**3GPP TSG-SA5 Meeting #162 *S5-253866***

**Goteborg, Sweden, 25 - 29 Aug 2025**

**Source: Samsung**

**Title: Rel-19 pCR 28.567 Historical CCL**

**Document for: Agreement**

**Agenda Item: 6.19.4.1**

# 1 Decision/action requested.

***The group is asked to endorse the proposal.***

# 2 References

None

# 3 Rationale

This contribution provides the solution for the agreed use case

\* \* \* First Change \* \* \* \*

## 6.2 Class diagram

### 6.2.1 Relationships



Figure 6.2.1-1: Relations for common information models for CCLmanagement

Editor’s Note: The handling of Goal, targets or objectives for the general closed control loops is FFS



Figure 6.2.1-2: NRM fragment for conflict management and Coordination entity



Figure 6.2.1-x: NRM fragment for Historical CCL

\* \* \* Next of Change \* \* \* \*

### 6.2.2 Inheritance



Figure 6.2.2-1: Inheritance Hierarchy for Closed Control Loops and for conflict management and Coordination entity

\* \* \* Next Change \* \* \* \*

### 6.3.x HistoricalCCLInfo

#### 6.3.x.1 Definition

This IOC defines the historical information specific for a particular CCL. This IOC is instantiated by the producre as appropriate.

#### 6.3.x.2 Attributes

The HistoricalCCLInfo IOC includes attributes inherited from Top IOC (defined TS 28.622[5]) and the following attributes:

Table 6.3.3.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | S | isReadable | isWritable | isInvariant | isNotifyable |
| cCLObjectClass | M | T | F | F | T |
| cCLInstanceIdentifier | M | T | F | F | T |
| satisfactionScore | M | T | F | F | T |
| metricBreachInformation | M | T | F | F | T |

#### 6.3.x.3 Attribute constraints

None

#### 6.3.x.4 Notifications

The common notifications defined in clauses 6.1 are valid for this IOC, without exceptions.

### 6.3.x MetricBreachInformation <<data type>>

#### 6.3.x.1 Definition

This defines the requirements breach information related with the CCL.

#### 6.3.x.2 Attributes

Table 6.3.3.2-1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | S | isReadable | isWritable | isInvariant | isNotifyable |
| breachedMetricIdentification | M | T | F | F | T |
| breachTime | M | T | F | F | T |
| mitigationAction | M | T | F | F | T |

#### 6.3.x.3 Attribute constraints

None

#### 6.3.x.4 Notifications

The common notifications defined in clauses 6.1 are valid for this IOC, without exceptions.

