**3GPP TSG-SA5 Meeting #157S5-245403**

**14 - 18 October 2024, Hyderabad, India**

**Source: Nokia**

**Title:** **pCR 28.867 Rapporteur cleanup**

**Document for: Approval**

**Agenda Item: 6.19.4**

# 1 Decision/action requested

***The group is asked to discuss and agree on the proposal.***

# 2 References

[1] 3GPP TR 28.867-041 “Closed control loop management”.

# 3 Rationale

TR28.867 has study several use cases on closed control loops. This pCR is to add conclusions and recommendations for normative work

# 4 Detailed proposal

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

# 5. Use Cases

# 1 Scope

The present document will study closed control loop management.

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 28.535: "Management and orchestration; Management services for communication service assurance; Requirements".

[3] 3GPP TS 28.104: "Management and orchestration; Management Data Analytics (MDA)".

[4] 3GPP TS 28.536: "Management and orchestration; Management services for communication service assurance; Stage 2 and stage 3".

[5] 3GPP TS 28.312: "Management and orchestration; Intent driven management services for mobile networks".

[6] 3GPP TS 32.422: "Telecommunication management; Subscriber and equipment trace; Trace control and configuration management".

[7] 3GPP TS 28.872: "Study on Management of planned configurations".

[8] 3GPP TS 28.552: "Management and orchestration; 5G performance measurements".

[9] 3GPP TS 28.554: "Management and orchestration; 5G end to end Key Performance Indicators (KPI)".

[10] 3GPP TS 32.161: "Management and orchestration; JSON expressions (Jex)".

[11] 3GPP TS 28.622: "Telecommunication management; Generic Network Resource Model (NRM) Integration Reference Point (IRP); Information Service (IS)".

# 3 Definitions of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

**Historical CCL:** CCL that existed in past

## 3.2 Symbols

Void.

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1]. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

# 4 Concepts and Background

## 4.1 Introduction and Overview

### 4.1.1 Closed Control loops

Extending the definitions in 3GPP TS 28.535 [2] and 3GPP TS 28.536 [4], A Closed Control Loop (CCL) is a type of control mechanism that monitors and regulates a set of managed entities with the objective of achieving a specific goal. A CCL can be logically decomposed into several stages, each providing a specific functionality and where the stages work together to achieve the stated goal. Any two CCLs with the same functionality may have the functionality supported in different count of stages implementing the functionality and similarly, any two CCLs with the same functionality and same count of stages, the respective stages may not have the same functionality.

A control loop is a building block for management of networks and services. The basic principle of any control loop is to adjust the value of an observed variable (expressed as for example an attribute) to control/influence the value of a desired goal (expressed as for example an attribute) for a controlled entity, such as a managed entity or managed function. The producer of the measurements or observations, the control service, and the controlled entity are all required to fully realize and use a control loop.

A control loop can be an open control loop in which case a human operator or other management entity intervenes inside the loop. A control loop can be Closed Control Loop (CCL) which operates without any intervention from a human operator or any other management entity other than possibly the initial configuration of the measurement producer and configuration of the control loop. In a closed control loop the input to the control loop provided by human operator or other management entity may include the goal or policies. Besides the provisioning needed to realize the goal, the output of the closed control loop may also include closed control loop status to a human operator or other management entity.

Examples of well-known Closed Loop types are OODA loop, composed of 4 stages (Observe, Orient, Decide, Act) and MAPE-K, also composed of 4 stages (Monitor, Analyse, Plan, Execute) plus Knowledge.

1. b)

Figure 4.1.1-1: Open control loop entities versus Closed control loop entities (see TS28.535)

The NRM Model defined for assurance closed control loop in 3GPP TS 28.536 [4] shows that the current CCL mechanism defined in 3GPP can only enable creation of a CCL for SLS assurance purpose only and the SLS assurance can only be done either for a slice or for a slice subnet. This is considered to be very restrictive and demeaning the potential of a CCL. A CCL can very well be instantiated to deliver in various other fields related improving the overall efficiency of the network including for the RAN, Core network or edge network. CCLs may be used for, e.g. performance assurance of the network and its nodes, coverage optimization of the radio network, improving energy efficiency and consumption of the network, maintaining guaranteed UE specific requirements e.g. UE throughput, etc.



Figure 4.1.1-2: Assurance management NRM fragment (3GPP TS 28.536 [4])

### 4.1.2 Functional stages of a closed control loop

A closed control loop may manage any managed entity, e.g. a network resource or a communication service. Generally, the control loop consists of the steps Monitoring/data collection, Analysis, Decision and Execution. The adjustment of the resources of the managed entity used is completed by the continuous iteration of the steps in a management control loop.



Figure 4.1.2-1: Stages of a Control Loop  
a) the four functional stages; and  
b) 2 stages combined into a single management function

The "Monitoring/data collection" stage is responsible for collecting and pre-processing data from managed entities or from external sources.

The "Analysis" stage derives insights from the available data obtained in the monitoring/data collection stage. The insights provide answers to the questions:

- "What happened?"; or

- "What is likely to happen?";

- "Where did it happen or is it likely to happen?";

- "When did it happen or is it likely to happen?"; and

- "Why?".

The "Decision" stage is responsible for deriving workflows from insights provided by the analysis stage. It decides which reactive, proactive or predictive actions should be taken in consideration of insights obtained in the analysis stage.

The execution stage manages the activation of commands on the controlled resources or entities. The decision stage should decide which actions are required, but not necessarily how they should be taken in the managed entities. So, the translation from actions to commands is a responsibility of the execution stage.

## 4.2 Realizations of Closed Control Loops

### 4.2.1 Closed-Box Closed Control Loops - SON Functions as CCLs

Closed-Box Closed Control Loops (CB-CCL) are assembled prior to their use in the Management system, e.g. as SON functionality. The CB-CCLs components, as well as the communication and interoperation between the CB-CCL components, are out of-scope of the present specification. As such, only the external interactions and capabilities of the CB-CCLs are in-scope of standardization.

A SON function may be used to realize a closed control loop, i.e. as illustrated by Figure 4.2.1-1, the SON function may be used as the implementation of the functional logic of an abstract CCL.



Figure 4.2.1-1: SON function as a closed control loop

### 4.2.2 Open-Box Closed Control Loops

Open-Box CCLs (OB-CCL) are assembled on demand by MnS consumers, using capabilities offered by the Management system, e.g. from independent management functions or management services. The OB-CCLs stages or the components accomplishing those stages, as well as the communication and interoperation between these OB‑CCL components, are based the different 3GPP management services. Examples management functions include MDA and SON functions while example management services include MDA capabilities, PM jobs, and data management jobs.

## 4.3 Closed Control Loop conflicts management

### 4.3.1 CCL conflict scenarios

Multiple CCLs could co-exist and concurrently act within the same environment. The CCLs can affect one another, in the worst cases leading to conflicts. The conflicts may occur among goals, control scopes or actions of the CCLs. The control scopes of a CCL are the set of managed entities and controlled parameters on those managed entities for which a CCL instance takes responsibility. Actions of the CCL are changes that a CCL can perform over a managed entity such as configuring an attribute. The possible conflict scenarios include:

- Conflicts among the targets within the goals of the individual CCLs sharing a given scope, i.e. (where applicable) that a target network configuration parameter is only part of the goals of one CCL in the given scope.

NOTE 1: The scope is the set of managed objects and their properties which the CCL measures or is responsible to configure.

- Conflicts among the performance metrics even where there are no goal conflicts, i.e. that for two CCLs which have different goals or targets within their goals but share a given scope, one CCL will not affect the network performance metrics that the other CCL is responsible for. For example, a conflict could occur among the metrics if a CCL that optimizes energy consumption affects handover performance metrics which are supposed to be optimized by another CCL.

- Conflicts among the scopes of the CCLs, e.g. where the measurement scope of CCL is the control scope of another CCL, so (where applicable) the spaces should be allocated such that that two CCLs will not control/adjust the same set of parameters on the same set of managed objects.

- Conflicts on when the CCLs may be triggered for execution, for example, where 2 CCLs have a related scope, to ensure that one CCL (CCL A) does not influence a scope that is used as input/measurement scope by another CCL (CCL B). In that case CCLA and CCL B need to be triggered in different times.

To address the conflicts, coordination interactions are required between the CCLs and one or more higher hierarchy coordination functions to avoid or detect and resolve the conflicts.

NOTE 2: The descriptions of the conflict scenarios may need to be improved after detailed descriptions of the conflict detection and resolution use cases.

## 4.4 Management Service for Closed control Loops

### 4.4.1 Overview

The closed control loop can be viewed as an entity to be managed. Accordingly, some management capabilities related to the closed control loop will be exposed by the MnS producer that is associated with the closed control loops to enable the MnS consumer to manage the closed control loops. The Management Service exposes governance and monitoring capabilities as described in 3GPP TS 28.535 [2], clause 4.2.5. Moreover, the MnS also exposes capabilities for coordinating the CL's activities, e.g. for providing information on conflict resolution and feedback on monitoring the impact of the CL's actions on other closed control loops or management functions.

### 4.4.2 Closed Control Loop Management Capabilities

CCLs automate the management of network resources thereby taking control away from operators. The behaviours of the CCLs need thus to be directed by operators as consumers of CCL management services. The characteristics and behaviours of the CCLs can be managed by the Mns consumer. The 3GPP management system should provide capabilities that enable a consumer to:

- Manage the execution of CCLs. E.g. to request for and be notified about the instantiation of CCLs. For instance, if the consumer wants to request for instantiation of an Energy saving CL for 10,000 cells.

- Compose or request for and be notified about the composition of a CCL from a set of specific components (such as analytics services or SON functions).

- Manage a closed loop composed from multiple components.

## 4.5 Closed control loop as enabler for Intent handling

A closed control loop can be described as an autonomous "entity" (after it has been configured in the network and associated with the controlled entity) with goals and descriptions what it aims to achieve through the phases of monitoring, analysis, decide and execute. The producer reports after an observation window if the goal is fulfilled or not fulfilled, see for further details 3GPP TS 28.536 [4].

An intent handler function can be described as an autonomous "entity" handling an intent containing a list of expectations, see 3GPP TS 28.312 [5]. An expectation includes knowledge about the desired outcomes. An intent handler function is producer of intent MnS. The producer of the intent MnS, reports on the fulfilment of an expectation according to the reporting requirements from the MnS consumer.

In contrast to closed control loop an intent has no information on how the producer realizes the intent expectations. Intent handling includes initial fulfilment and continuous assurance of the intent requirements. In many but not all cases closed control loops can be used to implement intent handling. The relationship between Intents and CCLs is shown in Figure 4.5-1.



Figure 4.5-1: Venn diagram showing relationship between set of Intent and set of CCLs

The main difference between intent handling and closed control loop concepts are demonstrated in Table 4.5-1

Table 4.5-1: Comparison between intent handling and closed control loop concepts

|  | Intent handling | Closed control loops | Comments |
| --- | --- | --- | --- |
| **Objective** | Focuses on how to comply to service and domain requirements as well as knowledge about desired outcomes as communicated by intent. | Focuses on maintaining optimal network performance through continuous feedback and adjustments i.e. implementing the means to achieve desired outcomes. | An intent can be used to specify what a closed control loop considers to be optimal. |
| **Level of abstraction** | Operates at an appropriate level, providing abstract requirements, focusing on requirements and communicate knowledge about desired outcomes. | Implementing the means to achieve and assure desired outcomes. | Intent handler for business, service or resource might be different |
| **Automation focus** | Multi-stage translation of higher-level requirements to lower-level requirements.  Monitoring of compliance to desired outcomes and translation into policies, actions, and configurations to reach compliance. | Automates the continuous monitoring and immediate adjustment processes. |  |
| **Integration** | Intent-based systems can utilize closed control loops to ensure that intents are continuously assured by dynamically adjusting network parameters based on immediate feedback. | Closed Control loops can participate in intent handling if they get the task to monitor and assure intent based requirements and detailed objectives derived from intent via an intent handler. |  |
| **Functional scope** | Includes negotiation capabilities see Annex B in 3GPP TS 28.312 [5]. | Closed Control loop management does not provide negotiation capabilities. |  |
| Deployment and assurance of networks and services (see for example clause 5.1.8 of 3GPP TS 28.312 [5]). | Assurance of communication services (see 3GPP TS 28.535 [2] and 3GPP TS 28.536 [4]). |  |

# 5 Use Cases

## 5.1 Use case 1: Dynamic CCL Creation

### 5.1.1 Description

#### 5.1.1.1 Overview

CCLs may be dynamically realized. There are two aspects to dynamically realization of CCLs - dynamic instantiation of a CCL from an existing template and dynamically composing the CCL.

#### 5.1.1.2 Dynamic composition of CCLs

A CCL may be composed on stages provided by different management functions or management services. i.e. the CCLs is assembled on demand by MnS consumers, using capabilities offered by the Management system, e.g. from independent management functions. The CCLs components, as well as the communication and interoperation between components, are based the different 3GPP management services. Accordingly, the MnS consumer should be able to identify and indicate the MnFs or MnS producers that should be used to compose a CCL. Moreover, the MnS consumer may indicate towards the MnS producer the request to compose the CL of a particular type (e.g. for optimizing energy efficiency) without requiring to state the specific components that should be used.

#### 5.1.1.3 Examples for scenarios for Dynamic composition of CCLs

##### 5.1.1.3.1 Composition from management Functions

Different management functions may be used to realize the different stages of a closed loop, for example, an MDA function may realize the analytics stage of the CCL while another management function may realize the decision stage of the CCL.



Figure 5.1.1.1-1: Management functions as stages of a closed control loop

##### 5.1.1.3.2 Composition from management services

Different management services may be used to realize the different stages of a closed loop, i.e. the management service provides the output expected from a specific stage.

