**3GPP TSG-SA5 Meeting #143-e *S5-223329***

e-meeting, 9 - 17 May 2022

**Source: Huawei**

**Title: Rapporteur clean up**

**Document for: Approval**

**Agenda Item: 6.6.3**

# 1 Decision/action requested

***The group is asked to discuss and approval.***

# 2 References

[1] 3GPP draft TS 28.312: “Management and orchestration; Intent driven management services for mobile networks v1.1.0”.

# 3 Rationale

This contribution proposes to do the final rapporteur clean up, mainly includes:

* Add the missing reference

- Update the value for Properties "isOrdered" and "isUnique" to align with TS 32.156.

# 4 Detailed proposal

It proposes to make the following changes to TS 28.312[1].

|  |
| --- |
| **1st Change** |

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 28.531: "Management and orchestration; Provisioning".

[3] 3GPP TS 28.532: " Management and orchestration; Generic management services".

[4] 3GPP TS 28.530: " Management and orchestration; Concept, use cases and requirements"

[5] 3GPP TS 28.541: " Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3".

[6] 3GPP TS 28.622: "Telecommunication management; Generic Network Resource Model (NRM); Integration Reference Point (IRP); Information Service (IS)".

[7] TM Forum IG1253A: "IG1253A\_Intent\_Common\_Model\_v1.1.0".

[X] 3GPP TS 38.104: "Technical Specification Group Radio Access Network; NR; Base Station (BS) radio transmission and reception".

[Y] 3GPP TS 28.538: "Management and orchestration; Edge Computing Management".

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

### 4.2.2 Intent driven MnS

Introduction of service-based architecture for 5G, in combination with functional model of business roles, exceeds the level of complexity for managing network in different scenarios (including scenarios for design/planning, deployment, maintenance and optimization) both in a single and multivendor network. New/simpler ways of managing are needed.

Actions of an intent driven MnS related to the fulfilment of intents may be categorized as intent deployment and intent assurance. Intent fulfilment refers to the steps taken to satisfy a newly received intent or an update to an existing intent. The goal of intent fulfilment is to bring the network or service's state to satisfy the new or updated intent. The fulfilment of some intents may end at the intent deployment, the case, if the intent's goal simply describes the availability or presence of a service. In other cases, the intent's goal describes the assurance requirements for a network or service (e.g. quality of service, end user experience, SLS, etc.) in addition to the need of existence of a service. Those intents have their fulfilment tied to the operation of the referred service or network function and may require frequent recurring actions to keep those assurance requirements achieved. This part of the intent fulfilment is referred to as intent assurance.

An Intent driven MnS allows its consumer to express intents for managing the network and services and obtain the feedback of intent evaluation result. The Intent-driven MnS producer have the following capabilities:

- Validate the intent.

- Translate the received intent to executable actions as follows:

- Performing service or network management tasks

- Identifying, formulating and activating service or network management policies

- Evaluate the result/information about the intent fulfilment (e.g. the intent is initially satisfied or not) and intent assurance (e.g. the intent is continuously satisfied).

The following figure 4.2-1 shows the model of Intent-driven MnS.

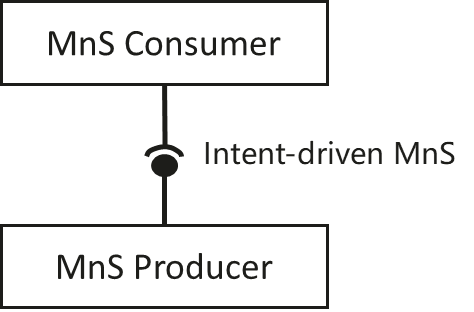


Figure 4.2-1: Intent-driven MnS

The intents may be fulfilled by utilizing multiple mechanisms including among others: Rule-based mechanisms, closed loop mechanisms and AI/ML based mechanisms. These mechanisms can be combined in solutions of various complexity, ranging from a simple approach rule-based mechanisms, to more elaborate solutions combining AI/ML, closed loop automation to ensure the fulfilment of intents.

