# Introduction

Handling the physical equipment for the WireInterface is a bit complex.

The Capability information of the WireInterface instance depends on the type of SFP (SFP shall represent all types of pluggables in this text), which shall carry the WireInterface. The concrete SFP might be replaced due to failure, and the replacing SFP might just be similar, but not of identical type. Nevertheless the instantiated WireInterface shall continue existing and also operating, in case original and replacing SFP have sufficiently identical characteristics. On the other hand, operating WireInterface instances on SFPs with different characteristics shall be strictly prevented.

The situation is additionally complicated by the fact that the vendor doesn’t know all the SFP types, which will be operated with the Device, during design time of its management interface.

# Operational Requirements

Data modeling and processing has to comply with many operational requirements, e.g. it must be possible …

… to just plug an SFP and the WireInterface comes up for configuration

… to upfront plan a WireInterface including its configuration and it gets automatically operative as soon as the necessary hardware gets plugged

… to replace a broken SFP without deleting the existing WireInterface configuration on the device

… to restrict the types of SFP, which get operative in the network, to a device type specific list of approved SFP. This list shall be updatable on the Device from remote

# Terms

Several sets of information about WireEquipment are required to support the processes, which are required for serving these operational requirements:

|  |  |
| --- | --- |
| List\_of\_compatible\_SFPs | List of SFPs, which can be operated with the Device and are known to the vendor during design time of the management interface. This list is to be provided to the operator and to be stored inside the device. |
| List\_of\_approved\_SFPs | List of SFPs, which are allowed to be operated with the Device in the operator’s network. This list is just available on the application layer and a subset of similar SFPs (SFP\_Kind\_Group) has to be configured into the Device as ExpectedEquipment. |
| SFP\_Kind\_Group | Group of SFPs, which have identical values in all relevant Capability attributes. |
| ActualEquipmentCharacteristics | Describe the concrete SFP, which is currently plugged into the Device. The ActualEquipment gets instantiated at the moment the hardware gets plugged and the instance stops existing at the moment the hardware gets pulled. ActualEquipmentCharacteristics might include instant values like e.g. current temperature. |
| ExpectedEquipmentCharacteristics | A WireInterface instance cannot exist without definition of an ExpectedEquipment. In this document, the ExpectedEquipmentCharacteristics define a list of SFP types, which lead to sufficiently identical capability values of a WireInterface instance.  In general, the amount and type of information, which is concrete enough, but not over-defining the ExpectedEquipment, depends on the type of equipment. Wherever the ExpectedEquipment has values different from default, the ActualEquipment must have identical values, otherwise the ExpectedEquipment is not getting operational state Enabled and associated interfaces cannot become operational. ExpectedEquipmentCharacteristics are defining minimum and maximum of the information, which is to be provided. |

# Information

The List\_of\_compatible\_SFPs and the List\_of\_approved\_SFPs contain the same attributes.

