OPEN NETWORKING
FOUNDATION

Core Information Model (CoreModel)

TR-512.6

Physical

Version 1.3.1
January 2018

OpenFlow™

ONF Document Type: Technical Recommendation
ONF Document Name: Core Information Model version 1.3.1

Disclaimer

Open Networking Foundation
2275 E. Bayshore Road, Suite 103, Palo Alto, CA 94303
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

Disclaimer 2

Important note 2

Document History 6

1 Introduction 7

1.1 References 7
1.2 Definitions 7
1.3 Conventions 7
1.4 Viewing UML diagrams 7
1.5 Understanding the figures 7

2 Introduction to the Physical model 8

3 Physical model detail 8

3.1 Equipment Pattern 8
3.1.1 Equipment 11
3.1.2 Holder 12
3.1.3 Connector 12
3.1.4 Cable 13
3.2 Equipment Detail 14
3.2.1 Invariant Equipment Detail 15
3.2.1.1 EnvironmentalRating 15
3.2.1.2 EquipmentInstance 15
3.2.1.3 EquipmentStructure 16
3.2.1.4 EquipmentType 16
3.2.1.5 HolderStructure 16
3.2.1.6 ManufacturedThing 17
3.2.1.7 ManufacturerProperties 17
3.2.1.8 MechanicalFeatures 18
3.2.1.9 OperatorAugmentedEquipmentInstance 18
3.2.1.10 OperatorAugmentedEquipmentType 18
3.2.1.11 PhysicalCharacteristics 18
3.2.1.12 Position 19
3.2.1.13 SpatialPropertiesOfType 19
3.2.1.14 Swappability 20
3.2.2 Dynamic Equipment Detail 20
3.2.2.1 FunctionEnablers 20

3.2.2.2 HolderMonitor 20
3.2.2.3 Location 21
3.2.2.4 MechanicalFunctions 21
3.2.2.5 PhysicalProperties 21
3.3 Connector to LTP sketch 22
3.3.1 AccessPort 22
3.3.2 ElementalSignals 23
3.3.3 GroupOfPins 23
3.3.4 Pin 23
3.3.5 PinGroup 23
3.3.6 PinLayout 24
3.3.7 SignalRefPt 24
3.3.8 SignalRefPtGroup 24
3.4 Equipment to Function Sketch 24
3.4.1 Equipment to Function using ProcessingConstruct 24
3.4.2 Classes associated with Equipment to function mapping with resilience 26
3.4.2.1 ProcessingConstruct (PC) 26
3.4.2.2 PcPort 26
3.4.2.3 ResilienceSelector 26
3.4.2.4 ForwardingConstruct (FC) 26
3.4.2.5 FcSwitch 27
3.4.2.6 Equipment 28
3.4.3 Obsolete model from V1.2 use to illustrate the V1.3 model 28
3.4.3.1 AggregateFunction 29
3.4.3.2 AtomicFunction 29
3.4.3.3 FunctionBlock 29
3.4.3.4 ResilienceSelector 30
3.4.3.5 ResilientFunctionBlock 30
3.5 FRU and non-FRU 30
3.5.1 FieldReplaceable 31
3.5.2 NonFieldReplaceable 31
3.6 Connector Rules 32
3.6.1 ConectorInHolder 33
3.6.2 ConnectorCableEnd 33
3.6.3 ConnectorOnEquipmentForCable 34
3.6.4 ConnectorOnEquipmentForHolder 34
3.7 Expected and Actual 34
3.7.1 ActualEquipment 35
3.7.2 ActualHolder 35
3.7.3 ExpectedEquipment 36
3.7.4 ExpectedHolder 36

3.8 Specification....36
3.8.1 NonFruSupportPosition....37
3.8.2 SupportConstraints....37
3.8.3 SupportedEquipmentType....37
3.8.4 SupportedNonFruType....38
3.9 Physical Connector and conceptual Port....38

4 Work in progress (see also TR-512.FE)....41
4.1 Addressing....41
4.2 Physical to functional model....41
4.3 Actual v expected....43

List of Figures

Figure 3-1 Skeleton Class Diagram of key object classes....9
Figure 3-2 - Equipment and Holder example....10
Figure 3-3 – Inside the Equipment....11
Figure 3-4 Equipment Detail Structure....14
Figure 3-5 Connector to LTP....22
Figure 3-6 Equipment to Function using ProcessingConstruct and showing Resilience....25
Figure 3-7 Equipment to Function....29
Figure 3-8 FRU and Non-FRU rules....31
Figure 3-9 Connector rules....33
Figure 3-10 Expected and actual....35
Figure 3-11 Specification....37
Figure 3-12 Basic cases of Physical Port Reference....39
Figure 3-13 More Complex cases of intertwined Connectors....40
Figure 3-14 Unidirectional Cases....40
Figure 4-1 Connector/Port based addressing of LTPs....41
Figure 4-2 Simplified sketch of physical to functional....42
Figure 4-3 Simplified sketch of forms of resilience in an NE....43
Figure 4-4 Expectation v actual showing mismatch and blocking....44

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).
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	Version 1.3 [Published via wiki only]
1.3.1	January 2018	Addition of text related to approval status.

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%) or 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 Physical model

The focus of this document is the modeling of physical things, especially equipment, in the ONF-CIM.

Note:

- A majority of the Physical model is experimental at this stage (some key classes and attributes have been upgraded to Mature/Preliminary since V1.2). It was considered vital to publish the work in progress on equipment as it is clearly an important part of the overall model. Many of the attributes and classes are not fully documented in the model.
- The Physical model deals with physical things where a physical thing is something that can be "measured with a ruler"[1]

This document:

- Introduces the Physical model structure
- Describes the key classes of the Physical model
- Explains the attributes of the Physical model
- Describes the relationship between the Connector and the LogicalTerminationPoint (LTP)
- Shows how the model deals with the relationship between physical and functional views
- Explains how the specification model describes equipment schemes (rules etc)
- Highlights work in progress to further advance the Physical model

The Physical model relates to:

- The Core Network Model including Termination and Forwarding described in TR-512.2 and Topology described in TR-512.4.
- The generalized processing and constraint model described in TR-512.11
- The specification model described in TR-512.7.

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

3 Physical model detail

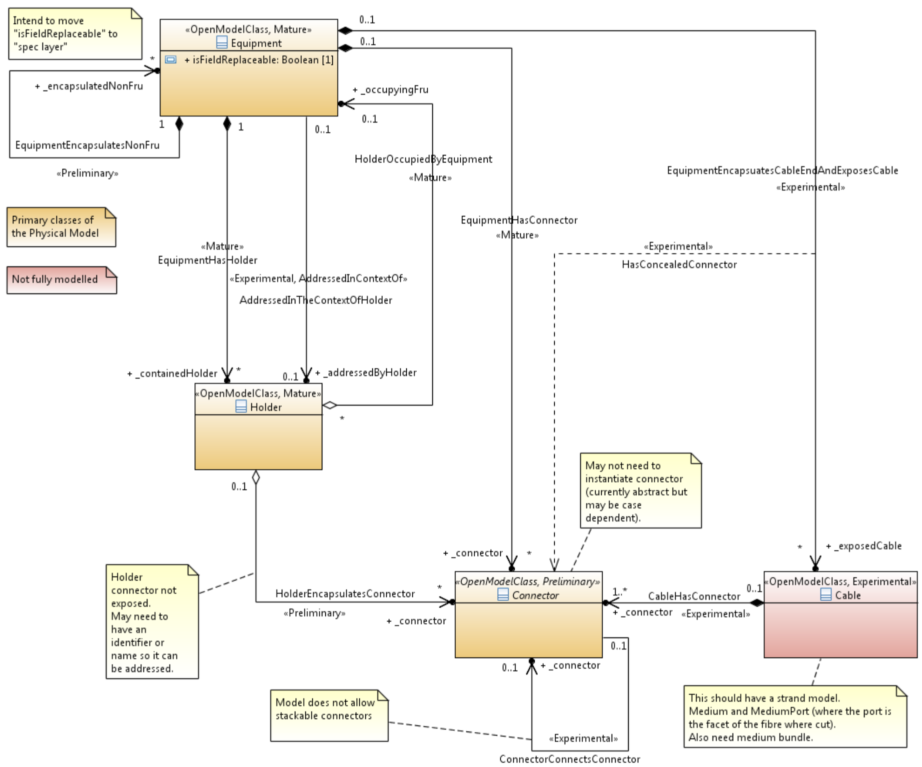
This section starts with a basic view of the equipment classes, then progresses through detail to sophisticated (and highly experimental) representations of equipment model constraints.

3.1 Equipment Pattern

The figure below sets out the basic equipment pattern.

[1] Often the word “physical” is used in the context of non-mechanical functional things. All non-mechanical functional things are considered under the functional model here regardless of how closely they are bound to the physical entities. All non-mechanical functional representations are considered as emergent abstractions and as virtualized (to some degree within a physical boundary). Mechanical functional things, such as fans, are not modelled in detail.

The classes of the model are described briefly after the figure. The associations are assumed to be sufficiently self-explanatory at this stage.



CoreModel diagram: Equipment-Pattern

Figure 3-1 Skeleton Class Diagram of key object classes

Taking a simple chassis, pictured in the figure below, as an example, we can consider the Holders to be the spaces within a piece of Equipment where a Holder is designed to accommodate another piece of Equipment. In a normal chassis it is possible that an Equipment may occupy several Holder, however it is not possible for a Holder to accommodate more than one Equipment at a time.

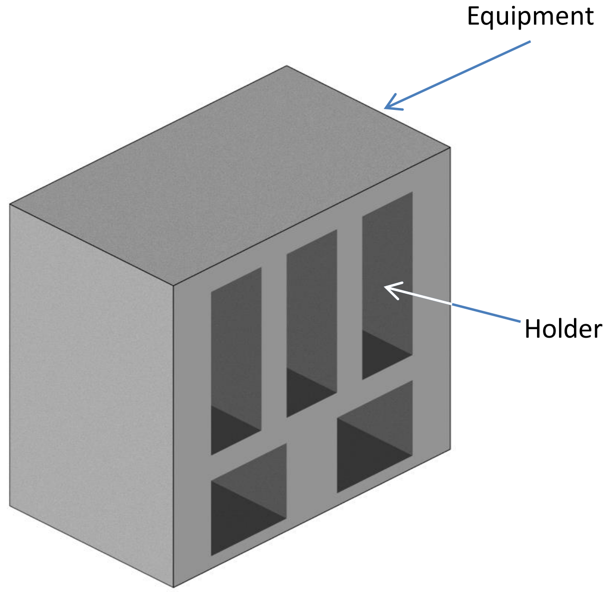


Figure 3-2 - Equipment and Holder example

There are two distinct roles for the Equipment entity controlled by the one attribute shown in the model figure above (isFieldReplaceable):

- Field Replaceable Unit (FRU):
  - Can be replaced in the field.
    - May be standalone
    - May plug in to a Holder in another Equipment (if not stand-alone)
- Non-Field Replaceable Unit (NFRU):
  - Cannot be replaced in the field. Is simply a subordinate part of an FRU (or another NFRU – where there must be an FRU at the top of the hierarchy).
  - Does not have any exposed Holders (any associated Holders are assumed to belong to the containing FRU).
  - Does not have any Connectors (any associated Connectors are assumed to belong to the containing FRU).