When the intent is created on the MnS producer, the MnS producer may consume other management services (including non-intent driven MnS and intent driven MnS) to fulfil or satisfy the intent, e.g. creating new assurance closed control loop instance(s) or using assurance closed control loop instance (s) to satisfy the intent. The internal implementation of the intent fulfilment will however not be standardized.

An Intent driven MnS includes the following management capabilities to support intent lifecycle management:

- Create an intent, a MnS Consumer request to create a new intent on the MnS producer.

- Activate an intent, MnS Consumer request to activate an intent on the MnS producer when the intent is suspended.

- De-activate an intent, MnS consumer request to de-activate an intent on the MnS producer for a temporary suspension.

- Delete an intent, MnS Consumer request to remove an intent on the MnS producer.

- Modify an intent, MnS Consumer request to modify the content of the intent (e.g. expectation targets) on the MnS producer.

- Query an intent, MnS Consumer request to return the content and state (e.g. active, inactive) of the intent on the MnS producer.

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

6.2.1.2.1.2 Attributes

The Intent IOC includes attributes inherited fromTOP IOC (defined in TS 28.622 [6]) and the following attributes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| intentExpectations | M | T | T | F | F |
| userLabel | M | T | T | F | F |
| intentContexts | O | T | T | F | F |
| intentFulfilmentinfo | M | T | F | F | T |

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

###### 6.2.1.3.4.2 Attributes

The ExpectationTarget includes the following attributes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute Name** | **Support Qualifier** | **isReadable** | **isWritable** | **isInvariant** | **isNotifyable** |
| targetName | M | T | T | F | F |
| targetCondition | M | T | T | F | F |
| targetValueRange | M | T | T | F | F |
| targetContexts | O | T | T | F | F |
| targetfulfilmentInfo | O | T | F | F | T |

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

##### 6.2.1.3.6 FulfilmentInfo << dataType >>

###### 6.2.1.3.6.1 Definition

This dataType represents the properties of a specific fulfilment information for an aspect of the intent (i.e. either an expectation, a target or the whole intent). The fulfilment information describes the MnS producer's assessment of the degree to which a specific aspect of the intent has been fulfilled. The MnS consumer may however assess the fulfilment differently e.g., the MnS consumer may evaluate the delivered outcome or network state to compute its fulfilment satisfaction.

The fulfilmentStatus field indicates whether the intent is fulfilled or not fulfilled. The possible values of the fulfilment include:

* NOTFULFILLED: This is the default status for any aspect of the intent and the fulfilmentStatus remains as "NOTFULFILLED" until the MnS producer is satisfied that the undertaken actions meets the requirements as stated by the MnS consumer.
* FULFILLED: This is the status if the MnS producer considers that the intent, expectation or target has been fulfilled as desired by the MnS consumer that created the intent. The consumer may provide a fulfilment satisfaction report that either confirms the fulfilment or describes its evaluation the fulfilment.

The degree of fulfilment of an intent with the NOTFULFILLED status may have multiple explanations and related states. These different progress states and conditions are recorded in the notFulfilledState field. The possible values of the notFulfilledState include:

* ACKNOWLEDGED: this is the default status and is the initial notFulfilledState right after the intent has been received.
* COMPLIANT: this is the state after the feasibility check has been run for the intent and the intent accepted as being compliant for fulfilment
* DEGRADED: this is the state if an intent that was previously fulfilled is found not be meeting the initially stated requirements.
* SUSPENDED: this is the state if the MnS producer decides to suspect the fulfilment of the intent, expectation or target for whatever reason. This is notFulfilledState must be supported by a reason such as the event(s) that were observed when fulfilment was attempted.
* TERMINATED: This state is registered if the respective aspect of the intent (i.e. either an expectation, a target or the whole intent) shall not be considered for fulfilment e.g. when an authorized MnS consumer sends an indication terminating the specific aspect of the intent. For instance, if the MnS consumer sends an update of the intent in which a particular target is eliminated, then that target shall be marked as cancelled.
* FULFILMENTFAILED: This is the state when the MnS producer decides that the intent, expectation or target cannot be fulfilled. This is state must be supported by a reason such as the event(s) that were observed when fulfilment was attempted.

