Core Information Model (CoreModel)

TR-512.FE

Future Enhancements

Version 1.4
November 2018

ONF Document Type: Technical Recommendation
ONF Document Name: Core Information Model version 1.4

Disclaimer

Open Networking Foundation
1000 El Camino Real, Suite 100, Menlo Park, CA 94025
www.opennetworking.org

Important note

This Technical Recommendations has been approved by the Project TST, but has not been approved by the ONF board. This Technical Recommendation is an update to a previously released TR specification, but it has been approved under the ONF publishing guidelines for 'Informational' publications that allow Project technical steering teams (TSTs) to authorize publication of Informational documents. The designation of '-info' at the end of the document ID also reflects that the project team (not the ONF board) approved this TR.

Table of Contents

List of Figures

Document History

Version	Date	Description of Change
1.0	March 30, 2015	Initial version of the base document of the "Core Information Model" fragment of the ONF Common Information Model (ONF-CIM).

Version	Date	Description of Change
1.1	November 24, 2015	Version 1.1
1.2	September 20, 2016	Version 1.2 [Note Version 1.1 was a single document whereas 1.2 is broken into a number of separate parts]
1.3	September 2017	Document name changed. Was TR-512.10 in Version 1.2. [Published via wiki only]
1.3.1	January 2018	Addition of text related to approval status.
1.4	November 2018	Alignment with release content.

1 Introduction

This document is an addendum to the TR-512 ONF Core Information Model and forms part of the description of the ONF-CIM. For general overview material and references to the other parts refer to TR-512.1.

1.1 References

For a full list of references see TR-512.1.

1.2 Definitions

For a full list of definition see TR-512.1.

1.3 Conventions

See TR-512.1 for an explanation of:

- UML conventions
- Lifecycle Stereotypes
- Diagram symbol set

1.4 Viewing UML diagrams

Some of the UML diagrams are very dense. To view them either zoom (sometimes to 400%), open the associated image file (and zoom appropriately) or open the corresponding UML diagram via Papyrus (for each figure with a UML diagram the UML model diagram name is provided under the figure or within the figure).

1.5 Understanding the figures

Figures showing fragments of the model using standard UML symbols as well as figures illustrating application of the model are provided throughout this document. Many of the application-oriented figures also provide UML class diagrams for the corresponding model fragments (see TR-512.1 for diagram symbol sets). All UML diagrams depict a subset of the relationships between the classes, such as inheritance (i.e. specialization), association relationships (such as aggregation and composition), and conditional features or capabilities. Some UML diagrams also show further details of the individual classes, such as their attributes and the data types used by the attributes.

2 Introduction to the Further Enhancements

The focus of this document is areas of ongoing work that are not represented in other TR-512 documents.

A data dictionary that sets out the details of all classes, data types and attributes is also provided (TR-512.DD).

3 Summary of some further enhancements

3.1 Future work areas[1]

Potential future areas of work in the CoreModel include, not in any particular order (highlighting document numbers within the TR-512 series):

- LTP Port and LP Port model (.2)
- Spec model pattern generalization (.7 and A.10)
    - Generalize the spec models of LTP, FC, FD, etc. and scheme/system spec to have a general spec pattern
    - Refactor LTP Spec to be Component-System Spec
        - In v2.n, refactor the LTP spec to be a full generalized Component-System spec with constraints on what components can be used. Depends on LTP Port work.
- Interaction and management-control messaging (.8, .A.6 and .A.7)
    - Developing models for interaction/messaging in the context of ECC/DCN and protection
        - This leads to the need to develop support for basic IP networking
    - Developing message sequence diagram (for operation sequences)
- Interaction patterns for advancement of TAPI etc (.10 and .A.7)
    - Completion of the generalized operations pattern covering range of cases including intend and CRUD
        - Support operations patterns including intent
        - Support TOSCA and policy
        - Operations interaction model
        - Operations temporality
    - Understand the relationship with Dynamic APIs and Strategic Mediation
    - Operation pattern for general task
        - Use TAPI OAM as the seed to investigate using the Operation Pattern to support general task, taking the G.8052 On-demand measurement job requirement into account as input.
- Support for specific interface development in TAPI (various)
    - More layer examples (.A.4, .A.5 and .A.6)
- OAM functions (.9, .A.7 and new .A.n)
    - Generalization of OAM functions, e.g., generalized MEPs
    - Consider use cases and scenarios to guide development of the generalized model
- Assurance (.10 and .A.7)
    - Modeling of events and the reporting of events
- Patterns and architectures (.A.2)
    - Continue construction of models that explore the pattern underlying Link/FC/FD and minimally represent that pattern and show derivation of Link/FC/FD from that pattern.

