OPEN NETWORKING FOUNDATION

Core Information Model (CoreModel)

TR-512.6

Physical

Version 1.2
September 20, 2016

OpenFlow™

ONF Document Type: Technical Recommendation
ONF Document Name: Core Information Model version 1.2

Disclaimer

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

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).
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 Introduction

This document is an addendum to the TR-512_v1.2 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 ONF Core IM - Overview.

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:

- The Physical model is experimental at this stage. However, it was considered important 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 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 equipment model

The physical model relates to the Core Network Model related to Termination and Forwarding described in TR-512.2 ONF Core IM - Forwarding and Termination and to Topology TR-512.4 ONF Core IM - Topology. Physical capability and other specification considerations are covered in TR-512.7 ONF Core IM - Specification.

A data dictionary that sets out the details of all classes, data types and attributes is also provided (TR-512.8).

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

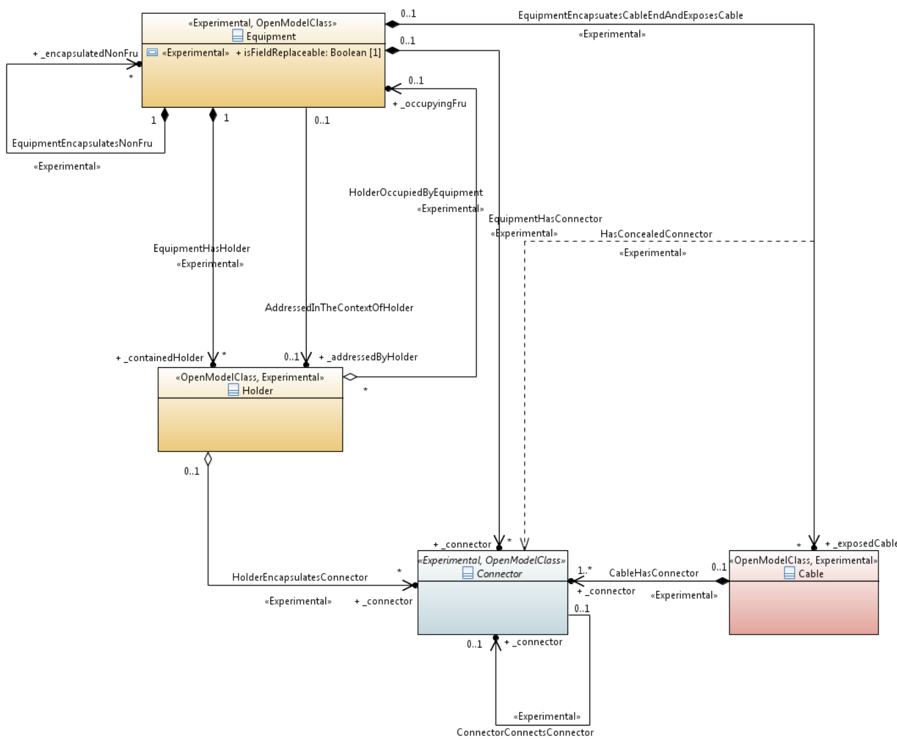
- Field Replaceable Unit (FRU):
    - Can be replaced in the field. May plug in to a holder in another equipment (if not stand-alone)

[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 functional representations are considered as abstractions and as, to some degree, virtualized. Mechanical functional things, such as fans, are not modelled in detail.

- 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 (an 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 21.

The classes of the model is 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

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 Experimental.

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 Experimental.

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).

This class is abstract.

Inherits properties from:

- LocalClass
- GroupOfPins

This class is Experimental.

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.

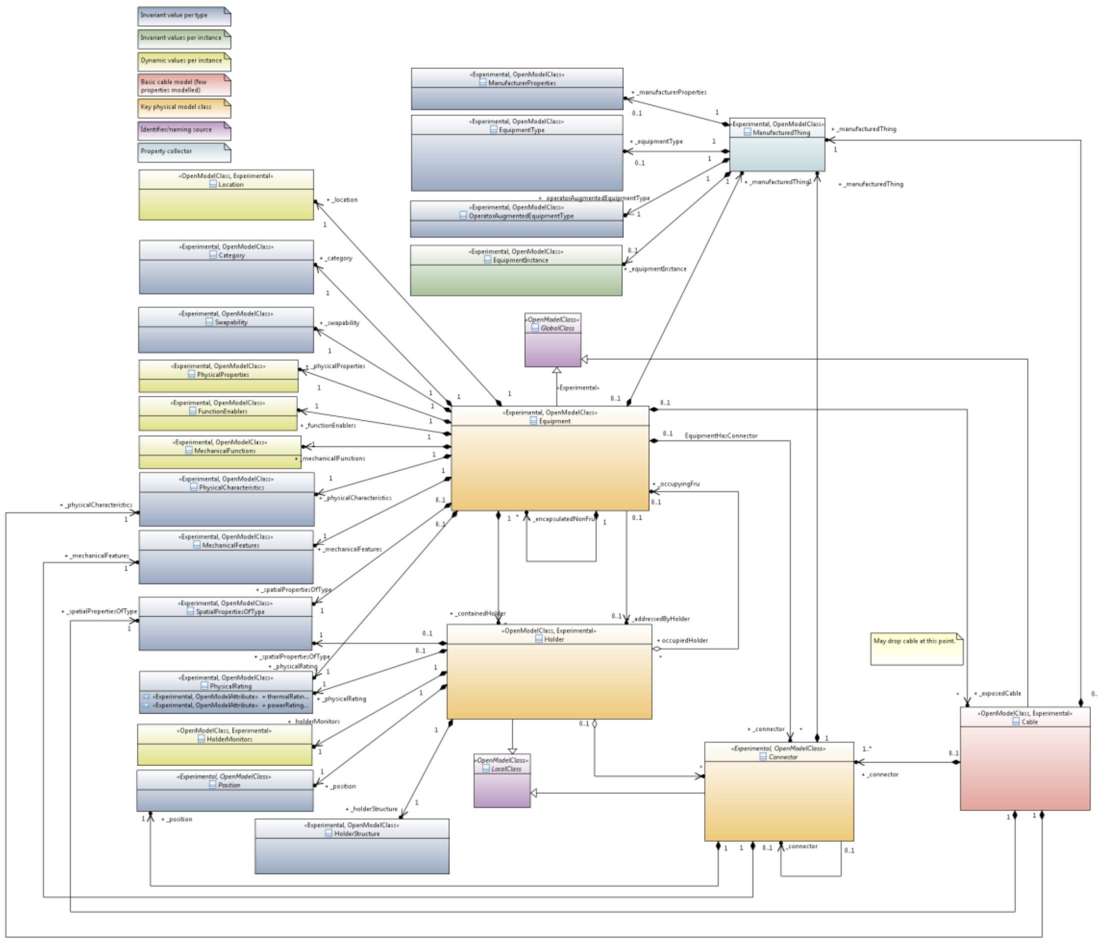
Inherits properties from:

