

TM Forum Technical Report

Proposal of Best Intent - Intent Extension Model

TR291C

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
1. Introduction	6
1.1. What is proposed?	6
1.2. Intent owner and handler interaction	7
1.3. Scope	8
1.4. Revision Information	8
2. Notation and namespaces.....	9
3. Requesting proposals.....	11
3.1. Integration into intent	11
3.2. Proposal Scope	12
3.3. Proposal scope with multiple entries	13
3.4. Multiple simultaneous proposals	14
3.5. Proposals for complex requirements	16
3.6. Proposal report initiation	16
4. Providing a best proposal	17
4.1. Expressing a best proposal	17
5. Administrative Appendix	19
5.1. Document History	19
5.1.1. Version History.....	19
5.1.2. Release History.....	19
5.2. Acknowledgments.....	19
6. Appendix A: Vocabulary Reference	21
6.1. BestProposalExpectation	21
6.2. BestProposalReport	21
6.3. Proposal	21
6.4. proposed	21
6.5. proposedFor	21
6.6. proposedBest	22
6.7. proposal	22
6.8. Vocabulary	22

Executive Summary

The proposal of best intent model is an intent extension model that introduces optional additional vocabulary for intent and intent report expression. This model enables the request for proposals about intent defined requirements. More specifically, an intent owner can request a proposal from an intent handler regarding the most challenging requirement of a certain type it would be able to fulfil. The model also covers the vocabulary needed for an intent handler sending its respective best proposal to the intent owner.

Introduction

If an intent handler receives requirements within an intent it would try to take actions necessary to fulfil these requirements. 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. The intent owner would therefore know if its requirements are met or not. But what the intent owner does not yet know is which requirements are the most challenging ones the intent handler can successfully reach compliance for. The proposal of best intent model allows an intent owner to explore this question. This is therefore exploring the edge cases of possible compliance.

What is proposed?

If an intent handler receives requirements, it would then choose a solution and take actions for reaching compliance. Typically, there is a range of different solutions available and the intent handler would sensibly choose the most economical one. This means, for example, the intent handler would not spend more resources than necessary. This is then the most optimal solution chosen for the current requirements and situation. If the intent was provided for operation, the intent handler would act according to this optimal solution and intent reports would reflect the results. If the intent was provided for exploration with an intent probing request, the intent handler would still choose the same optimal solutions, but it would not necessarily execute them. The respective intent reports for probing would reflect the potential results if the same intent with its requirements would be provided for operation.

In both cases of intent being provided for exploration or operation, the intent handler would have other solutions available that can potentially deliver even better results. This means, the requirements could be over-fulfilled. For example, if a latency of 100 ms is required, the intent handler might have solutions to reach 50, 90, 150 or 200 ms. The intent handler would choose the solution expected to deliver 90 ms, because it would meet the requirements. It would normally not choose the solution that can deliver 50 ms, because it would bind resources without a requirement actually demanding it. This is not optimal.

The proposal of best intent model allows an intent owner to request that the intent handler state the best result for a requirement possible. In the example given in the previous paragraph, regarding a latency requirement, the best possible result according to available solutions would be 50 ms. The intent handler would state this in an intent report. This means, for this example, the intent handler would state in the intent report that it has reached 90 ms with the chosen solution and in addition it would state that it has solutions available for reaching 50 ms if required. This is referred to as proposal.

The proposal of best intent model can be used with intent provided for operation as well as intent provided for exploration (probing). In both cases the proposal would reflect the edge compliance result for a requirement.

If the requirement cannot be fulfilled, the proposal for best intent would still show the best level of compliance that can be reached. However, in this case the proposed value would be below the required compliance level. For example, if a latency of 20 ms was required, but the intent handler would only have solutions to reach down to 50 ms, it would state that the intent is degraded, because the requirement cannot be fulfilled, and it would propose 50 ms as the edge compliance level possible.

The intent handler can in general reject new intent or intent modification, if it does not expect that it can reach compliance to the stated requirements. If the rejected intent contains a request for best proposals, the report that is communicating the rejection would then include a proposal for a requirement that can be met.