[1] This section was in TR-512.1 in V1.2.

- Dependency graph representation of telecommunications technology including flow semantics (.A.2)
    - For expression of detailed processes of a telecommunications technology to enable interpretation of a new technology
- Profiles, Templates and Specifications (.7 and .A.10)
    - Completion of spec model and addition of profiles model in the spec context
    - Further development of constraint models (also covering policy)
    - Complete pattern and migrate model to use pattern
    - Develop class based rule mechanism and consider more fluid approach to Core model
    - Provide further examples of usage
    - Develop detailed rules
    - Refine model to deal with rule interaction
    - Consider FD/FC spec convergence
        - FdPort added in V1.3 will help here
    - When dealing with Compound Links we need to consider whether rules are necessary for Link (the same structure will apply but an additional association from Link to FD will be necessary)
    - Development of a specification toolkit including standardized rules and structures
    - DSGL (Domain Specific Graphical Language to ease spec construction)
    - Model v specification (see also notes in .7):
        - Implication of the work so far is that the specification structure is the model structure and that the schema for any particular case has some parts of the structure in compile time form and other parts in runtime only form where the runtime form may have static parts only in the spec form
        - Is a replication of the model structure in formal model but that formal model should be decoupled at various points and extensible in a constrained way at various points.
        - Considerations of "model viscosity" (all models are fluid over some timeframe)
    - Dealing with LTP and LP formal sub-structuring challenge
        - Related to the previous bullet… should the LP sub structuring of the spec model be part of the LP model
    - Migration of operations from non-spec to spec
        - Continuum of usage approaches from "phrase book user" to "orator"
    - FcSpec refinements
        - Should the FC spec be rationalized to recognise that MSUF is essential an FC (implications etc)?
    - Moving to the generalized spec
        - How should constraints on each class spec be expressed in the context of a generalized spec?
    - Understanding the relationship to Dynamic APIs, Strategic Mediation and the Operations Patterns
    - Enhance the extension mechanism
    - Development of the scheme specification
        - Generalization to deal with specification of any network structure

        - Enhance the application to protection
- Resilience (.5 and .A.11)
    - Further development of the scheme specifications for resilience schemes
    - Carry out further work on the unexpected flow query
    - Use of partition v route and use in combination (see .5 for some notes)
    - Build on the model of signalling to deal specifically with resilience schemes (see .5 for some detailed considerations). This will be related to work in
        - .8 as this will be part of the MCC work as the C&SC is a controller and general messaging and signalling will be covered in .8
        - .7 as there is a specification aspect
    - Refine documentation on relationship to protectionGroups (see notes in .5)
    - Complete route feeds port relationship
    - Complete route lifecycle documentation (see .5)
    - Indication of encapsulated resilience on FC and Link
    - Further examples including dual homed cases
- Control (.8, .11, .A.7 and .A.9)
    - Improvements to documentation on replacement of NE
- Timing and Synchronization model including frequency and time/phase (.2 and .A.8)
    - Complete work and examples
- Physical Equipment (.6 and potentially a new .A.n)
    - Enhancements to the equipment model
    - Enhancements to the expectation v actual model approach (see detailed notes in .6)
    - Progress attribute details to «Preliminary» or «Mature»
    - Rationalize attribute groupings
    - Look for formal sources for physical properties and work towards a P&R relationship to these sources
    - Strengthen linkage and improve decoupling wrt other model areas
        - For functional work wrt ProcessingConstruct (.11 and .A.9) and OAM functions (.9 and .A.7)
            - Also see detail notes in .6
    - Separate out Management-Control parts into Management-Control model (.8 and .A.7)
    - Refine and move specification model detail from Physical model document to Specification model document and move model as appropriate (.7 and .A.10).
    - Addressing of ports (see .6 for some details)
- Software model enhancements (.12 and .A.13)
    - Representation of software and relationship to physical (.6) and functional (.11) models
- ProcessingConstruct (.11 and .A.9)
    - Document relationship to:
        - TOSCA meta-model
        - Component
        - VNF
    - Degree of specialization (related to .A.2)
        - PC v ControlComponent v C&SC