- GlobalClass

This class is Experimental.

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 ONF Core IM - Specification]() for more information)



CoreModel diagram: Equipment-DetailWithoutAttributes

Figure 3-2 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 equipments of the same type.

3.2.1.1 Category

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

Represents the form of the equipment.

This class is Experimental.

Table 1: Attributes for Category

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description

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

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 2: Attributes for EquipmentInstance

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
manufactureDate	Experimental	To be provided
serialNumber	Experimental	To be provided
assetInstanceIdentifier	Experimental	To be provided

3.2.1.3 EquipmentType

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

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

This class is Experimental.

Table 3: 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

3.2.1.4 HolderStructure

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

Represents the form of the holder.

This class is Experimental.

Table 4: Attributes for HolderStructure

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

3.2.1.5 ManufacturedThing

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

Collects all invariant aspects of a manufactured thing.

This class is Experimental.

Table 5: 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

3.2.1.6 ManufacturerProperties

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

Represents the properties of the manufacturer.

This class is Experimental.

Table 6: Attributes for ManufacturerProperties

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

3.2.1.7 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.

3.2.1.8 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 characterise the type.

This class is Experimental.

Table 7: Attributes for OperatorAugmentedEquipmentType

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

3.2.1.9 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 8: 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

3.2.1.10 PhysicalRating

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

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

This class is Experimental.

Table 9: Attributes for PhysicalRating

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

3.2.1.11 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 10: Attributes for Position

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

3.2.1.12 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 11: 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

3.2.1.13 Swapability

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

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

This class is Experimental.

Table 12: Attributes for Swapability

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 13: Attributes for FunctionEnablers

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

3.2.2.2 HolderMonitors

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

Represents the dynamic state of the holder instance.

This class is Experimental.

Table 14: Attributes for HolderMonitors

Attribute Name	Lifecycle Stereotype (empty = Mature)	Description

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

This class is Experimental.

Table 15: 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 16: 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 17: Attributes for PhysicalProperties

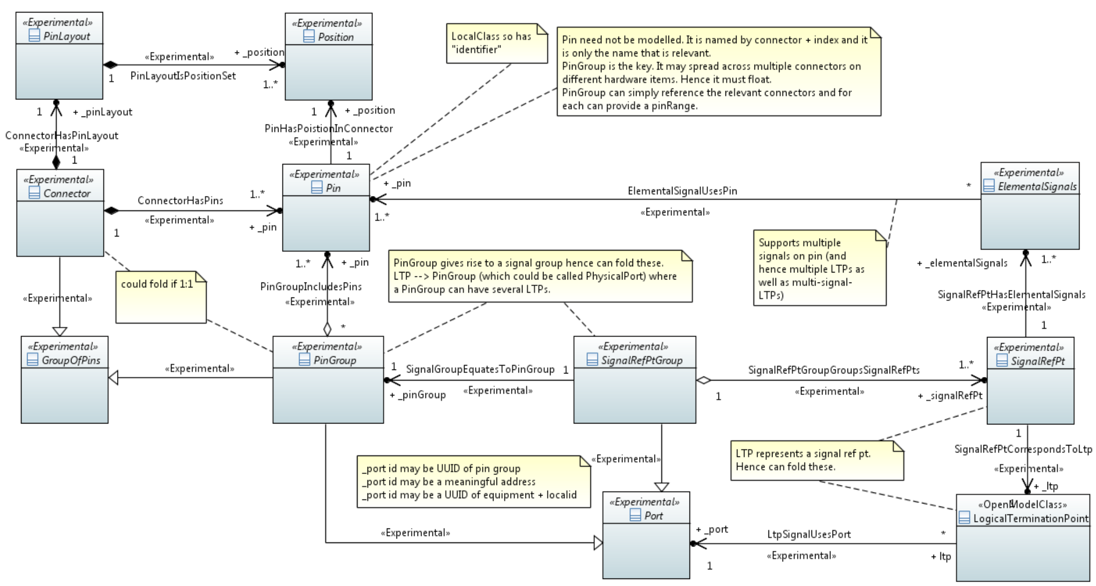
Attribute Name	Lifecycle Stereotype (empty = Mature)	Description
temperature	Experimental	To be provided