For some scenarios (in particular for the "SUSPENDED" and the "FULFILMENTFAILED" notFulfilledStates), the notFulfilledState should be supported by extra information describing or related to the state. This extra information is recorded into the notFulfilledReasons field.

###### 6.2.1.3.6.2 Attributes

The FulfilmentInfo includes the following attributes:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute Name** | **Support Qualifier** | **isReadable** | **isWritable** | **isInvariant** | **isNotifyable** |
| fulfilmentStatus | M | T | F | F | T |
| notFulfilledState | CM | T | F | F | T |
| notFulfilledReasons | CO | T | F | F | T |

###### 6.2.1.3.6.3 Attribute constraints

|  |  |
| --- | --- |
| Name | Definition |
| notFulfilledState  Support Qualifier | Condition: when FulfilmentInfo is implemented for IntentFulfilmentInfo |
| notFulfilledReasons  Support Qualifier | Condition: when FulfillmentInfo is implemented for IntentFulfilmentInfo |

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

#### 6.2.1.4 Attribute definition

| **Attribute Name** | **Documentation and Allowed Values** | **Properties** |
| --- | --- | --- |
| userLabel | A user-friendly (and user assignable) name of the intent.  allowedValues: Not Applicable | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| intentExpectations | It describes the expectations including requirements, goals and contexts (including constraints and filter information) given to a 3GPP system. It states the list of specific outcomes desired to be realized for expectation object(s).  allowedValues: Not Applicable | type: IntentExpectation  multiplicity: 1..\*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| intentFulfilmentInfo | It describes status of fulfilment of an intent and the related reasons for that status.  allowedValues: Not Applicable | type: FulfilmentInfo  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| expectationFulfilmentInfo | It describes status of fulfilment of an intentExpectation and the related reasons for that status.  allowedValues: Not Applicable | type: FulfilmentInfo  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| targetFulfilmentInfo | It describes status of fulfilment of an expectationTarget and the related reasons for that status.  allowedValues: Not Applicable | type: FulfilmentInfo  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| intentFulfillStatus | It describes the current status of the intent fulfilment result, which is configured by MnS producer and can be read by MnS consumer.  allowedValues: "FULFILLED", "NOT\_FULFILLED" | type: ENUM  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| notFulfilledState | It describes the current progress of or the reason for not achieving fulfilment for the intent, intentExpectation or expectationTarget. It is configured/written by MnS producer and can be read by MnS consumer.  allowedValues: " ACKNOWLEDGED", "COMPLIANT", "DEGRADED", "SUSPENDED", "TERMINATED" "FULFILMENTFAILED", | type: ENUM  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| notFulfilledReason | It describes the reasons/observations related to the specific noted notFulfilledState  allowedValues: Not Applicable | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| intentContexts | It describes the list of IntentContext(s) which represents the constraints and conditions that should apply for the entire intent even if there may be specific contexts defined for specific parts of the intent.  allowedValues: triple of (attribute, condition, value range) | type: Context  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| expectationId | A user-friendly (and user assignable) name of the intentExpectation.  allowedValues: Not Applicable | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| expectationVerb | It describes the characteristic of the intentExpectation and is the property that describes the types of intentExpectations. Examples of verbs and their related types of expectation are   * Deliver: DeliveryIntentExpectation, e.g. Deliver a RAN network, Service, Slice, function * Ensure: AssuranceintentExpectation, e.g. Ensure the performance targets values   allowedValues: Deliver, Ensure | type: String  multiplicity: 1  isOrdered:N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| expectationObject | It describes the expectation objects of the IntentExpectation that are required to be applied on.  allowedValues: Not Applicable | type: ExpectationObject  multiplicity: 1  isOrdered:N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| objectType | It describes the type of expectation object of the IntentExpectation that are required to be applied on. It can be class name of the managed object.  allowedValues: see scenario specific Intent Expectation | type: Enum  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| objectInstance | It describes a specific object instance (e.g. instance of managed object) to which the intentExpectation should apply.  allowedValues: Not Applicable | type: DN  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: None  isNullable: False |
