

TM Forum Technical Report

Intent Probing - Intent Extension Model

TR291B

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.....	5
Introduction.....	6
Scope.....	7
Revision Information.....	7
1. Notation and namespaces.....	8
2. Requesting intent probing.....	10
3. Reporting probing results	11
4. Administrative Appendix.....	13
4.1. Document History	13
4.1.1. Version History.....	13
4.1.2. Release History.....	13
4.2. Acknowledgments.....	13
5. Appendix A: Vocabulary Reference.....	15
5.1. expectedObservation	15
5.2. expectedResult	15
5.3. Vocabulary.....	15

Executive Summary

Probing of intent is a dialog between an intent owner and intent handler for the purpose of feasibility and potential outcome assessment of intent requirements. It allows the intent owner to send an intent to an intent handler asking for a prediction of the outcome and fulfillment. In this respect the intent handler is not obliged to actually meet the requirements. It is rather asked to provide an estimate of what it would achieve hypothetically, if it would need to meet the requirements within the intent. This provides the intent owner with valuable information regarding what it can expect if sending this intent for operation asking for actually complying to the intent.

Intent probing is optional and can therefore only be used between an intent owner and an intent handler, it both support it. Intent model vocabulary is provided in this intent extension model rather than being part of the mandatory intent common model.

Introduction

An intent handler receives requirements from an intent owner via an intent enabled API such as TMF921. If the intent owner provides the intent for operation, the handler is obliged to comply to the requirements stated within the intent. It would then take necessary action within the network and with the resources available to comply. This might or might not be possible, and the intent handler would report the results back to the intent owner. This is the basic intent handling covered by the intent common model.

However, the intent owner might benefit from having more information regarding feasibility of intent requirements and expected outcomes, before it decides which exact requirements to ask compliance for. For example, if the intent owner is distributing coordinated requirements to multiple intent handlers within multiple resource domains in parallel, it would set individual requirements in distinct domains to that their contribution to an end to end use case meets overall customer requirements. This is an intent requirements translation, breakdown and distribution process typical for intent owners. In such scenarios, an intent owner can send initial sets of requirements, observe the results as stated in intent reports and then optimize successively by gradually changing and re-distributing requirements.

This trial and error approach is possible with the base operations of intent sending and reporting covered by an intent enabled API and intent expression according to the vocabulary specified by the intent common model. However, it can lead to high system load due to repeated intent changes and reduced customer experience. The intent owner can avoid most of this using intent probing. For intent probing the intent is provided to the intent handler for exploration rather than operation. An intent API would provide the intent with a "probe intent" request rather than "set intent" is probing is wanted. An intent send with "probe intent" requests that the intent handler assesses and predicts if and to what extent it can comply to the requirements. The main difference between set intent and probed intent is that probing does not require that the intent handler actually meets the requirements with the resources it has available. The handling of a probed intent is therefore hypothetical.

The intent expression contains a fully defined and complete intent in both cases of providing intent for operation and providing intent for exploration. The distinction between operation and exploration is made by the request made as "set intent" or "probe intent" on the API. All intent expectation classes can be used in the same way as for intent requested for operation. The idea is that there is no difference between the intents used in operation and in exploration/probing.

The intent handler is providing its probing results through intent reports. The reporting is steered by reporting expectations within the intent. The same reporting events and criteria as provided by reporting expectations apply for probed intents as well as for set intent. This also means that intent probing would result in a sequence of intent reports similar to the intent reports expected to appear if the intent would be set for operation. Probing is therefore like intent handling in a sandbox. It looks like operational intent handling to the intent owner with respect to content and timing of intent reports, but without actually using network resources or changing configurations by the intent handler. The intent reports would however report as if resources would be used or configurations would have been changed. Depending on the scope of reporting and events for reporting specified in the reporting expectations, the requested details can vary from simple feasibility and acceptance of the intent by the handler to detailed predicted KPI values.

By receiving the probing results in intent reports, the intent owner can observe the hypothetical intent operation. It can choose to modify the probed intent to assess requirement variations. It is also possible to do multiple probing sessions with different intent variations in parallel. Also, multiple probing sessions towards multiple distinct intent handlers are possible. In this sense, the intent probing mechanism allows an intent owner to gather information and learn about current network operation and current feasibility with respect to new requirements it would use. At some point the intent owner can decide to transition from exploration to operation. The owner can request a removal of probed intent and then assemble a new intent as needed. The intent owner would provide this intent for operation. Alternatively, the intent owner can use the same intent with a set intent request after it was previously provided with a probe intent request and without removing the probed intent first. This would transition and intent from exploration to operation. This might happen, for example, if the intent owner has initiated intent probing, and is satisfied with the replies by the intent handler.