3.3 Connector to LTP sketch

The figure below is a sketch model relating connector and LTP. The connector is first decomposed into pins then the pins are then grouped across potentially multiple connectors with a granularity that corresponds to a 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 of the signals would be lost, and such that the signals are indivisible when they flow through the pins.

Following the figure there are definitions of each of the new classes not covered in this document in previous sections. See section 3.9 Physical Connector and conceptual Port 28 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 3-3 Connector to LTP

3.3.1 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.

3.3.2 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.

3.3.3 Pin

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

An individual physical connection point (male or female). May be capable of carrying electrical or optical signals. A pin may have more than one wire/fiber attached but is such that all attached things get exactly the same signal set.

This class is abstract.

This class is Experimental.

3.3.4 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:
- GroupOfPins
- Port

This class is Experimental.

3.3.5 PinLayout

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

The structuring of pins in a connector.

This class is abstract.

This class is Experimental.

3.3.6 Port

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

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

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:

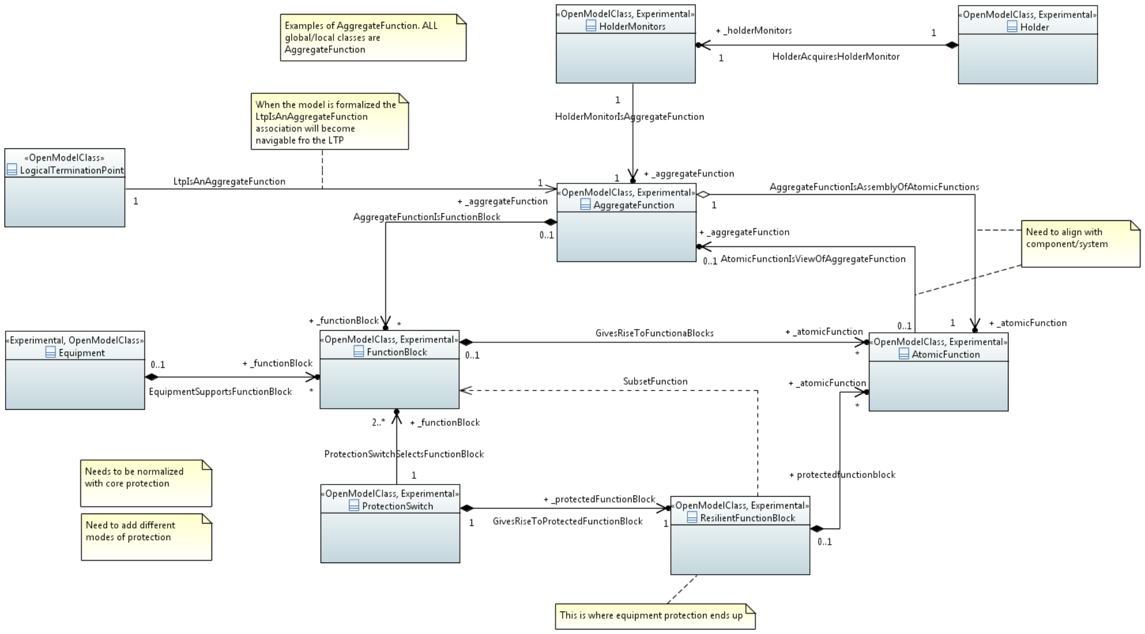
- Port

This class is Experimental.

3.4 Equipment to Function Sketch

The figure below is a sketch model relating Equipment to Function. 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 "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 30.

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



CoreModel diagram: Equipment-EquipmentToFunction

Figure 3-4 Equipment to Function

3.4.1 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 Experimental.

3.4.2 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 Experimental.

3.4.3 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 Experimental.

3.4.4 ProtectionSwitch

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

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 Experimental.

3.4.5 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 Experimental.

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.