A method for representation of these restrictions is covered in section 3.5 FRU and non-FRU on page 30.

Connectors allow for Cables to be plugged in. So, for example, an SFP is a piece of Equipment that plugs in a Holder, and has a Connector on its front for a Cable with a Connector to be plugged in.

Looking at the picture below, we have removed some of the chassis panel to show the inside. We see that the circuit-pack (Equipment) will actually plug into Connectors on the (non-FRU) backplane. If we wish to explicitly represent the Connectors on the backplane, then we use association HolderEncapsulatesConnector to relate the Connector to the Holder it is at the back of.

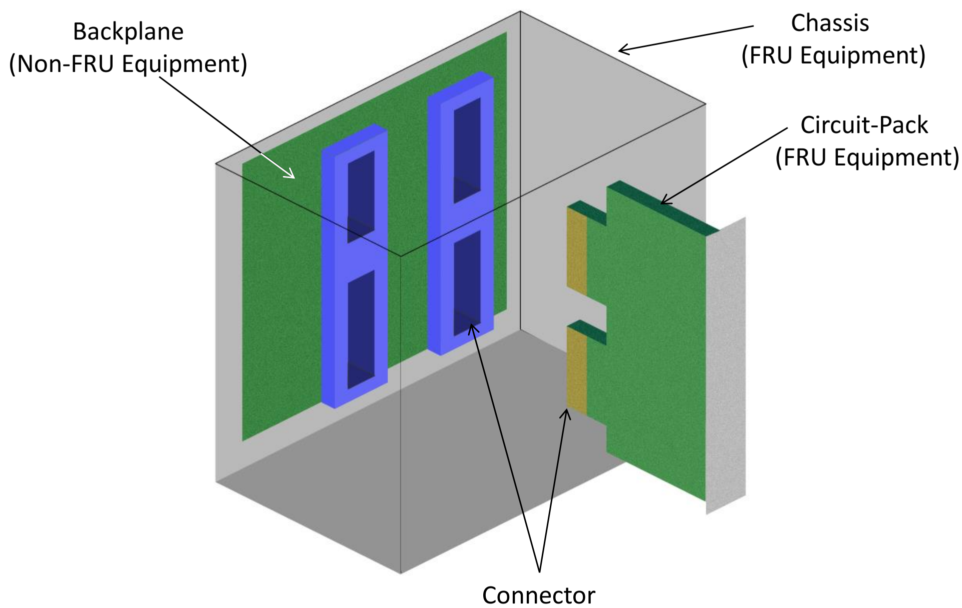


Figure 3-3 – Inside the Equipment

3.1.1 Equipment

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentPatternStructure::ObjectClasses::Equipment

Represents any relevant physical thing.
Can be either field replaceable or not field replaceable.
Note: The model is currently constrained to inside plant.

Inherits properties from:

- GlobalClass

This class is Mature.

Table 1: Attributes for Equipment

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_connector	Mature	A connector on the equipment.
_containedHolder	Mature	References the Holder in an Equipment that is available to take other Equipments. For example: - Slot in a sub-rack - Slot in a Field Replaceable Unit that can take a small form-factor pluggable.
_addressedByHolder	Experimental	See referenced class

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_encapsulatedNonFru	Preliminary	An Equipment that is part of this Equipment and that is not separately field replaceable (i.e. will be field replaced with this Equipment).
_exposedCable	Experimental	See referenced class
isFieldReplaceable	Preliminary	Indicates whether or not the equipment can be removed and replaced "in the field" (i.e. in a deployment) by normal operations personnel.

3.1.2 Holder

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentPatternStructure::ObjectClasses::Holder

Represents a space in an equipment in which another equipment can be fitted in the field.

Inherits properties from:

- LocalClass

This class is Mature.

Table 2: Attributes for Holder

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_connector	Preliminary	The connector associated with a Holder of an Equipment. May represent connector on a backplane that takes Field Replaceable Units or a connector on a circuit pack that takes an SFP (Small Form-factor Pluggable).
_occupyingFru	Mature	The FRU that is occupying the holder. A holder may be unoccupied. An FRU may occupy more hat one holder (using or blocking are intentionally not distinguished here).

3.1.3 Connector

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentPatternStructure::ObjectClasses::Connector

Represents a connector that may be fully exposed (e.g. to plug in a cable or on the end of a cable) or partially exposed (e.g. backplane to plug in another piece of equipment such as a module).

A physical port on the Equipment. A place where signals produced by the functionality of the Equipment may be accessed.

This class is abstract.

Inherits properties from:

- GroupOfPins
- LocalClass

This class is Preliminary.

Table 3: Attributes for Connector

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_connector	Preliminary	The Connector that is attached to this Connector so as to join the Equipment/Cable with this Connector to another Equipment/Cable. This may provide physical support and/or allow signal flow.
orientation	Experimental	To be provided
connectorType	Experimental	To be provided
role	Experimental	The purpose of the Connector in the physical space and the functional space.

3.1.4 Cable

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentPatternStructure::ObjectClasses::Cable

Basic model representing a cable with connectors fitted where the cable is "short" (e.g. patch cord, in-station cabling).
This is intentionally a very basic representation of a cable.
In a more sophisticated representation cable ends might be represented that then associate to the attached connector.
At this point it is assumed that the basic model is sufficient.

Inherits properties from:

- GlobalClass

This class is Experimental.

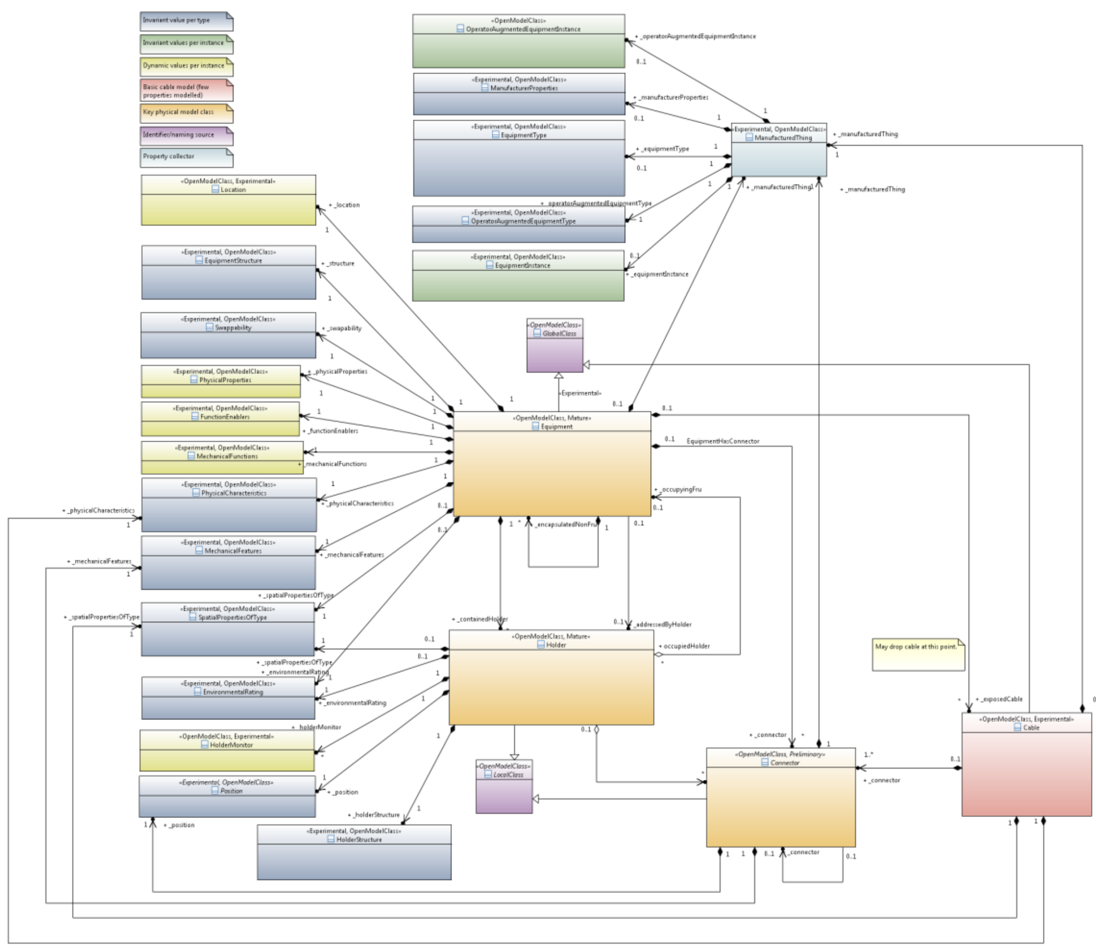
Table 4: Attributes for Cable

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_connector	Preliminary	A connector that terminates the Cable to support the cable and/or allow signal flow into/out of the Cable.

3.2 Equipment Detail

The figure below highlights classes that group together related attributes (related as suggested by the name of the class). As noted in the key to the diagram the attributes are also grouped on the degree of variation. This latter grouping will guide the construction of specifications indicating what can reside only in the spec and what has to be available per instance (see TR-512.7 for more information)



CoreModel diagram: Equipment-DetailWithoutAttributes

Figure 3-4 Equipment Detail Structure

3.2.1 Invariant Equipment Detail

The following classes have attributes that do not change in value for the life of the Equipment. Some are the same value across all Equipment of the same type.

3.2.1.1 EnvironmentalRating

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::InvariantDetails::EnvironmentalRating

Represents the invariant physical operational boundaries for the equipment/holder type.

This class is Experimental.

Table 5: Attributes for EnvironmentalRating

Attribute Name	**Lifecycle Stereotype (empty = Mature)**	**Description**
thermalRating	Experimental	This attribute represents the thermal characteristics (preferred maximum/minimum, absolute maximum/minimum etc) that the entity can tolerate.
powerRating	Experimental	This attribute represents the power chracteristics (peak and mean per power source) of the entity. For an Equipment this is the power consumption. For a Holder this is the power that can be conveyed.
humidityRating	Experimental	This attribute represents the humidity characteristics (preferred maximum/minimum, absolute maximum/minimum etc) that the entity can tolerate.

3.2.1.2 EquipmentInstance

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::InvariantDetails::EquipmentInstance

Represents the per instance invariant properties of the equipment.

This class is Experimental.