| objectContexts | It describes the list of ObjectContext(s) which represents the constraints and conditions to be used as filter information to identify the object(s) to which a given intentExpectation should apply. Note there may be other constraints and conditions defined either for the entire intent, for the specific intentExpectation or for the expectationTarget of the considered intentExpectation.  The concrete ObjectContext depends on the ExpectationObject, which is defined in clause 6.2.2. All the concrete ObjectContexts follow the common structure of ObjectContext | type: Context  multiplicity: \*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| expectionTargets | It describes the list of ExpectationTarget(s) which represent specific outcomes on the metrics that characterize the performance of the object(s) or some abstract index that expresses the behavior of the object(s) that are desired to be realized for a given intentExpectation.  The concrete ExpectationTarget depends on the ExpectationObject, which is defined in clause 6.2.2. All the concrete ExpectationTargets follow the common structure of ExpectationTarget | type: ExpectationTarget  multiplicity: 1..\*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| expectationContexts | It describes the list of context(s) which represents the constraints and conditions that should apply for a specific intentExpectation.  Note there may be other constraints and conditions defined for the entire intent or for specific parts of the intentExpectation.  allowedValues: depends on Expectation Object in the IntentExpectation | type: Context  multiplicity: 1..\*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| targetName | It describes the name of the Expectation of the expectation target which represents specific outcomes on the metrics that characterize the performance of the object(s) or some abstract index that expresses the behavior of the object(s) that are desired to be realized for a given intentExpectation.allowedValues: depends on ExpectationObject in the IntentExpectation | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: Null  isNullable: True |
| targetCondition | It expresses the limits within which the targetName is allowed/supposed to be  allowedValues: is equal to; is less than; is greater than: "is within the range" ; "is outside the range" | type: Enum  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: "is equal to"  isNullable: False |
| targetValueRange | It describes the range of values that applicable to the targetName and the targetcondition.  allowedValues: depends on the targetName | type: Real  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: Null  isNullable: True |
| targetContexts | It describes the list of constraints and conditions that should apply for a specific expectationTarget. Note there may be other constraints and conditions defined for the entire intent or the intentExpectation.  allowedValues: Not Applicable | type: Context  multiplicity: 1..\*  isOrdered: False  isUnique: True  defaultValue: None  isNullable: False |
| contextAttribute | It describes a specific attribute of or related to the object or to characteristics thereof (e.g. its control parameter, gauge, counter, KPI, weighted metric, etc) to which the expectation should apply or an attribute related to the operating conditions of the object (such as weather conditions, load conditions, etc). | type: String  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: Null  isNullable: True |
| contextCondition | It expresses the limits within which the ContextAttribute is allowed/supposed to be  allowedValues: is equal to; is less than; is greater than; "is within the range";"is outside the range" . | type: Enum  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: "is equal to"  isNullable: False |
| contextValueRange | It describes the range of values that applicable to the ContextAttribute and the ContextCondition.  AllowedValue: depends on the contextAttribute | type: Real  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: Null  isNullable: True |

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

##### 6.2.2.1.2 Service Support Expectation

###### 6.2.2.1.2.1 Definition

Service Support Expectation is an IntentExpectation which can be used to represent MnS consumer's expectations for service deployment.

The Service Support Expectation is defined utilizing the constructs of the generic IntentExpectation <<dataType>> with set of allowed values and concrete dataTypes specified.