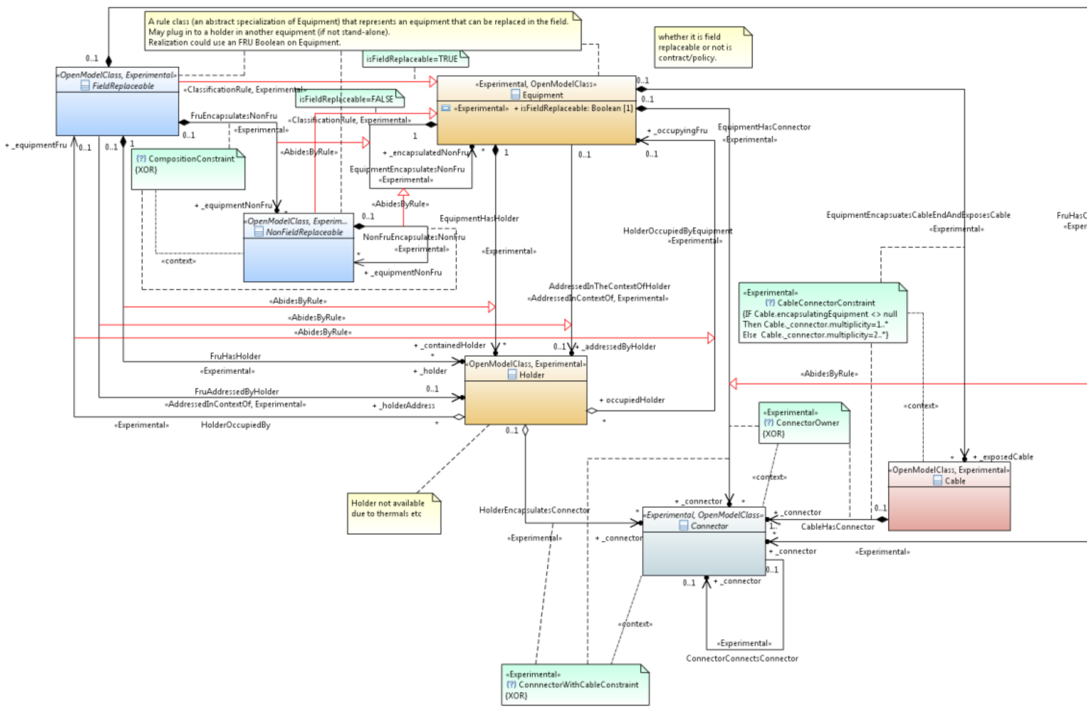
Considering the distinction between FRU and nonFRU there were two choices, model the FRU explicitly (and then expect instantiation of instances of distinct FRUs and non-FRUs classes) or model the FRU and non-FRU as Equipment with solely an attribute to indicate the case. The attribute approach was prefered 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 3-5 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 (an associated connectors are assumed to belong to the containing FRU).

This class is abstract.

Inherits properties from:

- Equipment

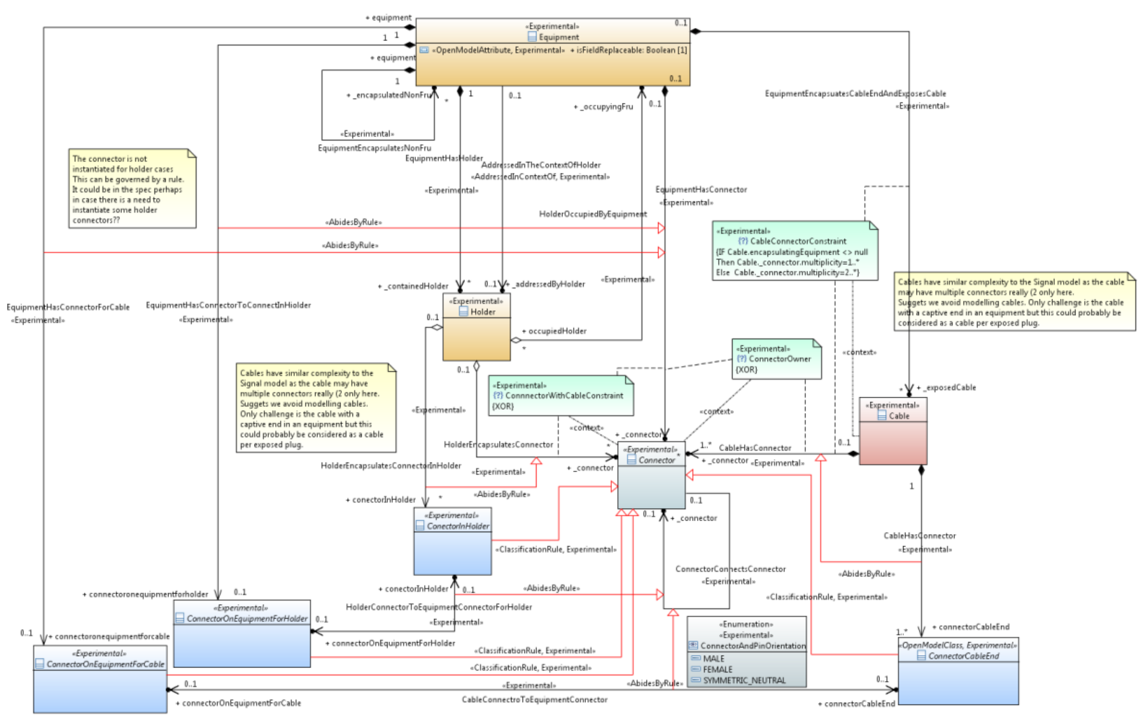
This class is Experimental.

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 v 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 3-6 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.