\* \* \* Next Change \* \* \* \*

## 6.4 Attribute definitions

### 6.4.1 Attribute properties

Table 6.4.1-1

| Attribute Name | Documentation and Allowed Values | Properties |
| --- | --- | --- |
| scopeType | It indicates the type of scope that represented by the particular scope instance. allowedValues: CCL\_MEASUREMENT\_SCOPE, CCL\_TARGET\_SCOPE, CCL\_CONTROL\_SCOPE, CCL\_IMPACT\_SCOPEEditor’s Note: The allowed values will be revisited | type: Enummultiplicity: 1..\*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| coordinationCapability | It indicates a capability of a coordination entity to coordinate CCL conflicts  | type: CoordinationCapabilitymultiplicity: \*isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| cCLCoordinationCapabilityID | It indicates an identifier for a specific CCL conflicts coordination capability  | type: Stringmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| closedControlLoopRefList | It indicates a list of DN for ClosedControlLoop Instances.allowedValues: N/A | type: DNmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| cCLScopeCoordinationCapability | It indicates a specific type of CCL conflict coordination capacity  | type: CCLScopeCoordinationCapabilitymultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| coordinatedCCLsScopes | It indicates the scopes of the CCL that are coordinated by the coordinationEntity It is a pair <string\_1, string\_2 > where string\_1 is the DN of a CCL being coordinated and string\_2 the DN of that CCL’s CCLScope. | type: pair <string, string >multiplicity: 2 ..\*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| operationalState | It indicates the operational state of the ClosedControlLoop instance. It describes whether the resource is installed and partially or fully operable (Enabled) or the resource is not installed or not operable (Disabled).AllowedValues; Enabled/DisabledallowedValues: "ENABLED", "DISABLED".The meaning of these values is as defined in 3GPP TS 28.625 [14] and ITU-T X.731 [15]. | type: ENUM multiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: DisabledisNullable: False |
| administrativeState | It indicates the administrative state of the ClosedControlLoop instance. It describes the permission to use or the prohibition against using the ClosedControlLoop instance. The administrative state is set by the MnS consumer. AllowedValues; Locked/UnlockedallowedValues: "LOCKED", "UNLOCKED".The meaning of these values is as defined in 3GPP TS 28.625 [14] and ITU-T X.731 [15]. | type: ENUM multiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: LockedisNullable: False |
| cCLComponentsInfo | It indicates information on the constituent components of a CCL. allowedValues: N/A | type: CCLComponentInfomultiplicity: 1..\*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| cCLComponentId | It indicates the identifier of a CCL component. It is the DN of a object instantiated to act as a component of the CCL | type: DNmultiplicity: 1..\*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| cCLSteps | It indicates the CCL steps or functionality that is accomplished by a CCL component. allowedValues: DATA\_COLLECTION, ANALYSIS, DECISION, EXECUTION | type: Enummultiplicity: 1..\*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| FaultManagementAlarmIdList | It describes the list of IDs of alarms to be managed by Fault Management CCL. allowedValues: A list of alarmIds as specified in TS 28.111 [4], clause 7.4.1 | type: Listmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: True |
| FaultManagementTimeWindow | It describes the information of a time window (including start and end time) specified by the consumer for fault management to carry out troubleshooting and to clear the alarms. allowedValues: timeWindow as defined in 3GPP TS 28.622 [5], clause 4.4.1 | type: TimeWindowmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: True |
| FaultManagementBackUpObjectRequirement | It describes whether to back-up the alarmed object is required by the consumer before fault management.allowedValues: True, False | type: Booelanmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| FaultManagementIsolateObjectRequirement | It describes whether to isolate the alarmed object from interaction with other objects is required by the consumer before fault management.allowedValues: True, False | type: Booelanmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| clearUserId | It carries the identity of the Fault Management CCL who is the consumer that invokes the clearAlarms operation.allowedValues: clearUserId as defined in 3GPP TS 28.111 [4], clause 7.4.1 | type: stringmultiplicity: 0..1isOrdered: N/AisUnique: N/A defaultValue: NoneisNullable: False |
| FaultManagementCCLReport | It describes the Fault Management CCL report.allowedValues: Not Applicable | type: FaultManagementCCLReportmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| GeneratedAlarmResultList | It describes the list of generated alarm results allowedValues: A list of GeneratedAlarmResult | type: Listmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| GeneratedAlarmResult | It describes the result for each alarmId listed in FaultManagemetAlarmIdListallowedValues: Not Applicable | type: GeneratedAlarmResultmultiplicity: 1..\*isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| FaultManagementCCLReportTime | It describes the time when the FaultManagementCCLReport is created.allowedValues: DateTime as specified in TS 28.622 [5]. | type: DateTimemultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| alarmId | It identifies an AlarmRecord as specified in TS 28.111 [4]allowedValues: A string as specified in TS 28.111 [4] | type: stringmultiplicity: 1isOrdered: N/AisUnique: N/A defaultValue: NoneisNullable: False |
| alarmClearedStatus | It describes whether an alarm is cleared by the Fault Management CCL when the identified root cause is resolved.allowedValues: True, False | type: Booelanmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| identifiedRootCauseInformation | It describes root cause information identified by the Fault Management CCL. allowedValues: String  | type: stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| enhancedCorrelationInformation | It describes the list of correlated alarm Ids identified by the Fault Management CCLallowedValues: A list of alarmId | type: Listmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| cCLActionConflictsHandling | This defines the handling of CCL action conflict between the two existing CCLs. | Type: cCLActionConflictsHandlingmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| conflictInformation | This defines the information related with a conflicting CCL. | Type: ConflictInformationmultiplicity: \*isOrdered: TrueisUnique: FalsedefaultValue: NoneisNullable: False |
| conflictResolution | This defines the information related with conflict resolution. | Type: ConflictResolutionmultiplicity: \*isOrdered: TrueisUnique: FalsedefaultValue: NoneisNullable: False |
| targetCCL | The identification of the CCL that need to be deleted or updated to resolve conflict. This will be decided as per the information ConflictResolution. | Type: Dnmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| conflictingCCLId | This indicates the CCL identification | Type: Dnmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| conflictingActions | This provides the set of actions that have been taken by the CCL as part of the Execute step. | Type: Stringmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| cCLPriority | This provides the priority of the CCL. This will be the numerical value between 1 to 10, with 1 being the least priority. | Type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| cCLMetricBreachPercentage | It defines the breach percentage per metric in terms of how bad the metric(s) is breached. For example, if the metric of guaranteed throughput is 200mbps and the actual throughput is coming to be 100mbps then the breach percentage would be 50%. The CCL that have higher percentage of breach will be prioritized | Type: Integermultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| cCLComponentList | It indicates the list of components ating as steps of the CCL, each either a MnF or a MnS producer whose services can be part of the CCL. The cCLComponent may have a role among MONITOR; ANALYSIS; DECISION; EXECUTION. Or OTHER. OTHER. Is used for example in the caes where a components fulfile more than 1 role or where the role can be siml y described by the four options.The cCLComponents are sequenced, i.e., cCLComponents is an ordred list. For example, if there are 2 steps that contribute to the analysis role, it is necessary to show how those steps are sequenced. The order in which they are listed indicates the order in which their services should be chained to complete the CCL | type: CCLComponentmultiplicity: 1..\*isOrdered: TrueisUnique: TruedefaultValue: NoneisNullable: False |
| cCLType | It indicates a type or Category of CCL that is to be instantiated or dynamically composition. It indicates the kind of capability that will be accomplished by the CCL instance, e.g. ENERGYOPTIMIZATION, SLICEASSURANCE, etc.The specific details, characteristics and behavior of a CCL for a given CCL type are then written into the CCL purpose.Editor’s Note: Documentation and Allowed values will be revisited | type: Stringmultiplicity: 1isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| cCLComponentRole | It indicates a role accomplished by CCL component. AllowedValues: MONITOR; ANALYSIS; DECISION; EXECUTION, OTHER. Is used for example in the caes where a components fulfile more than 1 role or where the role can be siml y described by the four options | type: Enummultiplicity: 1..\*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| cCLComponentIdentification | It indicates the entity accomplishing the component.It may be the the DN of an MOI or the combination of URI and DN that can be used to fulfil that role. | Type: Stringmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| cCLInstanceIdentifier | This defines the specific CCL instance | Type: Dnmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| satisfactionScore | The numerical value from 1 to 10 (1 being the worst), providing the consumer satisfaction with the CCL.  | Type: Integermultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| metricBreachInformation | This defines the requirement breach information related with the CCL. | Type: MetricBreachInformationmultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| breachedMetricIdentification | This defines the requirement which got breached | Type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| breachTime | This defines the time of the requirement breach | Type: DateTimemultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| mitigationAction | This defines the configuration actions that was performed by the CCL execution to mitigate the breach. | Type: Stringmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |
| cCLActionTrigger | This defines the criteria/conditions under which the CCL is allowed to take actions. | Type: CCLTriggermultiplicity: \*isOrdered: FalseisUnique: TruedefaultValue: NoneisNullable: False |
| desiredBehavior | This will define the corresponding behavior of the CCL. The behaviors can be represented by an ENUM to include:- DECISION\_ACTIVATION: The CCL executes the recommendations that it derives on to the network.- NOTIFY\_RCOMMENDATION: The CCL starts processing input to derive recommendations but without the corresponding actions executed on the network. Instead, the recommendation is notified to the consumer who then considers whether it should be applied or not.- DO\_NOTHING: do not do anything. | Type: ENUMmultiplicity: 1isOrdered: N/AisUnique: N/AdefaultValue: NoneisNullable: False |

