

TM Forum Technical Report

Preference of Intent Handling Outcomes - Intent Extension Model

TR291G

Maturity Level: General availability (GA)	Team Approved Date: 04-Jul-2024
Release Status: Production	Approval Status: TM Forum Approved
Version 3.6.0	IPR Mode: RAND

Notice

Copyright © TM Forum 2024. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to TM FORUM, except as needed for the purpose of developing any document or deliverable produced by a TM FORUM Collaboration Project Team (in which case the rules applicable to copyrights, as set forth in the [TM FORUM IPR Policy](#), must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by TM FORUM or its successors or assigns.

This document and the information contained herein is provided on an “AS IS” basis and TM FORUM DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

TM FORUM invites any TM FORUM Member or any other party that believes it has patent claims that would necessarily be infringed by implementations of this TM Forum Standards Final Deliverable, to notify the TM FORUM Team Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this deliverable.

The TM FORUM invites any party to contact the TM FORUM Team Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this TM FORUM Standards Final Deliverable by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the TM FORUM Collaboration Project Team that produced this TM FORUM Standards Final Deliverable. TM FORUM may include such claims on its website but disclaims any obligation to do so.

TM FORUM takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this TM FORUM Standards Final Deliverable or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on TM FORUM's procedures with respect to rights in any document or deliverable produced by a TM FORUM Collaboration Project Team can be found on the TM FORUM website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this TM FORUM Standards Final Deliverable, can be obtained from the TM FORUM Team Administrator. TM FORUM makes no representation that any information or list

of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

Direct inquiries to the TM Forum office:

181 New Road, Suite 304
Parsippany, NJ 07054, USA
Tel No. +1 862 227 1648
TM Forum Web Page: www.tmforum.org

Table of Contents

Notice	2
Table of Contents	4
Executive Summary.....	6
Introduction.....	7
Scope.....	8
Revision Information.....	8
1. Notation and namespaces.....	9
2. Requesting preference	11
2.1. Preference deadline.....	12
2.2. Coordinated judgement requests	13
3. Providing preference	14
3.1. Preference objects in intent.....	14
3.2. Direct selection of options	17
3.3. Quantitative preference.....	18
3.4. Score interpretation hints	19
4. Administrative Appendix	21
4.1. Document History	21
4.1.1. Version History.....	21
4.1.2. Release History.....	21
4.2. Acknowledgments.....	21
5. Appendix A: Vocabulary Reference	23
5.1. about.....	23
5.2. aboutOption	23
5.3. JudgementRequest.....	23
5.4. judgementRequest.....	23
5.5. maxScore	24
5.6. minScore	24
5.7. neutral.....	24
5.8. objected.....	24
5.9. objectedThreshold	24
5.10. Preference	25
5.11. preference	25
5.12. preferenceDeadline.....	25
5.13. PreferenceScore.....	25
5.14. preferred	25

5.15.	preferredThreshold	26
5.16.	score.....	26
5.17.	ScoreInterpretation	26
5.18.	scoreInterpretation	26
5.19.	scoreValue.....	26
5.20.	Vocabulary.....	26

Executive Summary

The main task of an intent handler is to determine and execute actions needed to comply to all its intent determined requirements. It typically evaluates multiple potential solutions and applies the one that best meets the requirements set by intent owners. In this process, the intent handler might find multiple solutions with different outcomes, were the information provided in the intent is not sufficient to decide, which is preferential for the intent owner. The intent extension model "preference of handling outcomes" allows requesting a preference statement from an intent owner. The intent handler would present multiple alternative intent reports to the intent owner. Each of these reports represents the expected outcome for an available solution option. The intent owner can then provide feedback to the handler which one it prefers, and it can also state a gradual score of preference. This intent extension model provides the vocabulary for this interaction through objects in intent and intent reports. The procedure to request a preference statement is often referred to as "judgement request" and used synonymously.

Introduction

A strong separation of concerns between intent owner and intent handler means that the owner does not have a detailed understanding of the resources an intent handler has available, the actions an intent handler can take and how an action taken effects the resources and results in a particular intent compliance level. This means an intent owner does not implement the necessary domain logic to assess intent handler actions directly. This also means an intent handler cannot describe the action it wants to take to an intent owner and expect a sensible evaluation from it if this is a preferential action or not. The intent owner can only be approached regarding resulting effects of actions on intent requirement fulfillment.

