TM Forum Model

Intent Compliance Latency - Intent Extension Model

TR291E

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

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

List of Figures

List of Tables

Executive Summary

The intent compliance latency model is an intent extension model that introduces vocabulary needed to steer the ambition and timing applied by an intent manager to reach compliance and counteract issues.

Introduction

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

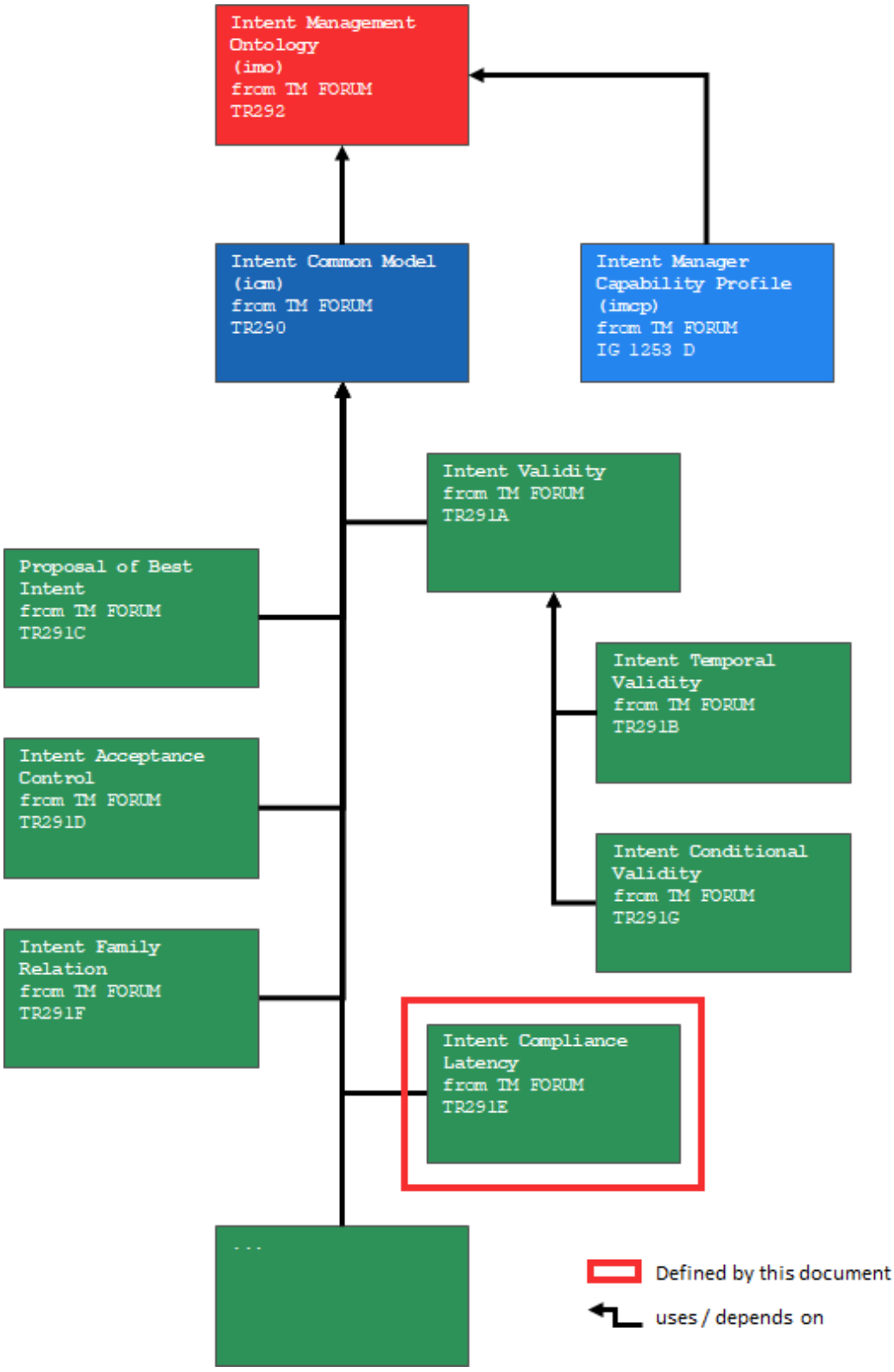

Figure 0.1: Intent model dependencies overview

1. Motivation and background

The system state might become degraded with respect to the requirements specified within the intent. If this is happening the intent handler is expected to take action for bringing the system back into a compliant state. It might take time for the action to show an effect. How much time this typically takes depends on the actions taken. For the intent owner it might not be acceptable that the system recovers eventually after a potentially long period of time. It might need fast recovery from degradation. The intent compliance latency model allows an intent owner to specify how much time the intent handler is given to reach "compliant" state. This means the intent owner can specify how long it would be willing to tolerate an intent degradation. The intent owner can express this by setting a recovery time budget. The handler can consider this for example by taking more aggressive action.

Next to defining specific time budgets per intent it might be useful to define general default compliance targets for an intent manager. This means an intent can be used with the intent manager as expectation target. This means, the intent manager is expected to comply to these requirements with all action it takes. Defining compliance latency this way sets the default compliance latency target.

This model distinguishes

- Setting a goal for initial compliance latency after the intent and its constituent expectations were initially received.
- Setting a goal for reaching compliance again after an intent or an expectation was updated.
- Setting a goal for recovery to compliance latency after an intent or an expectation got degraded.

The goals are temporal goals expressed as time duration.

2. Notation and namespaces

The intent compliance latency 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 Compliance Latency	icl	http://tio.models.tmforum.org/tio/v1.0.0/IntentComplianceLatency/	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
W3C time Ontology in OWL	t	http://www.w3.org/2006/time	W3C
XML Schema	xsd	http://www.w3.org/2001/XMLSchema#	W3C
Example namespace	ex	http://example.com/IntentModeling/	n/a

The proposed prefix label for the intent compliance latency model is "icl".