\* \* \* Next Change \* \* \* \*

\* \* \* Next Change \* \* \* \*

\* \* \* Next Change \* \* \* \*

# Annex A (informative):UML code for model diagrams

## A.1 UML code for CCL management model diagrams

This annex contains the PlantUML source code for the NRM diagrams defined in clause 6.2 of the present document.

## A.1.1 CCL NRM fragment (Figure 6.2.1-1)

@startuml

skinparam ClassStereotypeFontStyle normal

skinparam ClassBackgroundColor White

skinparam shadowing false

skinparam monochrome true

hide members

hide circle

class ManagedEntity <<ProxyClass>>

class ClosedControlLoop <<InformationObjectClass>>

class CCLPurpose << ProxyClass >>

class CCLScope << InformationObjectClass >>

class CCLReport <<InformationObjectClass>>

class CCLComponent<<InformationObjectClass>>

ManagedEntity "1" \*-- "\*" ClosedControlLoop: <<names>>

ClosedControlLoop "1" <--> "\*" CCLPurpose

ClosedControlLoop "1" \*-- "\*" CCLScope: <<names>>

ClosedControlLoop "1" \*-- "\*" CCLReport: <<names>>

ManagedEntity "1" \*-- "\*" CCLComponent: <<names>>

ClosedControlLoop "1" -r-> "\*" CCLComponent

note left of ManagedEntity

 Represents the following IOCs:

 SubNetwork or

 ManagedElement

 end note

note top of CCLPurpose

 Can be any of these CCL purposes:

 NetworkProblemRecovery

 FaultManagement

 ...