EXAMPLE: A capability of the MDA MnS realizes an analytics stage of the CCL while another capability may realize a specific data collection stage of the CCL.

a) b)

Figure 5.1.1.3.2-1: Management services used as implementations of CCL stages:  
a) MDA MnS and PM job the respective implementations of the analysis and data collection stages; and b) MDA MnS as the implementation of the decision stage

The MnS consumer should be enabled to manage the composition of such a CCL. The MnS consumer could request for and be notified about the composition of a CCL from a set of specific components (i.e. specific management functions or management services). The request could indicate components with specific given capabilities (such as analytics services with specific analytics types) which should be combined to achieve the closed loop. Moreover, the request could be for composition of a CCL required to achieve a specific set of desired outcomes or goals.

### 5.1.2 Potential Requirements

**REQ-CCL-CRTN-1**: The CCL MnS Producer should support a capability enabling the MnS consumer to request for a CCL (instance) to be composed from a set of management function types or instances or management services.

**REQ-CCL-CRTN-2:** The MnS producer for CCL management should support a capability enabling the MnS consumer to request that a CCL of a specific type or fulfilling a stated goal should be composed from a set of management function types or instances or services.

**REQ-CCL-CRTN-3:** The MnS producer for CCL management should support a capability enabling the MnS consumer to provide conditions under which a CCL can be dynamically composed or instantiated triggered to execute.

**REQ-CCL-CRTN-4:** The MnS producer for CCL management should support a capability enabling the MnS consumer to be notified when a CCL is dynamically composed or instantiated or triggered to execute.

### 5.1.3 Potential Solutions

To enable dynamic composition of the CCL:

1. Extend the existing ACCL IOC to represent a general Closed Control Loop, say named CCL:

NOTE 1: The best name for this IOC and how to extend is For Further Study.

- The CCL may include information on a CCL Category, which indicates the CCL Category to be instantiated:

1. Introduce an attribute for a type of CCL, say called CCLCategory. The CCL Category indicates the kind of capability that is accomplished by the CCL instance, e.g. ENERGYOPTIMIZATION, SLICEASSURANCE, etc.

- The CCL may include an attribute that contains information on the conditions under which a CCL can be dynamically composed . The attribute can be of type condition.

1. Introduce a datatype representing a step of the CCL, say named cCLStep. The cCLStep represents either a MnF or a MnS producer which can be part of the CCL:

- The cCLStep may have a role among those identified, i.e. MONITOR; ANALYSIS; DECISION; EXECUTION:

1. Introduce an attribute for the role of the cCLStep.

NOTE 2: The CCL may combine the defined cCLSteps with other capabilities that are internal to the CCL for some of the roles.

- The cCLStep may indicate the DN of an MOI or the combination of URI and DN that can be used to fulfil that role.

1. Introduce an attribute for the entity that plays the role of the cCLStep.
2. The identifier may also be a query of the MnS registry defined as a condition expressed as JEX/XPATH expression.

- The cCLStep may contain information on how the specific step can be constructed. This may be represented as configuration plan as will be agreed in 3GPP TS 28.872 [7] or as a JEX/XPATH expression.

NOTE 3: The representation of the cCLStep as configuration plan as will be agreed in 3GPP TS 28.872 [7] or as a JEX/XPATH expression is For Further Study.

1. Introduce on the CCL IOC, an attribute representing the sequence of steps of the CCL, say called cCLSequence. The MnS consumer can provide the list of MnFs or MnS producers that should be combined into a CCL. The cCLSequence indicates the sequence in which the CCL steps are executed.

EXAMLE: Is there are 2 steps that contribute to the analysis role, it is is necessary to show how those steps are sequenced:

- cCLSequence may be used as the request to compose a CCL (instance) through its members with each as a cCLStep identifying the DNs of the managed service producers or identifiers of management services (capabilities) that the MnS consumer wants to be used in a CCL (instance). The set of DNs may be 1 or more, e.g. only 1 if a single SON function fulfils the objectives of the CCL.

1. Introduce in CCL an attribute providing information related to the identifiers of the required management functions and the required configuration that make the CCL. When a combination of the sets of management functions and services are all defined to include their data sources, the combination is equivalent to a dynamically composed CCL.

### 5.1.4 Evaluation of solutions

The potential solution described in clause 5.1.3 is a fully NRM-based approach that extends the existing NRM fragments to realize dynamic composition of a CCL. The solution allows the MnS consumer to either directly compose the CCL or to request a MnS producer to compose the CCL. The solution enhances the existing ACCL with small straightforward implementable changes that reuse existing information elements like the condition monitor.

Therefore, the solution described in clause 5.1.3 is a feasible solution for dynamic composition of CCLs.

## 5.2 Use case 2: Triggered CCL

### 5.2.1 Description

#### 5.2.1.1 Overview

The existing CCL mechanism enables consumer to request the initiation of a CCL with the goal to maintain particular SLS (indicated by the AssuranceGoal). The CCL is expected to monitor the network to see if there have been some goal breaches. If there is, the consumer is notified, and the appropriate actions can be taken to mitigate the breach by the consumer. The consumer may also decide to update the existing CCL or create a new one to mitigate the breach. A CCL is always instantiated, updated and deleted on an explicit request from the consumer.

#### 5.2.1.2 Conditional instantiation of a CCL

Considering the autonomous nature of CCL, it is beneficial to study possible improvements to CCL management including automated instantiation, update and deletion of a CCL based on information provided by the consumer that could be used by the system to trigger CCL management. The existing CCL mechanism places a burden on the consumer to monitor the network and decide whether to instantiate a CCL, update a CCL, or delete a CCL. A possible improvement may be to allow the consumer to define trigger conditions for automated instantiation, update and deletion of a CCL.

The MnS consumer may want to request for a CCL to be dynamically instantiated when certain conditions are met. For example, the MnS consumer may want that for a CCL of a stated type or that matches a set of stated characteristics (e.g. goal) to be instantiated under conditions A and another with variations in goals to be instantiated under other conditions. The MnS consumer should be enabled to define those conditions so that the CCL is instantiated when the stated conditions are met.

The ConditionMonitor[11], post appropriate extensions, can be utilized to define triggering conditions for CCL management.

#### 5.2.1.3 Conditional execution of CCLs network changes

For the CCLs that have been instantiated, the MnS consumer may want to request for a CCL to be triggered to execute when certain conditions are met, e.g. when the performance on a certain threshold is crossed, or when the confidence is the decision is above a stated threshold. The consumer does not need to be aware of all decision, but providing conditions under which decisions may be activated or not, it is able to have supervision over the CCL without having to continuously track the decisions. The MnS consumer should be enabled to define those conditions for executing the CCL and that the CCL is triggered to execute when the stated conditions are met. Otherwise, the consumer should be enabled to define alternative actions, e.g. to notify the consumer of the decision that is not executed.

By supporting this, the execution can be affected by producer based on consumer's conditions or requirements.

### 5.2.2 Potential Requirements

**REQ-TRI-FUN-01:** The 3GPP management system should enable authorized consumers to provide information that can be used to trigger CCL instantiation.

**REQ-TRI-FUN-02:** The 3GPP management system should enable authorized consumers to provide information that can be used to trigger CCL update.

**REQ-TRI-FUN-03:** The 3GPP management system should enable authorized consumers to provide information that can be used to trigger CCL deletion.

### 5.2.3 Potential solutions

#### 5.2.3.1 Overview of LoopTrigger object

This solution proposes LoopTrigger object that would contain information a producer would use to trigger a CCL. The clause 5.2.3.1 specify the potential information to be present in this object. Clause 5.2.3.2 specify the usage of condition monitor to implement the LoopTrigger object.

#### 5.2.3.2 Information to be present in LoopTrigger object

Performance based criteria: This will define information related with performance measurements and KPIs that need to be monitored by the producer to see if the values have crossed the thresholds defined. This will include:

- Target Node: The identification of the Managed Object for which the performance is to be monitored.

- Measurement/KPI Name: Name of the measurement or the KPI.

- Trigger Value range: The CCL is triggered when the value of the measurement or KPI exceeds more that the trigger value or when the value decreases below the trigger value.

Once the Trigger Value has reach, the producer will send a notification to the consumer stating that an CCL is required. The notification will contain information needed to instantiate an CCL. The CCL is triggered appropriately.

Provisioning based criteria: This will define various provisioning events that need to be monitored by the producer to see if an CCL is to be initiated:

- Target Node: This can be a particular object or a DN, e.g. Intent.

- Provisioning Location: The CCL will be created only when the object created is targeting a specific location.

- Provisioning Event (e.g. Create{in case of an object}, Modify, Delete): The CCL will be created when the given event occurs on the given DN.

- Provisioning Time: The CCL will be created only when the given event occurs at a specified time.

- PreOrPostProvEvent: This will define if the CCL is to be instantiated before or after the provisioning event is completed.

Fault based criteria: This will define various fault related info that need to be monitored by the producer to see if an CCL is to be initiated:

- Target Node: This will define the node which need to be monitored for the emitted alarms (i.e. objectInstance in AlarmInformation).

- AlarmSeverityThreshold: This will define the "perceivedSeverity" threshold (i.e. threshold for each Severity). If total number of alarms, belonging to particular perceivedSeverity (e.g. critical, major etc.), goes beyond the threshold, an CCL will be instantiated.

- AlarmTypeThreshold: This will define the "AlarmType" threshold (i.e. threshold for each AlarmType). If total number of alarms, belonging to a particular alarmType, goes beyond the threshold, an CCL will be instantiated.

#### 5.2.3.3 Usage of ConditionMonitor to realize LoopTrigger object

This LoopTrigger object can be inherited from ConditionMonitor. The existing condition attribute will be extended to include various type of triggers provided in clause 5.2.3.

The condition will be defined as a datatype containing following information:

1. conditionObject: This is to represent the target node i.e. object for which the performance and fault is to be monitored.
2. conditionInfo: This is a set of multiple conditions that should be satisfied for a CCL to be instantiated:

- conditionItem: This will be the PM data name.

- conditionValue: This is to represent the expected value of the measurement or KPI.

- conditionString: This will be the logical assertion related to conditionItem and conditionValue ("is equal to", "is less than", etc.).

ConditionMonitor can be used to define the fault-based criteria as follows: the existing condition attribute will be defined as data type including the following information:

1. conditionObject: This is to represent the target node. This will define the node which need to be monitored for the emitted alarms (i.e. objectInstance in AlarmInformation).
2. conditionInfo: This is a set of multiple conditions that should be satisfied for a CCL to be instantiated:

- conditionItem: This may represent the total number of alarms with particular alarmType or perceivedSeverity.

- conditionValue: This is to represent the expected value.

- conditionString: This will be the logical assertion related to conditionItem and conditionValue ("is equal to", "is less than", etc.).

ConditionMonitor can be used to define the provisioning-based criteria as follows: the existing condition attribute will be defined as data type including the following information:

1. conditionInfo: This is a set of multiple conditions that should be satisfied for a CCL to be instantiated:

- conditionItem: This may represent the following a) The DN at which the provisioning operation is performed. b) the location of the instantiated DN c) the provisioning operation executed d) the time at which the provisioning operation is executed d) time detail specifying where it is the pre or post provisioning operation.

- conditionValue: This is to represent the expected value.

1. conditionString: This will be the logical assertion related to conditionItem and conditionValue ("is equal to", "is less than", etc.).

#### 5.2.3.4 Expressing trigger conditions using existing SA5 defined mechanisms

##### 5.2.3.4.1 Introduction

SA5 has established notations to express conditions:

- Jex for the HTTP/JSON solution.

- XPath for the NETCONF/YANG solution.

Conditions are expressed based on data nodes in the data node tree on a MnS producer.

Currently there are no data nodes defined for performance metrics and trace metrics. Therefore, it is not possible to express conditions based on these metrics.

Furthermore, it is not possible to build conditions that evaluate structural changes of the data node tree such as the creation or deletion of a data node.

This clause proposes mechanisms to address these gaps. The motivation for doing so is to have a single notation allowing to express conditions of all kinds and for all use cases, and to avoid silo solutions for the different kinds of conditions or silo solutions for some use case like triggering the creation of a CCL.

##### 5.2.3.4.2 Conditions based on performance metrics

**Stage 2 considerations**

Data nodes for performance metrics need to be introduced to allow for conditions based on performance metrics. For this purpose, the stage 2 NRM definitions are conceptually extended with pseudo attributes representing these performance metrics. These attributes are not readable and not writable by MnS consumers. They can be used only in condition expressions that are evaluated on MnS producers. The name of these attributes is equal to a measurement name defined in 3GPP TS 28.552 [8] or the KPI name defined in 3GPP TS 28.554 [9]. Its value is equal to the current value of the measurement or the KPI. A pseudo attribute for a specific performance metric can be (conceptionally) added only to the object class that is equal to the measurement object class of that measurement.

Pseudo attributes are not added explicitly to NRM stage 2 definitions. Alternatively, as a more formal approach, they could be added also to the stage 2 NRM definitions. Adding them directly to all measurement classes is feasible but not practical. A practical solution could be to add them to the "TopX" class, from which every other class inherits.

Table 5.2.3.3.2-1: Attributes of "TopX"

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | S | isReadable | isWritable | isInvariant | isNotifyable |
| objectClass | M | T | T | T | T |
| objectInstance | M | T | T | T | T |
| <measurementName1> | O | T (see note) | F | F | F |
| <measurementName2> | O | T (see note) | F | F | F |
| … | O | T (see note) | F | F | F |
| <measurementNameX> | O | T (see note) | F | F | F |
| NOTE: Attribute is not readable using CRUD operations. It can be accessed only by condition expressions. | | | | | |

In a condition expression the measurement pertaining to a measured object is identified by concatenating the "objectInstance" of the measured object with the "measurementName".

The MnS producer evaluating the condition expression should have access to the performance metric values. This is assumed here.