Table 6: Attributes for EquipmentInstance

Attribute Name	**Lifecycle Stereotype (empty = Mature)**	**Description**
manufactureDate	Experimental	This attribute represents the date on which this instance is manufactured.
serialNumber	Experimental	This attribute represents the serial number of this instance.
assetInstanceIdentifier	Experimental	This attribute represents the asset identifier of this instance from the manufacturer's perspective.

3.2.1.3 EquipmentStructure

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::InvariantDetails::EquipmentStructure

Represents the form of the equipment.

This class is Experimental.

Table 7: Attributes for EquipmentStructure

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
category	Experimental	This attribute provides the identifier for a category of equipments regarded as having particular shared characteristics.

3.2.1.4 EquipmentType

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::InvariantDetails::EquipmentType

Represents the invariant properties of the equipment that define and characterize the type.

This class is Experimental.

Table 8: Attributes for EquipmentType

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
description	Experimental	To be provided
modelIdentifier	Experimental	To be provided
partTypeIdentifier	Experimental	To be provided
typeName	Experimental	To be provided
version	Experimental	To be provided

3.2.1.5 HolderStructure

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::InvariantDetails::HolderStructure

Represents the form of the holder.

This class is Experimental.

Table 9: Attributes for HolderStructure

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
holderCategory	Experimental	To be provided

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
isCaptive	Experimental	To be provided
isGuided	Experimental	To be provided
isQuantisedSpace	Experimental	To be provided

3.2.1.6 ManufacturedThing

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::InvariantDetails::ManufacturedThing

Collects all invariant aspects of a manufactured thing.

This class is Experimental.

Table 10: Attributes for ManufacturedThing

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_manufacturerProperties	Experimental	See referenced class
_equipmentType	Experimental	See referenced class
_equipmentInstance	Experimental	See referenced class
_operatorAugmentedEquipmentType	Experimental	See referenced class
_operatorAugmentedEquipmentInstance	Experimental	See referenced class

3.2.1.7 ManufacturerProperties

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::InvariantDetails::ManufacturerProperties

Represents the properties of the manufacturer.

This class is Experimental.

Table 11: Attributes for ManufacturerProperties

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
manufacturerIdentifier	Experimental	To be provided
manufacturerName	Experimental	To be provided

3.2.1.8 MechanicalFeatures

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::InvariantDetails::MechanicalFeatures

Represents the invariant characteristics of dynamic mechanical features of a physical thing.

This class is Experimental.

3.2.1.9 OperatorAugmentedEquipmentInstance

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::InvariantDetails::OperatorAugmentedEquipmentInstance

Represents the invariant properties of the equipment asset allocated by the operator that define and characterize the type.

Table 12: Attributes for OperatorAugmentedEquipmentInstance

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
assetInstanceIdentifier		This attribute represents the asset identifier of this instance from the operator's perspective.

3.2.1.10 OperatorAugmentedEquipmentType

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::InvariantDetails::OperatorAugmentedEquipmentType

Represents the invariant properties of the equipment asset allocated by the operator that define and characterize the type.

This class is Experimental.

Table 13: Attributes for OperatorAugmentedEquipmentType

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
assetTypeIdentifier	Experimental	To be provided

3.2.1.11 PhysicalCharacteristics

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::InvariantDetails::PhysicalCharacteristics

Represents the invariant physical characteristics (including composition and physical robustness) of the type.

This class is Experimental.

Table 14: Attributes for PhysicalCharacteristics

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
weightCharaceristics	Experimental	To be provided
fireCharacteristics	Experimental	To be provided
materials	Experimental	To be provided

3.2.1.12 Position

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::InvariantDetails::Position

Represents the invariant relative position of the holder (with respect to some frame of reference in an equipment) or connector on an equipment or pin in a connector.

This class is abstract.

This class is Experimental.

Table 15: Attributes for Position

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
relativePosition	Experimental	To be provided

3.2.1.13 SpatialPropertiesOfType

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::InvariantDetails::SpatialPropertiesOfType

Represents the basic invariant spatial properties of a physical thing.

This class is Experimental.

Table 16: Attributes for SpatialPropertiesOfType

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
height	Experimental	To be provided
width	Experimental	To be provided
length	Experimental	To be provided

3.2.1.14 Swappability

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::InvariantDetails::Swappability

Represents the degree of field replacement that is possible for the equipment type.

This class is Experimental.

Table 17: Attributes for Swappability

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
isHotSwappable	Experimental	To be provided

3.2.2 Dynamic Equipment Detail

The following classes have attributes that can change in value during the life of the Equipment.

3.2.2.1 FunctionEnablers

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::DynamicDetails::FunctionEnablers

Represents the dynamic aspects of the properties that relate to the motive force that directly enable functionality to emerge from the equipment.

This class is Experimental.

Table 18: Attributes for FunctionEnablers

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
powerState	Experimental	To be provided

3.2.2.2 HolderMonitor

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::DynamicDetails::HolderMonitor

Represents the dynamic state of the holder instance.

This class is Experimental.

Table 19: Attributes for HolderMonitor

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
isActive	Experimental	To be provided
isActualMismatchWithExpected	Experimental	To be provided
_aggregateFunction	Experimental	See referenced class

3.2.2.3 Location

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::DynamicDetails::Location

Represents where the equipment is.

This class is Experimental.

Table 20: Attributes for Location

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
equipmentLocation	Experimental	To be provided
geographicalLocation	Experimental	To be provided

3.2.2.4 MechanicalFunctions

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::DynamicDetails::MechanicalFunctions

Represents the dynamic aspects of the mechanical functions of the equipment.

This class is Experimental.

Table 21: Attributes for MechanicalFunctions

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
rotationSpeed	Experimental	To be provided

3.2.2.5 PhysicalProperties

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentDetail::ObjectClasses::DynamicDetails::PhysicalProperties

Represents the dynamic aspects of the physical environmental properties of the equipment.

This class is Experimental.

Table 22: Attributes for PhysicalProperties

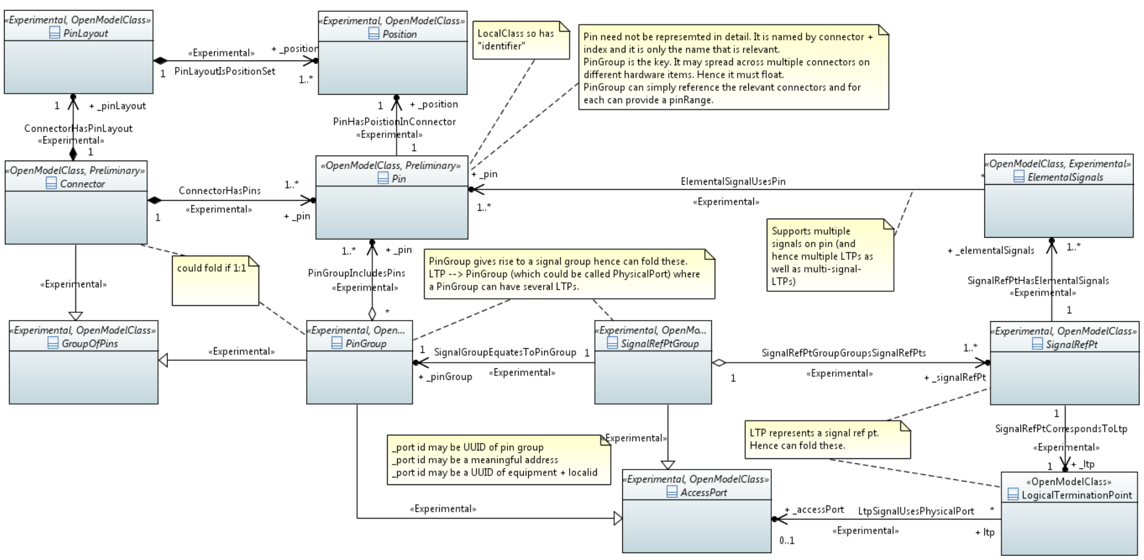
Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
temperature	Experimental	To be provided

3.3 Connector to LTP sketch

The figure below is a sketch model relating Connector and LTP. The Connector is first decomposed into pins. The pins are then grouped across potentially multiple Connectors with a granularity that corresponds to signal groups such that each defined group of pins corresponds to a group of signals. The groupings are such that if any pins were to be removed from the group, then all the signals of the group would be lost, and such that the signals of the pin group are indivisible when they flow through the pins.

Following the figure there are definitions of each of the new classes not covered in the previous sections of this document. See section 3.9 Physical Connector and conceptual Port on page 38 for some pictorial examples of the interaction of some of the entities discussed here.

The notes shown on the figure touch on some ongoing discussions.



CoreModel diagram: Equipment-ConnectorPinPortAndLTP

Figure 3-5 Connector to LTP

3.3.1 AccessPort

Qualified Name: CoreModel::CorePhysicalModel-Initial::ConnectorAndPin::ObjectClasses::AccessPort

A conceptual access for a group of signals (where that group of signals cannot be separated).

This class is abstract.

This class is Experimental.

3.3.2 ElementalSignals

Qualified Name: CoreModel::CorePhysicalModel-Initial::ConnectorAndPin::ObjectClasses::ElementalSignals

The elemental (sub-atomic) parts of an "indivisible" signal where processing in the LTP is required to extract the elemental signals.

This class is abstract.

This class is Experimental.

3.3.3 GroupOfPins

Qualified Name: CoreModel::CorePhysicalModel-Initial::ConnectorAndPin::ObjectClasses::GroupOfPins

A group of pins from one or more connectors relevant for some purpose.

This class is abstract.

This class is Experimental.

3.3.4 Pin

Qualified Name: CoreModel::CorePhysicalModel-Initial::ConnectorAndPin::ObjectClasses::Pin

An individual physical connection point (male or female) that is not relevantly divisible.
May be capable of carrying electrical or optical signals.
A pin normally has only one wire/fiber strand attached.
It may have more than one wire/fiber attached but is such that the attachment forms a physical merge (all attached things receive exactly the same signal and any inputs to the pin are electrically/optically merged).

This class is abstract.

This class is Preliminary.

3.3.5 PinGroup

Qualified Name: CoreModel::CorePhysicalModel-Initial::ConnectorAndPin::ObjectClasses::PinGroup

A group of pins that together provide signal group where any one pin removed from the group will prevent the signals of the signal group from flowing successfully.

This class is abstract.

Inherits properties from:
- AccessPort
- GroupOfPins

This class is Experimental.

3.3.6 PinLayout

Qualified Name: CoreModel::CorePhysicalModel-Initial::ConnectorAndPin::ObjectClasses::PinLayout

The structuring of pins in a connector.

This class is abstract.

This class is Experimental.

3.3.7 SignalRefPt

Qualified Name: CoreModel::CorePhysicalModel-Initial::ConnectorAndPin::ObjectClasses::SignalRefPt

A single coherent signal as processed by a single LTP.

This class is abstract.

This class is Experimental.