end note

@enduml

**Source code for Figure 6.2.1-1 CCL NRM fragment**

## A.1.2 NRM fragment for Coordination entity (Figure 6.2.1-2)

@startuml

skinparam ClassStereotypeFontStyle normal

skinparam ClassBackgroundColor White

skinparam shadowing false

skinparam monochrome true

hide members

hide circle

class ManagedEntity <<ProxyClass>>

class ConflictManagementAndCoordinationEntity <<InformationObjectClass>>

class CoordinationCapability <<dataType>>

class ClosedControlLoop <<InformationObjectClass>>

ManagedEntity "1" \*-- "1" ConflictManagementAndCoordinationEntity: <<names>>

ConflictManagementAndCoordinationEntity "1" -r- "\*" CoordinationCapability

ClosedControlLoop "\*" -r- "\*" ConflictManagementAndCoordinationEntity

note left of ManagedEntity

 Represents the following IOCs:

 Subnetwork or

 ManagedElement

 end note

note top of CoordinationCapability

 Represents the following capabilities: ScopeCoordinationCoordination

 TriggerCoordination

 ActionExecutionCoordination

 DirectActionsCoordination

 IndirectTargetsCoordination

end note

@enduml

**Source code for Figure 6.2.1-2 NRM fragment for Conflict management and Coordination entity**

## A.1.3 NRM fragment for Historical CCL (Figure 6.2.1-x)

@startuml

skinparam ClassStereotypeFontStyle normal

skinparam ClassBackgroundColor White

skinparam shadowing false

skinparam monochrome true

hide members

hide circle

class ManagedEntity <<ProxyClass>>

class HistoricalCCLInfo<<InformationObjectClass>>

ManagedEntity "1" \*-- "1" HistoricalCCLInfo: <<names>>

note left of ManagedEntity

 Represents the following IOCs:

 SubNetwork or

 ManagedElement

 end note

@enduml

**Source code for Figure 6.2.1-3 NRM fragment for CCLTrigger**

## A.2 CCL inheritance relationships (Figure 6.2.2-1)

@startuml

skinparam ClassStereotypeFontStyle normal

skinparam ClassBackgroundColor White

skinparam shadowing false

skinparam monochrome true

hide members

hide circle

class Top << InformationObjectClass >>

class ClosedControlLoop <<InformationObjectClass>>

class CCLReport <<InformationObjectClass>>

class CCLScope <<InformationObjectClass>>

class ConflictManagementAndCoordinationEntity <<InformationObjectClass>>

class CCLComponent<<InformationObjectClass>>

class CCLTrigger<<InformationObjectClass>>

class HistoricalCCLInfo<<InformationObjectClass>>

Top <|-- ClosedControlLoop

Top <|-- CCLScope

Top <|-- CCLReport

Top <|-- ConflictManagementAndCoordinationEntity

Top <|-- CCLComponent

Top <|-- CCLTrigger

Top <|-- HistoricalCCLInfo

@enduml

**Source code for Figure 6.2.2-1 CCL inheritance relationships**

\* \* \* Next Change \* \* \* \*

## 7.x CCL creation based on Historical CCL data



Figure 7.x-1: CCL creation based on Historical CCL data

Producer instantiate and provision a CCL as defined in 3GPP TS 28.536

1. Consumer send DeleteMOI request for a CCL.
2. Producer sends a response. Producer either instantiate or modify the HistoricalCCLInfo MOI with the information related with CCL being deleted.
3. Consumer may decides to initiate a CCL. Before that it would like to understand the historical CCL information.
4. It send getMOIAttributes for HistoricalCCLInfo MOI to read the information captured.
5. Producer send a response
6. Consumer develops the learning based on the historical CCL information received based on the HistoricalCCLInfo MOI attributes.
7. Based on the learning, the consumer send a createMOI request to create a new CCL. It enables the newly created CCL to move from a reactive mode to a proactive mode, where it anticipates and prevents problems based on historical trends and patterns. This proactive approach enhances network optimization, issue prevention and improves the overall efficiency of network operations.
8. Producer send a response.

\* \* \* Next Change \* \* \* \*

## B.x CCL creation based on Historical CCL data (Figure 7.x-1)

@startuml CCL creation based on Historical CCL data

skinparam Shadowing false

autonumber

skinparam monochrome true

participant "CCL MnS Consumer" as CL1

participant "CCL MnS producer" as CL2

Note over CL2: CCL Configuration

Loop

 CL1 -> CL2: DeleteMOI (CCL) Request

 CL2 -> CL1: DeleteMOI (CCL) Respone

 Note over CL2: Instantiate HistoricalCCLInfo

end

CL1 -> CL1: Consumer may decides to initiate a CCL

CL1 -> CL2: getMOIAttributes (HistoricalCCLInfo) request

CL2 -> CL1: getMOIAttributes (HistoricalCCLInfo) response

CL1 -> CL1: Learning by the consumer

Opt

 CL1 -> CL2: CreateMOI(CCL) Request

 CL2 -> CL1: CreateMOI(CCL) Respone

end

@enduml

**PlantUML source code for Figure 7.x-1 CCL creation based on Historical CCL data**

\* \* \* Next Change \* \* \* \*

\* \* \* End of Change \* \* \* \*