NOTE 1: So far conditions are designed to switch on and off a process: When the condition evaluates to false the process is (switched) off, when the condition evaluates to true, the process is (switched) on. For the present use case the condition needs to trigger an action. The triggering condition is therefore the transition of the condition evaluation result from false to true or true to false. The triggering condition needs to be specified together with the condition expression.

In the context of CCL, a CCL can be created when the condition expression transits from false to true and deleted when the condition expression transits from true to false.

**Stage 3 considerations**

Jex is used in the HTTP/JSON solution for expressing conditions. XPath is used in the NETCONF/YANG solution for expressing conditions.

**Jex example:**

The following example shows a Jex conditions expression with a pseudo attribute for the measurement "measurementA".

|  |
| --- |
| SubNetwork[id="SN1"]/ManagedElement[id="ME1"]/XyzFunction[id"=YXZF1"]/attributes/measurementA >10 |

When hysteresis is needed, the "ThresholdMonitor" can be used. It needs to be extended with the Boolean attribute "flag" flagging if the threshold is not crossed or crossed. For multi-level thresholds integers need to be used to flag the status. The "flag" attribute can be used in Jex conditions.

|  |
| --- |
| SubNetwork[id="SN1"]/ThresholdMonitor[id="TM1"]/attributes/flag = true |

NOTE 2: The MnS producer evaluating the condition expression should have access to the performance metric values. This is assumed here.

**XPath example:**

The XPath notation is the same as the Jex notation. Therefore, the Jex examples are examples for XPath, too.

##### 5.2.3.4.3 Conditions based on Trace metrics

**Stage 2 considerations**

To allow for conditions based on trace metrics the NRM is conceptually extended with pseudo attributes. These attributes are not readable and not writable by MnS consumers. They can be used only in condition expressions that are evaluated on MnS producers. The name of these pseudo attributes is equal to the trace metric identifier.

Pseudo attributes are not added to stage 2 NRM definitions. Alternatively, the approach described for measurements may be used:

- Adding the trace metric identifiers to "TopX".

NOTE: The MnS producer evaluating the condition expression should have access to the trace metric values. This is assumed here.

For trace metrics there are two use cases, which are described in the following. For signalling based MDT only use case 1 applies.

**Stage 3 considerations**

Jex is used in the HTTP/JSON solution for expressing conditions. XPath is used in the NETCONF/YANG solution for expressing conditions.

***Use case 1: MnS consumer is aware of corresponding "TraceJob" MOI***

The pseudo attribute is attached to the corresponding "TraceJob" MOI. The pseudo attribute name is constructed according to the metric identifier specified in 3GPP TS 32.422 [6]. Its value is equal to the current value of the measurement or information element.

Examples (stage 3):

The following example shows a Jex conditions expression to verify whether "REGISTRATION REQUEST" message on N1 interface between AMF and UE has been traced. It uses a pseudo attribute for the "REGISTRATION REQUEST" message.

|  |
| --- |
| SubNetwork[id="SN1"]/ManagedElement[id="ME1"]/TraceJob[id="YXZF1"]/attributes/ trace.amf.n1.registrationRequest=true |

The following example shows a Jex conditions expression to verify whether the Information Element (IE) "Requested NSSAI" of the "REGISTRATION REQUEST" message on N1 interface between AMF and UE has been included in the message content. It uses a pseudo attribute for Information Element (IE) "Requested NSSAI".

|  |
| --- |
| SubNetwork[id="SN1"]/ManagedElement[id="ME1"]/TraceJob[id="YXZF1"]/attributes/ trace.amf.n1.registrationRequest.requestedNssai=true |

The following example shows a Jex conditions expression with a pseudo attribute for the RRC counter "countMSB‑Uplink".

|  |
| --- |
| SubNetwork[id="SN1"]/ManagedElement[id="ME1"]/TraceJob[id="YXZF1"]/attributes/ trace.gnbCuCp.uu.CounterCheck.counterCheck.drb-CountMSB-InfoList.countMSB-Uplink>500 |

The following example shows a Jex conditions expression with a pseudo attribute for the RSRP value.

|  |
| --- |
| SubNetwork[id="SN1"]/ManagedElement[id="ME1"]/TraceJob[id="YXZF1"]/attributes/ immediateMdt.nr.m1.rsrp<90 |

***Use case 2: MnS consumer knows the corresponding network entity MOI***

The containment tree is used to identify the corresponding network entity (network element/interface). The pseudo attribute is attached to the corresponding network entity MOI. The pseudo attribute indicates the message name, message name and IE name or measurement name. The pseudo attribute name is constructed according to the metric identifier specified in 3GPP TS 32.422 [6] omitting, in case of trace, the items of job type, network element and interface.

If there is no corresponding interface IOC defined, the interface name is included in the pseudo attribute name.

Examples (stage 3):

The following example shows a Jex conditions expression to verify whether a "Nudm\_UECM\_Registration\_Request" message on N8 interface between AMF and UDM has been traced. It uses a pseudo attribute for message "Nudm\_UECM\_Registration\_Request".

|  |
| --- |
| SubNetwork[id="SN1"]/ManagedElement[id="ME1"]/UDMFunction[id="YXZF1"]/ EP\_N8[id="XXZ2"]/attributes/Nudm\_UECM\_Registration\_Request=true |

The following example shows a Jex conditions expression to verify whether a "REGISTRATION REQUEST" message on N1 interface between AMF and UE has been traced. As there is no IOC specified for the N1 interface, the N1 interface is included in the pseudo attribute name "n1.registrationRequest".

|  |
| --- |
| SubNetwork[id="SN1"]/ManagedElement[id="ME1"]/AMFFunction[id="YXZF1"]/attributes/ n1.registrationRequest=true |

The following example shows a Jex conditions expression to verify whether the information element (IE) "Requested NSSAI" of the "REGISTRATION REQUEST" message on N1 interface between AMF and UE has been included in the message content.

|  |
| --- |
| SubNetwork[id="SN1"]/ManagedElement[id="ME1"]/AMFFunction[id="YXZF1"]/attributes/ n1.registrationRequest.requestedNssai=true |

The following example shows a Jex conditions expression to verify whether the immediate MDT measurement M1 RSRP in NR is above a certain threshold.

|  |
| --- |
| SubNetwork[id="SN1"]/ManagedElement[id="ME1"]/GNBCUCPFunction[id="XYY"]/NRCellCU[id="YXZF1"]/ attributes/immediateMdt.nr.m1.rsrp>90 |

##### 5.2.3.4.4 Conditions based on data node tree changes

XPath allows to express conditions based on data node tree changes with the "count" function.

Jex does not feature the "count" function yet. It is proposed to add the "count" function as follows:

- Equality and relational expressions may have on the left side of the operator also the "count" function, defined in clause 4.1 of the W3C XPath1.0 specification. The "count" function returns the number of nodes in the argument node-set. The argument is an absolute location path or a relative location path:

EqualityExpr ::= CompOperandExpr EquaComp (String| Number | true | false | null)

RelationalExpr ::= CompOperandExpr RelaComp Number

CompOperandExpr ::= LocationPath | 'count' '(' LocationPath ')'

EquaComp ::= '=' | '!='

RelaComp ::= '<' | '>' | '<=' | '>='

Examples:

The "count" function can be used to test for the presence of objects. The following example evaluates to true only when at least one "ManagedElement" object exists under the "SubNetwork" object. The "ManagedElement" indicates the particular object or a DN, e.g. Intent.

|  |
| --- |
| count(/SubNetwork[id="SN1"]/ManagedElement)>=1 |

NOTE: An expression using the proposed count function can be combined with expressions testing other things (defined as part of Provisioning based criteria in clause 5.2.3.1) such as the value of one or more attribute. Note that the Jex notation currently specified in 3GPP TS 32.161 [10] already allows for testing these other things.

|  |
| --- |
| count(/SubNetwork[id="SN1"]/ManagedElement)>=1\  and /SubNetwork[id="SN1"]/ManagedElement/attributes/location="TV Tower"\  and /SubNetwork[id="SN1"]/ManagedElement/attributes/vendor="Company XY" |

In the context of the present use case testing on the presence of an intent object that triggers the creation of a closed loop is a realistic deployment scenario.

#### 5.2.3.4 Potential solution for Conditional execution of CCLs network changes

To enable dynamic Conditional decision activation of the CCL:

1. Introduce on the CCL IOC, an attribute representing the set of conditions to be monitored for activation of the CCL decisions. The condition may represent the context under which the CCL may execute actions or not. The attribute may be of type threshold monitor defined in 3GPP TS 28.622 [11], condition monitor as defined in 3GPP TS 28.622 [11], the condition expressed in form of a JEX/XPATH expression.

NOTE 1: The conditions could alternatively be part of the goal.

1. Introduce on the CCL IOC, an attribute representing the triggered behavior. This will define the corresponding behavior of the CCL. The behaviors can be represented by an ENUM to include:

- DECISION\_ACTIVATION: The Loop executed the recommendations that it derives on to the network.

- NOTIFY\_RCOMMENDATION: The Loop 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.

NOTE 2: The internal action of the CCL is not communicated to the MnS consumer.

- DO\_NOTHING: do not do anything.

NOTE 3: The trigger behavior may be added into the MnS consumer's defined condition.

### 5.2.4 Evaluation of solutions

It is recommended to express conditions triggering the creation of a CCL using the method that is used to express conditions for other use cases: Jex in HTTP/JSON and XPath in NETCONF/YANG.

For enabling to express all conditions required in the context of CCL creation, the stage 2 and stage 3 enhancements outlined in clause 5.2.3 are recommended to be standardized.

Furthermore, it is recommended to use conditional activation of planned configurations (3GPP TR 28.872 [7], clause 5.6) as a solution for the use case Triggered CCL.

## 5.3 Use case 3: CCL creation based on Historical CCL data

### 5.3.1 Description

This use case describes the need of maintaining information about the CCLs that existed in the past. Those CCLs are called Historical CCLs.

In an automation environment, before a consumer request to create a CCL it would like to know the data related with Historical CCLs that were available with the producer. This information will enable consumer to request for an optimal CCL. The information about historical CCL may include, scope of the CCL, configured goals/targets, controlled entity, etc.

Further, Historical CCL information serves as a valuable data source for predictive analytics within the CCL system executed as Analytics step. It enables the system to move from a reactive mode, where it responds to current issues, to a proactive mode, where it anticipates and prevents problems based on historical trends and patterns. This proactive approach enhances network reliability, minimizes downtime, and improves the overall efficiency of network operations.

The existing CCL mechanism has no means to enable historical CCL information that can be used to predict potential network issues and take proactive measures to prevent them. The absence of historical CCL information can be a significant limitation in network automation.

The breach and feedback information is provided in clause 5.3.3.

### 5.3.2 Potential Requirements

REG-HIS-REQ: The 3GPP management system should enable authorized MnS consumer to request for information (e.g. CCL identification, configured goals/targets and the related status, scope of the CCL, conflict information) related with Historical CCL.

### 5.3.3 Potential Solution

The solution involves introducing <<datatype>> (e.g HistoricalCCLInfo) to contain historical CCL information that can be queried by the consumer to understand the information related with previous CCL including the following:

1. CCL Information:

- CCL Identification.

- Initial Goals and Targets: It provides the initial goals/targets set provisioned for the CCL.

- Intermediate Goals and Targets: It provides the set if intermediate goals/targets set provisioned for the CCL.

- Last Goals and Targets: It provides the last goals/targets set provisioned for the CCL.

- CCL Scope: It indicates the scope of CCLe.g in terms of a location.

1. Breach Information related with goal breach. There will be multiple instance of this datatype for each breach instance:

- Time of breach: The time at which the breach happened.

- Breached Goals and Targets: The goal which got breached.

- Action Taken: The action(s) that was taken to mitigate the breach. This will provide the list of operations performed, MOI effected and attributes set/modified.

Editor's Note: The attributes in CCL information and Breach Information should be aligned with the attributes of CCL in other CCL solutions

Figure 5.3.3-1 shows the procedural flow.



Figure 5.3.3-1

1. Producer instantiate and provision a CCL as defined in 3GPP TS 28.536 [4].

2. Consumer send DeleteMOI request for a CCL.

3. Producer sends a response.

4. Producer either instantiate or modify the HistoricalCCLInfo MOI with the information related with CCL being deleted.

5. Consumer may decides to initiate a CCL. Before that it would like to understand the historical CCL information.

6. It send getMOIAttributes for HistoricalCCLInfo MOI to read the information captured.

7. Producer send a response.

8. Consumer develops the learning based on the historical CCL information received.

9. 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.

10. Producer send a response.

NOTE: The above procedure flow is for illustration only. It assume that the proposed information is modelled as an IOC. The actual modelling of the information will be decided as part of normative work, that may change the procedure flow.

## 5.4 Use case 4: closed control loop for problem recovery

### 5.4.1 Description

Based on the concept in 3GPP TS 28.104 [3], MDA reports may contain root cause analysis of ongoing issues, predictions of potential issues and corresponding relevant causes and recommended actions for preventions, and/or prediction of network and/or service demands. For example:

1. MDA for Coverage problem analysis can provide the following information in the MDA report:

- coverageProblemId;

- coverageProblemType;

- coverageProblemAreas; and

- recommendedActions.

1. MDA for Energy saving analysis can provide the following information in the MDA report:

- energyEfficiencyProblematicObject;

- energyEfficiencyProblemType;

- rANenergySavingRecommendations; and

- cNenergySavingRecommendations.