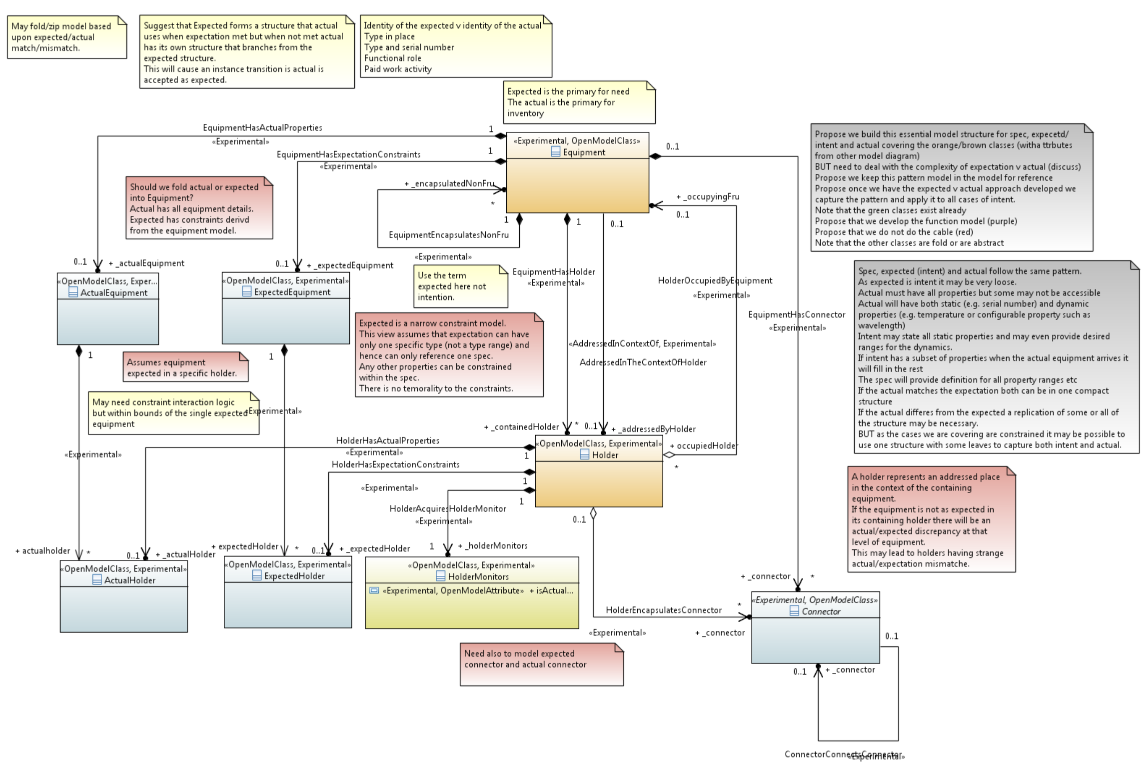
3.7 Expected and Actual

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.

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 32 for a pictorial example of work in progress.



CoreModel diagram: Equipment-ExpectedAndActual

Figure 3-7 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 dynami properties.

This class is Experimental.

3.7.4 ExpectedHolder

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

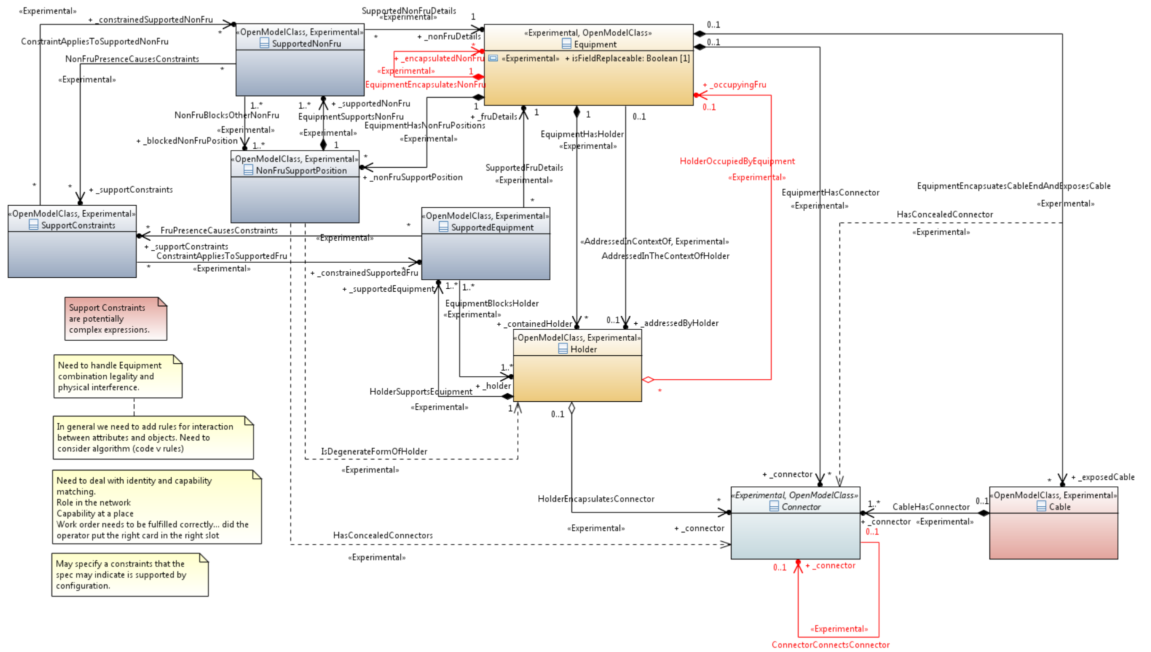
A definition of a holder expected in the ActualEquipment (i.e. an ActualHolder) as part of the constraints provided by the ExpectedEquipment.

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 supportedNonFru class and SupportedEquipment class (and their associated classes) respectively.