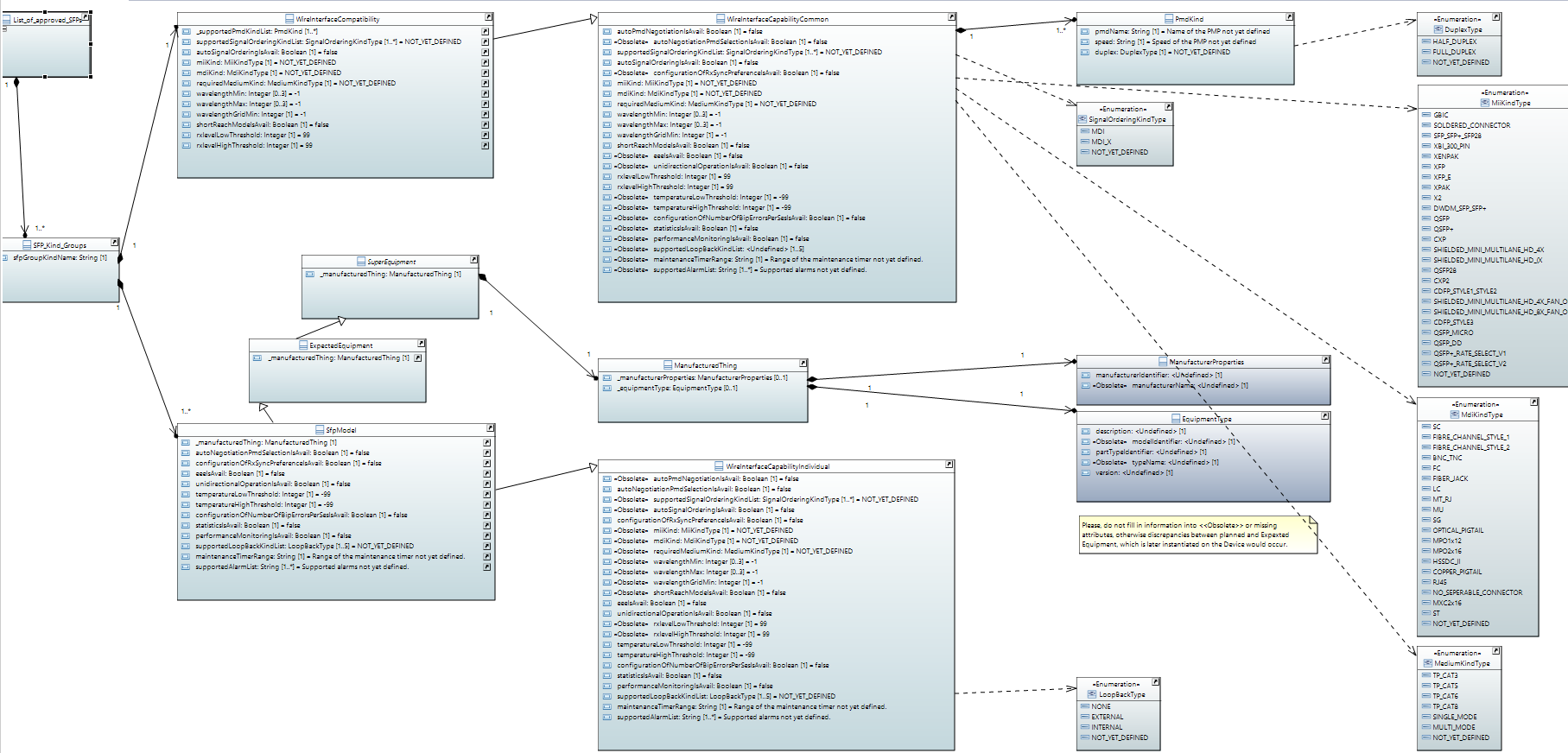
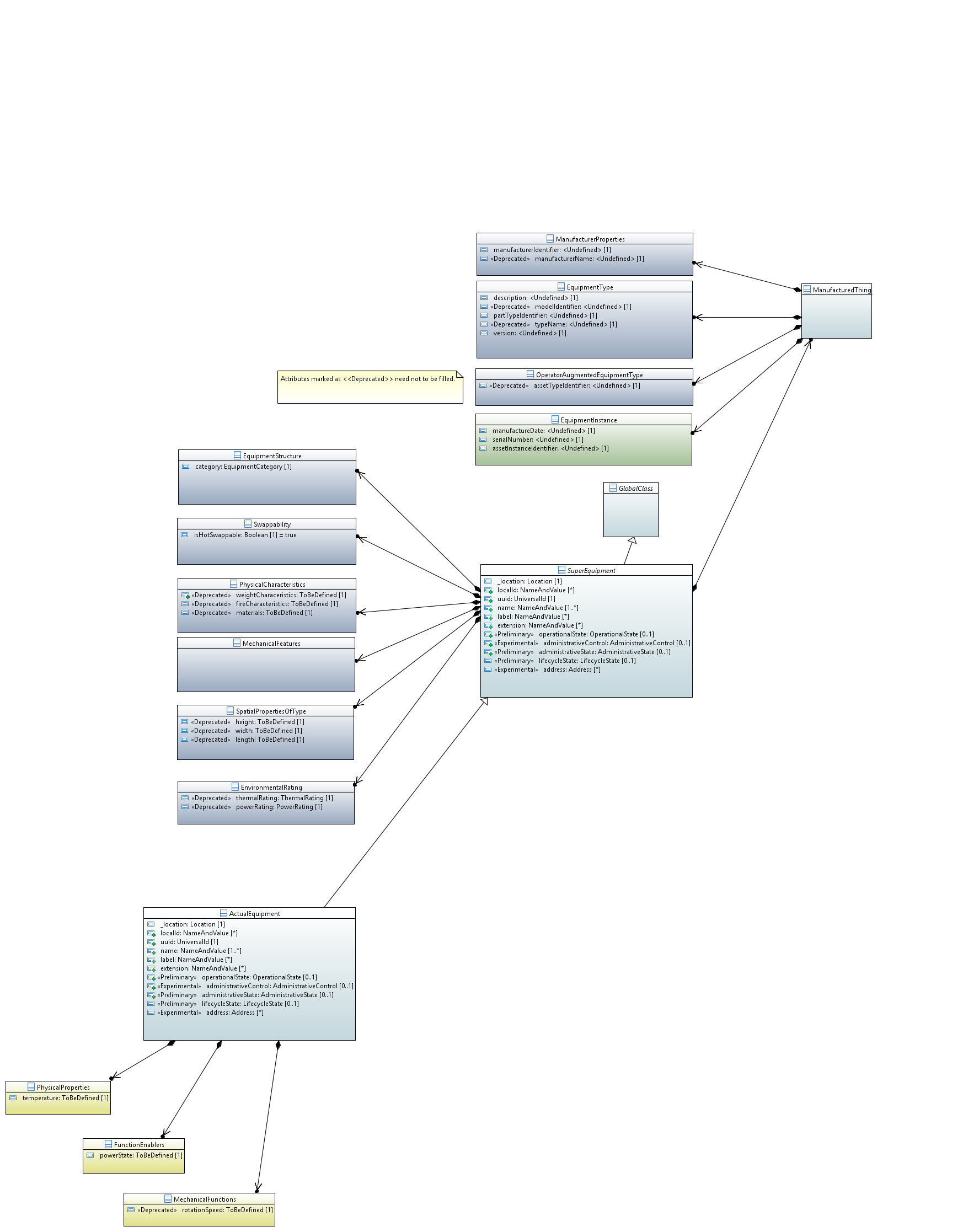