Following are the specific allowed values when implemented the IntentExpectation for Service Support Expectation

|  |  |
| --- | --- |
| Attribute | Allowed Values |
| ObjectType (CM) | ServiceSupport |
| objectInstance (CM) | DN of the ServiceSupport |

Note: following are the qualifier description for attribute "objectType" and "objectInstance":

- In case of the intent expectation is not for a specific service instance or/and MnS consumer have no knowledge of the DN of this service instance, the attribute "objectType" needs to be specified;

- In case of the intent expectation is for a specific service instance and MnS consumer have the knowledge of the DN of this service instance, the attribute "objectInstance" needs to be specified.

###### 6.2.2.1.2.2 ObjectContexts

Following provides the concrete ObjectContexts for Service Support Expectation based on the common structure of ObjectContext. The properties of the attributes in the following table should be same with properties of ObjectContexts defined in clause 6.2.1.3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute Name | Support Qualifier | isReadable | isWritable | isInvariant | isNotifyable |
| edgeIdenfiticationId | CM | T | T | F | F |
| edgeIdenfiticationLoc | CM | T | T | F | F |

Note: following are the qualifier description for attribute "edgeIdentificationId" and " edgeIdentificationLoc":

- In case of the Service deployment is needed at a particular edge data network, the attribute " edgeIdentificationId " needs to be specified;

- In case of the Service deployment is needed at a particular location, the attribute "edgeIdentificationLoc" needs to be specified;

###### 6.2.2.1.2.3 ExpectationTargets

Following provides the concrete ExpectationTargets for Service Support Expectation based on the common structure of ExpectationTarget. The attribute properties defined in the table below should be same with the properties defined for ExpectationTargets in section 6.2.1.3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute Name** | **Support Qualifier** | **isReadable** | **isWritable** | **isInvariant** | **isNotifyable** |
| dlThptPerUE | O | T | T | F | F |
| UlThptPerUE | O | T | T | F | F |
| dLLatency | O | T | T | F | F |
| uLLatency | O | T | T | F | F |

###### 6.2.2.1.2.4 ExpectationContext

Following provides the concrete ExpectationTargets for Service Deployment Expectation based on the common structure of ExpectationTarget. The attribute properties defined in the table below should be same with the properties defined for ExpectationTargets in section 6.2.1.3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Attribute Name** | **Support Qualifier** | **isReadable** | **isWritable** | **isInvariant** | **isNotifyable** |
| serviceStartTime | O | T | T | F | F |
| serviceEndTimeTarget | O | T | T | F | F |
| coverageAreaTAList | O | T | T | F | F |
| uEMobilityLevel | O | T | T | F | F |
| resourceSharingLevel | O | T | T | F | F |
| maxNumberofUEs | O | T | T | F | F |
| activityFactor | O | T | T | F | F |
| uESpeed | O | T | T | F | F |