        - PC v FC
        - CD v FD
    - Component-System recursion and projecting views
    - PC emergence from software (see also Software in this list)
    - CD/PC spec
- FC, FD, Link and Topology (.2, .4)
    - Various detailed enhancements including considerations of merging of FC and Link
    - Consider merging FD and Link
    - Improve derivation of FD/FC/Link from Component-System pattern (.A.2)
        - Consider FC/Link convergence (.4)
    - Further clarification of off-network "things" (could be a link topology)
    - Various illustration of FD/Link lifecycle
        - Consider generalized lifecycle state and state interaction
    - Cost algorithm
    - Illustrate use of model for inverse multiplexing cases (.A.4, .A.5 and .A.6)
    - Improve documentation on terminationState
    - Stating detailed properties on topology
    - Rationalize use of NearEnd/FarEnd, Input/Output, Ingress/Egress etc
    - Further develop Transitional Link examples (in .A.5 and .A.6) to cover complex Links (see .4)
    - Further develop non-orthogonal FDs (see .4) and provide examples (in .A.4, .A.5 and .A.6)
- View abstraction (.4)
    - Enhancements to view abstraction examples and cases, including FD view, FC view, Call view, Service view, Connection view
    - Further work on rules for virtualization (e.g. what from one view can be grouped in the same link from another view)
- Lifecycle (.3)
    - Entity lifecycles including splits and merges
- Mapping to other models (.TM)
    - Enhancements to the mapping to OpenFlow
    - Development of mappings to IETF models (e.g. TEAS)
    - Mapping to TOSCA
    - Aim for convergence of industry models
- Temporal properties (new .n and .A.n document)
    - Determining how to inject temporal aspects into the model process
- Stating rules and constraints (various including .2, .4)
    - Method for statement of inter-attribute and inter-association rules
    - See also subsections below
- Model restructuring (.1 and new .A.n)
    - Decoupling for improved modularity and independent working (new .A.n)
    - Model federation (new .A.n)
    - Model patterns (new .A.n)
- Model backward compatibility (new .W.2)
    - "W" series are white papers. .W.1 will be an introduction document

    - Develop from material originally targeted for a white paper
    - Semantic compatibility
- Event driven solution investigation
- Compute model
    - CPU storage
- Identity model investigation
    - Investigate the identify model (Global class & Local class). Instead of inheriting from Global or Local class, the tool will generate the necessary identifier attributes from the Global and Local classes and add to the entity classes at design/compile/run time.
- Minor enhancements
    - Rename the LayerProtocol class
- Documentation
    - Ongoing improvements
- Tooling (not in TR-512)
    - Enhancement to UML YANG to cover specification models
    - Enhancements to pruning and refactoring process and tooling