Because of the same separation of concerns, an intent handler does not and should not understand why the intent owner has chosen to provide an intent and its detailed requirements. For example an intent handler in resource operation does not implement end-to end service considerations. And an intent handler on service operation layer does not implement business level consideration regarding contracts and business value considerations. This means an intent handler can and should not assume preferences of the owner, which are rooted in higher level considerations.

Despite the rich information and guidance an intent owner can provide proactively in the intent expression, an intent handler might find multiple possible solutions without a clear direction to decide. For example, this can happen if the intent contains two KPI based requirements. The intent handler has one solution available, which would provide a better result for the first KPI, while another possible solution would provide a better outcome with respect to the second KPI. Both solutions meet the requirements stated in the intent. Therefore, any of them would lead to compliance, but there is further optimization potential if the solution is chosen that prefers the KPI which has a higher business value. This would require the intent owner to state its preference.

This intent extension model allows an intent handler to generate intent reports representing the outcomes of solution alternatives. Instead of sending the solution details to the intent owner, the intent handler would predict the potential effects of the evaluated solutions on the intent requirements and provide these report to the intent owner for evaluation. This means the intent owner is approached on the level of requirement fulfillment outcomes. The level of information provided is intent report content, which the intent owner can understand after using equivalent content in the intent expression. For example, if the intent owner uses a latency KPI to provide a requirement in the intent, it would be able to evaluate its preference regarding achieved values for this KPI. This preference might, for example, be rooted in business value or cost considerations.

Furthermore, this intent extension model allows the intent owner to provide its preference through intent updates. It would modify the intent over the intent API to provide information regarding the preference request. It can state which options it prefers, which options should not be taken, and it can quantify to which extent it prefers one option over another one.

This intent extension model specifies two ways to provide a preference after a request. One specified possibility is to categorize options as preferred or not. The other specified option allows assigning a numerical score to each option. Further ways to state the preference of an intent owner can be specified with additional intent extension models.

Scope

This document is part of the TR291 series specifying intent extension models published as optional models within the TM Forum Intent Ontology (TIO).

This document defines general purpose vocabulary extending the intent models for assigning validity to intents and intent elements.

Revision Information

This revision v3.6.0 of the Preference of Intent Handling Outcomes model is an optional part of the TM Forum Intent Ontology (TIO) v3.6.0.

The revision v3.6.0 of this document replaces v.3.5.0 with the following changes:

- Minor editorial corrections.

1. Notation and namespaces

The intent probing model depends on the following models:

Model	Prefix	Namespaces	Published by	Purpose in the model
Intent Common Model	icm	http://tio.models.tmforum.org/tio/v3.6.0/IntentCommonModel/	TM Forum	General ontology model of intent and intent report expression. This document is part of the intent common model specification.
Intent Management Ontology	imo	http://tio.models.tmforum.org/tio/v3.6.0/IntentManagementOntology/	TM Forum	Defines basic vocabulary and concepts of intent based operation. This document specifies vocabulary for intent management functions and their roles, as well as the types of intent models within the TM Forum Intent Ontology (TIO).
Preference of Intent Handling Outcomes	pre	http://tio.models.tmforum.org/tio/v3.6.0/PreferenceOfHandlingOutcomes/	TM Forum	(This model) Defines the ontology model for requesting and providing preference about potential intent handling outcomes.
Logical Operators Ontology	log	http://tio.models.tmforum.org/tio/v3.6.0/LogicalOperators/	TM Forum	Specifies logical operators to express logical relationships and the evaluation of truth values.
Quantity Ontology	quan	http://tio.models.tmforum.org/tio/v3.6.0/QuantityOntology/	TM Forum	Introduces quantities and quantity operators.
Set Operators	set	http://tio.models.tmforum.org/tio/v3.6.0/SetOperators/	TM Forum	Specification of set operators.
Function Definition Ontology	fun	http://tio.models.tmforum.org/tio/v3.6.0/FunctionOntology/	TM Forum	Basic expression of functions.
Time Ontology in OWL	t	http://www.w3.org/2006/time#	W3C	Expression of date and time [owltime]
RDF version 1.1	rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	W3C	Providing fundamental modeling basics [rdf11]

Model	Prefix	Namespaces	Published by	Purpose in the model
RDF Schema 1.1	rdfs	http://www.w3.org/2000/01/rdf-schema#	W3C	Providing schema for knowledge modeling [rdfs11]
XML Schema	xsd	http://www.w3.org/2001/XMLSchema#	W3C	Providing data types for literal objects [xsd-1] [xsd-2]
Examples	ex	http://www..example.org/	IANA	Reserved domain name for examples

Table 1: Model references

The Preference of Intent Handling Outcomes model is based on the Resource Description Framework (RDF) [rdf, rdf_mt, rdf_primer] and the Resource Description Framework Schema (RDFS) [rdfs] published by the World Wide Web Consortium (W3C).