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

#### 6.2.2.2 Attribute definition

| Attribute Name | Documentation and Allowed Values | Properties |
| --- | --- | --- |
| coverageAreaPolygonContext | It describes the coverage areas for the RAN SubNetwork that the intent expectation is applied in the form of polygon.  CoverageAreaPolygonContext is a Context including attributes: contextAtrribute, contextCondition and contextValueRange.  Following are the allowed values:  -contextAttribute: "CoverageAreaPolygon"  -contextCondition: "With the range"  -contextValueRange: a list of CoverageArea defined in TS 28.541[5] | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| coverageTACContext | It describes the coverage areas for the RAN SubNetwork that the intent expectation is applied in the form of TAC.  CoverageTACContext is a Context including attributes: contextAttribute, contextCondition and contextValueRange.  Following are the allowed values:  -contextAttribute: "CoverageAreaTAC"  -contextCondition: "With the range"  -contextValueRange: a list of nRTAC defined in TS 28.541[5] | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| plMNContext | It describes the PLMN(s) supported by the RAN SubNetwork that the intent expectation is applied.  PLMNContext is a Context including attributes: contextAtrribute, contextCondition and contextValueRange.  Following are the allowed values:  -contextAttribute: "PLMN"  -contextCondition:"With the range"  -contextValueRange: a list of PLMNId defined in TS 28.541[5] | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| nRFqBandContext | It describes the nRFqBands supported by the RAN SubNetwork that the intent expectation is applied.  nRFqBandContext is a Context including attributes: contextAtrribute, contextCondition and contextValueRange.  Following are the allowed values:  -contextAttribute: "NRFqBand"  -contextCondition: "With the range"  -contextValueRange: a list of NRFqBand expressed as string. Valid frequency band values are specified in sub-clause 5.4.2 in 3GPP TS 38.104[X]. | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| rATContext | It describes the RAT supported by the RAN SubNetwork that the intent expectation is applied.  RATContext is a Context including attributes: contextAtrribute, contextCondition and contextValueRange.  Following are the allowed values:  -contextAttribute: "RAT"  -contextCondition: "With the range"  -contextValueRange: a list of ENUM with allowed value: UTRAN, EUTRAN and NR. | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| weakRSRPRatioTarget | It describes the downlink weak coverage ratio target for the RAN SubNetwork that the intent expectation is applied.  WeakRSRPRatioTarget is an ExpectationTarget including attributes: targetName, targetCondition and targetValueRange.  Following are the allowed values:  -targetName: "WeakRSRPRatio"  -targetCondition: "is less than"  -targetValueRange: integer with allowed value [0,100].  -targetContext: WeakRSRPContext | type: ExpectationTarget  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| WeakRSRPRatioTarget.weakRSRPContext | It describes the threshold for downlink weak RSRP of the RAN SubNetwork that the intent expectation is applied.  WeakRSRPContext is a Context including attributes: contextAtrribute, contextCondition and contextValueRange.  Following are the allowed values:  -contextAttribute: "WeakRSRPThreshold"  -contextCondition: "is less than"  -contextValueRange: Float. | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| LowSINRRatioTarget | It describes the low SINR ratio target for the RAN SubNetwork that the intent expectation is applied.  LowSINRRatioTarget is an ExpectationTarget including attributes: targetName, targetCondition and targetValueRange.  Following are the allowed values:  -targetName: "WeakRSRPRatio"  -targetCondition: "is less than"  -targetValueRange: integer with allowed value [0,100].  - targetContext: LowSINRContext | type:ExpectationTarget  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| LowSINRRatioTarget.lowSINRContext | It describes the threshold for low SINR for RAN SubNetwork that the intent expectation is applied.  LowSINRContext is a Context including attributes: contextAtrribute, contextCondition and contextValueRange.  Following are the allowed values:  -contextAttribute: "LowSINRThreshold"  -contextCondition: "is less than"  -contextValueRange: integer. | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| aveULRANUEThptTarget | It describes the average UL RAN UE throughput target for RAN SubNetwork that the intent expectation is applied.  AveULRANUEThptTarget is an ExpectationTarget including attributes: targetName, targetCondition and targetValueRange.  Following are the allowed values:  -targetName: "AveULRANUEThpt"  -targetCondition: "is greater than"  -targetValueRange: integer | type: ExpectationTarget  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| aveDLRANUEThptTarget | It describes the average DL RAN UE throughput target for RAN SubNetwork that the intent expectation is applied.  AveDLRANUEThptTarget is an ExpectationTarget including attributes: targetName, targetCondition and targetValueRange.  