

TM Forum Model

Intent Acceptance Control - Intent Extension Model

TR291D

Maturity Level: Beta	Team Approved Date: 01-Jun-2022
Release Status: Pre-production	Approval Status: Member Evaluated
Version 1.1.0	IPR Mode: RAND

Notice

Copyright © TM Forum 2022. 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.

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	3
List of Figures	4
List of Tables	5
Executive Summary.....	6
Introduction.....	7
1.Motivation and background.....	8
2.Notation and namespaces	9
3.Principles and vocabulary overview	10
4.Vocabulary specification	12
4.1. Classes.....	12
4.2. Instances	12
4.3. Properties	13
5.Model usage and examples	15
5.1. Specifying time budget for acceptance decision and rejection override 15	
6.Administrative Appendix	16
6.1. Document History	16
6.1.1. Version History.....	16
6.1.2. Release History.....	16
6.2. Acknowledgments.....	16
6.2.1. Guide Lead & Author.....	16
6.2.2. Main Contributors.....	16
6.2.3. Additional Inputs	17

List of Figures

Figure 0.1: Intent model dependencies overview	7
Figure 3.1: Vocabulary of acceptance and rejection control	10

List of Tables

Table 2-1: Model references	9
-----------------------------------	---

Executive Summary

The intent acceptance control model is an intent extension model that introduces vocabulary needed to steer and override intent and intent update rejections.

Introduction

This document describes a model in the suit of models for intent based operation.