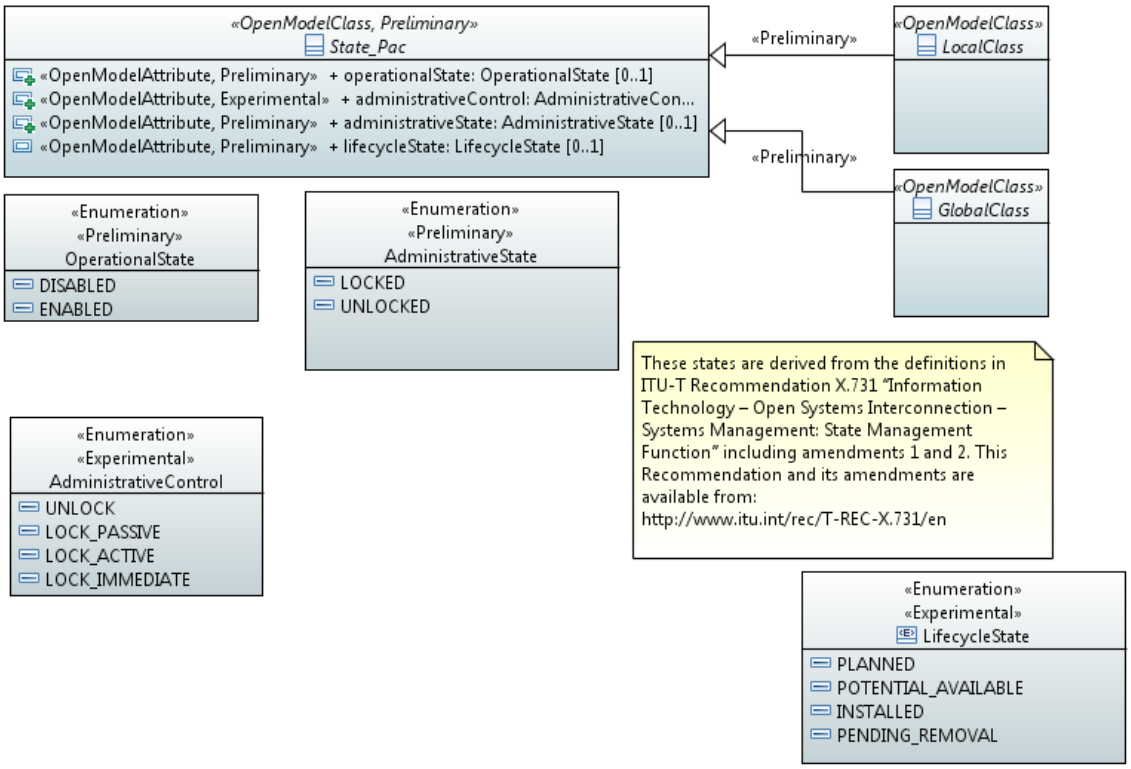
3.2 Some detailed notes on future work

3.2.1 Controller model

Work is underway to develop a model of management-control that replaces the SdnController, NetworkControlDomain and the NetworkElement with a unified model of control that also includes the ConfigurationAndSwitchController discussed in TR-512.5.

3.2.2 State extensions

There are a number of experimental items in the state model as shown below. These are included in the main data dictionary.

