**3GPP TSG-SA5 Meeting #155 *S5-243341d2***

**, , -**

|  |
| --- |
| *CR-Form-v12.2* |
| **CHANGE REQUEST** |
|  |
|  |  | **CR** |  **-** | **rev** |  | **Current version:** |  |  |
|  |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME |  | Radio Access Network | **X** | Core Network | **X** |

|  |
| --- |
|  |
| ***Title:***  | Input to DraftCR TS 28.105 Correct information models definition  |
|  |  |
| ***Source to WG:*** |  |
| ***Source to TSG:*** | S5 |
|  |  |
| ***Work item code:*** | AIML\_MGT |  | ***Date:*** |  |
|  |  |  |  |  |
| ***Category:*** | F |  | ***Release:*** |  |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)…Rel-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19) Rel-20 (Release 20)* |
|  |  |
| ***Reason for change:*** | It is specified in the Draft CR wheather a specific IOC is system created , or deleted by the system . However it is not adressed for all the cases where the IOC is system created that it is also deleted by the system only. Correct the description of information about system created or deleted by the system in order to be consistence.  |
|  |  |
| ***Summary of change:*** | Correct description of information about system created or deleted by the system in order to be consistence, and information about subscription to notifications. |
|  |  |
| ***Consequences if not approved:*** | leads to incorrect implementation. |
|  |  |
| ***Clauses affected:*** |  |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **X** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **X** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **X** |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** |  |

|  |
| --- |
| **Start of modification** |

### 7.2a.2 Class definitions

#### 7.2a.2.1 MLEntity

##### 7.2a.2.1.1 Definition

This IOC represents the ML entity. ML model or ML entity are not subjects for standardization. It is name-contained by MLEntityRepository.

This MLEntity MOI can be created by the system (MnS producer) or pre-installed. The MnS consumer can request the System to delete the MLEntity MOI.

The MLEntity may contain 3 types of contexts - TrainingContext, ExpectedRunTimeContext and RunTimeContext which represent status and conditions of the MLEntity. These contexts are of mLContext <<dataType>>, see clauses 7.4.3 and 7.5.1 for details.

 It also contains a reference named retrainingEventsMonitorRef which is a pointer to ThresholdMnonitor MOI. This indicates the list of performance measurements and the corresponding thresholds that are monitored and used to identify the need for re-training by the MnS Producer. After the MLEntity MOI has been instantiated, the MnS Consumer can request MnS producer to instantiate a ThresholdMonitor MOI and update the reference in the MLEntity MOI that can be used by the MnS producer to decide on the re-training of the MLEntity. The MnS producer can be ML Training MnS producer or ML Inference MnS Producer.

##### 7.2a.2.1.2 Attributes

Table 7.2a.2.1.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable  | isWritable | isInvariant | isNotifyable |
| mLEntityId | M | T | F | F | T |
| aIMLInferenceName | M | T | F | F | T |
| mLEntityVersion | M | T | F | F | T |
| expectedRunTimeContext | M | T | T | F | T |
| trainingContext | CM | T | F | F | T |
| runTimeContext | O | T | F | F | T |
| supportedPerformanceIndicators | O | T | F | F | T |
| mLCapabilitiesInfoList | M | T | F | F | T |
| **Attribute related to role** |  |  |  |  |  |
| retrainingEventsMonitorRef | O | T | T | F | T |
| sourceTrainedMLEntityRef | CM | T | F | F | T |

##### 7.2a.2.1.3 Attribute constraints

Table 7.2a.2.1.3-1

|  |  |
| --- | --- |
| Name | Definition |
| trainingContext Support Qualifier | Condition: The trainingContext represents the status and conditions related to training and should be added when training is completed. |
| sourceTrainedMLEntityRef Support Qualifier | Condition: The MLEntity MOI containing this attribute represents an ML entity loaded to an inference function. |

##### 7.2a.2.1.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

#### 7.2a.2.2 MLEntityRepository

##### 7.2a.2.2.1 Definition

The IOC MLEntityRepository represents the repository that contains the ML entities. It is name-contained by SubNetwork or ManagedElement.

This MLEntityRepository instance can be created by the system (MnS producer) or pre-installed.

The MLEntityRepository MOI may contain one or more MLEntity(s).

##### 7.2a.2.2.2 Attributes

Table 7.a.2.2.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

##### 7.2a.2.2.3 Attribute constraints

None.