This intent extension model for intent probing introduces vocabulary needed for probing in the intent expression model. It primarily introduces a way to distinguish observed results from operation from hypothetical and predicted results in probing.

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 Intent Probing 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).
Intent Probing	pro	http://tio.models.tmforum.org/tio/v3.6.0/IntentProbing/	TM Forum	(This model) Defines the ontology model for intent probing
Conditions and 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]
RDF Schema 1.1	rdfs	http://www.w3.org/2000/01/rdf-schema#	W3C	Providing schema for knowledge modeling [rdfs11]

Model	Prefix	Namespaces	Published by	Purpose in the model
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 intent probing 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 intent probing 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 intent probing

Intents used in intent probing are identical to intent used in operation. There is, for example, no extra marking that an intent is currently used in probing rather than operation. This is distinguished on the API used for managing the intent (e.g. TMF921). If the intent is requested using "probe intent" the intent shall be interpreted by the intent handlers as requested to exploration. Additional markings of probing within the intent are avoided to not create issues from consistency of information.

3. Reporting probing results

Intent reports for probed intent are generated according to reporting expectations as defined in the intent common model. This means the scope of reporting is determined by the target of the reporting expectation and the conditions for reporting are stated using intent handler events. However, in intent probing the handler would need to predict, when respective events would occur if the intent had been provided for operation. It would also need to simulate the usage of resources and changes in configurations from potential actions it would perform if the intent were requested for operation. In this respect the intent report would have similar content like an intent report issued in operation. However, in probed intent, the reported probing results are distinguished from reported observations.

According to the intent common model TR290B an intent report contains the property `icm:result` to state the logical evaluation result. When provided for expectation reports or as property of the intent report, it is providing the compliance to requirements. In intent reports used for stating probing results, the property `pro:expectedResult` would be used instead. This states that the provided result is not based on the system actually complying to the intent requirements, but it states that the intent handler expects that it can comply if asked.

The intent common model uses the property `icm:observed` to provide the measured or otherwise obtained values of metrics, states or KPIs through `icm:Observation` objects. In intent reports in probing, values need to be provided that represent predicted of expected results rather than actual observations from the operated domain. The property `pro:expectedObservation` provides a predicted value. It is also using `icm:Observation` objects in its range.

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:C1
  a log:Condition ;
  quan:smaller ( ex:Latency1
    [ rdf:value "10"^^xsd:decimal ;
      unit "ms" ]
  )
.
```

This is the intent used in the example. It specifies a condition regarding latency contributing to a property expectation. The example is simplified and leaves out elements not in focus of this chapter, such as the target definition or additional expectations. In this example, this intent is provided for exploration/probing.

```
ex:IR1
  a icm:IntentReport ;
```

```

icm:about ex:Intent1 ;
pro:expectedResult true ;
icm:resultFrom ex:ER1
.
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" ]
                             ] ;
  icm:observed [ a icm:Observation ;
                 icm:observedTopic ex:Latency1 ;
                 rdf:value [ a quan:Quantity ;
                             rdf:value "20"^^xsd:decimal ;
                             unit "ms" ]
                 ]
.

```

This is a possible report for the intent provided above if it is used in exploration/probing. It is using the property `pro:expectedResult` for the intent report, expectation report and condition report to identify the reported value as a predicted one within intent probing. Furthermore, the condition report provides an expected observation alongside an observed one. The expected observation states the predicted value for the latency KPI is the intent would be operational. With `icm:observed` it also states the currently observed value. This means the intent handler would state what latency it achieves without the intent setting a requirement for it, and what it would achieve if it would act on the requirement.

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	Initial Release
1.1.0	01-Jun-2022	Alan Pope	Updated to beta
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 (Beta)
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 (GA)
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. expectedObservation

The property `pro:expectedObservation` states predicted values for metrics rather than actually observed ones.

Instance of: `rdf:Property`

Range: `icm:Observation`

5.2. expectedResult

The property `pro:expectedResult` is used to state the predicted evaluation result for a probed intent.

Instance of: `rdf:Property`

Domain: `xsd:Boolean`

5.3. Vocabulary

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

Instance of: `rdfs:Container`