MnS consumer may make a decision to resolve the observed problems based on the analytics reports (e.g. provided by MDA) and other management data (e.g. historical decisions made previously) if necessary. It can be possible that one MnF (e.g. Domain MnF) is responsible for problem observation and recovery, while another MnF (e.g. Cross Domain MnF) is responsible for decision on whether the problem needs to be resolved. In this scenario, The Cross Domain MnF can decide whether needs the Domain MnF to recovery the observed problems (e.g. coverage problem) based on MDA report (e.g. root cause information, recommended solutions) and other information (e.g. user experience information, information from other domains). If decides to recovery the observed problems, Cross Domain MnF needs to request Domain MnF to recovery the specified problems observed from the MDA report by using closed control loop. MnS consumer may specifies the time window for problem recovery, which means the MnS producer needs to recovery the problem at the specified time window. During problem recovery phase, MnS consumer also needs to be obtain the progress information for the problem process. When the last step of the problem process is completed, MnS producer needs to send the result of this problem recovery process to the MnS consumers.

If a closed control loop instance can be used to resolve network problem, the MnS consumer may need to know the result of resolving the network problem by the closed control loop instance, including the network problems which are resovled by the closed control loop as well as network problem resolution statistics (e.g. the number of network problem resolved by the closed control loop in the specified period).

### 5.4.2 Potential requirements

**REQ-CSA-CON-1:** The 3GPP management system should have the capability to allow the MnS consumer to request a CCL for resolving the network problems identified in the MDA report.

**REQ-CSA-CON-2:** The 3GPP management system should have the capability to allow the MnS consumer to get the result of network problem resolved by the closed control loop.

### 5.4.3 Potential solutions

It proposes to introduce the ProblemRecoveryRequest IOC name contained by AssuranceClosedControlLoop defined in 3GPP TS 28.536 [4] to represent the MnS consumer's requirements for resolving network problem. The ProblemRecoveryProfile IOC may include following attribute:

1. ProblemIdList, a list of problem Ids to indicates which network problems are requested to be resolved by the closed control loop instance. The ProblemId is defined in 3GPP TS 28.104 [3].
2. ProblemRecoveryPolicyList, a list of ProblemRecoveryPolicy <<dataType>> for resolving the network problems. Each ProblemRecoveryPolicy <<dataType>> indicates which network problems are requested to be resolved by the closed control loop instance automatically. The ProblemRecoveryPolicy <<dataType>> can includes following attributes:

- networkProblemTypes, represent which network problem types (e.g. coverageProblem, energyEfficiencyProblem) are requested to be resolved by the ACCL. The allowed values for networkProblemType are defined in 3GPP TS 28.104 [3].

- networkProblemAreas, represent the network problem identified in the specified geographical areas are requested to be resolved by the ACCL.

- networkProblemOccurTimes, represent the network problem identified in the specified time windows are requested to be resolved by the ACCL.

It proposes to introduce the ProblemRecoveryReport IOC name contained by AssuranceClosedControlLoop defined in 3GPP TS 28.536 [4] to represent the result of network problem resolved by the closed control loop. The ProblemRecoveryResult IOC may include the following attributes:

- reportTimeWindow, indicates the ProblemRecoveryResult is observed in the specified time window.

- problemIdList, a list of problem Id to indicates which network problems are resolved by the closed control loop instance. The ProblemId is defined in 3GPP TS 28.104 [3].

### 5.4.4 Evaluation of potential solutions

Only one potential solution is proposed, which is feasible.

## 5.5 Use case 5: CCL for fault management

### 5.5.1 Description

Current fault management has some issues.

Alarms contain information like probableCause, specificProblem, and rootCauseIndicator, etc. However, in some scenarios, it is difficult for operators to directly identify the root cause just from these alarm attributes. Additional troubleshooting steps are usually required to dig into the root cause.

Fault management procedures often require operators to follow a sequence of troubleshooting steps to narrow down the issue before they can identify the root cause. This multi-step process introduces complexity and is resource intensive. Thus, fault recovery response time and network and service downtime may be prolonged.

CCL can be extended to automate and optimize fault management:

- Monitor: PM/KPIs，performance threshold monitoring events and fault supervision events.

- Analyse: analyse fault alarms and correlate them with other PMs/KPIs etc. to identify likely root causes.

- Decide: provide automated decision making according to the fault root causes and propose fixing solutions.

- Execute: execute through provisioning services to fix network and service faults.

### 5.5.2 Potential Requirements

**REQ-CCLM\_ FAULT -01:** The 3GPP management system should have the capability to identify the root cause and take actions using closed control loop to mitigate or solve the root cause.

### 5.5.3 Potential Solutions

#### 5.5.3.1 Solution-x

#### 5.5.3.2 Solution-y

### 5.5.4 Evaluation of solutions

## 5.6 Use case 6: CCL conflicts management

### 5.6.1 Description

#### 5.6.1.1 CCL Conflicts

Multiple CCLs could co-exist and concurrently act within the same environment. The CCLs can affect one another, in the worst cases leading to conflicts. The different kinds of conflicts are summarized by Table 5.6.1.1-1.

Table 5.6.1.1-1: Types of potential conflicts among CCL instances for goals g1, g2 and g3

| Conflict Type | Description | CCL-A | CCL-B | Comments |
| --- | --- | --- | --- | --- |
| Target Conflict | For CCLs C1 and C2, when same at least 1 target of a goal is present in both CCL asking for different outcomes on that target on same controlled entity (ME1). | Control Scope: ME1  Goal targets:  - Load > 90 % (to maximize resource utilization)  - Latency < 10 ms | Control Scope: ME1  Goal target:  - Load < 90 % (to avoid congestion) | Conflict among the targets within the goals - due to different required target outcomes |
| Action Conflict | For CCLs C1 and C2, when both C1 and C2 is trying to configure the same characteristics of same target entity (gNB‑g1) in contradiction. | Example 1 | | Conflict due to configuration actions at execution step because both CCL want different contradicting value for a particular characteristic of gNB-g1.  Effect: even when executed at different times, the value may ping-pong continuously. |
| Goals target:  - Throughput > 10 gbps  Actions:  - Target Entity: gNB-g1  - Target Change: scale-out virtual resource | Goals target:  - EC is < 10 KVA  Actions:  - Target Entity: gNB‑g1  - Target Change: scale-in virtual resource |
| Example 2 | |
| Goal target:  - HO failure is < 2 %  Actions:  - Target Entity: gNB-g1  - Target Change: set CIO to a small **positive** value{to guarantee HOs with low chances of HO failure} | Goal target:  - Load < 80 %  Actions:  - Target Entity: gNB-g1  - Target Change: set CIO to a small negative value [to advance HOs and move load to other cells] |
| Indirect target conflict | For CCLs C1 and C2, when C1 [optimize handover] and C2 [minimize interference] have different goals but the actions of C1 affect the goals of C2. | Goal target:  - HO failure is < 2 %  Actions:  - Target Entity: gNB-g1  - Target Change: reduce CIO {to reduce chances of HO failure} | Goal target:  - SINR > 10 dB  Actions:  - Target Entity: gNB‑g1  - Target Change: lower antenna tilt | By reducing antenna tilt to minimize interference C2 affect the HO goal target of C1 |
| Action Execution Time Conflict | For CCLs C1 and C2, when both C1 and C2 are trying to configure the same characteristics of same target entity (gNB-g1) in contradiction. | Goals:  - Throughput > 10 gbps  Actions:  - Target Entity: gNB-g1  - Target Change: scale-out  - Target Time: 04:00 | Goals:  - EC is < 10KVA  Actions:  - Target Entity: gNB-g1  - Target Change: scale-in  - Target Time: 04:00 | Conflict due to the time of executing the configuration actions at the execution step |
| Scope conflict | For CCLs C1 and C2, C1 and C2 have different goals and actions but their scopes are overlapping - e.g. C1's control scope (i.e. the controlled entities in the network) is part of C2's measurement scope (i.e. the measured entities in the network). | Measurement scope:  - cells g1  Control Scope:  - g1  Goal targets:  - EC/bit is < 1WA  Actions:  - Target Entity: gNB-g2  - Target Change: switch off g2 | Measurement scope: cells g1, g2, g3, g4  Control Scope:  - g2  Goals:  - Load < 80 %  Actions:  - Target Entity: gNB-g2  - Target Change: change CIO | By switching off g2, C1 affects the scope which C2 reads for its load distribution measurements |

The CCL may detect or observe events that identify the possibility of any one of the above conflicts. The conflict can be avoided using some information or the policies (e.g. priority) provided by the consumer. If the conflict actually occurs, the CCL MnS producer should support services to inform MnS consumers the confirmed detected conflicts. This may also include informing MnS consumer about the potential conflict.

#### 5.6.1.2 CCL conflicts management and coordination interactions

The coordination of CCLs includes the management services needed to detect, resolve, or avoid conflicts among goals and their targets, , control scopes or actions of the CCLs. To address the different conflict situations, coordination capabilities could be required for the following scenarios:

- Capabilities to identify different interaction types between CCLs such as cooperation (positive interaction), conflict (negative interaction) or dependency (neutral interaction).

- Capabilities to align targets among the goals of individual CCLs sharing a given scope.

- Capabilities to identify different types of conflicts between CCLs such as parameters conflict, metrics conflict, or any others.

- Capabilities to address the different interactions between CCLs with adequate mechanisms, such as conflict resolution mechanisms.

- Capabilities to identify before the execution of a proposed action of CCL that such an action could cause undesired effects to other CCLs or to managed entities (e.g. pre-execution and post-execution coordination, concurrency coordination, etc.).

- Capabilities to evaluate the impact and effectiveness of CCLs actions after their execution (e.g. impact assessment).

The coordination of CCLs could be required at different execution points of the CCL translating into different CCL coordination use cases with corresponding CCL coordination services required at those points as illustrated by example Figure 5.6.1.2-1. The coordination of CCLs could be achieved via direct interaction among the CCLs or via a third-party entity, say called the CCLs coordination Function (or simply CCL Coordinator).



NOTE: The terms at the top indicate general naming of the groupings of coordination interactions at the different execution points during the execution of the CCL. Action-space coordination implies coordinating the sets of actions that the different CCL can apply. Concurrency control implies coordinating the times at which different CCLs can execute actions. Action-impact assessment indicates interactions and processes on the evaluation of the impacts of the different CCLs.

Figure 5.6.1.2-1: Exemplary Closed Control Loop Coordination interaction points

#### 5.6.1.3 CCL conflicts resolution

##### 5.6.1.3.1 CCL Goal-conflicts resolution

The targets in the goals of Closed Control Loop should not contradict one another within that goal or contradict with other targets in goals of related CCLs, otherwise a goal conflict is observed. For such a goal conflict, goal coordination interactions are needed to resolve the conflict, i.e. to align goals (and related targets) that should be achieved by the various deployed Closed Control Loops. Given the potentially high number and diversity of Closed Control Loops, the process of setting and coordination goals for the Closed Control Loops should be accomplished using another CCL that consumes the CCL-related monitoring and governance services to coordinate the resolution of conflicts with the CCL.

The MnS producer for this CCL instance should inform the MnS consumer about a candidate goal conflict, e.g. about the values of the goal's targets that are in conflict with the targets of another goal. In response, the MnS consumer could revise the goals of that CCL instance, terminate the execution of the CCL instance, delete the CCL instance.

##### 5.6.1.3.2 CCL Trigger-time conflicts resolution

Typically, a CCL will be triggered to run at a specific time and terminate when certain conditions are met, to run when a certain performance threshold is crossed. If triggered independently, there may be conflicts among the CCLs. The triggers for different CCLs to be executed need to be coordinated to avoid conflicts among the CCLs. And in some instances, the conditions in the network may be such that it is not clear which CCL should be triggered, requiring to trigger multiple CCL in sequence. The triggering may be done by a coordination function that consumes the CCL-related monitoring and governance services to receive information with which to evaluate the conditions and determines which CCL to be triggered.

It may be the case that CCLs need to operate in a hierarchy with each CCL having an operational profile indicating the specific level of hierarchy. The operational profile describes characteristic sunder which the CCL operates, e.g. when or after which other CCLs, this CCL should be executed. For example, to ensure that handovers are always optimal, a CCL on handover optimization may need to be triggered every after a CCL on Energy saving has been executed to be sure that there are appropriate handover relations even when some cells may have been disabled. The MnS consumer that coordinates the execution times of the CCLs needs to configure the appropriate hierarchy for the CCLs. Using the operational profiles of the CCLs, , the MnS consumer evaluates the description of the third CCL against at least one of the profiles P1 and P2 and accordingly determines and configures the operational profile of the third CCL.

NOTE: A CCL may be involved in more than 1 hierarchies or within a single hierarchy, the CLL may relate to multiple other CCLS, which requires the hierarchies to be coordinated.

### 5.6.2 Potential Requirements

**REQ-CCL-CONFLICT-1:** The MnS Producer for CCL management should support a capability to detect a potential or actual conflict.

NOTE: A potential conflict is where some events are observed that indicate that there may be a conflict, but the CCL MnS Producer cannot conclude that it is a conflict. So, the CCL can indicate this so that some other entity e.g. the MnS consumer takes responsibility to confirm the conflict.

**REQ-CCL-CONFLICT-2:** The MnS Producer for CCL management should support a capability to inform an authorized MnS consumer about a potential conflict that has been detected.

**REQ-CCL-CONFLICT-3:** The MnS Producer for CCL management should support a capability to confirm a detected potential goal, action, indirect target, action execution time, scope conflict.

**REQ-CCL-CONFLICT-4:** The MnS Producer for CCL management should support a capability to resolve a goal, action, indirect target, action execution time, scope conflict that has been detected.

**REQ-CCL-CONFLICT-5:** The MnS Producer for CCL management should enable authorized MnS consumers to provide information that can be used to avoid the conflict.

**REQ-CCL-CONFLICT-6:** The MnS Producer for CCL management should enable authorized MnS consumers to provide information that can be used to resolve the conflict.

**REQ-CCL-CONF\_RES-1:** The MnS producer should support a capability to coordinate the resolution of conflicts on the CCLs goals.

