**3GPP TSG- Meeting # *9***

**, , -**

**Source: Samsung**

**Title: Solution for triggered CCL**

**Document for: Approval**

**Agenda Item: 6.19.4**

# 1 Decision/action requested

***In this box give a very clear / short /concise statement of what is wanted.***

# 2 References

None

# 3 Rationale

This provides the solution for triggered CCL.

# 4 Detailed proposal

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

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.

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.

5.2.2 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 occur on the given DN.
    - Provisioning Time: The CCL will be created only when the given event occur 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 alarm 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.).