##### 7.2a.2.2.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

#### 7.2a.2.3 MLEntityCoordinationGroup

##### 7.2a.2.3.1 Definition

This IOC represents the group of ML entities, which can be trained and tested jointly and used to perform inference in a coordinated way. It is name-contained by MLEntityRepository.

This MLEntityCoordinationGroup instance is created by the system (MnS producer) or pre-installed, The MnS consumer can request the System to delete the MLEntity MOI.

One ML entity may have dependencies on one or more of the other ML entities of the same group.

One group is associated with at least two ML entities.

##### 7.2a.2.3.2 Attributes

Table 7.2a.2.3.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable  | isWritable | isInvariant | isNotifyable |
|  |  |  |  |  |  |
| **Attribute related to role** |  |  |  |  |  |
| memberMLEntityRefList | M | T | F | F | T |

|  |
| --- |
| **Next modification** |

##### 7.3a.1.2.1 MLTrainingFunction

###### 7.3a.1.2.1.1 Definition

The IOC MLTrainingFunction represents the entity that undertakes ML training. The MOI of MLTrainingFunction is also the container of the MLTrainingRequest, MLTrainingReport, MLTrainingProcess and ThresholdMonitor MOI(s).

This MLTrainingFunction instance is created by the system (MnS producer) or pre-installed, it can only be deleted by the system

The ThresholdMonitor contains the list of performance measurements and the corresponding thresholds that are monitored and used to identify the need for re-training by the MnS Producer.

The entity represented by MLTrainingFunction MOI supports training of one or more MLEntity(s).

###### 7.3a.1.2.1.2 Attributes

Table 7.3a.1.2.1.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| **Attribute related to role** |  |  |  |  |  |
| mLEntityRepositoryRef | M | T | F | F | T |

###### 7.3a.1.2.1.3 Attribute constraints

None.

###### 7.3a.1.2.1.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

##### 7.3a.1.2.2 MLTrainingRequest

###### 7.3a.1.2.2.1 Definition

The IOC MLTrainingRequest represents the ML model training request that is triggered by the ML training MnS consumer.

To trigger the ML model training process, ML training MnS consumer needs to create MLTrainingRequest object instances on the ML training MnS producer.The MLTrainingRequest MOI is contained under one MLTrainingFunction MOI.

The MLTrainingRequest MOI may represent the request for initial ML training or re-training. For ML re-training, the MLTrainingRequest is associated to one MLEntity, or associated to one MLEntityCoordinationGroup.

The MLTrainingRequest may have a source to identify its origin, which may be used to prioritize the training resources for different sources. The sources may be for example the network functions, operator roles, or other functional differentiations.

Each MLTrainingRequest indicates the expectedRunTimeContext that describes the specific conditions for which the MLEntity should be trained.

In case the request is accepted, the ML training MnS producer decides when to start the ML training based on consumer requirements. Once the MnS producer decides to start the training based on the request, the ML training MnS producer instantiates one or more MLTrainingProcess MOI(s) that are responsible to perform the followings:

- collects (more) data for training, if the training data are not available or the data are available but not sufficient for the training;

- prepares and selects the required training data, with consideration of the consumer’s request provided candidate training data if any. The ML training MnS producer may examine the consumer's provided candidate training data and select none, some or all of them for training. In addition, the ML training MnS producer may select some other training data that are available in order to meet the consumer’s requirements for the MLentity training;

- trains the MLEntity using the selected and prepared training data.

The MLTrainingRequest may have a requestStatus field to represent the status of the specific MLTrainingRequest:

- The attribute values are "NOT\_STARTED", " IN\_PROGRESS", "SUSPENDED", "FINISHED", and "CANCELLED".

- When value turns to " IN\_PROGRESS", the ML training MnS producer instantiates one or more MLTrainingProcess MOI(s) representing the training process(es) being performed per the request and notifies the MLT MnS consumer(s) who subscribed to the notification.

When all of the training process associated to this request are completed, the value turns to "FINISHED".

The ML training MnS prodcuer shall delete the corresponding MLTrainingRequest instance in case of the status value turns to "FINISHED" or "CANCELLED". The MnS producer may notify the status of the request to MnS consumer after deleting MLTrainingRequest instance.

|  |
| --- |
| **Next modification** |

##### 7.3a.1.2.4 MLTrainingProcess

###### 7.3a.1.2.4.1 Definition

The IOC MLTrainingProcess represents the ML training process.