3.3.8 SignalRefPtGroup

Qualified Name: CoreModel::CorePhysicalModel-Initial::ConnectorAndPin::ObjectClasses::SignalRefPtGroup

A physical indivisible group of signals.

This class is abstract.

Inherits properties from:
- AccessPort

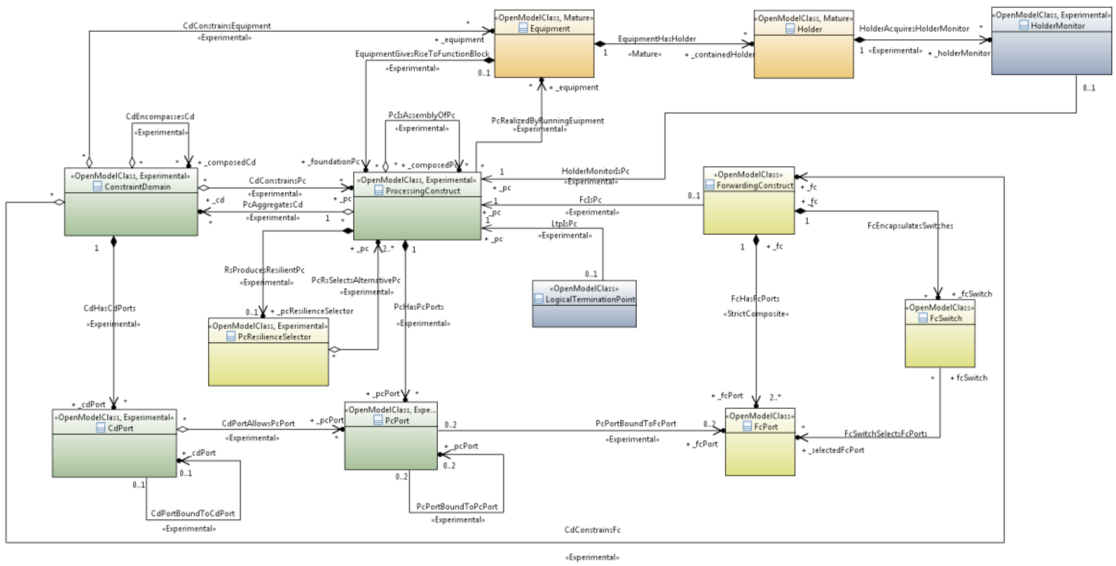
This class is Experimental.

3.4 Equipment to Function Sketch

As ProcessingConstruct (see TR-512.11) has been added to this release of the model the Equipment to Function model can be updated.

3.4.1 Equipment to Function using ProcessingConstruct

The following figure shows the use of PC to model the functionality emergent from an Equipment assembly.



CoreModel diagram: Equipment-ProcessingConstructAndResilience

Figure 3-6 Equipment to Function using ProcessingConstruct and showing Resilience

The ProcessingConstruct is used to represent:

- The function blocks supported by an Equipment
- The decomposition of each function block into atomic parts (via PcIsAssemblyOfPc)
- The assembly of the atomic parts (also via PcIsAssemblyOfPc) into more aggregate functions (across Equipment boundaries potentially) that can be seen as function blocks for further decomposition and aggregation as appropriate

The above is recursed until standard telecoms functions are achieved, such as LTP etc (shown in the figure via the "..IsPc" associations (that are intentionally not navigable from the LTP etc)).

The "via PcIsAssemblyOfPc" is essentially a compact form of the Component-System pattern (see TR-512.A.2).

The figure also shows two forms of representation of resilience:

- The ResilienceSelector which is a somewhat crude model allowing a basic representation of many PCs of the same function to provide a resilient form of that function. The model could also cover load sharing. The ResilienceSelector essentially substitutes for and expands upon the PcIsAssemblyOfPc allowing
- The ForwardingConstruct and FcSwitch which is a more sophisticated and versatile model that can be used to describe complex switching schemes (which is a standard part of the PC model) where the PCs are represented with PcPorts and it is the essential functionality available at those ports provides the protected function (via one of its ports).

The model from V1.2 is described in the next section as an illustration of the capability now supported by PC.

The following is a summary of the key classes and attributes related to Equipment to function mapping with resilience.

3.4.2 Classes associated with Equipment to function mapping with resilience

3.4.2.1 ProcessingConstruct (PC)

This section focuses on the aspects that support functional decomposition, functional aggregation/assembly and functional abstraction.

3.4.2.2 PcPort

This section focuses on assembly of components with no need for LTPs where the component association in the assembly is direct and indirect via a resilience mechanism.

3.4.2.3 ResilienceSelector

3.4.2.4 ForwardingConstruct (FC)

This section focuses on the protection aspects of the FC.

Qualified Name: CoreModel::CoreNetworkModel::ObjectClasses::ForwardingConstruct

The ForwardingConstruct (FC) class models enabled constrained potential for forwarding between two or more FcPorts at a particular specific layerProtocol .
Like the LTP, the FC supports any transport protocol including all analogue, circuit and packet forms.
For digital layer networks it is used to effect forwarding of transport characteristic (layer protocol) information.
An FC can be in only one ForwardingDomain (FD).
The FC is a forwarding entity.
At a low level of the recursion, a FC represents a cross-connection within an NE. It may also represent a fragment of a cross-connection under certain circumstances.
The FC object can be used to represent many different structures including point-to-point (P2P), point-to-multipoint (P2MP), rooted-multipoint (RMP) and multipoint-to-multipoint (MP2MP) bridge and selector structures for linear, ring or mesh protection schemes.
When applied to media, the FC represents the ability for a flow/wave (potentially containing information), to be propagated between FcPorts.
The existence of a FC instance is independent of the presence (or absence) of a flow/wave (and any information encoded within it).
A flow/wave cannot propagate in the absence of a FC instance.

Inherits properties from:

- GlobalClass
- ForwardingEntity

Table 23: Attributes for ForwardingConstruct

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_fcPort		The FcPorts define the boundary of the FC. The FC is accessed via the FcPorts. Flow within the FC is defined in terms of its FcPorts.

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_fcSwitch		If an FC exposes protection (having two FcPorts that provide alternative identical inputs/outputs), the FC will have one or more associated FcSwitch objects. The arrangement of switches for a particular instance is described by a referenced FcSpec

3.4.2.5 FcSwitch

Qualified Name: CoreModel::CoreNetworkModel::ObjectClasses::Resilience::FcSwitch

The FcSwitch class models the switched forwarding of traffic (traffic flow) between FcPorts and is present where there is protection functionality in the FC.
If an FC exposes protection (having two or more FcPorts that provide alternative identical inputs/outputs), the FC will have one or more associated FcSwitch objects to represent the alternative flow choices visible at the edge of the FC.
The FC switch represents and defines a protection switch structure encapsulated in the FC and essentially "decorates" FCs that are involved in resilience schemes that use switching in a protection mechanism.
Essentially FcSwitch performs one of the functions of the Protection Group in a traditional model. It associates 2 or more FcPorts each playing the role of a Protection Unit.
One or more protection, i.e. standby/backup, FcPorts provide protection for one or more working (i.e. regular/main/preferred) FcPorts where either protection or working can feed one or more protected FcPort.
The switch may be used in revertive or non-revertive (symmetric) mode. When in revertive mode it may define a waitToRestore time.
It may be used in one of several modes including source switched, destination switched, source and destination switched etc. (covering cases such as 1+1 and 1:1).
It may be locked out (prevented from switching), force switched or manual switched.
It will indicate switch state and change of state.
The switch can be switched away from all sources such that it becomes open and hence two coordinated switches can both feed the same LTP so long as at least one of the two is switched away from all sources (is "open").
The ability for a Switch to be "high impedance" allows bidirectional ForwardingConstructs to be overlaid on the same bidirectional LTP where the appropriate control is enabled to prevent signal conflict.
This ability allows multiple alternate routes to be present that otherwise would be in conflict.

Inherits properties from:

- LocalClass

Table 24: Attributes for FcSwitch

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_selectedFcPort		Indicates which points are selected by the switch. Depending on the switch spec (via FcSpec) - more than one FcPort can be selected at any one time (e.g. egress switch, ingress packet switch) - zero FcPorts can be selected. For an ingress switch this indicates that the switch common (egress) is "high impedance" .

3.4.2.6 Equipment

This section focuses on physical support for functionality.

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentPatternStructure::ObjectClasses::Equipment

Represents any relevant physical thing.
Can be either field replaceable or not field replaceable.
Note: The model is currently constrained to inside plant.

Inherits properties from:

- GlobalClass

This class is Mature.

Table 25: Attributes for Equipment

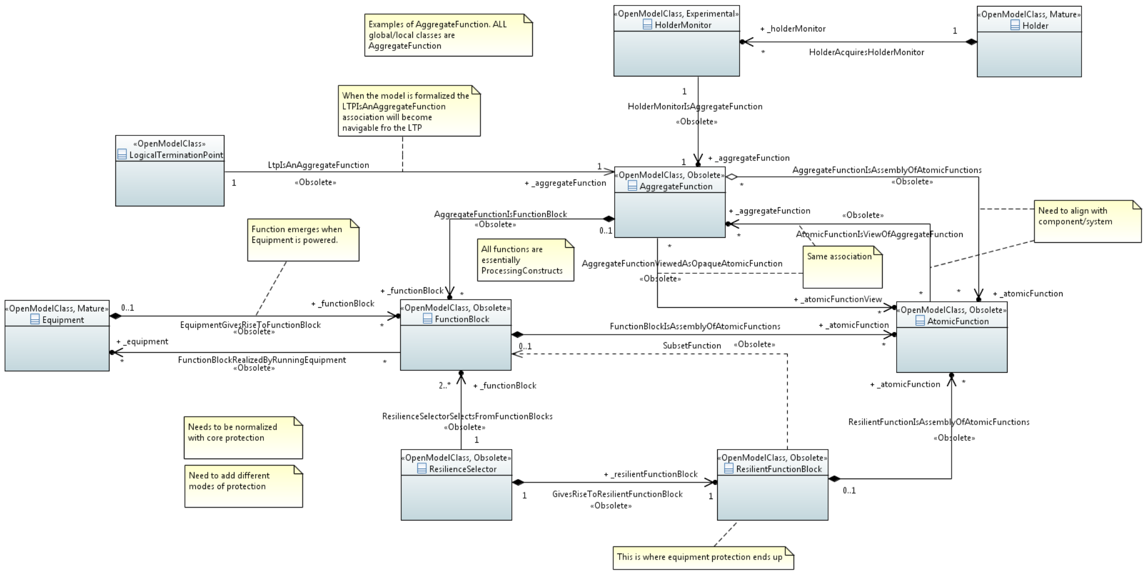
Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
_foundationPc	Experimental	Lowest level functional block with 1:1 to equipment.