Intent owner and handler interaction

In any case a proposal for best intent need to be requested by the intent owner. The intent handler will only make proposals according to this request. This means that the intent owner is leading the conversation. This is important, to avoid the intent handler makes assumptions of what the intent owner needs or does not need. The intent handler would and should not have knowledge about why an intent owner is setting a requirement.

Intent owner guided proposals are also important for another reason. An intent typically states multiple requirements and solutions chosen by the intent handler find a balance of resource usage to comply to all of them. But if an intent handler is supposed to make proposals for multiple requirements in combination, it would need to prioritize. For example, if the intent contains throughput and latency requirements and the intent handler is supposed make a combined proposal, it would typically need to decide if it is proposing according to a solution that favors latency or a solution that favors throughput. However, an intent handler cannot make this decision, because it would require deeper understanding of the reasons for the intent. This is knowledge based on a business logic implementation the intent owner would have, but the intent handler does not.

The intent owner guided proposal process is a direct consequence of the basic separation of concerns between intent managers in the role of intent owner and intent handler. The solution for intent fulfillment, which are available and chosen by the intent handler, are transparent to the intent owner. On the other hand, the business reasons and motivation for the intent owner to set a requirement are not a concern of the intent handler. Consequently, the intent owner would need to make focused best proposal requests to get sensible proposals from the intent handler. A focused request means that the intent handler is asked to make proposal for a single requirement or a very limited number of requirements. The request is interpreted as asking for a proposal for a requirement under the assumption that all other requirements are also fulfilled. This means the intent owner would know exactly what to maximize without the need to assume and guess what the owner needs.

If the intent owner wants to explore multiple requirements, it can do so with a sequence of multiple different requests. This means the intent owner and intent handler would have a conversation exploring multiple options. With this approach, at any point in time the intent handler would have a clearly scoped and focused task of what to generate proposals for. The intent owner leading the conversation process can gather information about the edge requirement compliance capabilities of the intent handler and use this in its decisions.

The request of proposals is expressed within the intent model. This means a change of proposal request is an intent modification over the intent management API. The best proposals are communicated from the intent handler to the intent owner through intent reports.

Support for the best intent proposal model is optional. Intent managers would announce their support through the intent manager capability profile by stating support for this intent extension model.

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 Proposal of Best Intent 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 proposal of best intent 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).
Proposal of best intent	pbi	http://tio.models.tmforum.org/tio/v3.6.0/IntentValidityOntology/	TM Forum	(This model) Defines ontology model for expression proposal of best intent requests and providing a proposal.
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]

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 [rdfs1]
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 Proposal of Best Intent 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 proposal of best intent 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. The Time Ontology in OWL is the base for expressing temporal conditions.

2. Requesting proposals

A best proposal is requested using the best proposal expectation. The best proposal expectation is expressed by an instance of class pbi:BestProposalExpectation, which is a subclass of class icm:ReportingExpectation. As such it has inherited typical elements of a reporting expectation, such as the target for scoping the report or the use of events for initiating report generation.

In the context of best requests, the target of the best proposal expectation sets the scope of the request. This means the target container refers to the elements within an intent for which a best proposal is requested. For example, it can refer to a condition object within the intent to indicate that a best proposal is needed for this object. Events determine when the best proposal shall be made.

2.1. Integration into intent

Expectations are associated with an intent through logical operators as described in the intent common model. The best proposal expectation is associated with an intent exactly the same way. It therefore represents additional requirements in the same way as other reporting expectations are requirements to generate and distribute intent reports with specified content and under defined conditions including timing. In case of the best proposal expectation the added requirement is to generate and distribute intent reports containing proposals if and when defined conditions are met. Furthermore, the best proposal expectation defines scope and expected content of the proposal.

For example:

```
ex:I1
  a icm:Intent ;
  log:allOf ( ex:P1 ex:R1 ex:BP1 )
```

```
ex:P1
  a icm:PropertyExpectation ;
  <...>
```

```
ex:R1
  a icm:ReportingExpectation ;
  <...>
```

```
ex:BP1
  a pbi:BestProposalExpectation ;
  <...>
```

This example shows a structure of an intent expression using multiple expectations including a best proposal expectation. The Best Proposal Expectation is associated with an intent in the same way as other expectations.