CoreModel diagram: Equipment-ConstraintsOnEquipmentPattern

Figure 3-8 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 whilt 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 additonal 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 SupportedEquipment

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

The FRU equipment types supported by the holder.

This class is Experimental.

3.8.4 SupportedNonFru

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

The non-FRU equipment types supported by the a 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.

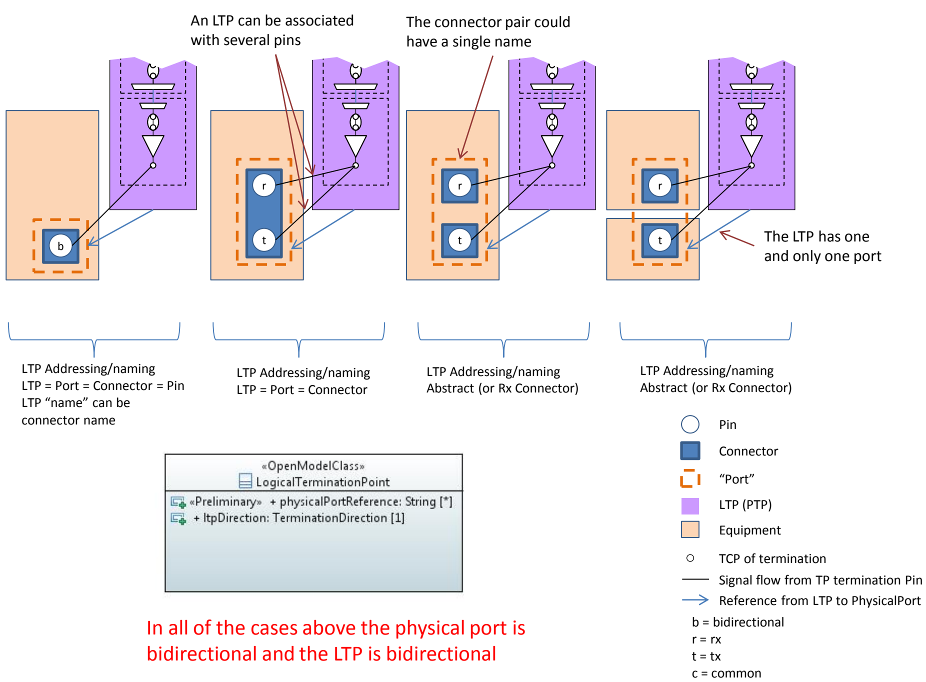


Figure 3-9 Basic cases of Physical Port Reference

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 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.

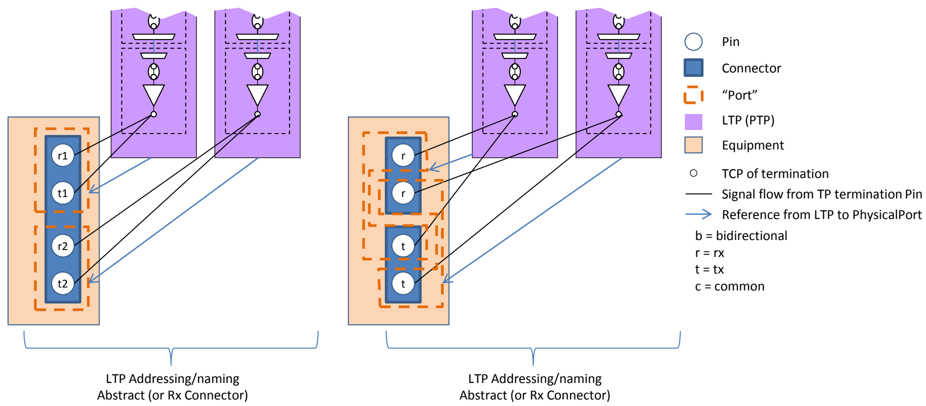


Figure 3-10 More Complex cases of intertwined connectors

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 on direction of signal. 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

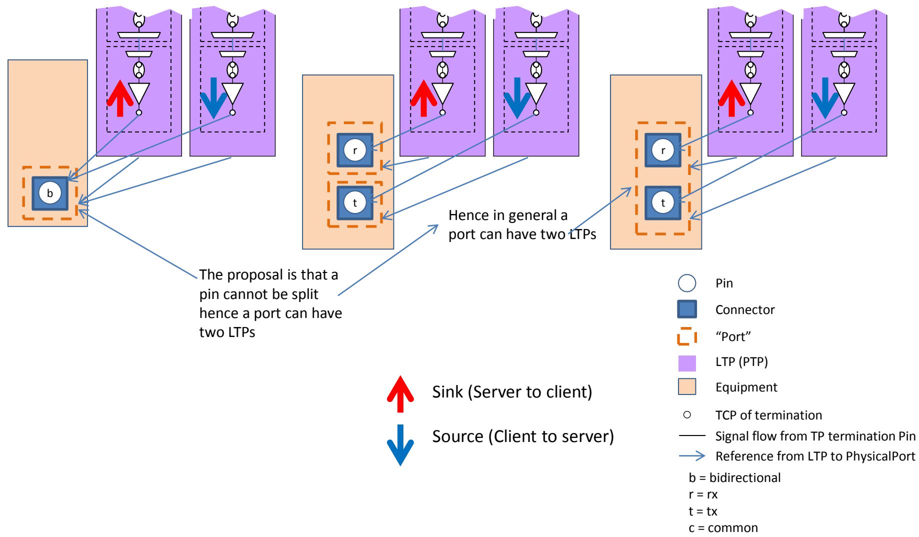


Figure 3-11 Unidirectional Cases

Other cases to consider:

- Multiple LTPs per port
- Multiple ports per pin

4 Work in progress related to the physical model

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.