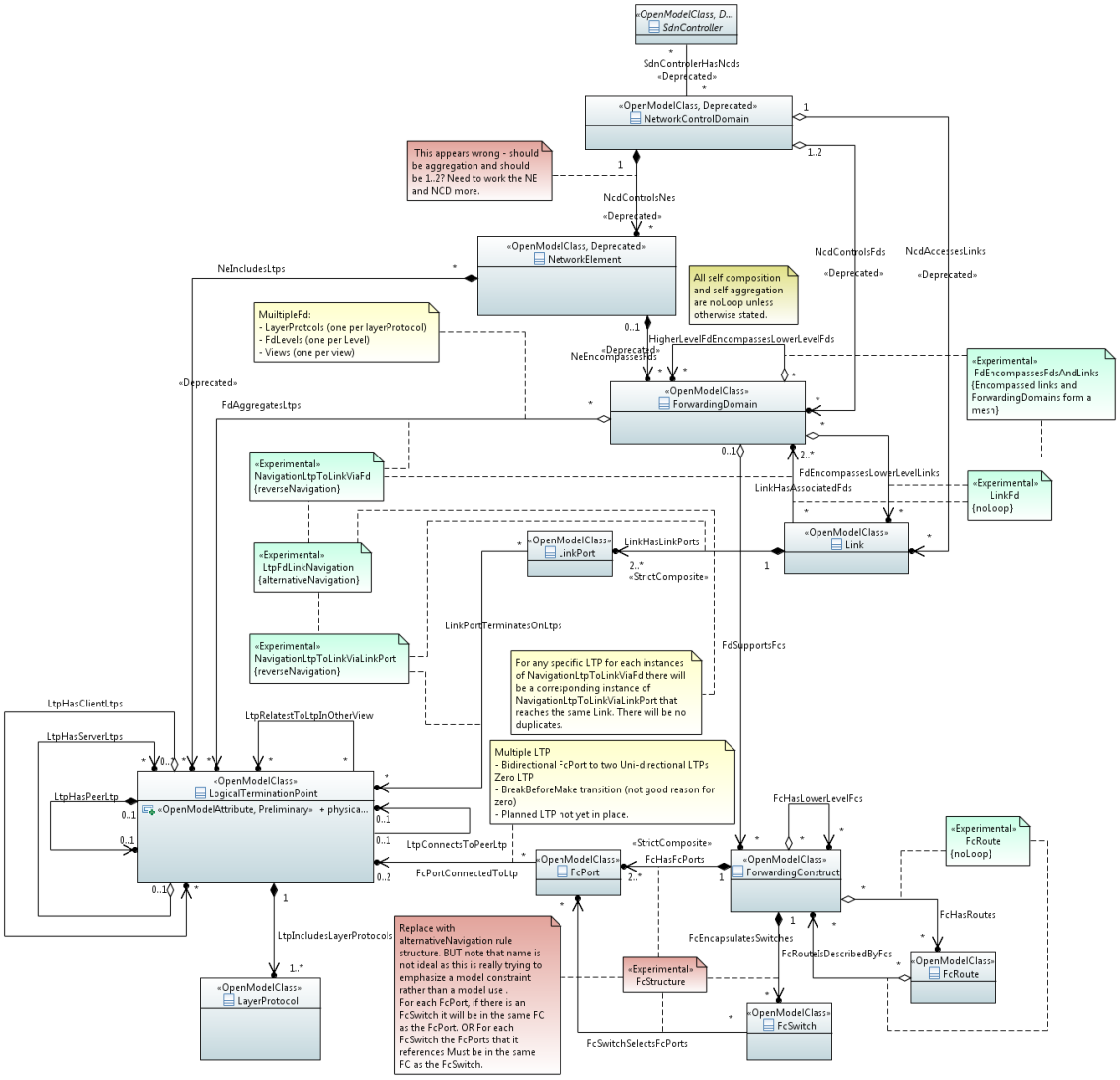


CoreModel diagram: EnhancedStates

Figure 3-1 State model with enhancements

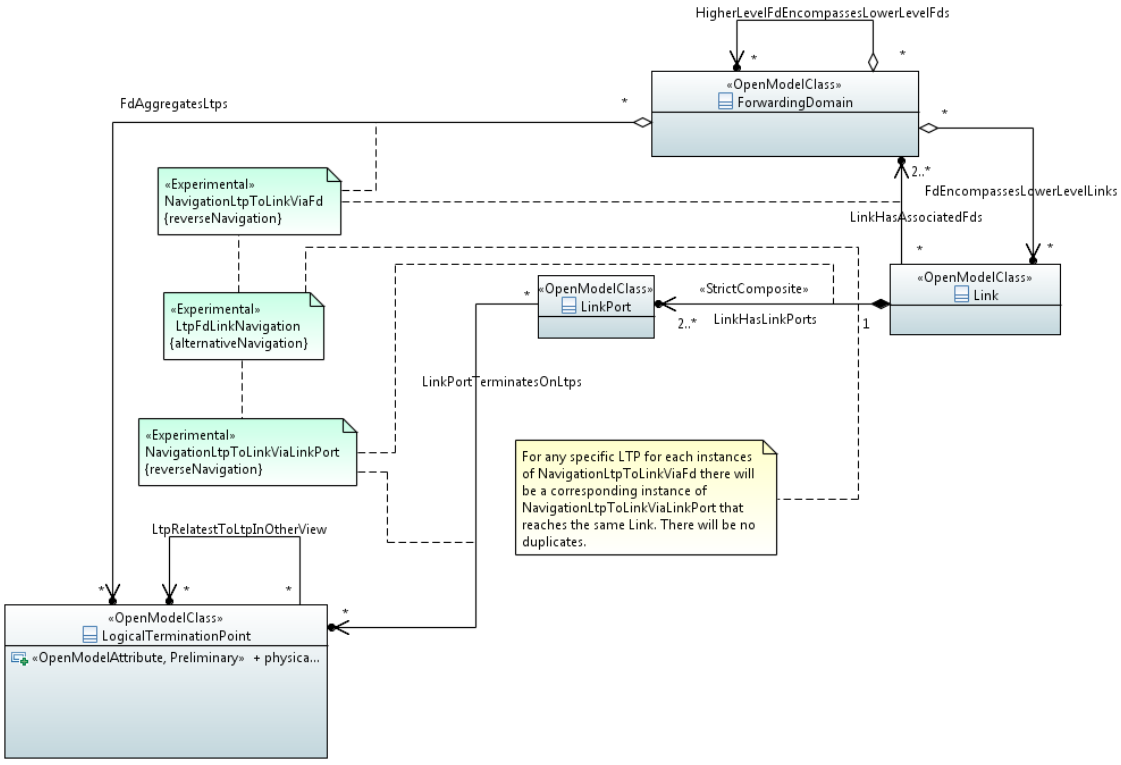
3.2.3 Model structure rules

Some of the associations in the model are interrelated by some pattern. The following figures explore ways of expressing the patterns and interrelationships.