Furthermore, the preference of handling outcomes model depends on select models from the TM Forum Intent Ontology such as the intent common model for definition of context and conditions. Further models, such as the Logical Operators Ontology, the Quantity Ontology, the Set Operators model and the Function Definition Ontology can be used to express validity conditions.

2. Requesting preference

Intent handlers generate an intent report for every solution option they want to include in the evaluation by the intent owner. This means the intent handler would predict what effects a solution option would have on the intent requirements and generate an intent report that would reflect these hypothetical results. The intent handler would then generate a judgement request involving these hypothetical intent reports. A judgement request is an instance of class `pre:JudgementRequest`. It is defined as container of intent reports. This means, each intent report that shall be included in the judgement request is a member of the judgement request container. The property `pre:judgementRequest` associates a judgement request with an intent report. This means that each intent report that is part of this judgement request contains a set of all intent reports that are part of this request.

An intent report that is part of a judgement request is hypothetical and based on predictions rather than observations. This is similar to intent probing and therefore the properties `pro:expectedResult` and `pro:expectedObservation` as defined in the intent extension model TR291B for intent probing are used rather than `icm:result` and `icm:observed`.

For example:

```
ex:IR1
  a icm:IntentReport ;
  pre:judgementRequest ex:JR1 ;
  icm:about ex:Intent1 ;
  pro:expectedResult true ;
  icm:resultFrom ex:ER1
.
ex:JR1
  a pre:JudgementRequest ;
  rdfs:member ex:IR1 , ex:IR2 , ex:IR3
.
ex:TR1
  e icm:TargetReport ;
  rdfs:member ex:Slice12345
.
ex:ER1
  a icm:PropertyExpectationReport ;
  icm:targetReport ex:TR1 ;
  icm:about ex:E1 ;
  pro:expectedResult true ;
  icm:resultFrom ex:CR1
.
ex:CR1
  a icm:ConditionReport ;
  icm:about ex:C1 ;
  pro:expectedResult true ;
  pro:expectedObservation [ a icm:Observation ;
                           icm:observedTopic ex:Latency1 ;
                           rdf:value [ a quan:Quantity ;
                                       rdf:value "8"^^xsd:decimal ;
```

```

        unit "ms" ]
    ];

```

This example shows a possible intent report that is part of a judgement request. Other intent reports that are part of this request are `ex:IR2` and `ex:IR3`, however, they are not explicitly shown in this example. The full scope of the judgement request is specified by `ex:JR1`. The intent report `ex:IR1` is interpreted as part of a judgement request using the `pre:judgementRequest` property to assign the judgement request individual `ex:JR1` of the intent report. This intent report states that the intent handler is considering three solution options. The expected results of one of the options is represented by this intent report `ex:IR1`. It states that the operated network would comply to the intent after executing this solution, and it would result in a latency on the selected slice of 8ms.

2.1. Preference deadline

The intent owner might need some time to evaluate the intent reports associated with a judgement request. However, the intent handler would typically wait until it receives a preference statement by the intent owner. After the preference is received, the intent handler would select a solution accordingly and proceed to its execution. This process is usually time-critical, because the intent handler tries to solve an issue and the longer an issue remains unsolved, the longer users might be exposed to degraded service or the resources are not used as optimally as possible. For this reason it is possible to assign a preference deadline to a judgement request.

The property `pre:preferenceDeadline` states the point in time until the preference needs to be provided. The intent handler waits for a preference statement provided by the intent owner until this point in time. If until then none or incomplete preference statements are received, the intent handler will proceed and select a solution with the information available. The range of `pre:preferenceDeadline` is `t:Instant`. This means a temporal instance according to the time ontology in OWL would be used to specify the point in time that marks the preference deadline.