When a ML training process starts, an instance of the MLTrainingProcess is created by the MnS Producer and notification is sent to to MnS consumer who has subscribed to it.

The MnS producer can delete the MLTrainingProcess instance whose attribute status equals to "FINISHED" or or "CANCELLED" automatically.

One MLTrainingProcess MOI may be instantiated for each MLTrainingRequest MOI or a set of MLTrainingRequest MOIs.

For each MLEntity under training, a MLTrainingProcess is instantiated, i.e. an MLTrainingProcess is associated with one MLEntity or one MLEntityCoordinationGroup.The MLTrainingProcess may be associated with one or more MLTrainingRequest MOI.

The MLTrainingProcess does not have to correspond to a specific MLTrainingRequest, i.e. a MLTrainingRequest does not have to be associated to a specific MLTrainingProcess. The MLTrainingProcess may be managed separately from the MLTrainingRequest MOIs, e.g. the MLTrainingRequest MOI may come from consumers which are network functions while the operator may wish to manage the MLTrainingProcess that is instantiated following the requests. Thus, the MLTrainingProcess may be associated to either one or more MLTrainingRequest MOI.

Each MLTrainingProcess instance needs to be managed differently from the related MLEntity, although the MLTrainingProcess may be associated to only one MLEntity. For example, the MLTrainingProcess may be triggered to start with a specific version of the MLEntity and multiple MLTrainingProcess instances may be triggered for different versions of the MLEntity. In either case the MLTrainingProcess instances are still associated with the same MLEntity but are managed separately from the MLEntity.

Each MLTrainingProcess has a priority that may be used to prioritize the execution of different MLTrainingProcess instances.

Each MLTrainingProcess may have one or more termination conditions used to define the points at which the MLTrainingProcess may terminate.

The "progressStatus" attribute represents the status of the ML model training and includes information the ML training MnS consumer can use to monitor the progress and results. The data type of this attribute is "ProcessMonitor" (see 3GPP TS 28.622 [12]). The following specializations are provided for this data type for the ML training process:

- The "status" attribute values are "RUNNING", "CANCELLING", "SUSPENDED", "FINISHED", and "CANCELLED". The other values are not used.

- The "timer" attribute is not used.

- When the "status" is equal to "RUNNING" the "progressStateInfo" attribute shall indicate one of the following states: "COLLECTING\_DATA", "PREPARING\_TRAINING\_DATA", "TRAINING".

- No specifications are provided for the "resultStateInfo" attribute. Vendor specific information may be provided though.

When the training is completed with "status" equal to "FINISHED", the MLT MnS producer provides the training report, by creating an MLTrainingReport MOI, to the MLT MnS consumer.

###### 7.3a.1.2.4.2 Attributes

Table 7.3a.1.2.4.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable  | isWritable | isInvariant | isNotifyable |
|  |  |  |  |  |  |
| priority | M | T | T | F | T |
| terminationConditions | M | T | T | F | T |
| progressStatus | M | T | F | F | T |
| cancelProcess | O | T | T | F | T |
| suspendProcess | O | T | T | F | T |
| **Attribute related to role** |  |  |  |  |  |
| trainingRequestRef | CM | T | F | F | T |
| trainingReportRef | M | T | F | F | T |
| mLEntityRef | M | T | F | F | T |

###### 7.3a.1.2.4.3 Attribute constraints

Table 7.3a.1.2.4.3-1

|  |  |
| --- | --- |
| Name | Definition |
| trainingRequestRef Support Qualifier | Condition: The MLTrainingReport MOI represents the report for the ML model training that was requested by the training MnS consumer (via MLTrainingRequest MOI). |

###### 7.3a.1.2.4.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

##### 7.3a.1.2.5 MLTestingFunction

###### 7.3a.1.2.5.1 Definition

The ML entity testing may be conducted by the ML training function, or by a separate function.

This MLTestingFunction instance is created by the system ( MnS Producer) or pre-installed , it can only be deleted by the system.

In case the ML entity testing is conducted by a function separate from the ML training function, the IOC MLTestingFunction is instantiated and represents the logical function that undertakes ML entity testing.

The entity represented by MLTestingFunction MOI supports testing of one or more MLEntity(s).