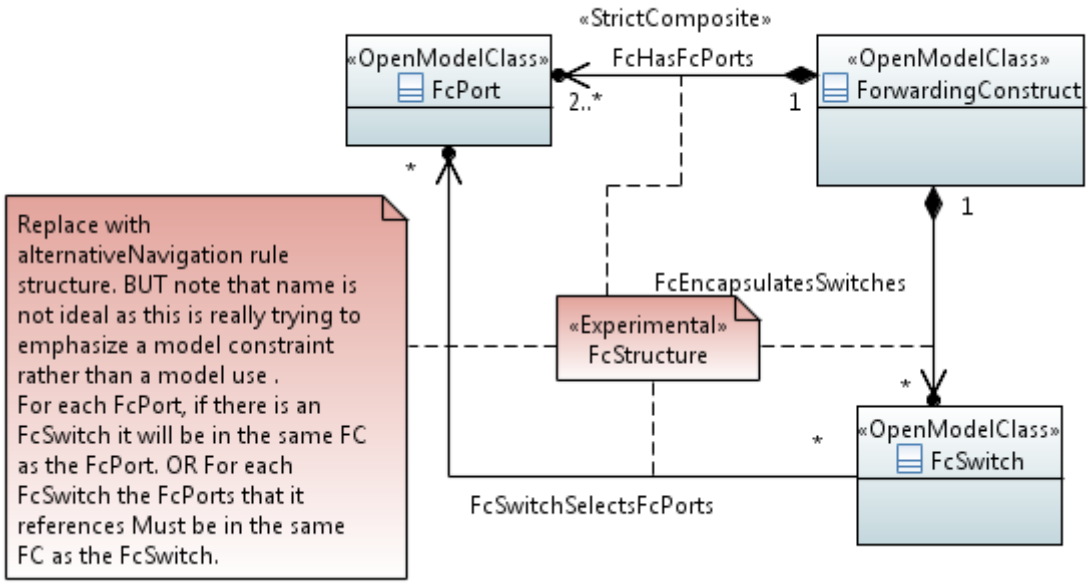
CoreModel diagram: HighLevelSkeltonOverviewWithLoopsHighlighted-Altenative1

Figure 3-2 Association interrelationship rules alternative 1



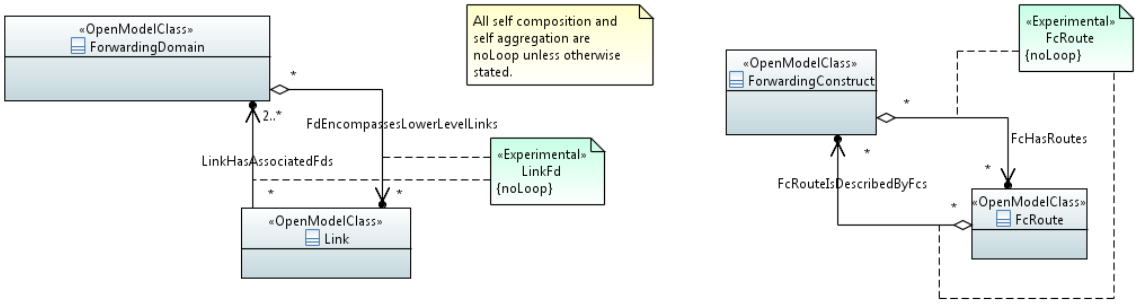
CoreModel diagram: AlternativeNavigationRuleExample-LinkFdLtp