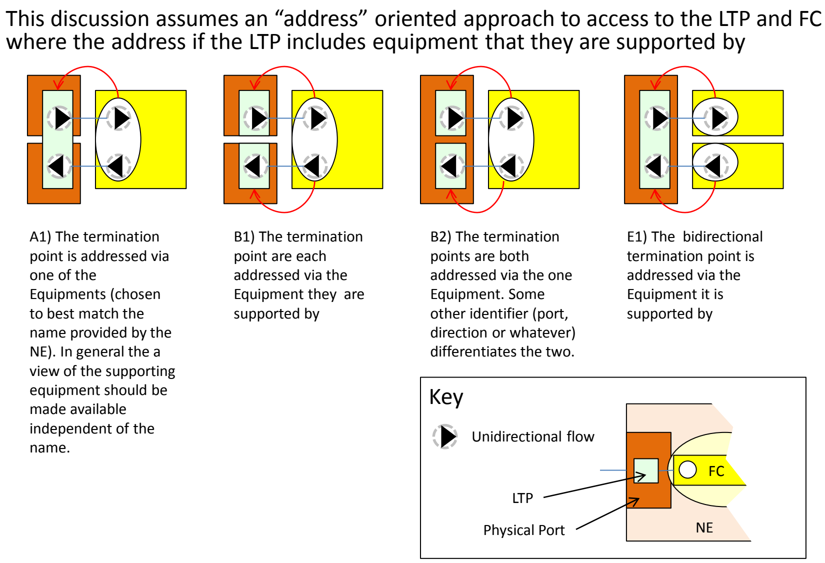
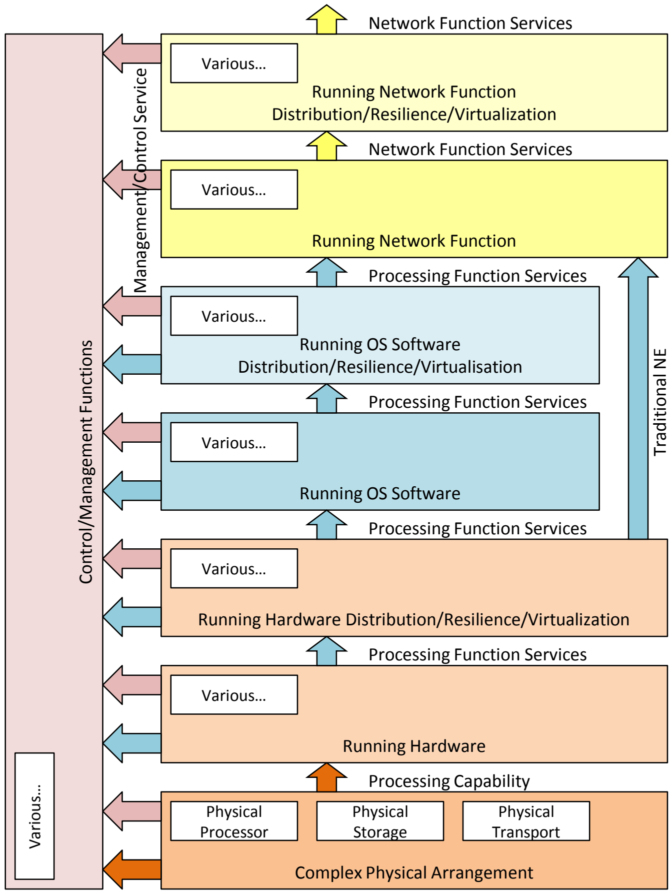


Figure 4-1 Connector/Port based addressing of LTPs

4.2 Physical to functional model

The figures in this section highlight aspects of the physical to functional relationship.

The 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.



- Each block (Function, Software, Hardware, Physical and Control/Management) may be a single component or may be many components interacting to form one or more systems
- Each component may expose detail as if many components where the exposed components are interacting to form one or more systems
- Each set of services may be subdivided and may be complex reflecting the exposed components and their encapsulated functions
- Running OS Software is generalized to any OS including the negligible OS of a FPGA
- Distribution and resilience may not be present or may be present for only some of the services (management/control may also have distribution/resilience (not shown)
- A network function may be built directly on the Processing Function Services of the Running Hardware (i.e. is essentially a hardware realization) or some combination of running hardware and OS software
- Processing capability is a generalized term that could also apply to a parallel compute engine and a scattered set of dedicated engines such as any assembly of logic gates and buffers
- Each block is controlled/managed by a function that itself is built on Processing Capability and Processing Function Services
- The following figures use this colour scheme and often omit Management/Control

Figure 4-2 Simplified sketch of physical to functional

The figure 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)