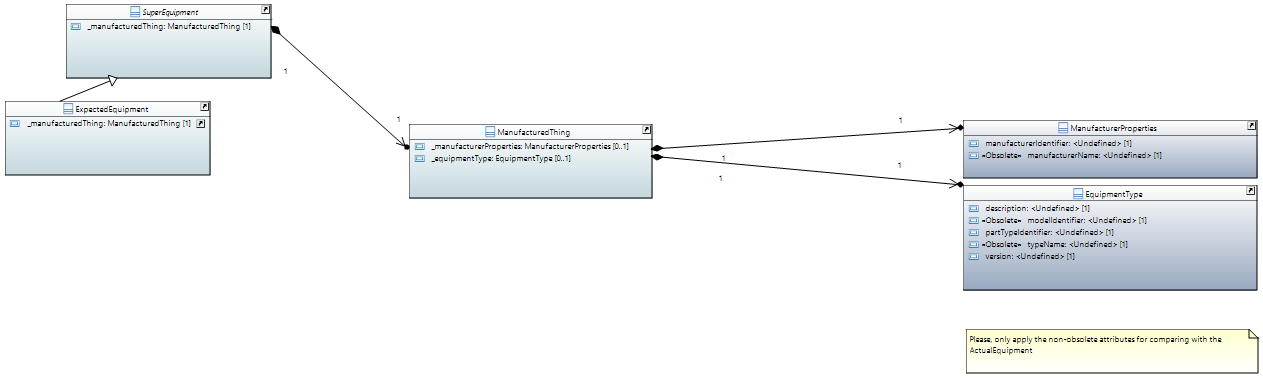