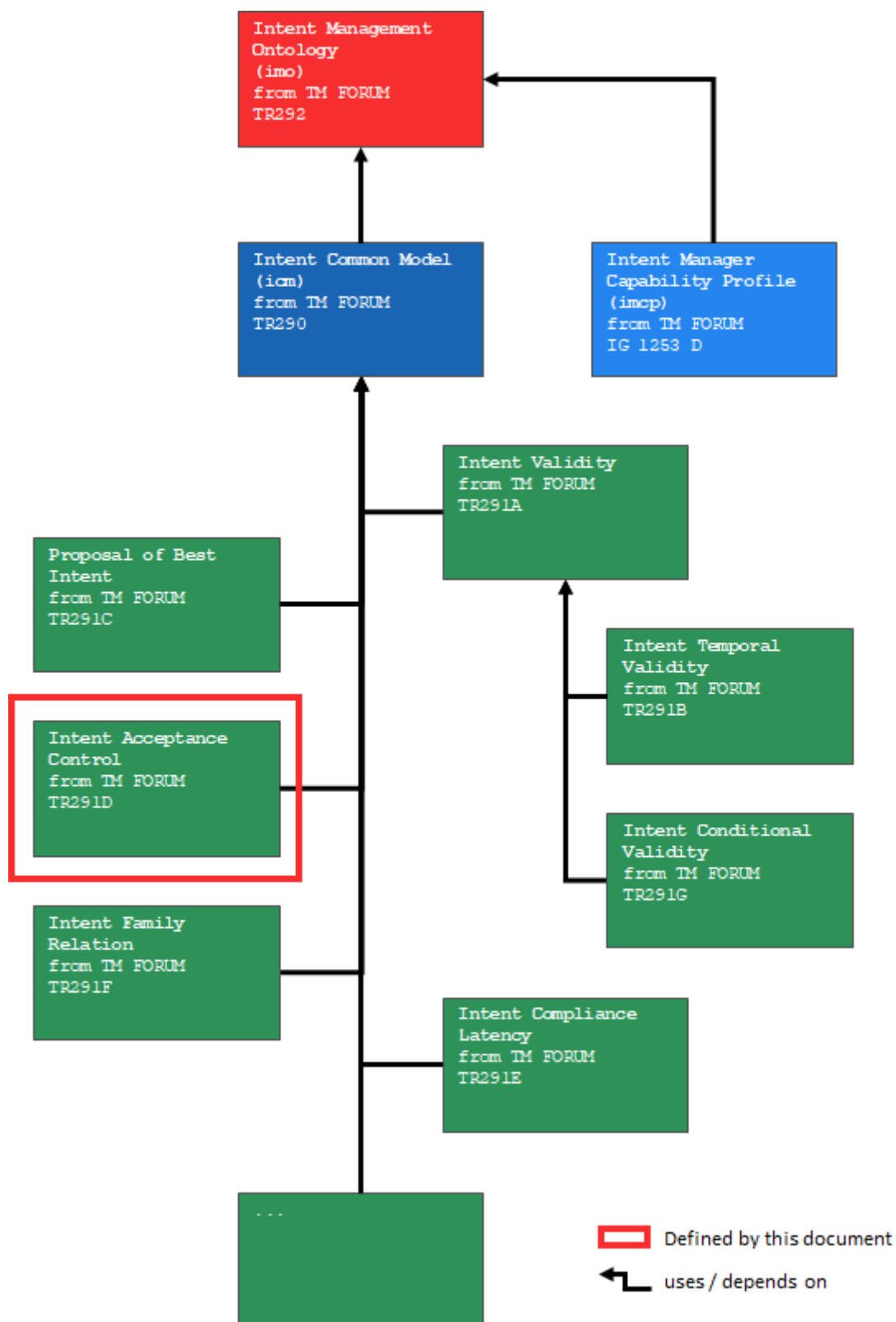


Figure 0.1: Intent model dependencies overview

1. Motivation and background

When receiving an intent for the first time or with new content as update, the intent handler needs to decide if it accepts or rejects the intent or its update. Rejection might have formal reasons. The intent handler might not understand some content, because it does not support a model the intent owner has used in the intent expression. This indicates a problem in the owner, as it has not interpreted the intent manager capability profile of the handler correctly.

Another reason for rejection might be that the intent handler does not believe it can successfully meet all expectations given the resources it has available. This is a soft rejection because the intent handler could in principle start trying to meet the expectations, although it would most likely not succeed. In this case the intent owner might ask the handler to accept it anyway. This implies that the owner accepts the consequence of not getting a system state that is fully compliant. This might make sense to do if the intent is about delivering an important service. It might be more important that the service is somehow working with limitations than not having it at all.

The acceptance and rejection control model allows the intent owner to specify a rejection override for certain rejection reasons. The handshake between the owner and handler might start with a first intent being sent to the handler and results in a rejection. If needed the intent owner would then send a new intent with the same requirements, but this time with a rejection override for the rejection reason from the first try.

Another concern of intent acceptance is the time the handler has for this initial decision. The intent common model does not define a maximum time for the first reply of the intent handler. The intent owner can however send a removal request for the intent if the decision has taken too long. The acceptance and rejection control model provides another possibility. The intent owner can specify a time budget for the acceptance decision. After this time has expired without an acceptance decision by the handler, the intent is automatically rejected. This time budget can be specified individually for the initial acceptance of the intent or for intent updates.

2. Notation and namespaces

The acceptance and rejection control model is defined in a namespace under the TM Forum domain. This intent extension model depends on the following models and uses the respective namespaces.

Table 2-1: Model references

Model	Prefix	Namespace	Published by
Intent Acceptance and Rejection Control	arc	http://tio.models.tmforum.org/tio/v1.0.0/IntentAcceptanControl/	TM Forum
W3C RDF version 1.1	rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	W3C
W3C RDF Schema 1.1	rdfs	http://www.w3.org/2000/01/rdf-schema#	W3C
Intent Common Model	icm	http://tio.models.tmforum.org/tio/v2.0.0/IntentCommonModel/	TM Forum
Intent Management Ontology	imo	http://tio.models.tmforum.org/tio/v1.0.0/IntentManagmentOntology/	TM Forum
W3C time Ontology in OWL	t	http://www.w3.org/2006/time	W3C
XML Schema	xsd	http://www.w3.org/2001/XMLSchema#	W3C

The proposed prefix label for the intent acceptance and rejection control model is "arc".

The model has a dependency to RDF and RDFS, because they are the chosen base standards for all intent and intent report models. It uses datatype defined in XML Schema. Furthermore, it gains expressiveness for time from the OWL time Ontology.

The acceptance and rejection control model specializes and extents the definitions of the intent common model.

The Example namespace is used for separating the namespace for example objects within this document.

3. Principles and vocabulary overview

By attaching the properties `arc:intentAcceptanceLatency` and `arc:updateAcceptanceLatency` to an intent, the owner sets a time budget for the intent handler to make an acceptance or rejection decision.