Following are the allowed values:  -targetName: "AveDLRANUEThpt"  -targetCondition: "is greater than"  -targetValueRange: integer | type: ExpectationTarget  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| lowULRANUEThptRatioTarget | It describes the low UL RAN UE throughput ratio target for the RAN SubNetwork that the intent expectation is applied.  LowULRANUEThptRatioTarget is an ExpectationTarget including attributes: targetName, targetCondition and targetValueRange.  Following are the allowed values:  -targetName: "LowULRANUEThptRatio"  -targetCondition: "is less than"  -targetValueRange: integer with allowed value [0,100].  -targetContext: LowULRANUEThptContext | type: ExpectationTarget  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| LowULRANUEThptRatioTarget.lowULRANUEThptContext | It describes the threshold for the low UL RAN UE throughput of the RAN SubNetwork that the intent expectation is applied  LowULRANUEThptContext is a Context including attributes: contextAtrribute, contextCondition and contextValueRange.  Following are the allowed values:  -contextAttribute: "LowULRANUEThptThreshold"  -contextCondition: "is less than"  -contextValueRange: Float. | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| lowDLRANUEThptRatioTarget | It describes the low DL RAN UE throughput ratio target for the RAN SubNetwork that the intent expectation is applied.  LowDLRANUEThptRatioTarget is an ExpectationTarget including attributes: targetName, targetCondition and targetValueRange.  Following are the allowed values:  -targetName: "LowDLRANUEThptRatio"  -targetCondition: "is less than"  -targetValueRange: integer with allowed value [0,100].  -targetContext: LowDLRANUEThptContext | type: ExpectationTarget  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| owDLRANUEThptRatioTarget.lowDLRANUEThptContext | It describes the threshold for the low DL RAN UE throughput of the RAN SubNetwork that the intent expectation is applied  LowDLRANUEThptContext is a Context including attributes: contextAtrribute, contextCondition and contextValueRange.  Following are the allowed values:  -contextAttribute: "LowDLRANUEThptThreshold"  -contextCondition: "is less than"  -contextValueRange: Float. | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| serviceStartTime | This describes the start time at which the service shall be available. This contributes to the selection of the appropriate edge data network to be used for service deployment.  Following are the allowed values:  -contextAttribute: "serviceStartTime"  -contextCondition: "is equal than"  -contextValueRange: start time stamp | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| serviceEndTime | This describes the end time after which the service shall not be available. This contributes to the selection of the appropriate edge data network to be used for service deployment.  Following are the allowed values:  -contextAttribute: "serviceEndTime"  -contextCondition: "is equal than"  -contextValueRange: end time stamp | type:Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| edgeIdenfiticationId | This identifies the edge network where the service need to be deployed. This should be used when the edge identification is known to the consumer  Following are the allowed values:  -contextAttribute: "edgeIdentificationId"  -contextCondition: "is equal than"  -contextValueRange: EDNidentifier as defined in 28.538 [Y]. | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| edgeIdenfiticationLoc | This identifies the location where the service need to be deployed. This should be used when the edge identification is not known to the consumer  Following are the allowed values:  -contextAttribute: "edgeIdentificationTarget"  -contextCondition: "is equal than"  -contextValueRange: geographical target location. This will take a form of either single latitude & longitude or a TAI | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| dlThptPerUE | See clause 6.3.1 of TS 28.541[5] | type: ExpectationTarget  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| UlThptPerUE | See clause 6.3.1 of TS 28.541[5] | type: ExpectationTarget  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| dLLatency | See clause 6.3.1 of TS 28.541 [5] | type: ExpectationTarget  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| uLLatency | See clause 6.3.1 of TS 28.541 [5] | type: ExpectationTarget  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| coverageAreaTAList | See clause 6.3.1 of TS 28.541 [5] | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| uEMobilityLevel | See clause 6.3.1 of TS 28.541 [5] | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| resourceSharingLevel | See clause 6.3.1 of TS 28.541 [5] | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| maxNumberofUEs | See clause 6.3.1 of TS 28.541 [5] | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| activityFactor | See clause 6.3.1 of TS 28.541 [5] | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |
| uESpeed | See clause 6.3.1 of TS 28.541 [5] | type: Context  multiplicity: 1  isOrdered: N/A  isUnique: N/A  defaultValue: False  isNullable: True |

