of the CIM so that they are available for reuse.

To provide maximum reuse a purpose specific view should be developed in two steps:

- a) Prune and refactor the artifacts of the CIM to provide a model of the network to be managed. Only those artifacts that represent the network capabilities that are both necessary to satisfy the purpose and are supported by the network are included in the purpose specific IM.
- b) Define the access rights for the various groups of users that will manage that network

Pruning and refactoring provides a purpose specific IM 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 IM. For example a user group responsible for network configuration could be provided full read/write access and the ability to create or delete object instances; while a user group responsible for inventory may only be allowed read access (i.e. can see the network but cannot make changes).

### 5.1 Pruning and refactoring

Pruning, i.e. remove the objects/packages/attributes that are not required:

- Select the required object classes from the common IM
  - All mandatory (non-optional) attributes and packages must be included
- Select the required conditional packages and optional attributes
  - Where appropriate, conditional packages and optional attributes in the common IM may be declared mandatory in the purpose specific IM
- Remove any optional associations that are not required

Refactoring, i.e. reduce association flexibility:

- Reducing multiplicity (for example 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)
- Add 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 purpose 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
  - Reducing legal value ranges
  - Defining which (if any) attributes should be read only (for all users)
  - Defining constraints between attributes

## 5.2 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

## 6 Data Schema

A Data Schema (DS) is developed in the context of 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. Note that it is possible to map directly from the purpose specific information model to interface encoding. Data schema should be developed by the relevant application, forwarding technology or interface specific project team. The DS is constructed by mapping the purpose specific information model into the DS together with the operations patterns from the CIM 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.

The development of the DS should consider the following:

- The operations should act on the information in a way consistent with the modeled object lifecycle interdependency rules as defined in the common IM.
  - Instance lifecycle dependencies to ensure sensible interface operation structuring and interface flow rules
  - Use transaction approach style of interface to account for instance lifecycle dependencies of the model
- The operations should abide by 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.
- Use of attribute value ranges, etc. to allow “effort” statement, optionality and negotiation to be supported by the interface

## 7 Interface encoding

This step encodes either, a purpose specific data schema or, a purpose specific information model into a 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. This encoding should be developed by the relevant application, forwarding technology or interface specific project team.

## 8 Common information model team

The ONF wide CIM will be maintained by a sub-team of the Information Modeling Project (IMP) team. It is expected that each of the other ONF project teams that will use (and contribute to) the CIM will provide a representative to ensure that the work of the IM sub team meets the needs of their project.

The IMP IM sub-team will develop and maintain the core model and maintain the library of all of the model fragments in GitHub. They will provide guidance (when requested) to the other IM sub-teams that are generating the application/forwarding technology specific fragments of the common IM or developing a purpose specific view of the common IM.