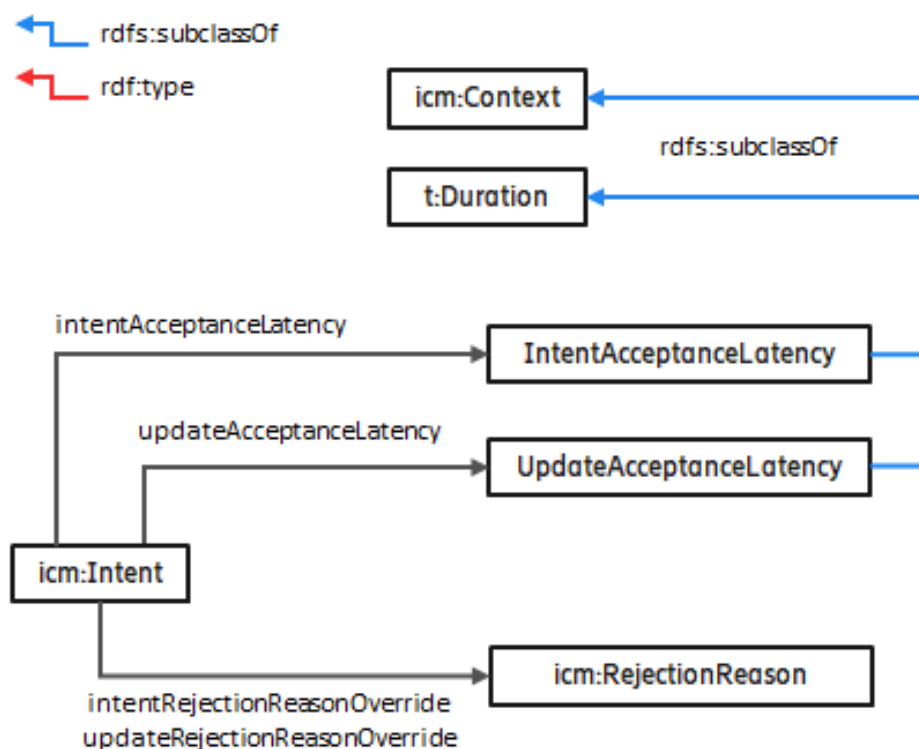


Figure 3.1: Vocabulary of acceptance and rejection control

The acceptance time is a `t:Duration` object according to the owl time ontology model. The timer related to this duration starts at reception of the intent or intent update. If the timer exceeds the time budget, the intent owner is supposed to interrupt its considerations and immediately reject the intent or intent update.

If an intent or an intent update gets rejected because the decision time expired, this causes an `icm:intentRejected` or `icm:updateRejected` event. No additional events are defined specific to this rejection cause. However, the rejection reason is then reported to be `arc:IntentAcceptanceLatencyExpired` or `arc:UpdateAcceptanceLatencyExpired`. The acceptance and rejection control model defines these additional individuals of class `icm:RejectionReason` to cover the forced rejection at expiry of the given time budget.

Using the properties `arc:intentRejectionReasonOverride` and `arc:updateRejectionReasonOverride` the intent owner can define rejection reasons for which the handler should accept anyway. The owner implicitly accepts potential state degradation.

The acceptance and rejection control model defines a set of additional individuals of class `icm:RejectionReason` to communicate that the reason for rejection was that a rejection override was not possible. This is usually an additional rejection reason next to the reason why the handler originally wanted to reject. A rejection override might not be possible if, for example, the intent handler does not support some models used in the intent. This means override would work for rejections because of a negative projection of handling success. But an override is not possible for reasons where the handler has no sensible way forward.

In intent reports the acceptance or rejection would appear as defined in the intent common model through information about events, states and rejection reasons. The acceptance and rejection control model contributes additional rejection reasons to intent reporting.

4. Vocabulary specification

4.1. Classes

Class:	arc:IntentAcceptanceLatency
Definition:	Instances of this class define how long it is allowed to take from intent reception until the intent handler has decided on acceptance of the intent. If this time expires without the intent handler having concluded to accept, the intent will get rejected with rejection reason arc:IntentAcceptanceLatencyExpired.
Instance of:	rdfs:Class
Subclass of:	icm:Context t:Duration

Class:	arc:UpdateAcceptanceLatency
Definition:	Instances of this class define how long it is allowed to take from intent update reception until the intent handler has decided on acceptance of the update. If this time expires without the intent handler having concluded to accept, the update will get rejected with rejection reason arc:UpdateAcceptanceLatencyExpired.
Instance of:	rdfs:Class
Subclass of:	icm:Context t:Duration

4.2. Instances

imo:RejectionReason individuals	Description
arc:IntentAcceptanceLatencyExpired	The defined time budget for the decision to accept an intent has expired.
arc:UpdateAcceptanceLatencyExpired	The defined time budget for the decision to accept an intent has expired.
arc:IntentRejectionOverrideNotPossible	The intent could not be accepted although the intent owner has asked to accept anyway. This might happen, for example, if the intent handler does not fully understand the intent content.
arc:UpdateRejectionOverrideNotPossible	The intent update could not be accepted although the intent owner has asked to accept

imo:RejectionReason individuals	Description
	anyway. This might happen, for example, if the intent handler does not fully understand the intent content.