For example:

```

ex:IR1
  a icm:IntentReport ;
  pre:judgementRequest ex:JR1 ;
  icm:about ex:Intent1 ;
  pro:expectedResult true ;
  icm:resultFrom ex:ER1
.
ex:JR1
  a pre:JudgementRequest ;
  rdfs:member ex:IR1 , ex:IR2 , ex:IR3 ;
  pre:preferenceDeadline [ t:inXSDDateTimeStamp "2023-07-
05T01:00+01:00"^^xsd:dateTimeStamp ]
.

```

This example shows the setting of a preference deadline as part of the judgement request. Here the preference is expected from the intent owner until the 5th of July 2023 at 01:00 at night according to the timezone GMT+1. The deadline is specified using time specifications according to XML Schema.

2.2. Coordinated judgement requests

Intent handlers typically handle multiple intents containing requirements from different sources and regarding different use cases, but with overlapping impacts on the operated resources and services within the scope of responsibility of the intent handler within an intent handling domain. Intents are therefore not handled in isolation from each other because solutions and actions and intent handler applies to the managed resources and services frequently have impacts on the compliance results of multiple intents. This is either intended if the action taken is supposed to improve the outcome or a particular intent, or it is a collateral effect. The impact on an intent might be a shift in observed results, such as higher or lower levels for KPIs. Often an improvement on one intent might cause a degradation to another intent. And if this results in losing compliance to some intent requirements, the applied action constitutes a conflict. The intent handler will seek to find conflict free actions or, if this is not possible, find the solution with a globally most preferential result.

With multiple intents involved in intent handling, partial degradation or conflict might also concern multiple intent owners. In order to assess the situation and come to an overall optimized solution, Intent handlers can use the judgement request towards multiple intent owners concerning multiple intent simultaneously. However, the judgement request is issued as described in this document on a per intent basis. This therefore involves to create a judgement request per intent. They can be coordinated, for example by providing the same preference deadline and the intent handler might also preserve the information that all these individual judgement requests are concerning the same set of alternative solutions. However, this information of parallel judgement requests is not shared with the intent owner. Therefore, additional vocabulary for coordinated judgement requests is needed and not defined in this intent extension model.

3. Providing preference

After receiving intent reports that are part of a judgement request, an intent owner would evaluate the reported outcomes. This will determine which reported results are preferential and which are not. The intent owner might rank the presented outcomes, identify the most preferred ones and the outcomes it would like to avoid. Potentially the preference can also be quantified with a score that establishes a relative degree of preference. The intent owner will then communicate its preference to the intent handler by adding its preference information to the intent. The following chapters specify the intent model vocabulary for presenting the intent owner's preferences in reply to a judgement request.

This intent extension model specifies further embodiments of preference can be added by additional intent extension models.

3.1. Preference objects in intent

To communicate its preferences regarding a judgement request to the requesting intent handler, the intent owner would update the respective intent and add an instance of class `pre:Preference`. This follows the principle that intent handlers communicate to the intent owner through intent reports and intent owners to the handler through intent.

The property `pre:preference` is used to assign the preference object to an intent. This is the preference object that states the preference of the intent owner with respect to a distinct judgement request.

The property `pre:about` is a reference to the judgement request instance stated in the intent reports that are part of the judgement request. The judgement request object uniquely identifies the request and the intent owner can correlate its request to it using `pre:about`. Preferences provided without a reference to a judgement request can be ignored or rejected by the intent handler.

For example:

```
ex:Intent1
  a icm:Intent ;
  log:allOf ( ex:E1 ex:E2 )
.
ex:E1
  a icm:PropertyExpectation ;
  icm:target ex:T1 ;
  log:allOf ( ex:C1 ex:C2 )
.
ex:C1
  a log:Condition ;
  quan:smaller ( ex:Latency1
    [ rdf:value "10"^^xsd:decimal ;
      unit "ms" ]
  )
.
ex:C2
  a log:Condition ;
  quan:atLeast ( ex:TroughputPerUser1
```

```
[ rdf:value "10"^^xsd:decimal ;
  unit "Mbps" ]
)
```