###### 7.3a.1.2.5.2 Attributes

Table 7.3a.1.2.5.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| mLEntityRef | M | T | F | F | F |

###### 7.3a.1.2.5.3 Attribute constraints

None.

###### 7.3a.1.2.5.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

##### 7.3a.1.2.6 MLTestingRequest

###### 7.3a.1.2.6.1 Definition

The IOC MLTestingRequest represents the ML entity testing request that is triggered by the ML testing MnS consumer.

To trigger the ML model testing process, ML testing MnS consumer needs to create MLTrainingRequest..

The MLTestingRequest MOI is contained under one MLTestingFunction MOI or MLTrainingFunction MOI which represents the logical function that conducts the ML entity testing. Each MLTestingRequest is associated to at least one MLEntity.

In case the request is accepted, the ML testing MnS producer decides when to start the ML testing. Once the MnS producer decides to start the testing based on the request, the ML testing MnS producer:

- collects (more) data for testing, if the testing data are not available or the data are available but not sufficient for the testing;

- prepares and selects the required testing data;

- tests the MLEntity by performing inference using the selected testing data, and

- reports the performance of the MLEntity when it performs on the selected testing data.

The MLTestingRequest may have a requestStatus field to represent the status of the request:

- The attribute values are "NOT\_STARTED", "IN\_PROGRESS", "SUSPENDED", "FINISHED", and "CANCELLED".

The ML testing MnS prodcuer shall delete the corresponding MLTestingRequest instance in case of the status value turns to "FINISHED" or "CANCELLED". The MnS producer may notify the status of the request to MnS consumer before deleting MLTestingRequest instance.

###### 7.3a.1.2.6.2 Attributes

Table 7.3a.1.2.6.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable  | isWritable | isInvariant | isNotifyable |
| requestStatus | M | T | F | F | T |
| cancelRequest | O | T | T | F | T |
| suspendRequest | O | T | T | F | T |
| **Attribute related to role** |  |  |  |  |  |
| mLEntityRef | M | T | F | F | T |
| mLEntityCoordinationGroupRef | M | T | F | F | T |

###### 7.3a.1.2.6.3 Attribute constraints

Void

###### 7.3a.1.2.6.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

##### 7.3a.1.2.7 MLTestingReport

###### 7.3a.1.2.7.1 Definition

The IOC MLTestingReport represents the ML testing report that is provided by the ML testing MnS producer.

The MLTestingReport MOI is contained under one MLTestingFunction MOI or MLTrainingFunction MOI which represents the logical function that conducts the ML entity testing.

For the joint testing of a group of ML entities, the ML testing report contains the testing results for every ML entity in the group.

The MLTestingReport instance is created by the ML testing MnS producer and notification is sent to ML testing Consumer who has subscribed to it.

###### 7.3a.1.2.7.2 Attributes

Table 7.3a.1.2.7.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable  | isWritable | isInvariant | isNotifyable |
| modelPerformanceTesting | M | T | F | F | T |
| mLTestingResult | M | T | F | F | T |
| **Attribute related to role** |  |  |  |  |  |
| testingRequestRef | CM | T | F | F | T |

###### 7.3a.1.2.7.3 Attribute constraints

Table 7.3a.1.2.7.3-1

|  |  |
| --- | --- |
| Name | Definition |
| testingRequestRef Support Qualifier | Condition: The MLTestingReport MOI represents the report for the ML model testing that was requested by the MnS consumer (via MLTestingRequest MOI). |

###### 7.3a.1.2.7.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions

|  |
| --- |
| **Next modification** |

##### 7.3a.2.2.1 AIMLInferenceEmulationFunction

###### 7.3a.2.2.1.1 Definition

This IOC represents the properties of a function that undertakes AI/ML Inference Emulation.

This AIMLInferenceEmulationFunction instance is created by the system (AI/ML inference emulation MnS producer) or pre-installed, it can only be deleted by the system

An AIMLInferenceEmulationFunction may be associated with one or more MLEntity(s). AIMLInferenceEmulationFunction is name contained with AIMLInferenceEmulationReport(s) that delivers the outcomes of the emulation processes.

NOTE: The way of triggering of an AI/ML inference emulation and the instantiation of the related AI/ML inference emulation process is not in the scope of the present document.

|  |
| --- |
| **Next modification** |

##### 7.3a.3.2.1 MLEntityLoadingRequest

###### 7.3a.3.2.1.1 Definition

This IOC represents the ML entity loading request that is created by the MnS consumer. Using this IOC, the MnS consumer requests the MnS producer to load an ML entity to the target inference function.