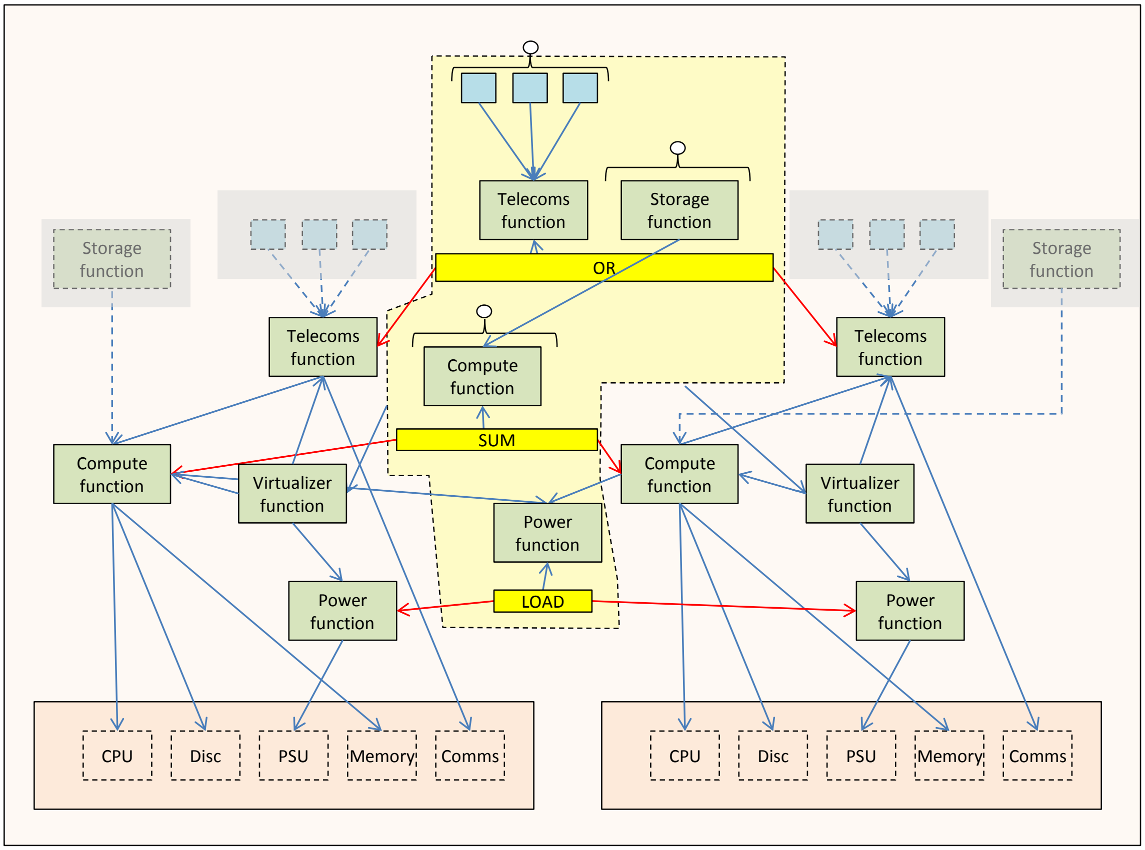


Figure 4-3 Simplified sketch of forms of resilience in an NE

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

.

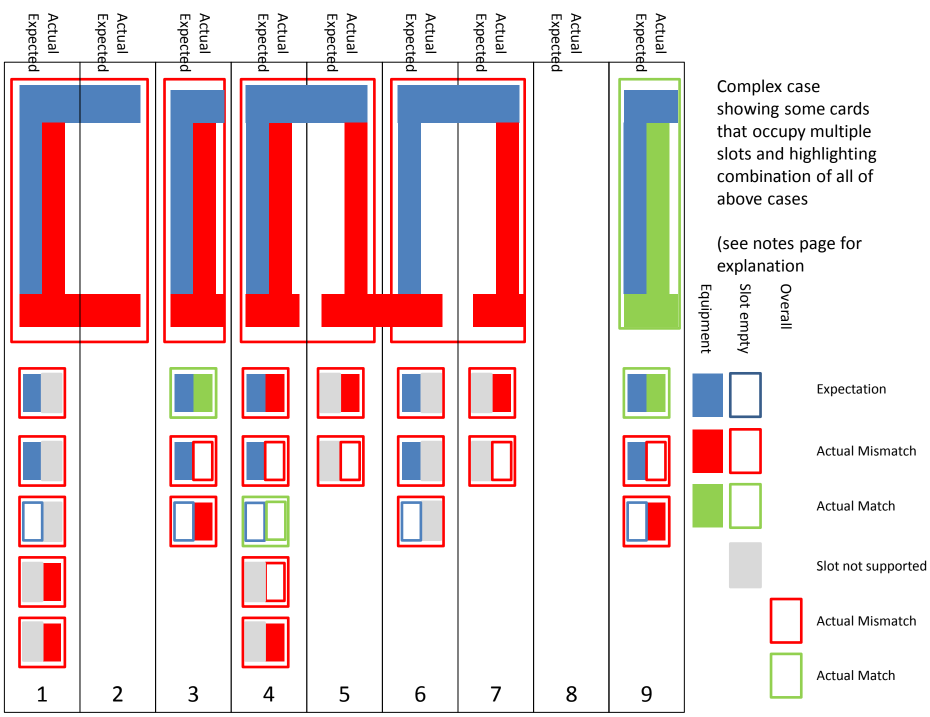


Figure 4-4 Expectation v actual showing mismatch and blocking

This figure requires further development.

End of document