2.2. Proposal Scope

A best proposal is typically requested for certain requirements within the intent. This scope of proposal would be determined by the target of the Best Proposal Expectation. The target would contain references to the elements of the intent a proposal is required for.

For example:

```
ex:I1
  a icm:Intent ;
  log:allOf ( ex:P1 ex:R1 ex:BP1 )

.
.

ex:TP1
  a icm:Target ;
  rdfs:member ex:Slice1 ;

ex:P1
  a icm:PropertyExpectation ;
  icm:target ex:TP1 ;
  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:greater ( ex:Availability1
    [ rdf:value "0.999"^^xsd:decimal ]
  )

.
.

ex:TBP1
  a icm:Target ;
  rdfs:member ex:C1

ex:BP1
  a pbi:BestProposalExpectation ;
  icm:target ex:TBP1 ;
  <...>
```

This example shows how to set the scope of best proposals. A proposal is requested for the condition ex:C1. This is a condition about the latency, thus a proposal regarding latency is wanted. The target ex:TBP1 is used as the target of a Best Proposal

Expectation. This target has the condition ex:C1 as single member. This means, this best proposal expectation is requesting a best proposal for condition ex:C1. There is another condition ex:C2 in this intent, and it is also contributing to the compliance evaluation. No proposal for ex:C2 is requested, but this condition is still considered. The best proposal for ex:C1 should reflect the best condition and value for the latency KPI possible to comply to under the assumption that all other requirements are also met. In this example this means that the proposal for ex:C1 would provide a new condition with a new value for the latency under the condition that the availability use in ex:C2 is also met.

2.3. Proposal scope with multiple entries

It is possible to define scopes of proposals for multiple distinct requirements at once. This means a combined proposal is wanted for multiple requirements. As this potentially requires the intent handler to assume a certain prioritization between the requirements, this does not always provide the desired result. In many use case it is therefore recommended that the intent owner uses a single requirement proposals in parallel or in sequence, rather than a single multi requirement proposal.

For example:

```

ex:I1
a icm:Intent ;
log:allOf ( ex:P1 ex:BP1 )

.

ex:TP1
a icm:Target ;
rdfs:member ex:Slice1 ;

ex:P1
a icm:PropertyExpectation ;
icm:target ex:TP1 ;
log:allOf ( ex:C1 ex:C2 ex:C3 )

.

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

.

ex:C2
a log:Condition ;
quan:greater ( ex:Availability1
    [ rdf:value "0.999"^^xsd:decimal ]
)

.

ex:C3

```

```

a log:Condition ;
quan:greater ( ex:Throughput1
    [ rdf:value "1"^^xsd:decimal ;
      unit "Mbps" ]
)

ex:TBP1
a icm:Target ;
rdfs:member ex:C1 , ex:C1

ex:BP1
a pbi:BestProposalExpectation ;
icm:target ex:TBP1 ;
<...>
.
.
```

In this example the target of the proposal expectation ex:BP1 points at the conditions ex:C1 and ex:C2 but does not include ex:C3. This means a proposal shall be made for the best combined latency and availability the system considers to be achievable at the time of generating the proposal. Remaining requirements such as ex:C3 for throughput in the example need to be considered as well. A proposal shall be made for possible values for latency and availability under the condition that other requirements of the intent can be met as well. This means, the intent handler assumes that an intent that would require the proposed values for latency and availability in combination with the existing value for throughput would be an intent it can comply to. In other words, values for latency and availability, which would not allow meeting the required throughput requirements would not be a sensible proposal.

2.4. Multiple simultaneous proposals

Within a single intent multiple proposals can be requested. This is expressed by using multiple best proposal expectations. Each of them is interpreted and acted on individually and independent of each other by the intent handler. They represent additional requirements for proposals and can have different scopes and generation conditions.