**REQ-CCL-CONF\_RES-2:** The MnS producer should support a capability to coordinate the resolution of conflicts on the triggers for execution of the CCL instances.

**REQ-CCL-CONF\_RES-3:** The MnS producer should support a capability enabling an MnS consumer to define and coordinate the hierarchies of the CCL.

### 5.6.3 Potential Solutions

#### 5.6.3.1 Alternative CCL coordination Approaches

The coordination of CCLs could be accomplished via one of three approaches illustrated by Figure 5.6.3.1-1:

- Distributed coordination with distributed execution (Figure 5.6.3.1-1 a), where the CCLs directly coordinate with one another, and each manages execution of its decisions.

- Hierarchical coordination with distributed execution (Figure 5.6.3.1-1 b), where the CCLs coordinate through a separate coordination layer, say via a coordination CCL, but each manages execution of its coordinated decisions.

- Hierarchical coordination and execution (Figure 5.6.3.1-1 c), where the CCLs coordinate through a separate coordination layer, say via a coordination CCL that besides coordination also manages execution of the coordinated decisions.



Figure 5.6.3.1-1: Closed Control Loop Coordination approaches

Distributed coordination can lead to too many exchanges between the CCLs which may unnecessarily clog the system. On the other hand, "Hierarchical coordination and execution" implies that too much responsibility is concentrated in a single CCL. A desired behavior is that the individual CCLs are responsible for their own decision execution, so it is recommended that to follow the "hierarchical coordination with distributed execution" approach. In this approach, the CCLs are responsible for making their decisions and executing actions but they coordinate with the CCL coordinator before, during or after execution.

The CoordinationCCL supports interactions with different CCLs for the following CCL conflicts

- Goal target Conflicts.

- Scope conflicts.

- Direct-action conflicts.

- Indirect target conflict.

- Action execution-time conflict.

For each conflict, the CoordinationCCL supports interactions to:

- Detect potential conflicts.

- Avoid potential conflicts.

- Detect actual conflicts.

- Resolve actual conflicts.

Accordingly, as illustrated by Figure 5.6.3.1-2, an IOC could be introduced for the coordination CCL with child IOCs for the specific capabilities for each conflict type.



Figure 5.6.3.1-2: Closed Control Loop Coordination NRM fragment

Closed Control Loop Coordination Inheritance Relations are For Further Study.

#### 5.6.3.2 General solution

The solution provides a baseline for all conflicts and needs (avoidance, detection and resolution) on which the specific solutions can be added. It involves introducing an IOC or datatype to contain conflict related information and mechanism to mitigate any conflict between CCL that may arise during instantiation of a new CCL or between two existing CCL. This IOC will also support interactions with different CCLs to detect, avoid and resolve all the conflicts defined in Table 5.6.1.1-1.

Extend the ACCL report to be a general report that applies to all types of CCLs. The alternatives are:

- Rename ACCL report to CCLReport and then extend the new CCL report.

- Introduce the CCLReport as an abstract IOC from which the ACCL report inherits.

To support reporting for conflict:

1. Introduce attributes for target conflict information through which the producer provides information about the conflict between an existing CCL and a requested CCL. The producer provided information includes:

a) For all conflicts:

1. Existing and new CCL identification.

b) For goals/target conflicts:

1. Conflict Information (conflicting goals/target).
2. Target CCL: The identification of the CCL that need to be deleted or modified. This will be decided as per the conflict resolution information.

c) For Action Conflict Information: This provides information about the conflict between two existing CCL:

1. Conflicting CCLs identification.
2. Conflict Information:

- Conflicting Goal.

- Conflicting Execute actions: This provides the set of actions that have been taken by the CCL as part of the Execute step.

1. Target CCL: The identification of the CCL that need to be deleted or modified. This will be decided as per the conflict resolution information.

To support resolution of conflicts:

1. Introduce on the CCL a datatype for conflict resolution information. Through this, the consumer provides information that may be used by the producer to resolve conflicts. It includes:

- For all conflicts:

1. Priority: This provides the priority of the CCL. This will be the numerical value between 1 to 10, with 1 being the least priority.
2. OverridingCapable: Whether the CCL can override other CCL.
3. OverrideProtect: Whether CCL can be overridden.

- For Action Conflicts:

1. GoalBreachPercentage: In case the priority of both the conflicting CCL is same then it defines the breach percentage per goal, in terms of how bad the goal(s) is breached, that should be used to prioritize one of the conflicting CCL. For example, if the goal of guaranteed throughput is 200 mbps and the actual throughput is coming to be 100 mbps then the breach percentage would be 50 %. The CCL that have higher percentage of breach will be prioritized.

Procedure flow



Figure 5.6.3.2-1

1. The CCL(s) are deployed and running.

2. The interactions happens for detection and avoidance of the conflicts.

3. Producer checks for the conflicts that has already happened. This will include checking for both Target Conflict and Action Conflict. Producer will decide which CCL is to be deleted, among the conflicting CCL, as per the conflict resolution logic provided by the consumer.

4. The producer send a notification to the consumer providing details on the conflict. This notification will also identify the CCL which need to be deleted in order to mitigate the conflict.

5. Based on the recommendation the consumer may delete a CCL. Consumer sends a deleteMOI request for the same.

6. Producer sends a reply.

7. Alternative to 6 above, consumer may decide to modify the CCL instead of deleting it. It sends modifyMOIattribute request.

8. Producer sends a response.

#### 5.6.3.3 Goal targets coordination

##### 5.6.3.3.1 Required capabilities and interactions

This solution focusses on the requirement on goal targets conflicts as well as detection and resolution of actual goal targets conflicts

CCL instances will be responsible for related, adjacent or in some cases overlapping scopes. In such cases it is good to ensure that the goals of two CCLs are not contradictory or conflicting or leading to contradictory or conflicting outcomes. The goal may be seen as a set of network measurements and KPIs (i.e. targets) to be concurrently achieved by the CCL. A coordination layer, say a coordination CCL, may have a goal management capability responsible for managing and optimizing the goals of the CCLs based on general objectives for the network scope, where the general objective describe the priorities among the different target values on the specific KPIs. For example, as illustrated by Figure 5.6.3.3.1-1, the input network scope objectives may simply require ensuring, for KPI K1, that with priorities, p1, p2, p3, the value of K1 should respectively be less than values V1, V2, V3. This may be provided for different KPIs on a network level (e.g. by the operator). Then, the goal management functionality compares these objectives to choose the appropriate KPI targets for each CCL. In the example, the CCL goals are set as K1 < 0.1 and K2 > 85 %.



Figure 5.6.3.3.1-1: Example prioritized goal targets on a set of KPIs that need to be coordinated  
among a group of CCL instances.Note: this solution assumes that the scopes are prefixed but  
the outcomes of the solution may a recommendation to adjust the scopes

To support detection and avoidance of potential goal targets conflicts:

- The CCL may register its goal targets with the coordinationCCL which triggers an evaluation of potential conflict, i.e. whether those targets are likely to conflict with the targets of another CCL.

- In case of a potential conflict, the goal management functionality of coordination CCL sends the selected new or revised goal targets to each CCL ensuring to minimize contradictions or conflicts among the targets in the different goals of different CCLs, for example, that for a given scope a specific target is assigned to only one CCL.

To support detection and resolution of actual goal targets conflicts:

- The CCLs attempt to fulfil its set targets, and where they ae unable to, the CCL sends feedback to the goal management functionality in the coordinationCCL indicating which targets cannot be fulfilled. A CCL may for example indicate that there are ping-pong effects on a target, i.e. whenever the target is pushed in a given direction, it flips back to a previous state. The flipflop is an indication of a potential goal conflict which the CCL should notify to the goal management.

- Based on the feedback, the goal management functionality acting as CCL MnS producer can then confirm the existence of goal-target conflict and may revise the targets by setting new target values.

##### 5.6.3.3.2 Information objects to realize required capabilities and interactions

The coordinationCCL should be extended with the capability to for coordinating CCLs goal:

1. Introduce an attribute on the coordinationCCL to capture the goal set of a CCL instance. A CCL that requires its goals to be evaluated for conflicts can add its goal set into the list of goal sets:

- For each introduced goal set, introduce a Boolean attribute to indicate if a potential conflict is observed for the goal set.

1. Introduce a datatype and corresponding attribute on the coordinationCCL to represent the the full set of goal-targets and their priorities for a given scope from which individual CCLs may be assigned their goal targets. This set may be called networkScopeObjectives and are used by the coordinationCCL to identify instances in which 2 CCLs have goals or goal targets that are conflicting.
2. It can also be used to decide how to reallocate the goal targets in away that avoids or minimizes conflicts.

Extend the assuranceGoal dataType with information to support goal coordination:

1. Add for each target in the goal target list an attribute to reflect the value of achievement of the target.
2. Add for each target in the goal target list a Boolean attribute, say called flipFlopNoted to indicate if flipflops are observed on the target. The flipFlopNoted can be notifiable which provides a way for the CCL to indicate to the coordinationCCL that there are flip flops observed on the goal target.

#### 5.6.3.4 Direct actions conflicts

##### 5.6.3.4.1 Detection and avoidance of actions conflicts

5.6.3.4.1.1 Required capabilities and interactions.

If two CCLs execute their actions within the same time period, the actions could cause undesirable effects, e.g. by conflicting for the same parameter on the managed entities. Before the interacting CCLs execute their actions, the Pre-execution coordination can be used to detect potential direct-actions conflicts.

To detect and avoid potential direct-actions conflicts:

- A CCL intending to take an action, sends its proposed configuration management changes to the coordination CCL prior to execution of those configuration changes. The configuration changes contain information of target resources and scheduled time for execution.

- The coordination CCL checks the submitted configuration changes against other previous configuration changes from other CCLs (that have been executed) to see if there are any potential conflicting actions based on the provided information. This ensures to check planned configuration changes against actions that have already been executed.

To avoid potential direct-actions conflicts:

- The coordination CCL notifies the detected conflict(s) to the related CCLs.

- The CCL may adjust its planned configurations to a new set that could have less conflicts.

5.6.3.4.1.2 Information objects to realize required capabilities and interactions

- Introduce a datatype, and related attribute on the CCL, representing the CCL desired changes. The desired changes, indicates the objects that are planned to be configured, the attributes on those objects that would be configured, the values to which they would be configured and the time at which those plans are expected to be executed. The CMPlan should be notifiable, the coordination CCL is notified by the CCL when the action has been drawn up.

- Introduce an attribute on the CCL , say called detectedConflict, representing a conflict for a given action in the plannedAction. The detectedConflict may be a Boolean flag which is by default FALSE but is toggled to TRUE when a conflicts is detected. It may also be pair which adds information about the other actions to which the said action conflicts.

NOTE: After the potential conflict is detected, a different solution is needed to resolve those conflicts e.g. using priorities among CCLs.

##### 5.6.3.4.2 Detecting actual conflicts based on counter-productiveness

5.6.3.4.2.1 Required capabilities and interactions

In certain cases, two CCLs may work together on the same managed entity, maybe at different times scales or involving different aspects/sub functionalities of the managed entity. However, there may be some known or unknown interdependence between actions taken by the two CCLs, E.g. in the cases where the scopes of the two CCLs cannot be separated. Furthermore, two CCLs may change the same parameter one after the other. For multi-aspect optimization, such interdependence is often expected, and it should be tolerated by the system as long as it is not harmful in terms of the overall performance of the managed entity.

A way to detect actual conflicts and minimize their impacts is detect counter-productiveness. CCLs that operate on the same managed object monitor any counter-productiveness and if observed, maintain it within some tolerance limits.

For this a CCL instance A that is likely to be affected, needs to monitor a specific scope or context that could be affected by another CCL instance B. CCL B can provide a conflict monitoring context/scope to CCL A informing CCL A about CCL B's latest actions on the managed entity and its tolerance w.r.t to its parameters and metrics in this managed entity. CCL A (the context recipient CCL) should work within these bounds, i.e. its actions should not violate the said tolerances to avoid counter-productiveness. CCL A observes the conflict monitoring context, so that if it observes the violations of the said tolerances, it reports the conflict to the CCL B.

5.6.3.4.2.2 Information objects to realize required capabilities and interactions

- Introduce an attribute on the CCL representing the scope that is being monitored for counter productiveness by the CCL, say called CCLMonitorScope. The CCLMonitorScope is notifiable, when an action is executed, the responsible CCL (acting as CCL B) updates the CCLMonitorScope and notifies the other CCLs (e.g. via the coordination CCL) about the current CCLMonitorScope.

- Introduce an attribute in the CCLMonitorScope for the monitored parameters, say called monitoredScopeParameters with corresponding tolerance for each monitored parameter, say called monitoredScopeParameterTolerance. The monitoredScopeParameters and their monitoredScopeParameterTolerance are notifiable, notified together with the CCLMonitorScope notification.

##### 5.6.3.4.3 Bargaining as resolution of potential direct actions conflicts

5.6.3.4.3.1 Required capabilities and interactions

The simplest way to resolve direct parameter conflicts is to separate the control spaces of the CCLs, i.e. to allocate each parameter to a specific CCL. However, this is not guaranteed to always be possible, i.e. in some cases two or more CCLs may want to set different values for the same parameter and the parameter cannot be assigned to only one CCL. In these cases, a coordinator functionality, e.g. a coordinator CCL should compute a compromise value for the parameter, a value which can be considered to be equally good for all the CCLs. However, since different CCLs have different goals, it is necessary for the coordinator CCL to understand the importance of the parameter to each CCL. For this purpose, the CCLs provide their usefulness for the parameter to the coordinator CCL. The usefulness provided by a CCL shows the relative goodness of different values of the parameter to the CCL in a pre-defined scale, e.g. [0:1]. Since all the CCLs used the same scale, when the CCL coordinator selects a parameter value, it can clearly understand how important this value is for each CCL. The CCL coordinator can then derive the compromise values which is then (provided to the CCLs to be) executed onto the managed object. An example way to compute the compromise is to use the Nash Social Welfare Function since it provides equal fairness to all competing entities.