Figure 1: List\_of\_compatible\_SFPs and List\_of\_approved\_SFPs

The ActualEquipment describes the concrete instance of the hardware and its status information.



The ExpectedEquipment is for the interface instances being able to exist independently from the availability of actual hardware and has to assure that the interface instances are matching the physical hardware in the Device. It is only type related information.



# Processes

|  |  |  |
| --- | --- | --- |
| During Device and Management Interface Design | | |
|  | Vendor | Testing and approving one or several SFP models (e.g. different versions, different manufacturers) of a specific kind of SFP (e.g. 1310nm short range) |
|  | Vendor | Grouping of the compatible SFPs into SFP\_Kind\_Groups, which lead to identical values of the WireInterfaceCapability attributes |
|  | Vendor | For each of the SFP\_Kind\_Groups: Storing the identical values of the WireInterfaceCapability attributes inside the device and associating them with the name of the SFP\_Kind\_Group (sfpGroupKindName) |
| During Device Homologation and Introduction | | |
|  | Vendor | Providing the List\_of\_compatible\_SFPs to the operator |
|  | Operator | Potentially complementing SFP\_Kind\_Groups with further models, which have the same capability values and are tested to be compliant with the device by the operator  Potentially adding additional SFP\_Kind\_Groups, which are also compliant with the device (but not documented inside the device) |
|  | Operator | Potentially shortening the List\_of\_compatible\_SFPs by models, which the operator does not want to allow to be operated in his network |
|  | Operator | Making the List\_of\_approved\_SFPs available to the application layer, e.g. to be able to instantiate ExpectedEquipment and WireInterface\_Pacs in a planning tool, but also for being able to add and delete the operator specific SFPs in the SFP\_Group inside the Device |
| During Pre-Planning (before the actual device exists in the network) | | |
|  | Operator (in Application) | Instantiating the new network element in the Controller. Instantiating the physics (e.g. rack, slot, board, SFP cage, …) according to the respective device type as ExpectedEquipment in the Controller’s inventory |
|  | Operator (in Application) | Instantiating one of the SFPs of the SFP\_Group, which has the required characteristics, as ExpectedEquipment in the Controller’s inventory |
|  | Controller | Instantiating LTP, LP and WireInterface\_Pac including Capabilities according to the ExpectedEquipment. Capability information shall be taken from a repository holding the information provided in the List\_of\_approved\_SFPs |
|  | Operator (in Application) | Configuring the WireInterface and potentially instantiating and configuring of higher layers in the Controller’s inventory |
| During Commissioning | | |
|  | Field Technician | Connecting the Device to power supply |
|  | Device | Automatically instantiating the already existing physics (e.g. rack, slot, board, SFP cache, …, SFP) as ActualEquipment inside the Device |
|  | Device | ( Because there is no ExpectedEquipment defined: ) Automatically instantiating the already existing physics as ExpectedEquipment inside the Device |
|  | Device | Automatically comparing ExpectedEquipment with ActualEquipment and setting the operational status of the ExpectedEquipment. (Operational status will presumably be Enabled, because ExpectedEquipment is equivalent to ActualEquipment after former steps.) |
|  | Device | If some SFP is actually plugged:  (As a consequence of its instantiation as ExpectedEquipment: ) Automatically instantiating the WireInterface\_Pac including the Capability information, which is matching the ExpectedEquipment SFP. |
|  | Device | If some SFP is actually plugged:  (Because the operational status of the ExpectedEquipment is Enabled: ) Setting the operational status of the WireInterface on Enabled (but not switching on transmitters or something).[[1]](#footnote-1) |
|  | Device | Physically connecting the management interface of the Device with the management network.  Establishing a Netconf connection to the Controller either by automated process described in TR-545 (only management port):  Executing the DHCP process … Netconf Call-Home …  Or by manual configuration of the management interface (either management port or payload port). |
