**3GPP TSG-SA5 Meeting #157 *S5-246020***

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

**Source: Samsung, Nokia**

**Title: Solution Evaluations Dynamic and 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

1. The concept of Trigger is used in both the UC. Hence, shall follow the same solution i.e Jex and/or XPath.
2. The concept of plan (defiled in 3GPP TR 28.872) shall be applicable to both the use case of “Dynamic CCL Creation” and “Triggered CCL”.
3. The following statement is repetitive and incorrect (adding to TopX should also be considered addition to stage.
   * “*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*.”
   * “*Pseudo attributes are not added to stage 2 NRM definitions. Alternatively, the approach described for measurements may be used*”

# 4 Detailed proposal

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

### 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. The conditions under which a CCL can be dynamically composed can be defined using the mechanism provided in clause 5.2.3.3.

Furthermore, it is recommended to use activation of planned configurations (3GPP TR 28.872 [7], clause 5.3) as a solution for the use case Dynamic CCL Creation.

Following is the wayforward for the normative work:

1) LoopTriggerObject is of type Plan (TR 28.872), so the name containment of plans is applied.

2) The operation in the LoopTriggerObject plan would specify “creation/activation for a CCL”

4) The LoopTriggerObject includes special profile containing information on the possible elements in the conditions, i.e., the ones which can be evaluated for triggering the loop. Only the elements in the profile are the ones on which conditions can be expressed, i.e. conditions on other elements will be considered invalid.

5) The conditions attached to the plan are the conditions to be evaluated for the CCL creation. The condition is expressed as described in the solution.

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

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

##### 5.2.3.3.1 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.

##### 5.2.3.3.2 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 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 must 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 must 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.

##### 5.2.3.3.3 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 must 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 |

##### 5.2.3.3.4 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.

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

### 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. The stage 3 enhancements requires defining Jex and/or XPath based conditions.

Note: Usage of Jex conditions related to fault based triggers, specified in clause 5.2.3 is to be analysed in the normative work.

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.

Following is the wayforward for the normative work:

1) LoopTriggerObject is of type Plan (TR 28.872), so the name containment of plans is applied.

2) The operation in the LoopTriggerObject plan would specify “creation/activation for a CCL”

4) The LoopTriggerObject includes special profile containing information on the possible elements in the conditions, i.e., the ones which can be evaluated for triggering the loop. Only the elements in the profile are the ones on which conditions can be expressed, i.e. conditions on other elements will be considered invalid.

5) The conditions attached to the plan are the conditions to be evaluated for the CCL creation. The condition is expressed as described in the solution.

|  |
| --- |
| **Last Change** |