A compromise based only on usefulness does not consider the relative (level of) interest of the CCLs in the parameter. To account for the interests, the CCLs should provide to the CCL coordinator their relative interest in the parameter, so that the computed compromise value accounts for the combined interests of the CCLs. The relative interest may be computed based on a fixed scale. For example, for a CCL on cell interference management on a scale of [0-10], a cell's transmit power has a goodness of say 9 than the cells load which has a goodness of 3.

NOTE 1: The coordination CCL does not have to calculate the compromise value all the time as this requires information exchange among the CCLs and computational energy. It should be possible to configure the coordination CCL such that it calculates the compromise values only when certain conditions are met. The coordination CCL should be able to expose required services to the MnS consumer to configure such conditions.

NOTE 2: For a given CCL, the usefulness may be equivalent to the level of interest but it is not always the case. It is possible that a CCL has high interest in a parameter that has low usefulness.

5.6.3.4.3.2 Information objects to realize required capabilities and interactions

1. Introduce for each control parameter of a CCL, an attribute presenting the usefullness of that parameter. The usefullness may be called parameterUsefullness , indicates the utility of different values of the parameter to the CCL:

- The parameterUsefullness should be notifiable, so that when the CCL sends its action plan, it can notify the coordinator CCL of the parameterUsefullness.

1. Introduce for each control parameter of a CCL, an attribute representing the compromise computed by the coordinator CCL.
2. Introduce for each control parameter of a CCL, an attribute presenting the degree of interest of the CCL in that parameter. The interest may be called parameterInterestLevel, indicates the CCL's level on interest in the parameter. For a given CCL, the usefulness may be equivalent to the level of interest but it is not always the case. It is possible that a CCL has high interest in a parameter that has low usefulness:

- The parameterInterestLevel should be notifiable, so that when the CCL sends its action plan, it can notify the coordinator CCL of the parameterInterestLevel.

#### 5.6.3.5 Indirect targets conflicts

##### 5.6.3.5.1 Detecting potential and actual indirect targets conflicts

5.6.3.5.1.1 Required capabilities and interactions

Two CCLs (CCL1 and CCL2) may optimize 2 target metrics Y1 and Y2, e.g. one intending to ensure "HO failure is < 2 %" while the other wants "SINR > 10dB". Due to coupling between Y1 and Y2, actions to optimize these by CCLs may lead to correlated oscillations/degradations in Y1 or Y2. The correlated oscillations indicate a potential conflict, but the CCLs may not see the oscillations in the metric that is not of their interest. The coordinator CCL may analyse the behavior of Y1 and Y2 to see if there are correlated oscillations as result of actions by CCL1 and CCL2 which then indicates potential conflict between CCL1 and CCL2. When the oscillations are observed, the coordination CCL MnS producer should be able to inform the related MnS Consumer(s) (i.e. CCL1 and CCL2) about the detected potential conflict represented by the correlated oscillations.

For detected potential conflict the CCL coordination service producer needs to confirm that it is an actual harmful conflict. This can be determined based on the severity of degradation in the performance metrics of the related CCLs. The threshold to determine the severity may be defined by the MnS consumer (e.g. the operator or coordinator CCL). If the degree of degradation is higher than the threshold then it is a confirmed conflict that requires resolution. Otherwise, no action is needed.

5.6.3.5.1.2 Information objects to realize required capabilities and interactions

1. To provide information on potential conflicts, introduce a datatype and corresponding attribute on the CCL (specifically on the coordination CCL) representing a detected potential conflict. It may be called detectedPotentialConflict and includes a list of targets which have been detected to have correlated oscillations and thus likely to be conflicting:

- The detectedPotentialConflict is a list that is notifiable; when an entry is added, a notification is sent to the CCLs who metrics are monitored by the respective coordination CCL.

1. To support confirmation of potential conflicts as actual conflicts, introduce an attribute for each goal target on the CCL that represents the threshold for the severity of degradation in the performance metrics at which a real conflict is declared by the CCL. The threshold is the percentage by which the performance metrics have to change form their desired value for the CCL to declare that the change is due to another CCL affecting the target, but not other "normal" changes.

##### 5.6.3.5.2 Avoiding indirect targets conflicts

For a detected indirect targets conflict, the coordinator CCL can trigger one or more CCLs to respond to the detected potential conflict. If the CCLs that has been requested to resolve potential conflict is unable to resolve that conflict, the CCL should inform the coordination CCL MnS producer about the failure to resolve the problem:

- Add in the detectedPotentialConflict, an attribute, say called resolutionCCL, that represents the CCL that should take action for the respective detectedPotentialConflict. The resolutionCCL is notifiable, when updated a notification is sent to the related CCL so that the CCL(s) whose DN appears as resolutionCCL can then start the resolution process.

- Add in the detectedPotentialConflict, an attribute, say called prioritizedCCL, that represents the DN of the CCL that has been prioritized over the others.

##### 5.6.3.5.3 Avoiding potential indirect target conflicts through likely-impact of planned actions

5.6.3.5.3.1 Required capabilities and interactions

For any CCL, large and frequent changes to network parameters may affect network stability since they increase the probability of occurrence of conflicts, i.e. avoiding making unnecessary configuration changes to the managed objects guarantees network stability and minimize the probability of conflicts between CCLs. This may then imply that executing large changes, e.g. to quickly improve the performance, in case of a poor decision, may also result in significant degradation. So, it is preferred to take small smooth changes in the case where the impact is not so clear, and only make the large changes when the CCL is sure that the impact is positive.

For any planned action, the CCL sends to the coordinator CCL the planned change, its claimed/predicted performance improvement and reliability/confidence in that action/decision. The coordinator CCL evaluates the claimed performance improvement and reliability/confidence to determine if the action should be allowed or not. This ensures to avoid counter-productive actions - if the CCL demands to make large changes, it proves high reliability/ confidence and significant improvement in performance. The criteria applied by the coordinator CCL to match acceptance/rejection of a planned action to the reliability and performance may be implementation-specific or defined by the operator.

The coordinator CCL then sends the decision and possibly the failed criteria to the CCL - to either be executed or to be used to compute better decisions. It is assumed that based on feedback on the quality of its decisions, the CCL updates it decision-making engine and repeats the decision evaluation process. Then if the CCL has consistently made good large action-decisions, the coordinator CCL can consider the CCL as trusted to make such large decisions. The coordinator CCL informs the CCL that the CCL has consistently made good decisions and achieved its ultimate trust.

5.6.3.5.3.2 Information objects to realize required capabilities and interactions

To support avoidance of indirect target conflicts by evaluating likely-impact of planned actions:

1. Re-use the attribute for planned action, say called CMPlan. A CCL can request a coordinator CCL for an evaluation of the CMPlan:

- The CMPlan includes information on the desired change, the predicted impact/effect of the decision on the related metrics as well as the CCL's confidence in that decision.

1. Introduce an attribute representing the coordinator CCL's evaluation of the CM plan, say called CMPlanReport. The CMPlanReportinforms the CCL of whether the decision is acceptable or not:

- A positive decision may indicate that the CCL can use/reuse that CMPlan.

- In case the decision is unacceptable, the response may include criteria on why the decision is bad/untrustworthy, e.g. how far the CCL predicted impact/effect on the network metrics is from the true value or what is the maximum change (in the current network parameters) that is allowed.

- The decision trust report may include in an indication for when the CCL has consistently made good decisions and achieved ultimate trust. The report may include an indication for how the CCL may behave thereafter - e.g. that the CCLs decisions will go without checking via the coordinator or that the CCL may directly execute its decisions on to the network.sthe CMPlanReport may include in an indication to pause or unpause the CCL, where the "pause" indicates that the CCL may cease to propose new actions until it is unpaused.

#### 5.6.3.6 Action-execution-time conflict coordination

##### 5.6.3.6.1 Required capabilities and interactions for detecting and avoiding Action-execution-time conflicts

5.6.3.6.1.1 Information on action plans for alignment and selection

Each CCL deployed in the network has a set of scopes for which it takes responsibility and actions that it can execute within those scopes and from which the CCL derives the decision and action plans that should be executed. An action plan is the combination of a set of actions that can be taken and the scopes under which those actions can be applied. To minimize conflicts (e.g. where the scopes overlap), the CCLs align the action plans, for example, through a coordinator CCL that selects which action plan to execute and when. Thereby:

- The CCLs inform the coordinator CCL about their respective action plans. The action plans contain information of target resources, scheduled time for execution, and may include other additional information such as historical results of the proposed actions.

- The coordinator CCL assesses each plan and choose the most appropriate combination of action plan(s) based on the selection policy. The appropriateness of action plan(s) or their combinations can be evaluated by multiple means and by using, for instance, historical data and/or operational data.

- Notify the selected action plan(s) to the related CCLs and/or the coordination CCL.

5.6.3.6.1.2 Information on detected action-execution-time conflict

Coordinating actions among multiple CCLs requires that there is a supervisory action-critic functionality that oversees the actions of the different CCLs. The action-critic functionality which may be part of coordinator CCL that takes the responsibility for the end-to-end performance of the Network.

For a given CCL, the action-critic receives the recommended changes from the CCLs, evaluates them to see:

1) if they overlap with other proposed changes from other CCLs; and

2) what their likely effects may be.

To determine the likely impacts, the action-critic may rely on network states analytics capabilities which discretize the state of the network into specific discrete scenarios and provide insights on performance characteristics in those scenarios. Such insights may for example characterize whether the network is in a scenario of low traffic and normal performance or scenario of normal traffic and anomalous performance.

Where there are likely conflicts and expected undesired impacts, the orchestration functionality decides the changes that should be executed on the network to minimize concurrent changes on the same network resources. The selection of actions to be accepted may be based on the priorities of the CCLs or the priorities of their goals and targets.

The coordination CCL then provides feedback to the CCL instance (s) regarding their recommended actions. The feedback may include information on which actions can be executed or not as well as information on the expected effects of the CCLs actions. Feedback may also include redefining the allowed control parameter spaces and ranges of the individual CCLs (i.e. which parameters the CCL should not control any further or the range in which the CCL may set the value of a control parameter).

##### 5.6.3.6.2 Information objects to realize required capabilities and interactions

Introduce a datatype on the coordinationCCL IOC for a profile representing capabilities for critiquing the actions of the CCLs, say called CCLExecutionTimeCoordination. It may be name contained in subnetwork or a managed Function, e.g. in a CoordinationCCL:

1. Introduce a datatype and corresponding attribute on the CCLExecutionTimeCoordination profile to represent the proposed actions from a given CCL, say called CCLCMPlan. The CCLCMPlan should as minimum include:

- the identifier of the CCL instance;

- the set of network resources that are targeted to be reconfigured by the CCL, i.e. the set of managed objects on which the CCL wants to execute actions and the set of attributes to be reconfigured and their desired new values;

- the time and/or conditions under which the reconfiguration is planned to be executed;

- where applicable, the expected impact of those actions;

- an indication, say called CMPlanEvaluationInterval interval which indicates the time within which the CCL expects feedback, e.g. corresponding to the time by which the CCL may need to execute the panned actions.

NOTE 1: The CCLCMPlan may be a list of provisioning management operations or may use the constructs developed in the ongoing study on plan management.

NOTE 2: The CCLCMPlan may also be an attribute on the CCL. If so, the CCLCMPlan should be notifiable, so that when the CCL constructs the CCLCMPlan, it notifies the CCLs that have subscribed to it (e.g. the coordinator CCL) of the said CCLCMPlan which when updated is then notified to the CCLExecutionTimeCoordination.

Extend the CCL with information needed to support the evaluation of the proposed actions of the CCL:

1. Add to the CCL an attribute for the proposed actions as described above.
2. Introduce an attribute for the priority of the CCL or its goals. For example, the priorities may be assigned based on a fixed scale of say [1,10] where the lowest number indicates the lowest priority. The priority may indicate a general priority of the CCL that is not specific to the resolution of execution-time conflicts.
3. Introduce a datatype and corresponding attribute on the CCL to represent the feedback from the CCLActionCritic, say called CCLCMFeedback:

- The CCLCMFeedback includes, for the list of proposed actions, an indication of which those actions should not be executed.

- The CCLCMFeedback may include, for the list of proposed actions, an indication of the expected impact of each of those actions.

- The CCLCMFeedback includes for the set of managed objects that are controlled by the CCL, the set of control parameters that should never be controlled by the CCL and/or the ranges of values which the control parameter can never be set into.

## 5.7 Use case 7: CCL scope management

### 5.7.1 Description

Each CCL should have specific scopes for which it is responsible.

The network may be assumed to be a p-dimensional space *SP* from which subregions d*P* Є D maybe created. Accordingly, *SP* is the full scope space whose dimension may include time, geography, etc. as showed in Table 5.7.1‑1 while d*p* Є D can be CCL's scope. In that respect, scope assignment is the mapping of CCLs to regions dЄD that are part of the network's full scope S. There may be 2 types of scopes - the measurement scope where related measurements are collected and the impact or control scope which is the scope to which the CCL's actions may have impact. The scopes for the different CCLs can be managed by the MnS consumer.

Table 5.7.1-1: Example scope-space map from which the scope of CCL may be derived

|  |  |  |
| --- | --- | --- |
| Scope dimension | Granularity | Example values to be assigned |
| Time | Seconds, minutes, days | Every hour,  Every Saturday at 2:00 hours |
| Network domains |  | Radio  Core |
| Geography | Region/City | City x  Street y in City x |
| Network Elements | gNB | gNB X |
| Cells | Cell A on gNB X |
| Terminals, e.g. types of users | users |
| Resources | Slices |  |
| Network Function | Virtual Network Function A  Physical Network Function B |
| Transport containers (links, flows, etc.) | an identifiable link,  a specific flow |
| Target Purpose | The purpose of the CCL target | Coverage Targets, Performance Targets, Energy Efficiency Targets, Fault Management Targets, UE specific Targets |

NOTE: Table 5.7.1-1 is not complete and can be improved and/or extended as needed.