3.4.3 Obsolete model from V1.2 use to illustrate the V1.3 model

The figure below is a sketch model relating Equipment to Function from V1.2. Whilst the classes are obsolete they still provide a helpful illustration of the model. As the comment in the figure says, "All functions are essentially ProcessingConstruct" (i.e. any class with the word "function" in its name is replaces by PC in the V1.3 model described above). The Equipment functionality support is first exposed as coarse FunctionBlocks (e.g. arithmetic process, traffic process etc). The function block may be made resilient via a complex protection switch which can select the functionality from one or more instances[2]. The FunctionBlock is then decomposed into AtomicFunctions which can then be assembled to form AggregateFunctions (e.g. the LTP). These two steps allow versatile mapping from the hardware oriented function blocks to the conceptual functions such as LTP and LayerProtocol where the conceptual function may be

[2] The protection model has only had very limited development so far and the model is clumsy in this area.

"smeared" across several FunctionBlocks. The model also allows for recursive decomposition and assembly to any depth to allow for cases where intermediate representations are necessary to describe the functional emergence. For some illustrative figures of this see section 4.2 Physical to functional model on page 41.



CoreModel diagram: Equipment-EquipmentToFunction

Figure 3-7 Equipment to Function

3.4.3.1 AggregateFunction

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentToFunction::ObjectClasses::AggregateFunction

Represents some assembly of atomic functions that can be considered as useful from some perspective. Can be viewed as one or more functional blocks (essential leading to a recursive cycle of Block --> Atomic --> Aggregate --> Block).
Each of the functional entities in the model is a view of a single AggregateFunction.

This class is Obsolete.

3.4.3.2 AtomicFunction

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentToFunction::ObjectClasses::AtomicFunction

Represents the micro-function that is the largest function of the functional block that will not need to be subdivided when forming the relevant abstract views (i.e., it can just be assembled).

This class is Obsolete.

3.4.3.3 FunctionBlock

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentToFunction::ObjectClasses::FunctionBlock

Represents the chunks of base functionality provided by the equipment.
The chunks of base functionality are likely to relate to the hardware layout and be quite distinct from the functions of the familiar abstract representation.
The functions are necessarily abstract and, to a degree, virtualized.

This class is Obsolete.

3.4.3.4 ResilienceSelector

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentToFunction::ObjectClasses::ResilienceSelector

Represents the ability to select capability from two or more identical FunctionalBlocks so as to give rise to an equivalent emergent resilient function.

This class is Obsolete.

3.4.3.5 ResilientFunctionBlock

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentToFunction::ObjectClasses::ResilientFunctionBlock

Represents the functions emergent from a function protection process.
The emergent functions are necessarily significantly virtualized.

This class is Obsolete.

3.5 FRU and non-FRU

Note that the approach in this subsection to representation of the constraints using inheritance with stereotype is highly experimental. It is likely that this approach will change as the model progresses.

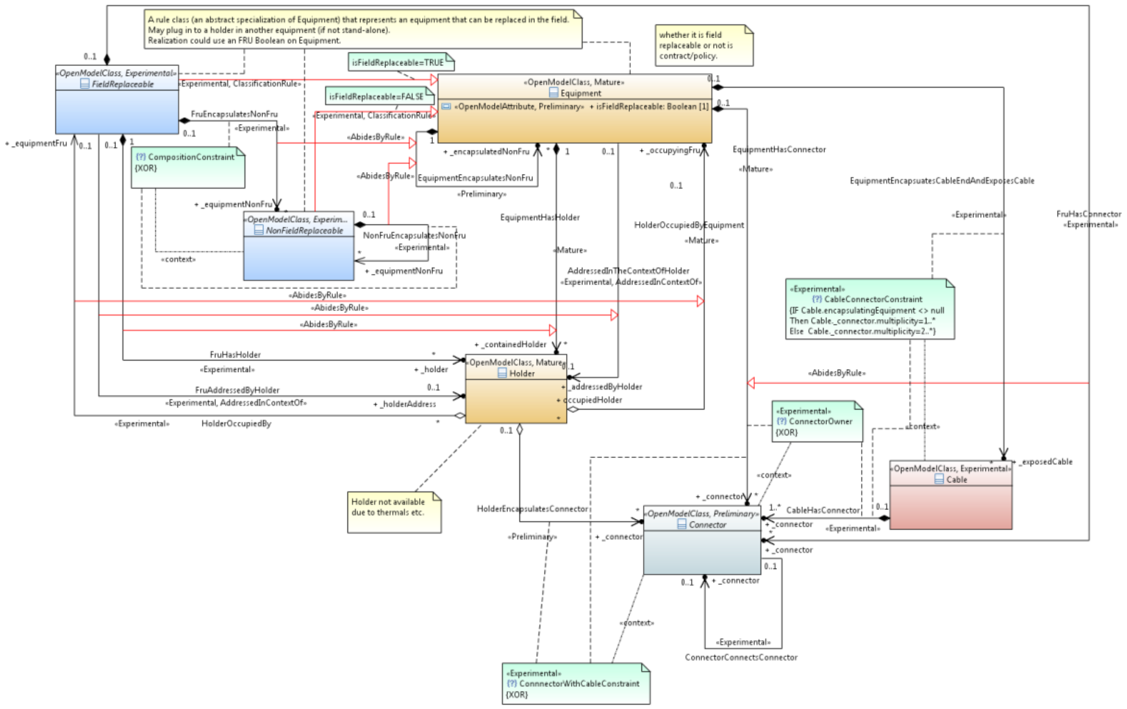
Considering the distinction between FRU and nonFRU there were two choices, model the FRU explicitly (and then expect instantiation of instances of distinct FRUs and non-FRUs classes) or model the FRU and non-FRU as Equipment with solely an attribute to indicate the case. The attribute approach was preferred but that lost rule detail present in the distinct class case.

So as not to lose the constraints, the FRU/non-FRU class distinctions were kept, those classes were made abstract and then applied as shown to the attribute based model developing an experimental technique

This technique is to use generalization modulated with stereotypes to represent the narrowing of a class to cover a defined case. The narrowed class does not gain attributes, the general class is fully populated whereas in the narrowed class attributes take specific fixed values where the specializations are all abstract and the generalization is concrete. The aim is to develop machine interpretable rule systems that allow the behavior of an instance of a generalized class to be constrained based upon the case.

The experimental stereotype « ClassificationRule » carries the properties that define the case, the stereotype « AbidesByRule » identifies the generalized association that constrained by the specialized association.

This model is experimental and requires significant further development. It is likely that an alternative form will eventually be used.



CoreModel diagram: Equipment-EquipmentToHolderRules

Figure 3-8 FRU and Non-FRU rules

3.5.1 FieldReplaceable

Qualified Name: CoreModel::CorePhysicalModel-Initial::RuleModels::FruNonFruRules::ObjectClasses::FieldReplaceable

A rule class (an abstract specialization of Equipment) that represents an equipment that can be replaced in the field.
May plug in to a holder in another equipment (if not stand-alone).
Realization could use an FRU Boolean on Equipment.

This class is abstract.

Inherits properties from:

- Equipment

This class is Experimental.

3.5.2 NonFieldReplaceable

Qualified Name: CoreModel::CorePhysicalModel-Initial::RuleModels::FruNonFruRules::ObjectClasses::NonFieldReplaceable

A rule class (an abstract specialization of Equipment) that represents an equipment that cannot be replaced in the field.
Is simply a subordinate part of an FRU (or another NFRU – where there must be an FRU at the top of the hierarchy).
Does not have any exposed holders (any associated holders are assumed to belong to the containing FRU).
Does not have any connectors (any associated connectors are assumed to belong to the containing FRU).

This class is abstract.

Inherits properties from:

- Equipment

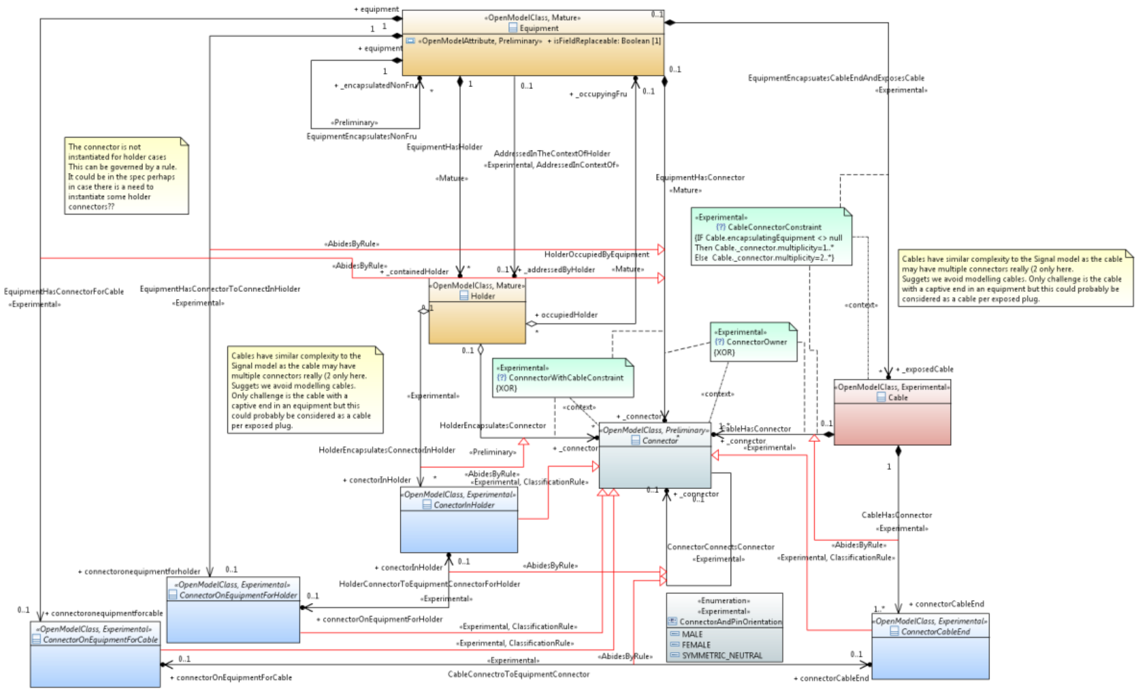
This class is Experimental.

3.6 Connector Rules

Note that the approach in this subsection to representation of the constraints using inheritance with stereotype is highly experimental. It is likely that this approach will change as the model progresses.

Similar to the previous section, the figure below shows an experimental method for representation of restrictions in the model. The figure below shows a representation of the rules for the exposure of Connectors on an Equipment accounting for both the FRU/Non-FRU differences and also for the differences between Connectors related to a Holder versus visible Connectors available to connect cables to.

Essentially the figure shows that Connectors related to Holders have different lifecycles and visibilities than Connectors related to cables.



CoreModel diagram: Equipment-ConnectorRules

Figure 3-9 Connector rules

3.6.1 ConectorInHolder

Qualified Name: CoreModel::CorePhysicalModel-Initial::RuleModels::ConnectorRules::ObjectClasses::ConectorInHolder

A rule class (an abstract specialization of Connector) that represents a connector that are only accessible to an equipment inserted in the holder.