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

### 6.3.2 Create an intent

The Figure 6.3.2-1 illustrates the procedure for create a new intent.

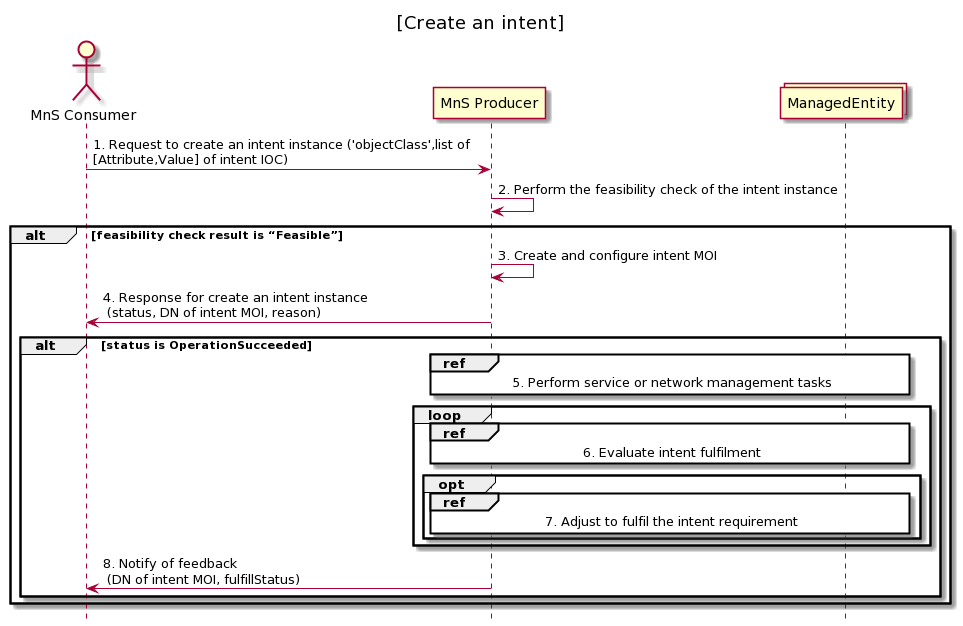


Figure 6.3.2-1 Procedure for create an intent

1. MnS Consumer sends a request to create an intent instance to MnS Producer with ‘objectClass’ and list of [Attribute,Value] for the new intent to be created. The detailed [Attribute,Value] see the concrete intent IOC defined in clause 6.2. ‘objectClass’ is the name for the concrete intent IOC.

2. MnS Producer perform the feasibility check of the intent instance. MnS Producer can perform the feasibility check and get the results based on latest statistics of network or service performance metrics, historical experience (e.g. experience based feasible value range or threshold of performance gain), current operating status including network resource utilization and availability, prediction results based on network simulation system, and predefined checking rules or policies.

Note: Whether to perform the feasibility check can be determined according to the feasibility check enabling policy (e.g. enforce to perform feasibility check in any case, enforce to perform feasibility check in specific cases, not to perform feasibility check in specific cases, not to perform feasibility check in any case). And the feasibility check enabling policy can be predefined/configured in the MnS Producer or sent with the intent creation request from the MnS Consumer.

If the feasibility check result is ‘feasible’,

3. Based on the request, the MnS Producer creates the concrete intent MOI (i.e. instance of intent IOC) with value for attribute ‘objectInstance’ allocated, and configure the new created intent MOI with the received list of [Attribute, Value]. ‘objectInstance’ is the identifier (DN) for the concrete intent MOI.

4. MnS Producer sends a response to the MnS Consumer with status (OperationSucceeded or OperationFailed) and ‘objectInstance’ of the created intent MOI. The response information may also include the possible reasons for the unsuccessful executions (e.g., conflicting with existing intents).

5. Based on the intent, MnS Producer identifies the MOI for managed entities (e.g. ManagedElement, ManagedFunction) and derives one or more executable management tasks (including deployment and configuration requirements) for these managed entities, then MnS producer deploys or configures corresponding managed entities to satisfy the intent.

6. During the execution of the intent, MnS Producer continuously monitors intent fulfilment status.

7. MnS Producer analyses and adjusts the managed entities to ensure the intent is continuously satisfied.

8. MnS Producer may notify MnS Consumer about the intent fulfilment information, including DN of intent MOI, and fulfillStatus.

If the feasibility check result is ‘infeasible’, MnS Producer does not create an intent MOI and feedback the feasibility check result information to MnS Consumer.

|  |
| --- |
| **End of Changes** |