Figure 3-3 Highlighting loops and spirals – FD, Link and LTP example



CoreModel diagram: AlternativeNavigationRuleExample-FcPort

Figure 3-4 Highlighting loops and spirals – FC, FcPort and FcSwitch example



CoreModel diagram: NoLoopRuleExample-FdLinkAndFcFcRoute

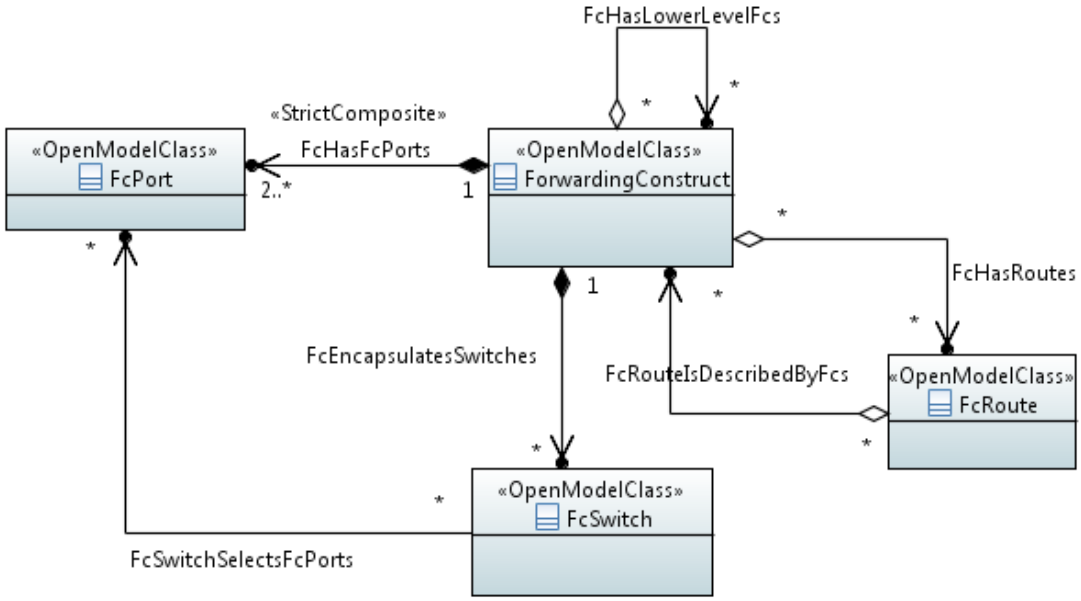
Figure 3-5 No Loops Rule

3.2.4 Strict Composition

Although the concept is now relatively stable there are some potential enhancements. The Strict Composite form is shown below.

The approach uses a stereotype to identify complete dependency. In the model below reporting the FC without its FcPorts is not useful whereas reporting the FC without its FcRoute is useful

A route has a life bounded by that of the containing FC as does an FcPort

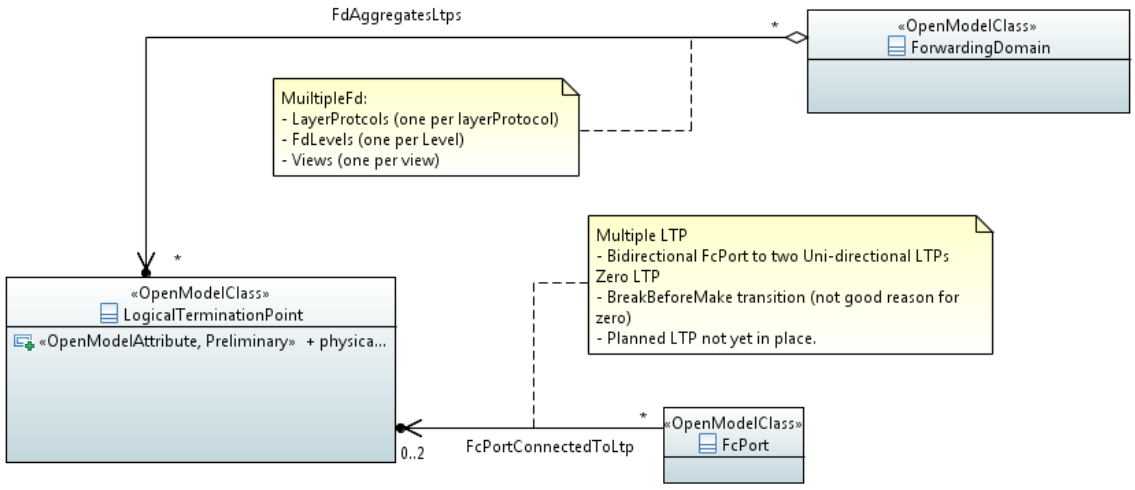


CoreModel diagram: StrictCompositionFcDetailExample

Figure 3-6 Strict Composite

3.2.5 Multiplicity restrictions

Many multiplicity ranges have subtle case based restrictions that are not reflected in the model. The following figure provides a view of a simple approach to recording multiplicity restrictions.



CoreModel diagram: MultiplicityRuleExampes

Figure 3-7 Multiplicity rule example

End of Document