This class is abstract.

Inherits properties from:

- Connector

This class is Experimental.

3.6.2 ConnectorCableEnd

Qualified Name: CoreModel::CorePhysicalModel-Initial::RuleModels::ConnectorRules::ObjectClasses::ConnectorCableEnd

A rule class (an abstract specialization of Connector) that represents a connector on the end of a cable.

This class is abstract.

Inherits properties from:

- Connector

This class is Experimental.

3.6.3 ConnectorOnEquipmentForCable

Qualified Name: CoreModel::CorePhysicalModel-Initial::RuleModels::ConnectorRules::ObjectClasses::ConnectorOnEquipmentForCable

A rule class (an abstract specialization of Connector) that represents a connector exposed on an equipment such that a cable may be plugged in.

This class is abstract.

Inherits properties from:

- Connector

This class is Experimental.

3.6.4 ConnectorOnEquipmentForHolder

Qualified Name: CoreModel::CorePhysicalModel-Initial::RuleModels::ConnectorRules::ObjectClasses::ConnectorOnEquipmentForHolder

A rule class (an abstract specialization of Connector) that represents a connector on an equipment that is intended to mate with a connector in a holder.

This class is abstract.

Inherits properties from:

- Connector

This class is Experimental.

3.7 Expected and Actual

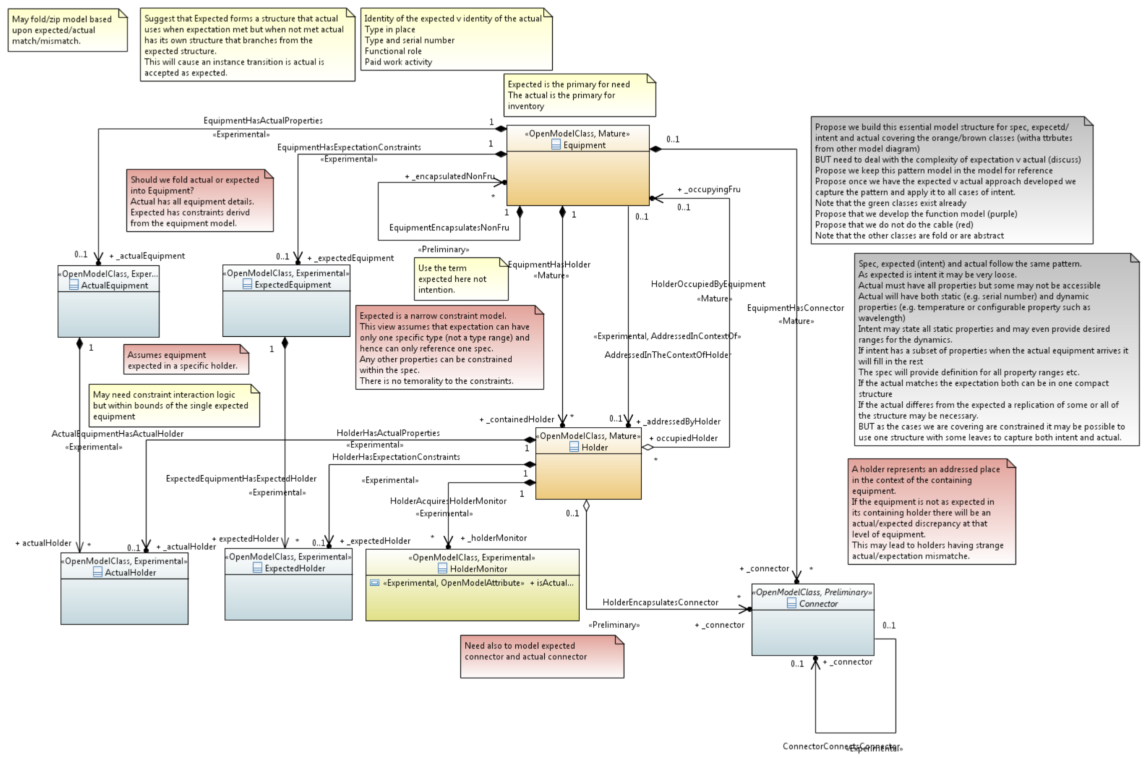
This model fragment explores a representation of expected and actual where the Equipment pattern is augmented with composed parts representing the expected and actual settings.

The ExpectedEquipment is likely to be in terms of constraints, with many "don't care" values (such as serial number) but is assumed to have a single precise definition of type and of position with respect to the Equipment/Holder it is to be contained in[3].

[3] In a client-provider context Expectation is the client view of an agreement that the provider intends to satisfy. As it is not possible to remotely install a physical thing there is a decoupling between the request and the actual realization. This is very similar to the decoupling at a contractual interface where there is an agreement to provide something and hence an intention by the provider and an expectation from the client. The agreement is satisfied once the provider provides what was agreed. In the general case any properties could be constraints. In the case of the Equipment the agreement is more precise providing no choice of positioning. It is possible under some circumstances that position flexibility may also be allowed. This aspect is for further study.

The assumption is that when there is a mismatch, the expected Equipment will have a set of expected Holders and the actual Equipment a potentially overlapping set of actual Holders. The hierarchy is driven by Holder position.

Where the Equipment is stand-alone (not in a Holder) it is assumed that geographical location or other statement of place will enable detection of expectation/actual mismatch. Hence stand-alone Equipment can also have an expected and actual value. Also see section 4.3 Actual v expected on page 43 for a pictorial example of work in progress.



CoreModel diagram: Equipment-ExpectedAndActual

Figure 3-10 Expected and actual

3.7.1 ActualEquipment

Qualified Name: CoreModel::CorePhysicalModel-Initial::ExpectedAndActual::ObjectClasses::ActualEquipment

The equipment that is actually present in the physical network.
It will expose all dynamic properties and some critical static properties.

This class is Experimental.

3.7.2 ActualHolder

Qualified Name: CoreModel::CorePhysicalModel-Initial::ExpectedAndActual::ObjectClasses::ActualHolder

A holder in the ActualEquipment.

This class is Experimental.

3.7.3 ExpectedEquipment

Qualified Name: CoreModel::CorePhysicalModel-Initial::ExpectedAndActual::ObjectClasses::ExpectedEquipment

A definition of the restrictions on the equipment that is expected to be present in the physical network at a particular "place".
The expected equipment will state the type and may constrain any other invariant properties.
It may also provide desired ranges for dynamic properties.

This class is Experimental.

3.7.4 ExpectedHolder

Qualified Name: CoreModel::CorePhysicalModel-Initial::ExpectedAndActual::ObjectClasses::ExpectedHolder

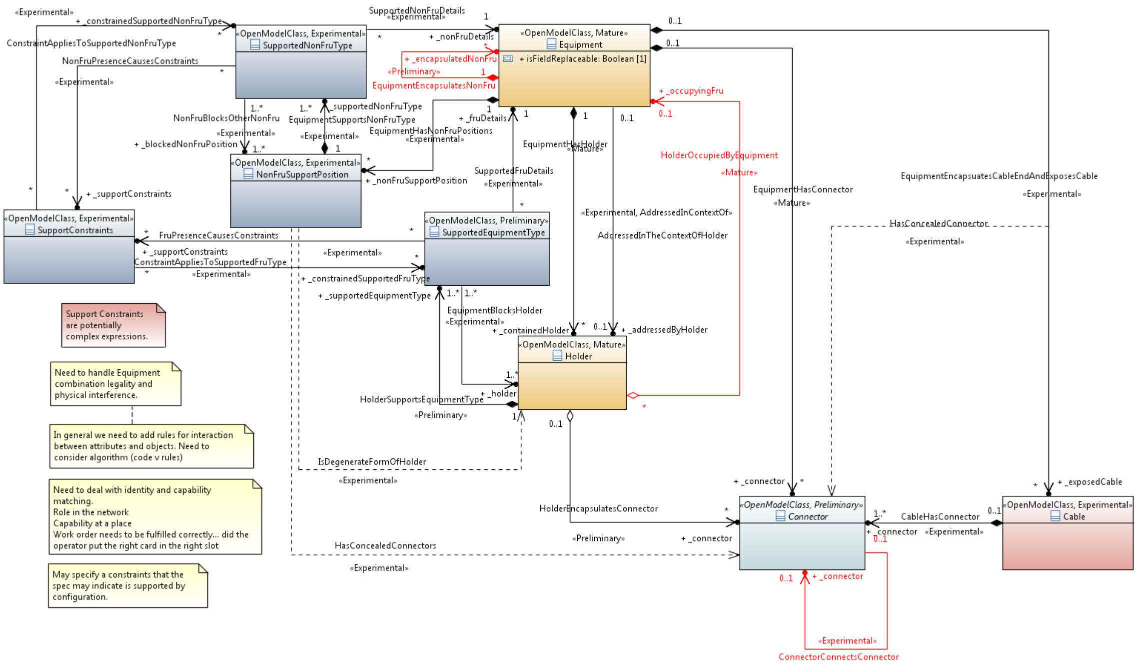
A definition of a holder expected in the ActualEquipment (i.e. an ActualHolder) as part of the constraints provided by the ExpectedEquipment.

This class is Experimental.

3.8 Specification

The figure below provides a fragment of a candidate specification model for Equipment, focusing on Holder compatibility and supported non-FRUs.

Note that the EquipmentEncapsulatesNonFru association and the HolderOccupiedByEquipment association (both shown in red) are essentially governed by the rules stated in the supportedNonFruType class and SupportedEquipmentType class (and their associated classes) respectively.



CoreModel diagram: Equipment-ConstraintsOnEquipmentPattern

Figure 3-11 Specification

3.8.1 NonFruSupportPosition

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentSpecification::ObjectClasses::NonFruSupportPosition

Equivalent to the holder for the FRU, represents in the specification a place where one or more types of non-FRU could be present.
Unlike the holder what is present is fixed whilst the equipment is in the field.

This class is Experimental.

3.8.2 SupportConstraints

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentSpecification::ObjectClasses::SupportConstraints

Rules related to how both non-FRU and FRU presence restricts the potential for additional equipments to be installed.
An FRU type installed in one holder may limit the FRU types that can be installed in another holder etc.

This class is Experimental.

3.8.3 SupportedEquipmentType

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentSpecification::ObjectClasses::SupportedEquipmentType

The FRU equipment types supported by the holder.

This class is Preliminary.

3.8.4 SupportedNonFruType

Qualified Name: CoreModel::CorePhysicalModel-Initial::EquipmentSpecification::ObjectClasses::SupportedNonFruType

The non-FRU equipment types supported by the non-FRU support position.

This class is Experimental.

3.9 Physical Connector and conceptual Port

The Connector and Port are modeled as distinct abstract entities. It is likely that in both cases only the "name" is required in an instance realization. Both the Port and Connector represent fixed rules of grouping and these rules could be completely contained in the spec. The Connector also has physical properties but again these are fixed and could be contained in the spec.