This is an example intent containing two requirements based on KPIs. It requires a latency of smaller than 10ms and available throughput per user of at least 10Mbps. The intent handler is looking for configuration actions that would meet these requirements. It found two options, which both meet the intent requirements, but with different expected outcomes on the KPIs. The intent handler decides to seek decision help from the intent owner using a judgement request. It would generate two intent reports that reflect the expected values the two identified configuration actions would deliver:

```
ex:IR1
  a icm:IntentReport ;
  pre:judgementRequest ex:JR1 ;
  icm:about ex:Intent1 ;
  pro:expectedResult true ;
  icm:resultFrom ex:ER1
.

ex:JR1
  a pre:JudgementRequest ;
  rdfs:member ex:IR1 , ex:IR2 ;
  pre:preferenceDeadline [ t:inXSDDateTimeStamp "2023-07-
05T01:00+01:00"^^xsd:dateTimeStamp ]
.

ex:ER1
  a icm:PropertyExpectationReport ;
  icm:about ex:E1 ;
  pro:expectedResult true ;
  icm:resultFrom ex:CR1
.

ex:CR1
  a icm:ConditionReport ;
  icm:about ex:C1 ;
  pro:expectedResult true ;
  pro:expectedObservation [ a icm:Observation ;
    icm:observedTopic ex:Latency1 ;
    rdf:value [ a quan:Quantity ;
      rdf:value "8"^^xsd:decimal ;
      unit "ms" ]
  ] ;
  pro:expectedObservation [ a icm:Observation ;
    icm:observedTopic ex:ThroughputPerUser1 ;
    rdf:value [ a quan:Quantity ;
      rdf:value "20"^^xsd:decimal ;
      unit "Mbps" ]
  ] ;
.
```

This is the first intent report as part of a judgement request. It states that a solution option is available that is expected to result in a latency of 8ms and a throughput per user of 20Mbps.

```

ex:IR2
  a icm:IntentReport ;
  pre:judgementRequest ex:JR1 ;
  icm:about ex:Intent1 ;
  pro:expectedResult true ;
  icm:resultFrom ex:ER2
.
ex:JR1
  a pre:JudgementRequest ;
  rdfs:member ex:IR1 , ex:IR2 ;
  pre:preferenceDeadline [ t:inXSDDateTimeStamp "2023-07-
05T01:00+01:00"^^xsd:dateTimeStamp ]
.
ex:ER2
  a icm:PropertyExpectationReport ;
  icm:about ex:E1 ;
  pro:expectedResult true ;
  icm:resultFrom ex:CR2
.
ex:CR2
  a icm:ConditionReport ;
  icm:about ex:C1 ;
  pro:expectedResult true ;
  pro:expectedObservation [ a icm:Observation ;
    icm:observedTopic ex:Latency1 ;
    rdf:value [ a quan:Quantity ;
      rdf:value "5"^^xsd:decimal ;
      unit "ms" ]
    ] ;
  pro:expectedObservation [ a icm:Observation ;
    icm:observedTopic ex:ThroughputPerUser1 ;
    rdf:value [ a quan:Quantity ;
      rdf:value "11"^^xsd:decimal ;
      unit "Mbps" ]
    ] ;
.

```

This is the second intent report as part of a judgement request. It states that a solution option is available that is expected to result in a latency of 5ms and a throughput per user of 11Mbps. The judgement request individual is stated in both associated intent reports. This example judgement request only consists of two options.

In this example the intent owner is asked to provide its preference regarding these two options. The central question the intent owner needs to decide in this example would be if it prefers to focus on achieving a better latency or a better throughput. The following chapters define various ways the preference can be expressed. The example is continued in the respective chapters.

3.2. Direct selection of options

After evaluating the intent reports that are part of a judgement request, the intent handler may have identified reported expected results it considers preferential and those which are not. The intent owner can use the following properties of the preference added to the intent to communicate these evaluation results.

The property `pre:preferred` refers to an intent report the intent owner considers preferential. It is referring to an intent report presented by the judgement request. This means the respective solution action represented by this report is preferential.

The property `pre:neutral` refers to an intent report the intent owner does not mind, but also does not express an explicit preference for. This is the assumed default for options without a dedicated preference statement. It is also assumed if the intent owner does not provide a preference statement before the preference deadline.

The property `pre:objected` refers to an intent report the intent owner considers problematic from its perspective. The respective solution should not be chosen by the intent handler.

If the intent owner has provided a preference statement for every reported option, the intent handler can conclude that the preference is complete even before a preference deadline has expired. This avoids unnecessary delays in applying needed actions.