### 5.7.2 Potential Requirements

**REQ-CCL-COORD-1:** The 3GPP management system should support a capability enabling the MnS consumer to configure the scopes of a CCL, including the measurement scope, the control scope , and the impact scope.

NOTE: Measurement scope is where the related measurements are collected, the control scope is where the CCL acts and the impact scope is where the CLL actions causes effects.

**REQ-CCL-COORD-2:** The 3GPP management system should support a capability to detect, avoid and resolve conflicts among the scopes of multiple CCLs, including the measurement scope, the control scope and the impact scope.

### 5.7.3 Potential Solutions

#### 5.7.3.1 Required capabilities and interactions

To coordinate scope assignments, a CCL coordination functionality, say in Coordination CCL, needs a capability to coordinate the scope assignment across multiple CCLs, say called the scope assignment coordination capability. The scope assignment coordination capability considers a defined full scope space *Sp* and a set of scope rules to define the best scope to be assigned to each CCL. An example rule may be that the defined CCL scope should not overlap. The rules may for example be defined by an operator or can be implementation specific depending on the types of CCLs that are to be configured.

To support detection and avoidance of potential scope conflicts:

- Each CCL has a scope, e.g. the assuranceScope for the ACCL. The CCL may register its scopes with the coordinationCCL which triggers an evaluation of potential conflict, i.e. whether those scopes are likely to conflict with the scopes of another CCL.

- Applying the rules, the scope assignment coordination capability divides the scope space into regions such that each region is matched to a CCL in a way that maximizes fulfilment of the assignment rules defined in the scope assignment coordination capability. The For example, if the benefit is to avoid overlaps, the subregions are assigned to the different CCLs in a way that ensures no overlaps and that all the scope space has been assigned.

- A CCL may have four scopes - the measurement scope, target scope, control scope and impact scope with different rules applied for each scope. The selection of subregions of the scope space should consider the different rules for each type of scope.

- In case of a potential conflict, the new optimized subregions are selected and assigned to the individual CCLs. The scope assignment coordination capability should be enabled to configure the scopes of the CCLs.

Scope conflicts are only considered actual if their use results in negative outcomes. To support detection and resolution of actual scope conflicts:

- The CCLs should monitor changes in their scope and if the scope is changed, the CCL should be able to inform the scope assignment coordination capability of any observed changes in the scope.

- The CCL should notify differences between what was configured and the actual scopes e.g. if the considered scope for taking measurement data are affected by the actions of another CCL.

- The scope assignment coordination capability may subsequently trigger scope conflict evaluation based on the actual scope that is notified by the CCL.

#### 5.7.3.2 Information objects to realize required capabilities and interactions

Introduce on the CoordinationCCL a profile for the capabilities to coordinate different scopes across multiple CCLs, say as a datatype, say called CCLScopeCoordination:

- It collects all the functionality and capabilities related to coordinating scopes among multiple CCLs to detect, avoid or resolve potential and real conflicts.

- Introduce a datatype and corresponding attribute on the CCLScopeCoordination to represent the scope space which is to be considered by the CCLScopeCoordination to select allocations to different CCL instances.

Introduce a dataType and corresponding attribute on the CCL to represent the scope of the CCL. The datatype may be called CCLScope. A CCL instance can have 4 scopes, which may be called measurementScope, targetScope, controlScope and impactScope. The cCLScope should be configurable by the MnS consumer:

- The CCLScope may include an indication as to whether the assigned scope is exclusive or not. An exclusive scope implies that not other function or CCL should have impacts in that scope apart from the CCL to which it has been assigned.

- The scope should be an extension of the assuranceScope of the ACCL.

Introduce a dataType and corresponding attribute on the CCL to represent misalignments in any of the scopes of the CCL. The datatype may be called ScopeMissalignmentInfo. The ScopeMissalignmentInfo allows the CCL to indicate mis alignments in its scope and adjust the scope to remove conflicts:

- The ScopeMissalignmentInfo includes information on the type of scope in which there are misalignments and an indication of which one is smaller - the assigned or the required scope.

- The ScopeMissalignmentInfo may include information on scope conflicts, e.g. indication that the CCL has observed that its allocated scope being changed by another CCL or function.

### 5.7.4 Evaluation of solutions

TBD

## 5.8 Use case 8: CCL-impact assessment and resolution

### 5.8.1 Description

#### 5.8.1.1 Overview

Besides having direct conflicts for parameter values, CCLs may also have direct and indirect effects for their goals and metrics, i.e. where actions on one CCL affect the goals and metrics of other CCLs. Impact assessment includes capabilities for evaluating the direct and indirect effects of CCL actions and determining measures for remediation. The scope affected by the actions of the CCL is the impact-scope and is different from the measurement scope, i.e. the scope where the CCLs measure and control scope, i.e. the scope where they act.

#### 5.8.1.2 impact on known/bounded impact-scope

For some Closed Control Loops, the expected impact of the action may be known to the Closed Control Loop or coordination functionality governing the CCL. The scope affected by these actions is derived from the (candidate) actions executed by the CCL (or their descriptions). A CCL coordination functionality may wish to evaluate the known impact scope and needs to rely on information from MnS producers of other Closed Control Loops to:

1) determine if there are unwanted outcomes;

2) diagnose if the executed action(s) is/are responsible for those outcomes, especially for the case where multiple Closed Control Loops have concurrently taken actions; and

3) determine what needs to be done to undo the degradation and to avoid it in future.

#### 5.8.1.3 impact on unknown impact-scope

For some CCLs, the impact-scope affected by the actions of a CCL A may not be known a priori. For example, for a CCL A that adjusts transmit power of a cell (e.g. to minimize interference), the exact neighbour cells and related CCLs acting on those cells that would be affected by any transmit power decrease or increase cannot be explicitly enumerated. Any negative effects cannot be easily anticipated, and most may not be easily resolvable by simple if‑then-else rules. Instead, the MnS producer of a CCL A should interact with MnS producers of the other CLLs or with a coordination functionality to identify actions that lead to negative outcomes and flag them accordingly. Thereby, after the CCL A takes an action on the network:

1) MnS producer of CCL A or the coordination functionality notifies all other CCLs or MnS producers of all other CCLs that an action has been executed that may affect those CCLs. The action is expected to have impact is a specified time, called the impact-time, which depends on the use case. Fr example the impact of load balancing action can be evaluated in a few seconds while the impact of a handover decision can take several minutes or hours. The notification should include the length f this impact time which indicates the time at which an observed impacts should be reported.

2) After a preset monitoring period equivalent to the notified impact time, the MnS producers of the impacted CCLs report (directly or through the coordination functionality) the impact that CCL A had, (i.e. the impact that CCL A's action had had to their performance metrics or goals. The impact may be reported an index say in the range [0,10] where 0 implies an unacceptable action and 10 implies a good action.

3) MnS producer of CCL A or the coordination function derives an appropriate remediation, e.g. by reconfiguring the candidate actions of the acting CCL (i.e. CCL A) or by undoing the action.

NOTE: That this clause needs further clarification.

### 5.8.2 Potential Requirements

**REQ-CCL-IMPACT-1:** The CCL MnS producer should support a capability enabling an MnS consumer to receive information on the impacts of the CCL on a particular impact-scope and the actions that caused such impacts.

NOTE 1: The MnS consumer may for example be another CCL or a CCL impact coordination function.

NOTE 2: The information enables the MnS consumer to determine if there are unwanted outcomes resulting from actions of the CCL and to propose what needs to be done to undo the degradation.

**REQ-CCL- IMPACT-2:** The CCL MnS producer should support a capability enabling an MnS consumer to notify the MnS producer of the actions of another CCL that may affect the MnS producer's CCL.

NOTE 3: The MnS consumer could for example be a CCL impact coordination function.

NOTE 4: The MnS producer represents the CCL which may be potentially impacted when a CCL A executes an action that may affect the goals or metrics of the MnS producer's CCL.

**REQ-CCL- IMPACT-3:** The CCL MnS producer should support a capability to report to an MnS consumer what the impact that the action had to the goals of the MnS producer's CCL.

NOTE 5: The MnS producer represents impacted CCL or MnF or a coordination function representing the impacted CCL or MnF.

NOTE 6: MnS consumer may for example be a coordination function or an acting CCL that took an action that has impacted the MnS producer's metrics.

NOTE 7: The use of metrics for this requirement is For Further Study.

**REQ-CCL- IMPACT-4:** The CCL MnS producer should support a capability enabling an MnS consumer to propose to MnS producer the appropriate remediation against the noted impact, e.g. the reconfiguration of the candidate actions of the acting CCL.

NOTE 8: MnS consumer may be the CCL impact coordination function or another CCL or management function.

NOTE 9: The MnS producer may be the acting CCL or the impacted CCL.

### 5.8.3 Potential Solutions

#### 5.8.3.1 Solution for detection of actual indirect targets conflicts via impact on unknown or unbounded impact-scope

NOTE: This solution focusses on the requirement on:

- detection of actual indirect targets conflicts.

##### 5.8.3.1.1 Required capabilities and interactions

For impacts on an unknown impact-scope, the CCL that took action or its coordination CCL cannot determine the impact and has to collect that form the affected entities. So:

- The CCL that took action or its coordination CCL should be enabled to notify the affected MnS consumers (e.g. other CCLs) of:

1) the fact that an action has been taken that may affect them;

2) that they need to provide feedback on how impact there has been; and

3) when they need to provide that feedback.

- To enable the affecting CCL to determine how much impact it had on the other CCL, the affected CCLs can provide their evaluation of the impacts as an index that indicates the degree to which the action was good or bad to their objectives.

##### 5.8.3.1.2 Information objects to realize required capabilities and interactions

- Introduce on the CCL an attribute representing, for each action taken, the time at which any affected MnS consumers (e.g. other CCLs) should provide their evaluation of the impact of the action. The time may be called cCLActionImpactTime. The cCLActionImpactTime should be notifiable, e.g. the coordination CCL or other CCLs which may be affected can subscribe to notifications on the cCLActionImpactTime.

- Introduce on the CCL an attribute representing for each action taken, an index that quantifies the evaluation of the impacts from each affected CCL. The index may be called an Action Quality Indicator (AQI), say in the range [0,10] where "0" indicates that the action was completely unacceptable and should never be reused in that context while "10" indicates that the action had very good outcomes for the reporting CCL. The AQI may be used by a coordination CCL to compute the aggregate impact form multiple affected CCLs and configure the acting CCL with a single AQI value. Based on the aggregate AQI the coordination CCL may also propose a response action, e.g. to reverse the action that was taken.

NOTE: The AQI is specific to each CCL and to each scenario thar the CCL evaluates - since it is used to check how good or bad an action was for that CCL. Accordingly, its computation cannot be standardized, only its range can be standardized.

### 5.8.4 Evaluation of solutions

TBD

## 5.9 Consumers feedback on CCL actions

### 5.9.1 Description

In fully automated control loops, the CCL re-configures a particular NF to meet its stated goals without the involvement of any other entity. The actions executed by the CCL have different levels of satisfaction for the different consumers. Without a reliable means to gauge consumer Execution Satisfaction, the CCL lacks the feedback to fine‑tune and optimize functionality, and so is unable to improve the overall performance. To be able to provide gauge the satisfaction, the consumer should be able to receive information about the provisioning operations executed by the CCL. This information includes operation performed, MOIs updated, etc.

Based on some local policies, the consumer may prefer that a particular NF is not updated as part of the Execution step of CCL. The consumer should be enabled to revoke the changes made to a NF. Consumer may also update the CCL to ensure that a particular NF is never updated in future. The existing attribute aCCLDisallowedList can be used, as appropriate.

Alternatively, the consumer may want to provide feedback enabling the CCL to apply an alternative approach to achieve the objectives. The consumer should be able to provide its feedback on the execution indicating how satisfied the consumer is with the CCL actions.

EXAMPLE: The consumer feedback may grade the usefulness of the executed action on a fixed scale say from 0 (indicating a terrible and never to be re-used action) to 10 (indicating a very good action for the interests of the consumer).

NOTE: The actions that need to be provided to the consumer are decided by the producer.

### 5.9.2 Potential Requirements

**REQ-FED-FUN-01:** The 3GPP management system should enable consumer to provide its feedback on the action(s) taken by CCL.

**REQ-FED-FUN-02:** The 3GPP management system should enable consumer to request for revocation of the action(s) taken by the CCL.

**REQ-FED-FUN-03:** The 3GPP management system should have a capability enabling consumer to receive information (e.g. operation performed, MOIs updated) about the action(s) taken by the CCL.

## 5.10 CCL decision escalation

### 5.10.1 Description

CCLs will make decisions in different contexts (states, status, conditions, etc.) of the network and not all decisions are equally effective. In some cases, the CCL may need to provide feedback indication the challenges faced on reaching a definite decision. The feedback may include a request to escalate its decision making to another entity.

A CCL can be fully or partially autonomous derivation or execution of its decisions. The degree to which the CCL independently executes decisions or escalates them, should be flexibly configurable by the MnS consumer as a confidence threshold. The confidence threshold could be configured based on the sensitivity of the operations under its control, the trust level in the decisions of the CCL and the necessity to consider a bigger picture at times. Then, based on how much confidence the CCL has in its decisions, the CCL can escalate a decision or situation to an escalation recipient which has this bigger picture (say has wider scope), can execute a different(larger) set of actions or has better capabilities, e.g. a larger and more capable ML model.

Note: the computation of confidence within the CCL is up to implementation as it depends on the CCL's purpose and the scenario that the CCL is addressing.