An LTP instance provides a Port reference. The Port represents a PinGroup that relates to the Connectors via Pins as described in the appropriate spec. The Connector is part of an Equipment. An Equipment instance references a spec that identifies the Connectors.

There are many potential arrangements of association of LTP to Port/Connector. The following sequence of figures provides a view of some of the variety.

As noted previously the Port is a "virtual" concept related to some coherent traffic flow. As can be seen from the sequence of figures there is no fixed relationship from port to Connector/pin.

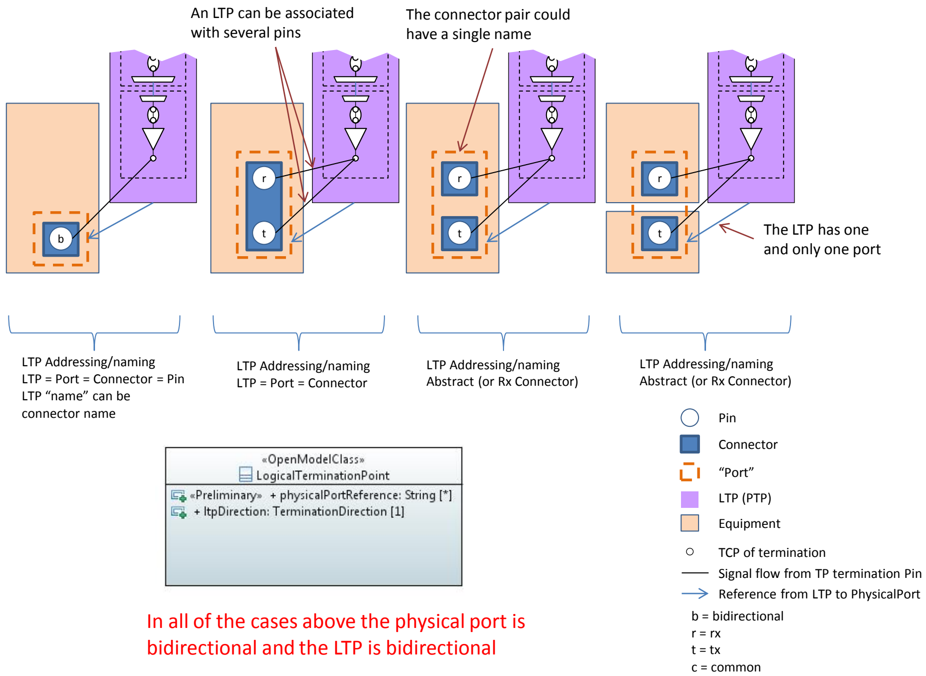


Figure 3-12 Basic cases of Physical Port Reference

The figure above shows some simple LTP – Port – Pin – Connector cases. The diagram case on the left could be where a single fiber is being used to convey a bidirectional signal (a coupler/splitter is within the Equipment) and hence only a single pin is required on a single Connector. In this case the Connector, Pin, Port and LTP all have multiplicity [1]. The diagram case on the right could be again an optical case (with one pin per Connector) where the LTP is being considered as bidirectional but there are separate dedicated Equipments for each direction of traffic.

The pin of the Connector is essentially omni-directional (i.e. the media has no directional limitation and the material would allow the photons/electrons etc to propagate in any direction – see documentation on media in TR-512.2 and TR-512.A.4). The construction of the Connector (and the associated strand of material) does in some cases provide such a narrow channel that the propagation is limited to essentially two directions but even in these cases the material is essentially omni-directional. The directionality designation associated with a Connector is "inherited" from the directionality of the termination function attached. In the figure above/below the rx and tx designations result from the attached terminations.

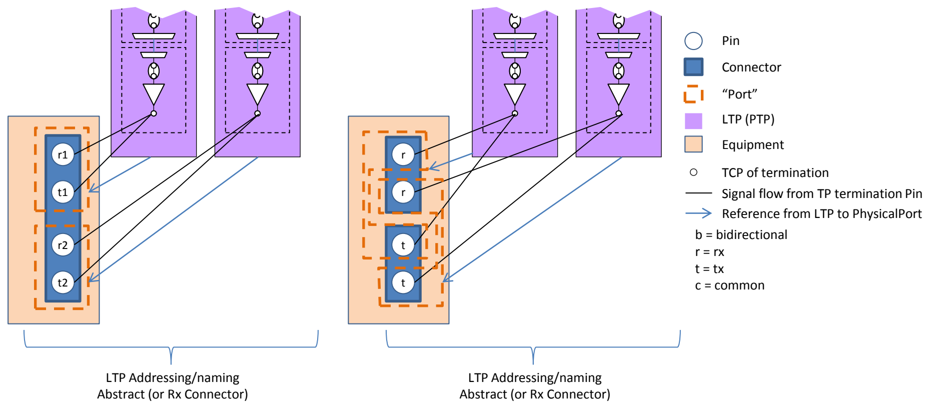


Figure 3-13 More Complex cases of intertwined Connectors

The diagram on the left in the figure above could represent a case where there is ribbon cable with multiple (in this case four) fibers terminated with one Connector and where each fiber is being used for only one direction of signal (but the fiber is inherently omni-directional as noted above). The two LTPs shown are both bidirectional and hence use two pins each. As the signal is bidirectional in nature the Port is also bidirectional

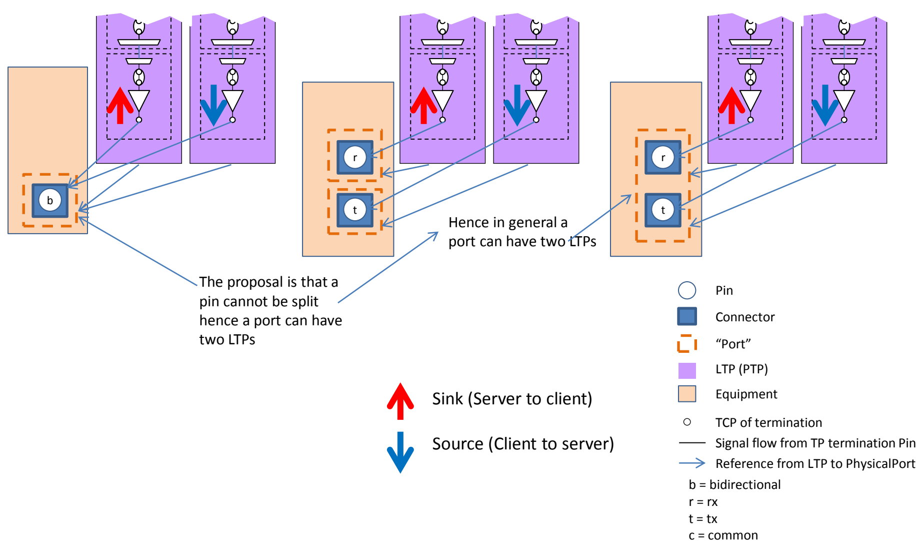


Figure 3-14 Unidirectional Cases

Other cases to consider:

- Multiple LTPs per port (as shown in the left and right cases in the diagram above)
- Multiple ports per pin

4 Work in progress (see also TR-512.FE)

4.1 Addressing

Traditionally ports have been identified using addressing schemes based on physical positioning, however there are challenges related to the complexity of potential spread across Equipments and ports as discussed in the previous section. The figure below discusses some cases. This work needs to be taken further.

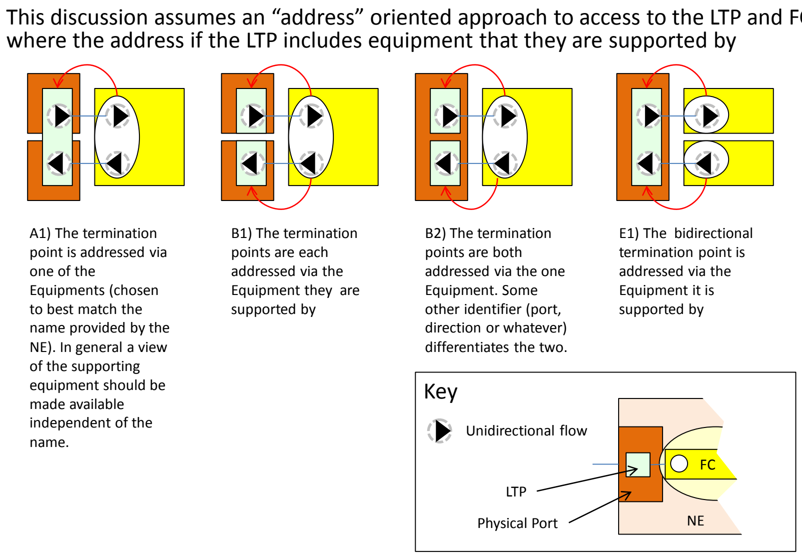


Figure 4-1 Connector/Port based addressing of LTPs

4.2 Physical to functional model

The figures in this section highlight aspects of the physical to functional relationship. The physical to functional model has been enhanced in V1.3 as a result of the addition of the Processing Construct (see TR-512.11). As a result the cases set out below can now be supported with the model. The specific examples have not yet been included in the detailed documentation. As a consequence this section has not yet been removed.

The figure below shows a simplified hierarchy of functions emerging from running hardware where the emergent behavior considered as "function services". The figure is a rough sketch.