|  | Controller | After Netconf connection has been established, retrieving the information about ActualEquipment from the Device and instantiating it in the Controller’s inventory |
|  | Controller | If Pre-planning is existing:  Comparing the ExpectedEquipment in the Controller’s repository with the ExpectedEquipment instantiated in the Device:  If different SFP\_Group:  WireInterface instance on the Device to be deleted  ExpectedEquipment on the Device to be deleted  ExpectedEquipment (entire SFP\_Group from the List\_of\_approved\_SFPs) from the Controller to be written into the Device  Comparing ExpectedEquipment with ActualEquipment and setting the operational status of the ExpectedEquipment on the Device  If operational status of the ExpectedEquipment would be Enabled:  WireInterface on the Device to be newly instantiated  Operational status of the WireInterface to be set on Enabled  Updating operational status of ExpectedEquipment and WireInterface (if any instantiated) inside the Controller  If equal SFP\_Group:  Updating operational status of ExpectedEquipment and WireInterface inside the Controller  If ExpectedEquipment and WireInterface would reach operational status Enabled inside Device and Controller:  Automatically configuring the WireInterface and potentially higher layer interfaces on the Device according to the Pre-Planning, which is stored in the Controller  If actually operational after successful instantiation and configuration, updating the operational status of the pre-planned interfaces inside the Controller  If no Pre-planning is existing:  Instantiating the ExpectedEquipment (e.g. rack, slot, board, SFP cache, …, SFP) inside the Controller according to the ExpectedEquipment on the Device  If some SFP is/are plugged and some application would exist for this purpose:  Comparing the ExpectedEquipment (inside the Controller), which is so far representing an SFP\_Group according to the List\_of\_compatible\_SFPs of the vendor inside the Device, with the ExpectedEquipment, which is representing the Group according to the List\_of\_approved\_SFPs of the operator  If SFP\_Group in List\_of\_compatible\_SFPs is different from the one in List\_of\_approved\_SFPs:  Updating the ExpectedEquipment (inside the Controller) according to the SFP\_Group as documented in the List\_of\_approved\_SFPs of the operator in the Controller  WireInterface instance on the Device to be deleted  ExpectedEquipment on the Device to be deleted  Writing the updated ExpectedEquipment from the Controller into the Device  (Automatically setting the operational status of the ExpectedEquipment on the Device)  (If operational status of the ExpectedEquipment would be Enabled (means ActualEquipment still matches) by the Device:  WireInterface to be automatically, newly instantiated by the Device  Operational status of the WireInterface to be automatically set on Enabled by the Device)  Updating the operational status information of the ExpectedEquipment inside the Controller  Instantiating and configuring interfaces as they are existing in the Device on the Controller  Setting the operational status of these interfaces according to the Device inside the Controller |
| During Failure and Replacement of an SFP (configured device with management connection) | | |
|  | Device | SFP is failing  Setting the operational status of the ExpectedEquipment, which is representing the SFP, on Disabled inside the Device  Notifying the Controller  Setting the operational status of the WireInterface on Disabled inside the Device  Notifying the Controller |
|  | Controller | Updating the operational status information of the ExpectedEquipment, which is representing the SFP, inside the Controller  Updating the operational status information of the WireInterface inside the Controller |
|  | Application | Initiating a site visit by a Technician |
|  | Technician | Removing the broken SFP from the Device |
|  | Device | Automatically deleting the existing instance of ActualEquipment, which is representing the broken SFP, inside the Device  Notifying the Controller |
|  | Controller | Deleting the existing instance of ActualEquipment, which is representing the broken SFP, inside the Controller |