For example:

```

ex:I1
a icm:Intent ;
log:allOf ( ex:P1 ex:BP1 ex:BP2 )

ex:TP1
a icm:Target ;
rdfs:member ex:Slice1 ;

ex:P1
a icm:PropertyExpectation ;
icm:target ex:TP1 ;
.
```

```
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:greater ( ex:Availability1
    [ rdf:value "0.999"^^xsd:decimal ]
)

ex:TBP1
a icm:Target ;
rdfs:member ex:C1

ex:BP1
a pbi:BestProposalExpectation ;
icm:target ex:TBP1 ;
<...>

ex:TBP2
a icm:Target ;
rdfs:member ex:C2

ex:BP2
a pbi:BestProposalExpectation ;
icm:target ex:TBP2 ;
<...>
```

In this example two proposals are requested using the best proposal expectations `ex:BP1` and `ex:BP2`. The proposal request `ex:BP1` asks for proposals about the condition `ex:C1`, which contains a latency requirement. The proposal therefore would state a latency value that can be required as the lowest latency currently possible to achieve given that other requirements are still achievable as well. Another requirement is in this case the availability requirement stated by the condition `ex:C2`. Similarly, the best proposal expectation `ex:BP2` asks for a proposal on availability. This proposal shall state the availability value that can be achieved given that other requirements, such as the latency requirement of the condition `ex:C1` would also be met.

Note that the proposed value is not necessarily representing a better outcome than the currently required one. In the previous example, the current requirement for latency is 10ms. A proposal for latency can state a better value such as 8ms if this is possible to achieve. But in other cases even the required 10ms might not be feasible, and the proposal would state for example 15ms is that is the best level of latency that can currently be achieved.

2.5. Proposals for complex requirements

Requirements are often complex combinations of several influencing factors. Expectations and entire intent are typically non-atomic requirements as they can combine multiple conditions to be met. Also conditions can have sub-conditions. Best proposal expectations can request proposals for complex, non-atomic requirements. If the scope of a best proposal expectation refers to atomic requirement elements such as expectations, then potentially all contributors to that expectation are in scope as well and a combined proposal for the entire expectation with all its contributors is needed. In this respect it is also possible to propose completely new atomic requirement contributions rather than only new values for the existing ones.

Furthermore, an intent handler might not only propose new values to existing requirements. An advanced intent handler can also propose new structures of expectation and conditions including new context. However, this level of proposal would require the intent handler to implement extensive understanding of the intent owner's domain and use cases. Systems designed with strong separation of concerns would usually avoid the resulting duplication spread out of implementations. This means, while the intent expression allows requesting complex proposals and provide them, it is usually more preferential to keep the intent owner driving the best proposal making. It should ask for focused proposals with single or a few requirements in scope at a time. The intent owner can then variate proposal requests and draw its conclusions about the current capabilities of the intent handler from multiple proposal reports.

2.6. Proposal report initiation

The intent handler monitors intent handling events stated in reporting expectations including the best proposal expectation. The reporting initiation for intent reports is described in the intent common model TR290A and also applicable for best proposals. If and when an event triggers the reporting conditions of a best proposal expectation, a new proposal is generated and added to an intent report. If only the best proposal expectation is triggered, an intent report is generated only for providing the proposal. If the event triggers multiple reporting expectations and best proposal expectations, a single report is generated containing basic intent reporting in combination with one or several proposals.

3. Providing a best proposal

The best proposal is provided through an intent report. This chapter specifies how the proposal would be expressed within an intent report.

3.1. Expressing a best proposal

Best proposals are provided using intent reports of class `icm:IntentReport` defined in the intent common model TR290. The intent report can be generated to only contain the proposal, or the proposal can be added to intent reports that also report other concerns. The combination of multiple reporting concerns in a single report can be chosen if the intent handler needs to report multiple concerns regarding an intent to the same receiver at the same time.

The property `pbi:proposal` refers to best proposal reports contained within an intent report. The best proposal report is an individual of class **`pbi:BestProposalReport`** and subclass of `icm:ExpectationReport`.

The target of the best proposal report refers to the requirement objects in the intent a best proposal is provided for. This is typically the same target specification used in the best proposal expectation for setting the proposal scope within the intent.

The property `icm:reportsAbout` refers to the best proposal expectation in the intent for which the respective best proposal report is stating proposals.