For example: (Continued example from chapter 3.1)

```
ex:Intent1
  a icm:Intent ;
  log:allOf ( ex:E1 ex:E2 ) ;
  ex:preference ex:Pre1
.
ex:Pre1
  a pre:Preference ;
  pre:about ex:JR1 ;
  pre:preferred ex:IR2 ;
  pre:neutral ex:IR1
.
ex:E1
  a icm:PropertyExpectation ;
  icm:target ex:T1 ;
  log:allOf ( ex:C1 ex:C2 )
.
ex:C1
  a log:Condition ;
  quan:smaller ( ex:Latency1
    [ rdf:value "10"^^xsd:decimal ;
      unit "ms" ]
  )
.
ex:C2
  a log:Condition ;
  quan:atLeast ( ex:TroughputPerUser1
    [ rdf:value "10"^^xsd:decimal ;
      unit "Mbps" ]
  )
```

)

This is an example shows modifications applied to intent `ex:Intent1` for providing a preference. The intent owner has added the preference instance `ex:Pre1` and assigned it to the intent using the `ex:preference` property. In this example the intent owner states that this preference is provided for the judgement request `ex:JR1`. It also states that it considers the results shown by intent report `ex:IR2` as preferential. This implicitly means that it prefers the intent handler would choose the actions associated with this intent report. Furthermore, the intent owner states that it has a neutral opinion about the results represented by the intent report `ex:IR1`. The intent handler therefore prefers to prioritize overachieving with respect to latency rather than throughput. It does not mind the result represented by `ex:IR1`, but this outcome does not have an increased business value. Based on this input the intent handler can now choose a preferred solution option. Note that the intent requirements or other parts of the intent were not modified. The intent owner has just added a preference object.

Adding preference objects would lead to accumulation of additional objects over time. It might be preferential to keep older preference objects in the intent, because the intent handler might learn from them about the preferences of an intent owner and optimize its decision-making. It is also possible that the intent owner regularly removes older preference objects from intent. It can for example perform a cleanup every time it modifies the intent. These are however considered to be implementation details of an intent manager and not subject to this specification.

3.3. Quantitative preference

The intent owner can provide a numerical score per solution. This can be used in addition to or instead of the preference statements provided according to chapter 3.2.

An instance of class `pre:PreferenceScore` is added to the preference object within the intent. It is assigned to a preference object using the property `pre:score`. The property `pre:aboutOption` refers to the intent report, for which this preference score object states the score. The property `pre:scoreValue` assigns a numerical value to the preference score object. This is the numerical score associated with an option.

For example: (Continued example from chapter 3.1)

```
ex:Intent1
  a icm:Intent ;
  log:allOf ( ex:E1 ex:E2 ) ;
  ex:preference ex:Pre1
.
ex:Pre1
  a pre:Preference ;
  pre:about ex:JR1 ;
  pre:score [ pre:aboutOption ex:IR1 ;
              pre:scoreValue 5
            ] ;
  pre:score [ pre:aboutOption ex:IR2 ;
              pre:scoreValue 9
            ]
.
```

```

ex:E1
  a icm:PropertyExpectation ;
  icm:target ex:T1 ;
  log:allOf ( ex:C1 ex:C2 )
.
ex:C1
  a log:Condition ;
  quan:smaller ( ex:Latency1
    [ rdf:value "10"^^xsd:decimal ;
      unit "ms" ]
  )
.
ex:C2
  a log:Condition ;
  quan:atLeast ( ex:TroughputPerUser1
    [ rdf:value "10"^^xsd:decimal ;
      unit "Mbps" ]
  )
.

```

This is an example shows a preference provided using numerical scores. The intent owner has assigned a higher score to the preferred option represented by intent report ex:IR2.

3.4. Score interpretation hints

An intent owner can provide its preference by assigning a score to each option as described in chapter 3.3. However, the intent handler would need to interpret the score. By default, it would consider a score of 0 to be the minimum score referring to a not preferred option. Furthermore, it would interpret the highest used score as maximum and most preferred. If this interpretation of the score is not correct, the intent owner can provide explanations regarding the score value range and thresholds.

The intent owner can provide further information about its preference scoring with a score interpretation object. It is specified by an individual of class `pre:ScoreInterpretation`. It is a subclass of the information class `icm:Information` as defined in the intent common model TR290A. A score interpretation is associated with a preference using the property `pre:scoreInterpretation`.

The following properties are available to provide information about the score, such as its value range and the interpretation of certain value ranges. Further information can be added by additional intent extension models.