To trigger the ML entityLoading process, MnS consumer has to create MLEntityLoadingRequest object instances on the MnS producer.

This IOC has a requestStatus field to represent the status of the request:

- The attribute value is one of "NOT\_STARTED", "IN\_PROGRESS", "SUSPENDED", "FINISHED\_SUCCESS ", FINISHED\_FAILED" and "CANCELLED".

- When value turns to "IN\_PROGRESS", the MnS producer instantiates one or more MLEntityLoadingProcess MOI(s) representing the loading process(es) being performed per the request and notifies the MnS consumer(s) who subscribed to the notification.

The MnS prodcuer shall delete the corresponding MLEntityLoadingRequest instance in case of the status value turns to "FINISHED" or "CANCELLED".

|  |
| --- |
| **Next modification** |

##### 7.3a.3.2.2 MLEntityLoadingPolicy

###### 7.3a.3.2.2.1 Definition

This IOC represents the ML entity loading policy set by the MnS consumer to the producer for loading an ML entity to the target inference function(s).

To specify ML entity loading policy for one or muiltiply ML entities, MnS consumer needs to create MLEntityLoadingPolicyobject instances .

To remove ML entity loading policy for one or muiltiply ML entities, MnS consumer needs to delete MLEntityLoadingPolicy object instances.

This IOC is used for the MnS consumer to set the conditions for the producer-initated ML entity loading. The MnS producer is only allowed to load the ML entity when all of the conditions are met.

|  |
| --- |
| **Next modification** |

###### 7.3a.3.2.2.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

##### 7.3a.3.2.3 MLEntityLoadingProcess

###### 7.3a.3.2.3.1 Definition

This IOC represents the ML entity loading process.

For the consumer requested ML entity loading, one or more MLEntityLoadingProcess MOI(s) may be instantiated for each ML entity loading request presented by the MLEntityLoadingRequest MOI.

For the producer-initiated ML entity loading, one or more MLEntityLoadingProcess MOI(s) may be instantiated and associated with each MLEntityLoadingPolicy MOI.

One MLEntityLoadingProcess MOI represent the ML entity loading process(es) corresponding to one or more target inference function(s).

The "progressStatus" attribute represents the status of the ML entity loading process and includes information the MnS consumer can use to monitor the progress and results. The data type of this attribute is "ProcessMonitor" (see 3GPP TS 28.622 [12]). The following specializations are provided for this data type for the ML entity loading process:

- The "status" attribute values are "RUNNING", "CANCELLING", "SUSPENDED", "FINISHED", and "CANCELLED". The other values are not used.

- The "timer" attribute is not used.

- When the "status" is equal to "RUNNING" the "progressStateInfo" attribute shall indicate one of the following state: "LOADING".

- No specifications are provided for the "resultStateInfo" attribute. Vendor specific information may be provided though.

When the loading is completed with "status" equal to "FINISHED", the MnS producer creates the MOI(s) of loaded MLEntity under each MOI of the target inference function(s).

When a ML entity loading process starts, an instance of the MLEntityLoadingProcess is created by the MnS Producer and notification is sent to MnS consumers who have subscribed. The MnS producer can delete the MLEntityLoadingProcess instance whose attribute status equals to "FINISHED" or or "CANCELLED" automatically.

|  |
| --- |
| **Next modification** |

##### 7.3a.4.2.1 MLUpdateFunction

###### 7.3a.4.2.1.1 Definition

This IOC represents the function responsible for ML update.

This MLUpdateFunction instance can be created by the system or pre-installed.

The MOI of MLUpdateFunction is name-contained in an MOI of either a subnetwork, a managedFunction or a managementFunction.

The MLUpdateFunction is be associated with one or more ML entities.

The MLUpdateFunction contains one or more MLUpdateRequest(s)as well as one or more MLUpdateProcess(s), where an MLUpdateProcess is instantiated corresponding to one received MLUpdateRequest.