The escalation recipient CCL enables the escalator CCL to request for escalation for a given network context or state with e.g. information about the escalator CCL preferences and observed constraints when driving decisions. Based on its evaluations, the escalation recipient CCL should provide to the escalator CCL a report that holds the outcomes that the CCL (acting as an escalation recipient) has derived for a given escalation request.

NOTE: The relation with existing Assurance closed loop execution supervision use case in 3GPP TS 28.535 [2] needs to be clarified.

### 5.10.2 Potential requirements

**REQ- CCL-ESC-1:** The CCL MnS producer should have a capability to enabling an authorized consumer to configure the degree of autonomy of the CCL as characterization of the conditions under which the ACCL should escalate a decision.

**REQ- CCL-ESC-2:** The CCL MnS producer should have a capability to enabling an authorized consumer to configure the entity to which a decision should be escalated.

**REQ-CCL-ESC-3:** The CCL MnS producer (acting as an escalation recipient CCL) should have a capability to enabling an authorized MnS consumer (e.g. an escalator CCL) to request escalation of a decision or escalation of decision-making for a given network context or state to the CCL associated with the CCL MnS producer.

**REQ-CCL-ESC-4**: The 3GPP management system (or the CCL Mns producer) should have a capability enabling an MnS consumer to provide information related to its previous decisions, decision constraints, preferences, … as input to be used in resolving escalations sent towards the CCL associated with the CCL MnS producer.

**REQ-CCL-ESC-5:** The CCL MnS producer (acting as an escalation recipient CCL) should have a capability to provide to an authorized MnSconsumer (e.g. an escalator CCL) a report that holds the outcomes that the CCL (acting as an escalation recipient) has derived for a given escalation request.

### 5.10.3 Possible solutions



Figure 5.10.3-1: End-to-end flow of CCL decision escalation

1. Introduce an attribute defining the entity to which the decision is escalated to, say called the escalationRecipient.
2. Introduce an attribute on the CCL for defining the condition that triggers the escalation. For example, the CCL may trigger escalation when its level of confidence in the derived decision is below some threshold, say called a confidence threshold. The confidence threshold attribute enables the CCL to autonomously make decisions for each situation and context based on its computed confidence level in the given situation. If the confidence level is lower than the confidence threshold the decision is escalated otherwise the decision is executed.
3. Introduce on the CCL representing the escalationRecipient. an IOC representing the request for escalation, say called EscalationRequest, that holds all information related to the request for escalation. The EscalationRequest may include:

- A proposedCMChange attribute which describes the configuration management changes that has been proposed by the escalator CCL.

- A decisionConstraints attribute indicating the constraints observed by the escalating CCL in making the decision(s). The constraints may be of type context as defined in 3GPP TS 28.312 [5] or conditions expressed using JEX/XPATH

- The EscalationReason attribute can provide an optional description of the reason behind the escalation to provide more context and further clarification.

1. The escalationRecipient may send an output representing the outcome of the escalation. Introduce in the AssuranceReport an escalationOutcome data type that holds the outcomes of the escalation. When the escalationOutcome attribute of the report is updated, the escalation recipient CCL sends a notification to the escalator CCL that the report is available:

- The escalation outcome indicates whether the CCL should take any action and what that action is. Introduce an attribute to indicate what should be done by the CCL. It is an ENUM with the values:

1. "DONOTHING"- indicating that the CCL does not ned to take any action, i.e. the escalationRecipient is addressing the scenario.
2. "APPLYACTION"- indicating that the CCL should apply a specific set of actions proposed by the escalationRecipient. The action may be placed a new attribute, say called "proposedActions that list as et of CM changes.
3. "APPLYGUIDANCE"- indicating that the CCL should compute a new CM change based on the guidance from the escalationRecipient. The guidance may be placed in the proposedActions attribute.

### 5.10.4 Evaluation of solutions

TBD

## 5.11 Use case 11: Performance Evaluation of a Closed Control Loop

### 5.11.1 Description

The advanced monitoring functionalities of a CCL can provide real-time insights into the performance and outcomes of a CCL. The monitoring activity for a Closed Control Loop may result in further actions that happen in the operation phase, e.g. evaluate and update, in order to change the closed control loop settings and improve its performance. So there is a need to evaluate performance of a Closed Control Loop itself. Such metrics are important to understand and change a CCL's behaviour and to improve its performance to pursue the assigned goal(s).

For example, certain performance aspects of a CCL can be very crucial to know in order to evaluate and decide upon a CCL's performance, such as number of breached goals, time taken to meet a breached goal, number of conflicts occurred by a CCL etc. With the knowledge of such performance aspects of an existing CCL a MnS consumer can more effectively update or create a new CCL.

An operator can also compare different CCLs based on these performances and choose the best one for its network deployment.

### 5.11.2 Potential Requirements

**REQ-CCL-PERF-1:** The 3GPP management system should be able to obtain a CCL's performance with respect to the total number of occurrences of an assurance goal breach.

**REQ-CCL-PERF-2:** The 3GPP management system should be able to obtain a CCL's performance with respect to the time taken by CCL to meet a breached goal.

**REQ-CCL-PERF-3:** The 3GPP management system should be able to obtain a CCL's performance with respect to the total number of conflicts occurred by a CCL.

### 5.11.3 Potential Solutions

This solution involves defining new performance metrics to evaluate performance of a CCL for its optimal execution. This enables operators to track the effectiveness of closed loop automation, identify areas for improvement, and make informed adjustments to CCL functionalities.

These new metrics are as follows:

1) Total number of occurrences of an assurance goal breach:

a) This measurement provides the total number of occurrences when an assurance goal, as defined in CCL is breached during an observation time period.

b) CC.

c) This is measured by counting each incidence when an assurance goal is breached and incrementing the corresponding counter by one for each such occurrence within an observation time period.

d) An integer value.

e) The measurement name has the form TotalAssuranceGoalBreach.

f) CCL Provider.

g) Valid for packet switched traffic.

h) 5GS.

2) Time taken by CCL to meet a breached goal:

a) This measurement provides the time taken by a CCL to meet a breached goal after activating it again.

b) DER.

c) This is measured by considering the time stamp when an assurance goal is breached and subtracting it from the time stamp when that goal is met after activating the CCL with required changes.

d) Each measurement is an integer representing the mean delay in milliseconds.

e) The measurement name has the form TimeBreachedGoalRecovery.

f) CCL Provider.

g) Valid for packet switched traffic.

h) 5GS.

3) Total number of conflicts occurred by a CCL:

a) This measurement provides the total number of conflicts that occur between a CCL under consideration and any other CCL during an observation time period.

b) CC.

c) This is measured by counting each incidence when conflict occurs between a CCL under consideration and the other CCL and incrementing the corresponding counter by one for each such occurrence within an observation time period.

d) An integer value.

e) The measurement name has the form TotalCclConflicts\_Filter, where filter is either Implicit or Explicit. Implicit represents the action conflict i.e. conflict between two existing CCL and target represents the explicit conflict i.e. conflict between an existing CCL and a requested CCL.

f) CCL Provider.

g) Valid for packet switched traffic.

h) 5GS.

These measurements can be used for performance evaluation of a closed control loop. For example when the SLS performance starts degrading for certain metric, the PA/CCL MnS consumer obtains information of attributes of all ACCLs from the PA/CCL MnS producer. Then PA/CCL MnS consumer identifies the ACCL 'n' which contains that SLS metric and obtains the status of its performance metrics (i.e. TotalAssuranceGoalBreach, TimeCorrectiveGoalMeet, TotalCclConflicts\_Filter) via PerfMetricJob MOI from producer.

Then PA/CCL MnS consumer can either update an existing ACCL 'n' in the producer (by sending modifyMOIAttributes or changeMOIs request message to PA/CCL MnS producer) or it can create a new ACCL for the desired SLS/assurance goal (by sending createMOI Request message to PA/CCL MnS producer). For both the scenarios a suitable response message is sent by PA/CCL MnS producer to consumer for the updated attributes in ACCL 'n' or for newly created ACCL MOI.

### 5.11.4 Evaluation of solutions

TBD

# 6 Conclusions and Recommendations

## 6.1 Closed control loop and intent

### 6.1.1 Conclusion

Intent and closed control loops are different, but complementary concepts as discussed in clause 4.x in Table 4.x.1. An intent manager may utilize one or several closed control loops for implementing the assurance of intent requirements. This means intent managers can become consumers of closed control loops.

A closed control loop that provides (MnS producer) an intent API is configured using intent. This closed control loop therefore meets the definition of being an intent manager in the role of intent handler. If the closed control loop acts by sending intent, it implements the intent owner role. This means, that closed control loop can either be utilized as integral part of intent management or directly implement intent management. However, the functional scope of intent management exceeds the proposed scope of closed control loop. Intent management considers, for example, the negotiation of requirements and intent handlers are already driving deployment decisions and processes. Furthermore, the reporting on achievements and results to the source of intent is an integral part of intent management and realized through intent reports.

### 6.1.2 Recommendation

The normative work should follow the principles in the conclusion.

Annex A:  
PlantUML Code for figures

# A.1 Relationship UML diagram for CCL (Figure 5.6.3.1-2)

@startuml

skinparam ClassStereotypeFontStyle normal

skinparam ClassBackgroundColor White

skinparam shadowing false

skinparam monochrome true

hide members

hide circle

class ManagedEntity <<ProxyClass>>

class CoordinationCCL <<InformationObjectClass>>

class CoordinationProfile <<dataType>>

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

CoordinationCCL "1" -- "\*" CoordinationProfile

note left of ManagedEntity

Represents the following IOCs:

Subnetwork or

ManagedFunction

end note

note right of CoordinationProfile

Represents the following capabilities:

GoalTargetCoordination

ScopeAssignmentsCoordination

DirectActionsCoordination

IndirectTargetsCoordination

executionTimeCoordination

end note

@enduml

Annex B:  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2024-02 | - | n/a | - | - | - | Initial skeleton | 0.0.0 |
| 2024-02 | SA5#153 | S5-241044  S5-241045 | pCR |  |  | Initial skeleton of TR 28.867  Rel-19 pCR 28.867 TR Structure | 0.1.0 |
| 2024-04 | SA5#154 | S5-241989  S5-241990  S5-241992  S5-241988  S5-241982 | pCR | - | - | Rel-19 pCR TR28.867 Update baseline TR 28.867-010  Rel-19 pCR TR28.867 Background on closed control loops  Rel-19 pCR TR28.867 Dynamic CCL creation  Rel-19 pCR 28.867 Triggered CCL Management  pCR TR 28.867 Add use case, requirement and solution for problem recovery to enable closed loop automation | 0.2.0 |
| 2024-06 | SA5#155 | S5-242444  S5-243119  S5-243120  S5-243121  S5-243137  S5-243138  S5-243139  S5-243140  S5-243142  S5-243143  S5-243144  S5-243145  S5-243147 | pCR |  |  | Rel-19 pCR TR28.867 Some editorial cleanup  Rel-19 pCR TR28.867 CCL MnS  Rel-19 pCR TR28.867 CCL Conflicts management  Rel-19 pCR TR28.867 CCL Conflicts  Rel-19 pCR TR28.867 CCL conflicts resolution  Rel-19 pCR TR28.867 CCL-impact assessment  Rel-19 pCR TR28.867 CCL scope management  Rel-19 pCR TR28.867 Add use case and requirements on CCL for single-domain fault management  Rel-19 pCR 28.867 Feedback Management  Rel-19 pCR 28.867 Historical CCL  Rel-19 pCR 28.867 solution for triggered CCL  Rel-19 pCR TR28.867 Dynamic CCL creation and execution  Rel-19 pCR TR28.867 CCL escalation | 0.3.0 |
| 2024-08 | SA5#156 | S5-244285  S5-244663  S5-244664  S5-244665  S5-244667  S5-244668  S5-244669  S5-244670  S5-244671  S5-244672  S5-244673  S5-244674  S5-244675  S5-244676  S5-244678  S5-244679  S5-244680  S5-244681  S5-244682  S5-244684  S5-244730 | pCR | - | - | pCR TR 28.867 Add clause on closed control loop as enabler for intent  Rel-19 CR 28.867 Add solution for conditions based on performance metrics.doc"  Rel-19 CR 28.867 Add solution for conditions based on Trace metrics.doc"  Rel-19 CR 28.867 Add solution for conditions based on data node tree changes.doc"  Rel-19 CR 28.867 Add conclusion for Triggered CCL.doc"  Rel-19 pCR TR28.867 CCL dynamic composition solution.docx"  Rel-19 pCR TR28.867 CCL conditinal execution solution.docx"  Rel-19 pCR 28.867 Conflict Management solutions.docx"  Rel-19 pCR TR28.867 Solution for CCL goal-target-conflicts.docx"  Rel-19 pCR TR28.867 Solution for direct actions conflicts.docx"  Rel-19 pCR TR28.867 Solution for CCL indirect-Actions conflicts.docx"  Rel-19 pCR TR28.867 Solution for CCL execution-Time conflicts.docx"  Rel-19 pCR TR28.867 Solution for CCL conflicts resolution.docx"  Rel-19 pCR TR28.867 Evaluating impact of planned actions.docx"  Rel-19 pCR TR28.867 CCL-impact-assessment solution.docx"  Rel-19 pCR TR28.867 CCL scope management solution.docx"  Rel-19 pCR TR28.867 Enhance CCL Escalation solution.docx"  pCR TR28.867 Update use case and solution for closed control loop for problem recovery.doc"  Rel-19 pCR 28.867 Historical CCL data solutions.docx"  Rel-19 pCR TR 28.867 Performance monitoring of Closed control loop.docx"  Rel-19 pCR TR28.867 Align CCL conflict management.docx" | 0.4.0 |
| 2024-09 |  |  |  |  |  | After editHelp cleanup | 0.4.1 |

|  |
| --- |
| **End of modifications** |