The property `pre:maxScore` states the maximum value or full score. If not provided, the highest score used within the preferences of this intent owner within this judgement request is considered to be the maximum score.

The property `pre:minScore` states the minimum value used. If not provided, a minimum score of 0 is assumed by default.

The property `pre:preferredThreshold` defines the minimum score value for preferred options. An option with at least this score is considered to be preferred. This is equivalent to an explicit `pre:preferred` statement as described in Chapter 3.2.

The property `pre:objectedException` defines the maximum score for objected options. An option with lower than this score should be avoided. This is equivalent to the explicit `pre:objectedException` statement as described in Chapter 3.2. A score of at least the objected threshold and lower than the preferred threshold is interpreted as neutral.

For example:

```
ex:Intent1
  a icm:Intent ;
  log:allOf ( ex:E1 ex:E2 ) ;
  ex:preference ex:Pre1
.
ex:Pre1
  a pre:Preference ;
  pre:about ex:JR1 ;
  pre:score [ pre:aboutOption ex:IR1 ;
    pre:scoreValue 5
  ] ;
  pre:score [ pre:aboutOption ex:IR2 ;
    pre:scoreValue 9
  ]
  pre:scoreInterpretation [ pre:maxScore 10 ;
    pre:minScore -10 ;
    pre:preferredThreshold 6 ;
    pre:objectedException -6 ;
  ]
.
```

This example shows a preference provided with scores and a score interpretation hint. It specifies that the used score value range is between -10 and 10. Options with scores of at least 6 are considered preferred and scores of smaller than -6 are objected.

4. Administrative Appendix

4.1. Document History

4.1.1. Version History

Version Number	Date Modified	Modified by:	Description of changes
1.0.0	31-Mar-2022	Alan Pope	Final edits prior to publication
1.1.0	01-Jun-2022	Alan Pope	Final edits prior to publication
3.4.0	29-Feb-2024	Alan Pope	Final edits prior to publication
3.5.0	03-May-2024	Alan Pope	Final edits prior to publication
3.6.0	04-Jul-2024	Alan Pope	Final edits prior to publication

4.1.2. Release History

Release Status	Date Modified	Modified by:	Description of changes
Pre-production	31-Mar-2022	Alan Pope	Initial Release
Pre-production	02-May-2022	Adrienne Walcott	Updated to reflect TM Forum Member Evaluated status
Pre-production	01-Jun-2022	Alan Pope	Updated to v1.1.0
Pre-production	04-Jul-2022	Adrienne Walcott	Updated to reflect TM Forum Member Evaluated status
Pre-production	29-Feb-2024	Alan Pope	Updated to v3.4.0
Production	26-Apr-2024	Adrienne Walcott	Updated to reflect TM Forum Approved status
Pre-production	03-May-2024	Alan Pope	Updated to v3.5.0
Production	28-Jun-2024	Adrienne Walcott	Updated to reflect TM Forum Approved status
Pre-production	04-Jul-2024	Alan Pope	Updated to v3.6.0
Production	30-Aug-2024	Adrienne Walcott	Updated to reflect TM Forum Approved status

4.2. Acknowledgments

Team Member (@mention)	Company	Role*
Jörg Niemöller	Ericsson	Author, Project Co-Chair
Kevin McDonnell	Huawei	Project Co-Chair
Yuval Stein	Amdocs	Project Co-Chair

Team Member (@mention)	Company	Role*
Kamal Maghsoudlou	Ericsson	Key Contributor
Leonid Mokrushin	Ericsson	Key Contributor
Marin Orlić	Ericsson	Key Contributor
Aaron Boasman-Patel	TM Forum	Additional Input
Alan Pope	TM Forum	Additional Input
Dave Milham	TM Forum	Additional Input
Xiao Hongmei	Inspur	Reviewer

**Select from: Project Chair, Project Co-Chair, Author, Editor, Key Contributor, Additional Input, Reviewer*

5. Appendix A: Vocabulary Reference

This chapter contains a reference definition of all model vocabulary. It is sorted alphabetically.

5.1. about

The property `pre:about` is a reference to the judgement request instance stated in the intent reports that are part of the judgement request.

Instance of: `rdf:Property`

Domain: `pre:Preference`

Range: `pre:JudgementRequest`

5.2. aboutOption

The property `pre:aboutOption` refers to the intent report, for which this preference score object states the score.