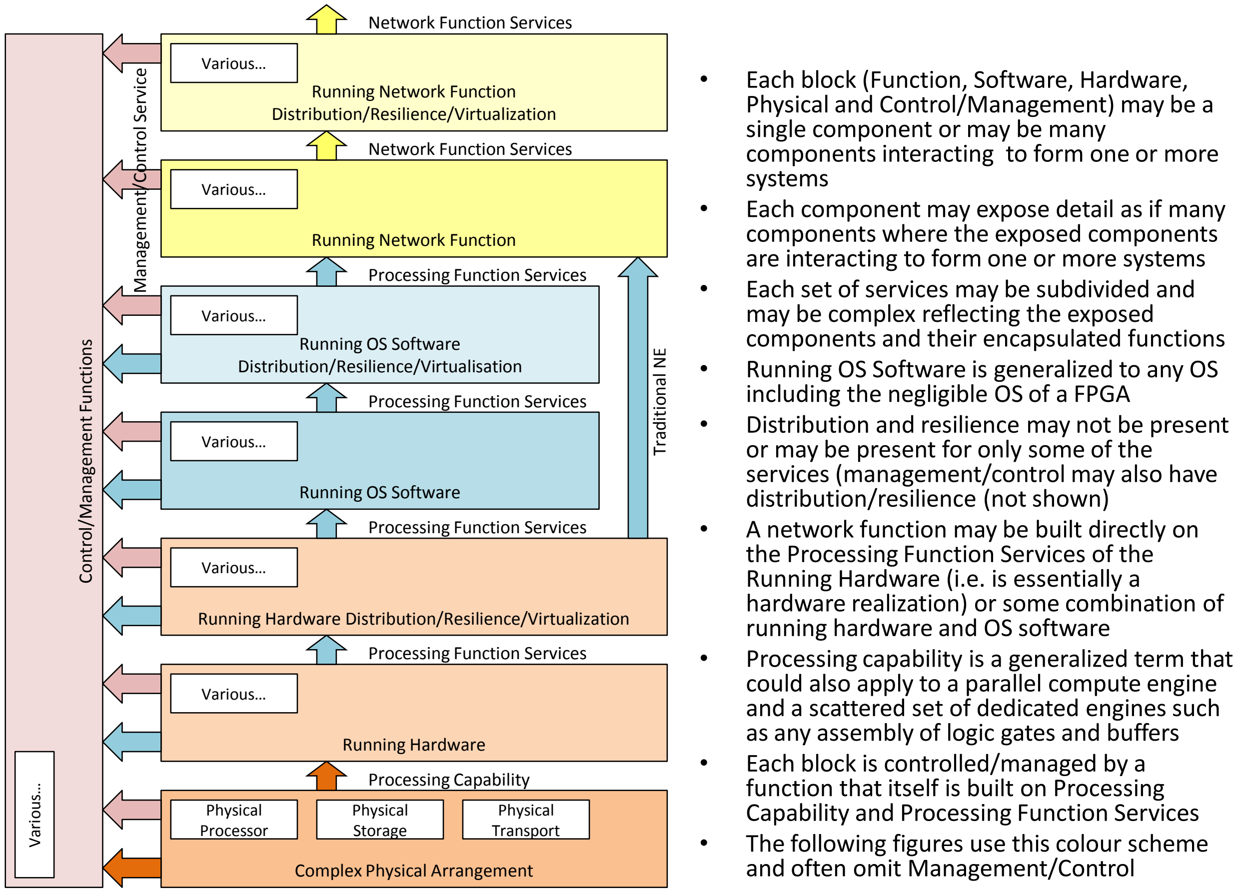


Figure 4-2 Simplified sketch of physical to functional

The figure below shows, very roughly, the emergence of various aggregate functions from functional blocks on two FRUs. In the figure:

- The FRUs are at the bottom (in brown)
- The inherent capabilities of the FRUs are shown as dotted boxes in the FRUs
- The emergent functional blocks are shown in green as are the aggregate functions
- A somewhat abbreviated progression from functional block to aggregate function via atomic function (not shown) to LTP (in blue at the top of the page) is shown
- The yellow bars represent different forms of protection essentially as FCs
- The functions in the yellow dotted shape are essentially virtual functions as the position of realization of the function is not fixed (and need not be known)

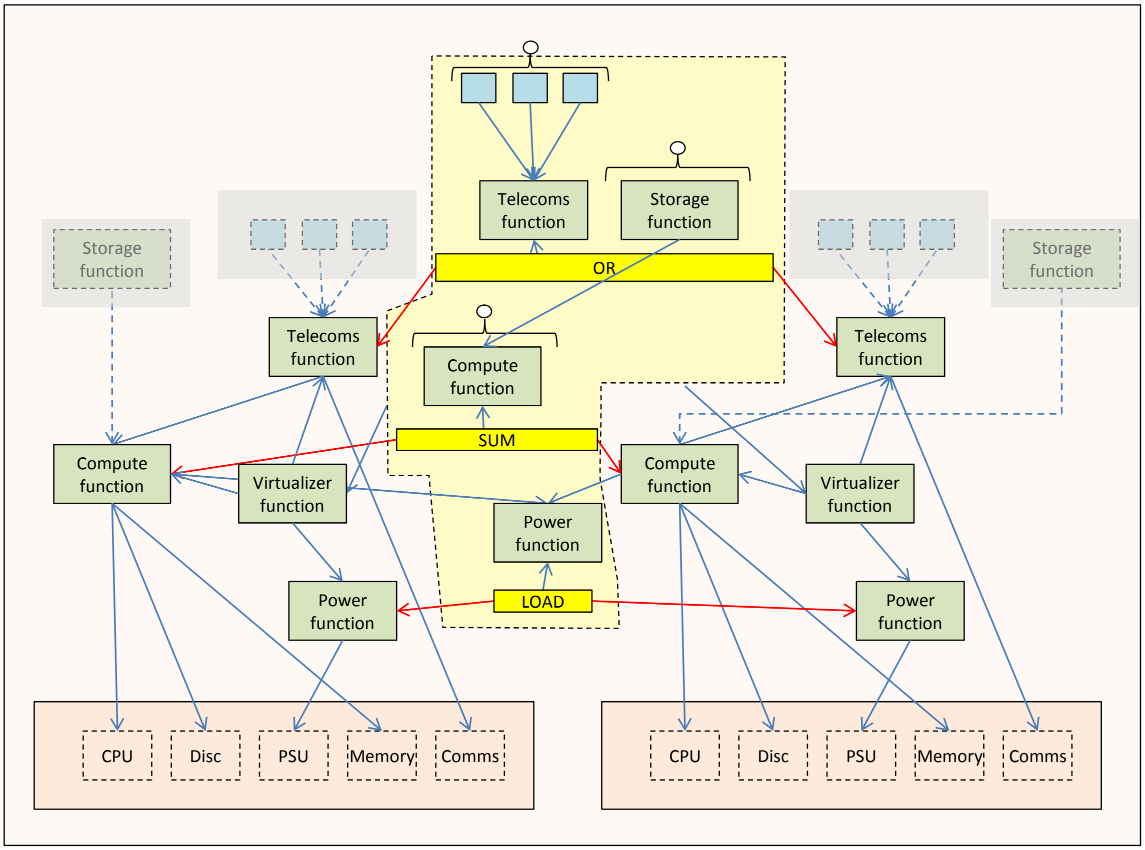


Figure 4-3 Simplified sketch of forms of resilience in an NE

4.3 Actual v expected

The figure below shows various cases of actual-expectation mismatch.

- Slot 1
  - Expectation: Double slot card with subslots (Holders) 1-3 (not numbered on diagram) where 1 and 2 are expected to be equipped
  - Actual: Double slot card of different type with subslots A & B where both are actually equipped
  - Model: One Equipment object in slot 1 with 5 subslots with 4 containing Equipment (two expected only and two actual only)
- Slot 2
  - Expectation: Blocked by expectation in slot 1
  - Actual: Blocked by actual in slot 1
  - Model: No Equipment
- Slot 3
  - Expectation: Single slot card with subslots 1-3 where 1 and 2 are expected to be equipped

    - Actual: Single slot card with subslots 1-3 where 1 is equipped not matching expectation, 2 is not equipped (not matching expectation) and 3 is unexpectedly equipped
    - Model: One Equipment object in slot 3 with 3 subslots with each containing an Equipment
- Slot 4
    - Expectation: Double slot card with subslots 1-3 (not numbered) where 1 and 2 are expected to be equipped
    - Actual: Single slot card of different type with subslots 1-5 where 1 is equipped with the wrong card type, 2 is not equipped 3 is not equipped as expected etc
    - Model: One Equipment object in slot with 5 subslots with 3 containing Equipments
- Slot 5
    - Expectation: Blocked by expectation in slot 4
    - Actual: Double slot card with subslots A & B where A is actually equipped
    - Model: One Equipment object in slot with 2 subslots with 1 containing an Equipment
- etc

.

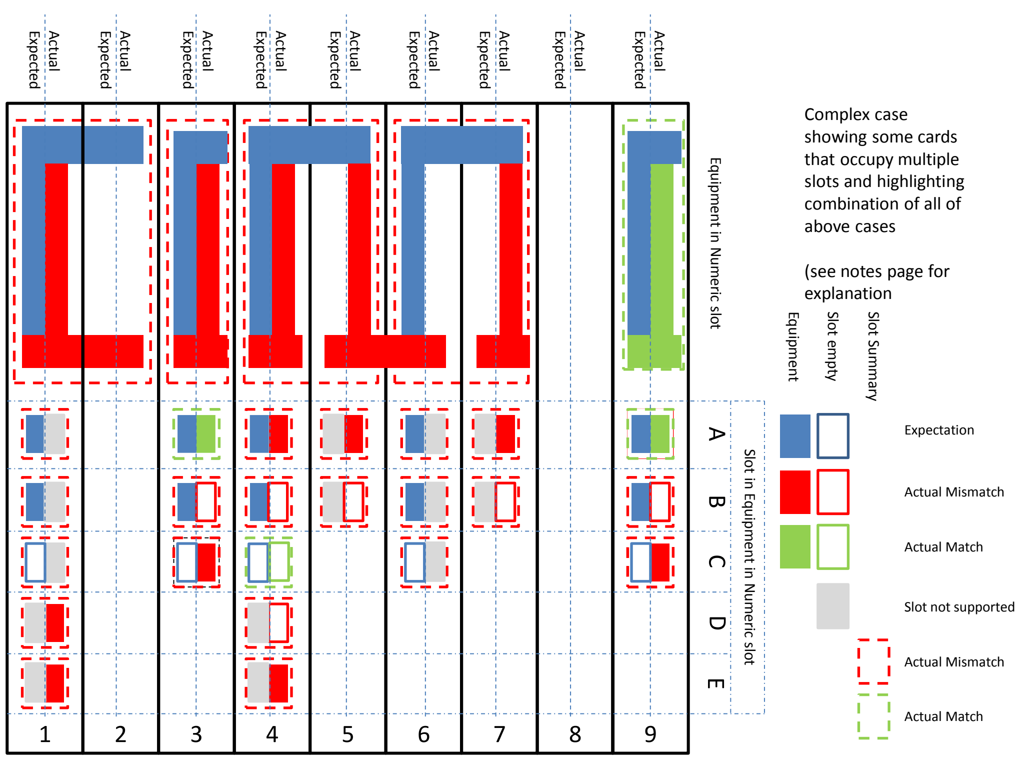


Figure 4-4 Expectation v actual showing mismatch and blocking

This figure above shows a representation of a SUBRACK (Equipment) with 9 SLOTS (Holders). Each Holder has a dotted line down the middle where the symbols in the slot to the left of the dotted line represents represent equipping expectation and to the right represent actual equipping. The upper large colored blocks represent the CIRCUIT_PACKs in the SLOTs (SLOT 8 is empty). The lower small colored rectangles represent the SLOTs in the CIRCUIT_PACKs (A – E) and the SMALL_FORMFACTOR_PLUGGABLEs (Equipment) in these SLOTS.

Considering the slots in turn

- SLOT 9 has a CIRCUIT_PACK that matches the expectation (hence green actual column). It also has:
    - An SFP in slot A that matches expectation
    - Unexpectedly no SFP in slot B hence a mismatch
    - Unexpectedly an SFP in slot C hence the mismatch

Further explanation to be added in a later release.

The figure requires further development.

End of document