4.3. Properties

Property:	arc:intentAcceptanceLatency
Definition:	This property assigns a time budget for making an intent acceptance decision after an intent is received.
Instance of:	rdf:Property
Domain:	icm:Intent
Range:	arc:IntentAcceptanceLatency

Property:	arc:intentRejectionReasonOverride
Definition:	<p>This property overrides a rejection decision an intent handler might make.</p> <p>The intent owner states the potential rejection reason for which the handler should accept the intent anyway.</p> <p>By asking for a rejection override an intent owner confirms that it would accept potential intent degradation for now.</p>
Instance of:	rdf:Property
Domain:	icm:Intent
Range:	icm:RejectionReason

Property:	arc:updateAcceptanceLatency
Definition:	<p>This property assigns a time budget for making an acceptance decision after an intent update is received.</p> <p>This is typically defined in the intent graph that constitutes the update. This information in the update has priority over the information that was potentially present in the previous and to be updated version of the intent.</p>
Instance of:	rdf:Property
Domain:	icm:Intent
Range:	arc:UpdateAcceptanceLatency

Property:	arc:updateRejectionReasonOverride
Definition:	<p>This property overrides a rejection decision an intent handler might make.</p> <p>The intent owner states the potential rejection reason for which the handler should accept the intent update anyway.</p> <p>By asking for a rejection override an intent owner confirms that it would accept potential intent degradation for now.</p>

Property:	arc:updateRejectionReasonOverride
Instance of:	rdf:Property
Domain:	icm:Intent
Range:	icm:RejectionReason

5. Model usage and examples

5.1. Specifying time budget for acceptance decision and rejection override

The following example intent shows how to express an intent acceptance latency.

Example 1: Time budget for acceptance and rejection override

```
@prefix rdfs: http://www.w3.org/2000/01/rdf-schema#
@prefix icm: https://tmforum.org/2020/07/IntentCommonModel/
@prefix arc: https://tmforum.org/2021/07/intentAcceptanceRejectionControl/
@prefix t: http://www.w3.org/2006/time
@prefix xsd: http://www.w3.org/2001/XMLSchema#
@prefix met: http://www.sdo2.org/TelecomMetrics/Version_1.0/
@prefix : http://example.org/IntentExample

:ExampleIntentXYZ
  a icm:Intent ;
  arc:intentAcceptanceLatency [ t:numericDuration 2 ;
                                t:temporalUnit t:unitSecond ;
                              ];
  arc:intentRejectionReasonOverride imo:SuccessfulHandlingNotExpected ;
  icm:hasExpectation :E1, :E2 ;
.

...
```

In this example the intent owner specifies that the handler has two seconds to decide if it accepts or rejects the intent.

Furthermore, the intent owner states that an intent shall still be accepted, although the handler wants to reject, because it does not expect it can successfully meet the intent expectations.

6. Administrative Appendix

6.1. Document History

6.1.1. 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

6.1.2. 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	Final edits prior to publication
Pre-production	04-Jul-2022	Adrienne Walcott	Updated to reflect TM Forum Member Evaluated status

6.2. Acknowledgments

6.2.1. Guide Lead & Author

Member	Title	Company
Jörg Niemöller	Expert of Analytics and Customer Experience	Ericsson

6.2.2. Main Contributors

Member	Title	Company
Jörg Niemöller	Expert of Analytics and Customer Experience	Ericsson
Kevin McDonnell	Senior Director, Intelligent Automation	Huawei
James O'Sullivan	Product Director, Intelligent Automation	Huawei
Dave Milham	Chief Architect	TM Forum
Vinay Devadatta	Practice Head (Innovation & Industry Relations)	Wipro Technologies
Azahar Machwe	OSS Automation	BT Group plc
Wang Lei	Systems Expert	Huawei
Tayeb Ben Meriem	Senior Standardization Manager (OSS)	Orange
Leonid Mokrushin	Principle Researcher	Ericsson

6.2.3. Additional Inputs

Member	Title	Company
Lester Thomas	Chief IT Systems Architect	Vodafone Group
Ankur Goyal	Lead Consultant	Infosys
Emmanuel A. Otchere	Chief Technical ExpertVP, Standards & Industry Development	Huawei
Min He	Chief Architect	Futurewei