**3GPP TSG-SA5 Meeting #157S5-245405**

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

**Source: Nokia, Deutsche Telecom**

**Title:** **pCR 28.867 Add information on severity conflicts**

**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

The CCL conflict solutions have been described with several textual descriptions. This pCR adds procedure flows to the CCL solutions to illustrated procedures described in text

# 4 Detailed proposal

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

## 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.

Some conflicts could infact have no impact on the network, in which case they can be classified as couplings. Those that qualify as conflicts are those which are observed to have significant impact on the network, and so can be associated with a severity that characterizes the degree to which they impact on the network and are critical for resolution. The MnS consumer should be enabled to define the severity levels (critical, major, minor, etc.) including, where applicable, as well as the rules and criteria for assigning the severity to conflicts. The management system should be able to provide information on the severity of the conflicts that are detected.

#### 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

Multiple conflicts are possible among CCL or their instances. The CCL MnS producer should be able to interactively coordinate with MnS consumers to resolve the conflicts.

##### 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-CONFLICT-7:** The MnS Producer for CCL management should enable to define the severity levels including, where applicable, as well as the rules and criteria for assigning the severity to conflicts.

Note: examples of severity levels could be critical, major, minor, etc.

**REQ-CCL-CONFLICT-8:** The MnS Producer for CCL management should enable authorized MnS consumers to receive information on the severity of the conflicts.**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

NOTE: This solution focusses on the requirement on:

- detection and avoidance of potential goal targets conflicts;

- detection and resolution of actual goal targets conflicts.

##### 5.6.3.3.1 Required capabilities and interactions

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.

- To resolve the conflict, the the goal management functionality may configure the CCL to operate in a sequence or in a hierarchy, with each CCL having a profile indicating the specific level of hierarchy. 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 CCL Coordination Function needs to configure the appropriate hierarchy for the CCLs.

##### 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. Introduce a dataType and corrsponding attribute on the CCL representing the hierarchy information of the CL. For each CCL, the hierarchy includes information on the CCLs in the lower or higher levels of hierarchy..
3. 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

NOTE: This solution focusses on the requirement on:

- detection and avoidance of potential direct actions conflicts;

- detection and avoidance of actual 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 shall 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

The CCL A or a coordination entity can also evaluate the severity of the conflict based on degree to which tolerances are violated. CCLA or a coordination entity can then notify CCL B about the severity of the CCL.

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.

- Introduce an optional attribute on the CCL coordination representing the severity of a detected conflict. It may be called CCLconflcitSeverity and defied as an enumeration with the values CRITICAL, MAJOR, MIMOR. The severity may not be possible to compute in all scenarios. Details on how it is computed are up to implementation.

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

NOTE: This solution focusses on the requirement on:

- avoidance 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

NOTE: This solution focusses on the requirement on:

- detection of potential indirect targets conflicts;

- avoidance of indirect targets conflicts;

- detection of actual 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 indirect target conflicts - Evaluating likely-impact of planned actions

NOTE: This solution focusses on the requirement on:

- avoidance of potential indirect targets conflicts.

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 must prove 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

NOTE: This solution focusses on the requirement on:

- detection of potential action-execution-time conflicts;

- avoidance of action-execution-time conflicts.

##### 5.6.3.6.1 Required capabilities and interactions

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.

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