Instance of: `rdf:Property`

Domain: `pre:PreferenceScore`

Range: `icm:IntentReport`

5.3. JudgementRequest

The class `pre:JudgementRequest` represents a request from the intent handler to an intent owner to provide its preference regarding the expected results of available solution options on the intent requirements. It is a subclass of `rdfs:Container` with members that refer to intent reports requesting the expected outcomes for available options.

Instance of: `rdfs:Class`

Subclass of: `rdfs:Container`

5.4. judgementRequest

The property `pre:judgementRequest` associates a judgement request of class `pre:JudgementRequest` with an intent report.

Instance of: `rdf:Property`

Domain: `icm:IntentReport`

Range: `pre:JudgementRequest`

5.5. maxScore

The property `pre:maxScore` states the maximum value or full score used in providing preference scores. If not provided, the highest score used within the preferences of this intent owner within this judgement request is considered to be the maximum score.

Instance of: `rdf:Property`

Domain: `pre:ScoreInterpretation`

5.6. minScore

The property `pre:minScore` states the minimum score value used in providing preference scores. If not provided, a minimum score of 0 is assumed by default.

Instance of: `rdf:Property`

Domain: `pre:ScoreInterpretation`

5.7. neutral

The property `pre:neutral` refers to an intent report the intent owner does not mind, but also does not express an explicit preference for. This is the assumed default for options without a dedicated preference statement. It is also assumed if the intent owner does not provide a preference statement before the preference deadline.

Instance of: `rdf:Property`

Domain: `pre:Preference`

Range: `icm:IntentReport`

5.8. objected

The property `pre:objected` refers to an intent report the intent owner considers problematic from its perspective. The respective solution should not be chosen by the intent handler.

Instance of: `rdf:Property`

Domain: `pre:Preference`

Range: `icm:IntentReport`

5.9. objectedThreshold

The property `pre:objectedThreshold` defines the maximum score for objected options. An option with lower than this score should be avoided.

Instance of: `rdf:Property`

Domain: `pre:ScoreInterpretation`

5.10. Preference

The class `pre:Preference` represents preferences of an intent owner with respect to the options presented in a judgement request.

Instance of: `rdfs:Class`

5.11. preference

The property `pre:preference` associates a preference provided by an intent owner to an intent.

Instance of: `rdf:Property`

Domain: `icm:Intent`

Range: `pre:Preference`

5.12. preferenceDeadline

The property `pre:preferenceDeadline` states the point in time until the preference needs to be provided. The intent handler waits for a preference statement provided by the intent owner until this point in time. If until then none or incomplete preference statements are received, the intent handler will proceed and select a solution with the information available.

Instance of: `rdf:Property`

Domain: `pre:JudgementRequest`

Range: `t:Instant`

5.13. PreferenceScore

An instance of class `pre:PreferenceScore` is added to the preference object within the intent by the intent owner to present its preferences regarding an option and quantify its preference with a numerical score.

Instance of: `rdfs:Class`

5.14. preferred

The property `pre:preferred` refers to an intent report the intent owner considers preferential. It is referring to an intent report presented by the judgement request.

Instance of: `rdf:Property`

Domain: `pre:Preference`

Range: `icm:IntentReport`

5.15. preferredThreshold

The property `pre:preferredThreshold` defines the minimum score value for preferred options. An option with at least this score is considered to be preferred.

Instance of: `rdf:Property`

Domain: `pre:ScoreInterpretation`

5.16. score

The property `pre:score` associates a preference score to a preference.

Instance of: `rdf:Property`

Domain: `pre:Preference`

Range: `pre:PreferenceScore`

5.17. ScoreInterpretation

An instance of class `pre:ScoreInterpretation` is added to the preference object within the intent by the intent owner to provide further information about the interpretation of scores used when specifying preferences

Instance of: `rdfs:ClassSubclass of: rdfs:Container`

5.18. scoreInterpretation

A score interpretation is associated with a preference using the property `pre:scoreInterpretation`.

Instance of: `rdf:PropertyDomain: pre:PreferenceRange:`

`pre:ScoreInterpretation`

5.19. scoreValue

The property `pre:scoreValue` assigns a numerical value to the preference score object. This is the numerical score associated with an option.

Instance of: `rdf:Property`

Domain: `pre:PreferenceScore`

5.20. Vocabulary

The object `pre:Vocabulary` is a container of all model elements.

Instance of: `rdfs:Container`