A distinct best proposal is an individual of class `pbi:Proposal`. The property `pbi:proposed` is used in the domain of a best proposal report to refer to a distinct proposal. Typically, there is a distinct proposal made for every member of the target.

A proposal states the object from the intent for which a proposal is made using the property `pbi:proposedFor`. The property `pbi:proposedBest` refers to an object that states the proposed values. This means, that `pbi:proposedFor` refers to the original requirement, while `pbi:proposedBest` refers to the proposed new requirement.

For example:

```
ex:IR1
a icm:IntentReport ;
  icm:reportsAbout ex:I1 ;
  pbi:proposal ex:BPR1
```

```
ex:C5
a log:Condition ;
  quan:smaller ( ex:Latency1
    [ rdf:value "8"^^xsd:decimal ;
      unit "ms" ]
  )
```

```
ex:P1
a pbi:Proposal ;
  pbi:proposedFor ex:C1 ;
  pbi:proposedBest ex:C5
```

```
.  
ex:BPR1  
  a pbi:BestProposalReport ;  
  icm:target ex:TP1 ;  
  icm:reportsAbout ex:BP1 ;  
  pbi:proposed ex:P1  
. 
```

In this example an intent report contains only a proposal. This means that only the reporting conditions of the best proposal expectation did apply. No other reasons and topics for reporting, such as regular intent reporting, did apply.

The example does not define a new target, but refers to the same target ex:TP1 that was defined in the intent for the respective best proposal expectation.

In this example the best proposal report ex:BPR1 is stating the proposals made according to the best proposal expectation ex:BP1. This relationship is stated using the property `icm:reportsAbout`.

In this example we assume that only a best proposal for the condition ex:C1 was requested. The respective proposal is made by the proposal object ex:P1. It states that the proposal is made for the condition ex:C1 in the intent. It proposes a different requirement expressed by a new condition ex:C5. More specifically it proposes that a latency of 8ms can be required, while the original intent did require 10ms. Please note that the requirement is still 10ms. The proposal made to the intent owner only states a possibility to successfully comply to a more severe latency requirement of 8ms at the time of proposal generation. The intent owner can use this information to potentially change the intent. This means the proposal for best intent states feasible requirement values.

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 (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*
<u>Kamal Maghsoudlou</u>	Ericsson	Key Contributor
<u>Leonid Mokrushin</u>	Ericsson	Key Contributor
<u>Marin Orlić</u>	Ercisson	Key Contributor
<u>Aaron Boasman-Patel</u>	TM Forum	Additional Input
<u>Alan Pope</u>	TM Forum	Additional Input
<u>Dave Milham</u>	TM Forum	Additional Input
<u>Xiao Hongmei</u>	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. BestProposalExpectation

The class pbi:BestProposalExpectation is used to request proposals of best intent. It is a subclass of icm:ReportingExpectation, which also implies that the proposal is communicated through an intent report.

Instance of: rdfs:Class Subclass of: icm:ReportingExpectation

5.2. BestProposalReport

The class pbi:BestProposalReport is used to provide the proposals corresponding to a best proposal expectation in the intent. It is a subclass of icm:ExpectationReport.

Instance of: rdfs:Class Subclass of: icm:ExpectationReport

5.3. Proposal

The class pbi:Proposal is used to provide an individual proposal. Typically, there is a proposal created for every requirement in scope of a best proposal expectation as stated by its target.

Instance of: rdfs:Class

5.4. proposed

The property pbi:proposed is used in the domain of a best proposal report to refer to an individual proposal.

Instance of: rdf:Property

Domain: pbi:BestProposalReport

Range: pbi:Proposal

5.5. proposedFor

The property pbi:proposedFor is used to state what requirement object from the intent the proposal is made for.

Instance of: rdf:Property

Domain: pbi:Proposal

5.6. proposedBest

The property pbi:proposedBest refers to an object that states the proposed values.

Instance of: rdf:Property

Domain: pbi:Proposal

5.7. proposal

The property pbi:Proposal refers to the best intent proposals contained in the intent report.

Instance of: rdf:Property

Domain: icm:IntentReport

Range: pbi:BestProposalReport

5.8. Vocabulary

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

Instance of: rdfs:Container