**3GPP TSG-SA5 Meeting #156S5-243536**

**19 - 23 August 2024, Maastricht, Netherlands**

**Source: Nokia**

**Title: Rel-19 pCR TR28.867 CCL conditional execution solution**

**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: “Closed control loop management” v0.3.0

# 3 Rationale

Use cases 5.1supports dynamic creation of a CCL but does not include conditional activation of CCL decisions. This pCR introduces the use case and solution for conditional activation of CCL decisions.

# 4 Detailed proposal

|  |
| --- |
| **First Change** |

# 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

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 5.1.3.1-1: 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. For 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 5.1.3.2-1: 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-1: 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-2: 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-3: 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

5.1.2.1 Solution-1

To enable dynamic composition of the CCL

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

Note: the best name for this IOC and how to extend is FFS

* 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.
* introduce on the CCL IOC, an attribute representing the sequence of steps of the CCL. The MnS consumer can provide the list of MnFs or MnS producers that should be combined into a CCL.

Introduce in CCL an attribute providing information related to the identifier of the required management function and the required configuration 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

TBD

5.2 Use case 2: Triggered CCL

5.2.1 Description

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.

5.2.1.1 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[x], post appropriate extensions, can be utilized to define triggering conditions for CCL management.

5.2.1.2 Conditional decision activation of CCLs

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.

5.2.2 Potential Requirements

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

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

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

5.2.3 Potential solutions

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. The clause 5.2.3.2 specify the usage of condition monitor to implement the LoopTrigger object.

5.2.3.1 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 shall be 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 shall be 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.

5.2.3.2 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.

* conditionObject: This is to represent the target node i.e., object for which the performance and fault is to be monitored.
* 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:

* 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).
* 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:

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

5.2.3.X Potential solution for Conditional decision activation of CCLs

To enable dynamic Conditional decision activation of the CCL

* 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 TS28.622, condition monitor as defined in TS28.622, the condition expressed in form of a JEX/XPATH expression
	+ Note: the conditions could alternatively be part of the goal
* 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
* DO\_NOTHING: do not do anything

5.2.4 Evaluation of solutions

The potential solution described in clause 5.2.3.2 is a fully NRM-based approach that extends the existing NRM fragments to realise dynamic conditional activation of CCL decisions. The solution extended the general proposal for triggering a CCL by adding activation of CCL decision or only recommendations as alternative behaviors to be triggered by conditions (besides triggering the instantiation of the CCL). The solution is implementable as it relies on already defined constructs like the condition monitor. Therefore, the solution described in clause 5.1.3.2 is a feasible solution for dynamic conditional activation of CCL decisions.

|  |
| --- |
| **Next Change** |

# 6 Conclusions and recommendations

The present technical report described potential enhancements to the closed control loop management specifications.

* It is recommended to move on to the normative specification development phase for the use case on dynamic conditional activation of CCL decisions, the normative specification development should follow the solution outlined in clause 5.2.3.2.