###### 7.3a.4.2.1.2 Attributes

The MLUpdateFunction IOC includes attributes inherited from ManagedFunction IOC (defined in TS 28.622 [12]) and the following attributes:

Table 7.3a.4.2.1.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| availMLCapabilityReport | M | T | F | F | F |
| **Attributes related to Role** |  |  |  |  |  |
| mLEntityRef | M | T | F | F | F |

###### 7.3a.4.2.1.3 Attribute constraints

None.

###### 7.3a.4.2.1.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

##### 7.3a.4.2.2 MLUpdateRequest

###### 7.3a.4.2.2.1 Definition

This IOC represents the properties of MLUpdateRequest.

For each request to update the ML capabilities, a consumer creates a new MOI of MLUpdateRequest on the MLUpdateFunction, i.e., MLUpdateRequest is instantiated for each request for updating ML capabilities:

- Each MLUpdateRequest is associated to at least one MLEntity

- Each MLUpdateRequest may have a RequestStatus field that is used to track the status of the specific MLUpdateRequest or the associated MLUpdateProcess. The RequestStatus is updated by MnS producer when there is a change in status of the update progress. The RequestStatus is an enumeration with the values: NOT\_STARTED, IN\_PROGRESS, CANCELLING, SUSPENDED, FINISHED, and CANCELLED

- Each MLUpdateRequest may contain specific reporting requirements including an mLUpdateReportingPeriod that defines the time duration upon which the MnS consumer expects the ML update is reported. The reporting requirements contained in the MLUpdateRequest are mapped to an existing MLUpdateProcess instance.

- The MLUpdateRequest may specify a performanceGainThreshold which defines the minimum performance gain that shall be achieved with the capability update. This implies that the difference in the performances between the existing capabilities and the new capabilities needs to be at least performanceGainThreshold, otherwise the new capabilities shall not be applied. A threshold of performanceGainThreshold=0% implies that the capabilities should be applied even if there is no noticeable performance gain.

- The MLUpdateRequest may indicates the maximum time that should be taken to complete the update.

To trigger the ML update process, MnS consumer needs to create MLUpdateRequest instances on the MnS producer.

The MnS producer shall delete the corresponding MLUpdateRequest instance in case of the status value turns to "FINISHED" or "CANCELLED".

###### 7.3a.4.2.2.2 Attributes

The MLUpdateRequest IOC includes attributes inherited from Top IOC (defined in TS 28.622 [30]) and the following attributes:

Table 7.3a.4.2.2.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| performanceGainThreshold | O | T | T | T | F |
| newCapabilityVersionId | O | T | T | T | F |
| updateTimeDeadline | O | T | T | T | F |
| requestStatus | M | T | T | F | T |
| mLUpdateReportingPeriod | O | T | T | F | T |
| cancelRequest | O | T | T | F | T |
| suspendRequest | O | T | T | F | T |
| **Attributes related to Role** |  |  |  |  |  |
| mLUpdateProcessRef | M | T | F | F | F |
| mLEntityRef | M | T | F | F | F |

###### 7.3a.4.2.2.3 Attribute constraints

None.

###### 7.3a.4.2.2.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

##### 7.3a.4.2.3 MLUpdateProcess

###### 7.3a.4.2.3.1 Definition

This IOC represents the ML update process.

For each MLUpdateRequest to update the ML capabilities, the MLUpdateProcess is instantiated for the MLUpdateRequest unless the MLUpdateRequest is associated with an ongoing MLUpdateProcess if the MLUpdateProcess is updating the same MLEntity(s) as stated in the MLUpdateRequest i.e., the MLUpdateProcess is associated with at least one MLUpdateRequest. Relatedly, the MLUpdateProcess is associated with at least one MLEntity.

- Each MLUpdateProcess may have a status attribute (i.e., progressStatus) used to indicate progress status of the update process.

- The MLUpdateProcess has the capability of compiling and delivering reports and notifications relating to the ML update request or process.

When a ML update process starts, an instance of the MLUpdateProcess is created by the MnS Producer and informed to MnS consumer who has subscribed to it. The MnS producer can delete the MLUpdateProcess instance whose attribute status equals to "FINISHED" or or "CANCELLED".

###### 7.3a.4.2.3.2 Attributes

The MLUpdateProcess IOC includes attributes inherited from Top IOC (defined in TS 28.622 [30]) and the following attributes:

Table 7.3a.4.2.3.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| cancelProcessOTTFTsuspendProcessOTTFTprogressStatus | M | T | F | F | T |
| **Attributes related to Role** |  |  |  |  |  |
| mLEntityRef | M | T | F | F | T |
| mLUpdateRequestRef | M | T | F | F | T |
| mLUpdateReportRef | M | T | F | F | T |

###### 7.3a.4.2.3.3 Attribute constraints

None.