|  | Technician | Plugging a new SFP into the Device |
|  | Device | Automatically instantiating the new SFP as ActualEquipment inside the Device  Notifying the Controller |
|  | Controller | Instantiating the new ActualEquipment inside the Controller |
|  | Device | Automatically comparing ActualEquipment with ExpectedEquipment inside the Device.  If ActualEquipment equals one of the ExpectedEquipment:  Operational status of the ExpectedEquipment to be set on Enabled inside the Device  Notifying the Controller  Setting the operational status of the WireInterface on Enabled inside the Device  Notifying the Controller |
|  | Controller | Updating the operational status information of the ExpectedEquipment, which is representing the SFP, inside the Controller  Updating the operational status information of the WireInterface inside the Controller |
| During plugging a new SFP (configured device with management connection) | | |
|  | Technician | Plugging a new SFP into the Device |
|  | Device | Instantiating the new SFP as ActualEquipment inside the Device  Notifying the Controller |
|  | Controller | Instantiating the new ActualEquipment inside the Controller |
|  | Device | If no ExpectedEquipment is already instantiated on the occupied resource in the Device (also means that there is no pre-planning inside the Controller):  Automatically instantiating the new SFP as ExpectedEquipment inside the Device  Notifying the Controller  Automatically setting the operational status of the ExpectedEquipment on the Device (would certainly be Enabled in this case)  Notifying the Controller  WireInterface to be automatically, newly instantiated by the Device  Notifying the Controller  Operational status of the WireInterface to be automatically set on Enabled by the Device  Notifying the Controller  If some ExpectedEquipment is already instantiated on the occupied resource in the Device (also means that there is a pre-planning inside the Controller):  Automatically comparing ActualEquipment with ExpectedEquipment  If new ActualEquipment equals one of the existing ExpectedEquipment:  Setting the operational status of the ExpectedEquipment on Enabled inside the Device  Notifying the Controller  Setting the operational status of the WireInterface on Enabled inside the Device  Notifying the Controller |
|  | Controller | If there is no pre-planning inside the Controller:  If some application would exist for this purpose:  Comparing the ExpectedEquipment (inside the Controller), which is so far representing an SFP\_Group according to the List\_of\_compatible\_SFPs of the vendor inside the Device, with the ExpectedEquipment, which is representing the Group according to the List\_of\_approved\_SFPs of the operator  If SFP\_Group in List\_of\_compatible\_SFPs is different from the one in List\_of\_approved\_SFPs:  Updating the ExpectedEquipment (inside the Controller) according to the SFP\_Group as documented in the List\_of\_approved\_SFPs of the operator in the Controller  WireInterface instance on the Device to be deleted  ExpectedEquipment on the Device to be deleted  Writing the updated ExpectedEquipment from the Controller into the Device  (Automatically setting the operational status of the ExpectedEquipment on the Device)  (If operational status of the ExpectedEquipment would be Enabled (means ActualEquipment still matches) by the Device:  WireInterface to be automatically, newly instantiated by the Device  Operational status of the WireInterface to be automatically set on Enabled by the Device)  Updating the operational status information of the ExpectedEquipment inside the Controller  Uploading information about the WireInterface interface and creating a corresponding instance in the Controller  Setting the operational status of the WireInterface according to the Device inside the Controller  If there is pre-planning inside the Controller (which also means that there were already instances for ExpectedEquipment and WireInterface inside the Device):  Updating operational status information of the ExpectedEquipment and the WireInterface based on received notifications inside the Controller |

1. Remark: Default values are hindering Plug&Play at payload interfaces, because automatically generated interface is Enabled, but transmitters etc. are off by default. [↑](#footnote-ref-1)