###### 7.3a.4.2.3.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

##### 7.3a.4.2.4 MLUpdateReport

###### 7.3a.4.2.4.1 Definition

This IOC represents the properties of ML update report.

- The ML update process may generate one or more MLUpdateReport(s),

- Each MLUpdateReport is associated to one or more MLEntity(s) to indicate ML entities that have been updated.

- The MLUpdateReport may indicate the achieved performance gain for the specific ML capability update, which is the gain in performance of the new capabilities compared with the original capabilities.

- MLUpdateReport provides reports about MLEntity(s) or MLUpdateProcess(s) that themselves are associated with MLEntity(s) for which update is requested and/or executed. Correspondingly, both the MLUpdateRequest(s)and the MLUpdateProcess(s) are conditionally mandatory in that at least one of them must be associated with an instance of MLUpdateReport.

The MLUpdateReport instance can be created by the MnS producer when creating an MLUpdateRequest instance.

When the MnS producer delete a MLUpdateRequest instance, the corresponding MLUpdateReport instance is also deleted by MnS producer. The MnS consumer cannot request to create nor delete the MLUpdateReport instance.

###### 7.3a.4.2.4.2 Attributes

Table 7.3a.4.2.4.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| UpdatedMLCapability | M | T | F | F | F |
| **Attributes related to Role** |  |
| mLEntityRef | M | T | F | F | F |
| mLUpdateProcessRef | M | T | F | F | F |

###### 7.3a.4.2.4.3 Attribute constraints

None.

###### 7.3a.4.2.4.4 Notifications

The notifications specified for the IOC using this <<datatype>> for its attribute(s), shall be applicable.

##### 7.3a.4.2.5 AIMLInferenceFunction

###### 7.3a.4.2.5.1 Definition

This IOC represents the common properties of the AI/ML inference function.

This AIMLInferenceFunction instance can be created by the system or pre-installed.

The AIMLInferenceFunction MOI may be associated with one or more MOIs that represent the functions/functionalities (Note) provided by the subject AIMLInferenceFunction MOI.

The AIMLInferenceFunction MOI can be only created by the MnS producer but not consumer.

The MOI of AIMLInferenceFunction or the MOI of the IOC inheriting from the AIMLInferenceFunction IOC contains one or more MOI(s) of MLEntity .

NOTE: The IOCs representing the functions/functionalities (Note) that use the AI/ML inference function include MDAFunction, AnLFFunction, DMROFunction, DLBOFunction, and DESManagementFunction.

The AIMLInferenceFunction MOI may be contained by either a SubNetwork MOI, a ManagedElement MOI, or an MOI of ManagedFunction’s subclass, and it is allowed for an MnS producer to support multiple AIMLInferenceFunction MOIs contained in different superordinated MOIs among SubNetwork, ManagedElement and the ManagedFunction’s subclass.

The generation of inference outputs is based on the configuration of inference, e.g., to start a stated time, or to be executed at all times. The observations of the inference function and information on derived Outputs is registered in the inference report.

###### 7.3a.4.2.5.2 Attributes

Table 7.3a.4.2.5.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| activationStatus | M | T | T | F | T |
| managedActivationScope | O | T | T | F | T |
| **Attributes related to role** |  |  |  |  |  |
| usedByFunctionRefList | M | T | F | F | T |
| MLEntityRef | M | T | F | T | T |

###### 7.3a.4.2.5.3 Attribute constraints

None.

###### 7.3a.4.2.5.4 Notifications

The common notifications defined in clause 7.6 are valid for this IOC, without exceptions or additions.

##### 7.3a.4.2.6 AIMLInferenceReport

###### 7.3a.4.2.6.1 Definition

This IOC represents a report from a AI/ML Inference.

An AIMLInferenceFunction may generate one or more AIMLInferenceReport(s).

Each AIMLInferenceReport provides information about inference outputs from one or more MLEntity.

The AIMLInferenceReport also provides historical inference outputs for a series of time stamps.

The AIMLInferenceReport instance can be created by the MnS producer when creating an AIMLInferenceFunction instance.

|  |
| --- |
| **End of modification** |