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 data types defined in XML Schema. Furthermore, it gains expressiveness for time from the OWL time Ontology.

The intent compliance latency model specializes and extents the definitions of the intent common model (TR290).

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

2.1. Principles and vocabulary overview

The intent compliance latency model defines the class icl:ComplianceLatency as a subclass of icm:Context. It is also a subclass t:Duration from the OWL time ontology. This means it represents a time duration.

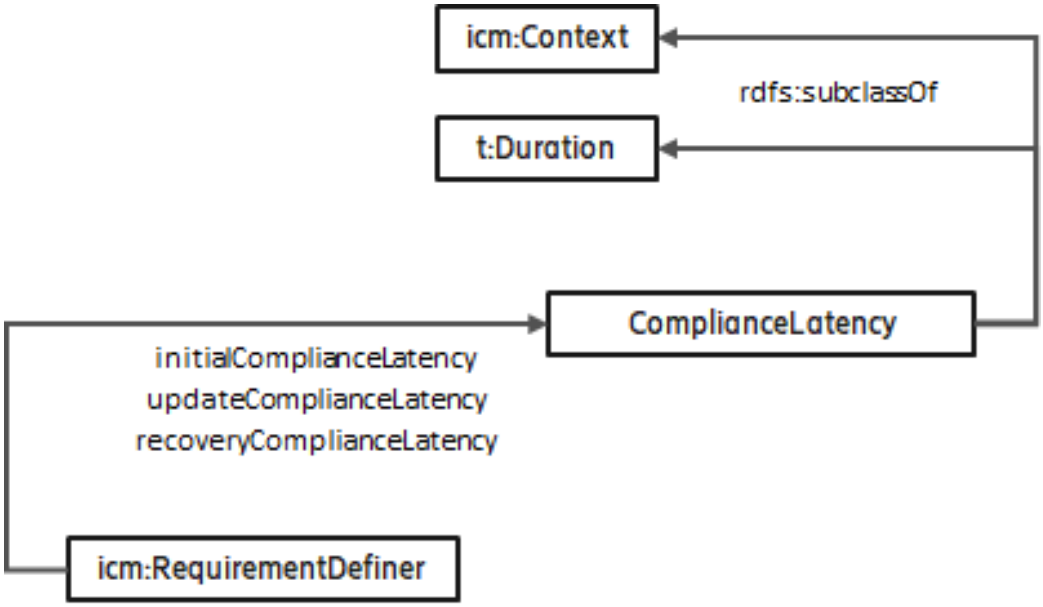

Figure 2.1: Vocabulary Overview

A compliance latency is a target time budget the intent handler has available to reach a compliant system state with its actions. This way the intent owner communicates and quantifies urgency of complying to the intent.

A compliance latency is assigned using the properties `icl:initialComplianceLatency`, `icl:updateComplianceLatency` and `icl:recoveryComplianceLatency`. This subject is a requirement definer according to the intent common model. These are intent, expectation, expectation params and context class subjects. For them compliance latency can be defined. If the subject is an intent, the respective compliance latency time budget means that the system is expected to become compliant to the entire intent within this time. If the subject is an expectation, then compliance within the time budget is only expected for the requirements of this particular expectation.

The property `icl:recoveryComplianceLatency` has `icl:ComplianceLatency` as range. It defines a time duration until the system is supposed to recover from degradation. .

The properties icl:initialComplianceLatency and icl:updateComplianceLatency define respective target latencies for reaching compliance after the intent was initially received or after it was updated. This means the model distinguishes three cases of degradation: after initial reception, after update reception or after degradation was reached for other or unknown reasons. It allows setting individual time targets for reaching complicate again.

If multiple compliance latencies are defined, their fulfillment is considered separately and this can lead to multiple events. For example, the intent owner can specify a compliance latency for recovery of 2 minutes and another one also for recovery of 5 minutes. With a reporting expectation that asks for a report at the icl:RecoveryComplianceLatencyExpired event, the first event would be generated if the system is not compliant 2 minutes after it was degraded. If it is still continuously degraded after 5 minutes, another expiry event would lead to yet another report.

The compliance latency always counts from the time degradation was discovered. This includes the degradation that was initially determined after intent reception or after an update. If the system fails to recover after the specified time, an

icm:IntentHandlingEvent is generated. Furthermore, the intent handler can reject an intent or intent update if it considers the compliance latency times specified are too short to be sensibly reached and it would state this as rejection reason. The intent compliance latency model defines additional individuals of class icm:IntentHandlingEvent and of class icm:Rejection reason to allow expressing this.

2.2. Vocabulary specification

2.2.1. Classes

Class:	`icl:ComplianceLatency`
Definition:	instances of this class define how long it is allowed to take until compliance with an intent or intent object is reached or reached again.
Instance of:	`rdfs:Class`
Subclass of:	`icm:Context` `t:Duration`

2.2.2. Instances

The following table defines additional individuals of class `imo:IntentHandlingEvent`:

`imo:IntentHandlingEvent` **individuals**	**Description**
`icl:InitialComplianceLatencyExpired`	The time specified for initial compliance has expired without the system becoming compliant.
`icl:UpdateComplianceLatencyExpired`	The time specified for reaching compliance after an update has expired without the system becoming compliant.
`icl:RecoveryComplianceLatencyExpired`	The time specified for reaching compliance after a degradation has expired without the system becoming compliant.

The following table defines additional reasons for intent or intent update rejection as individuals of class icm:RejectionReason

`imo:RejectionReason` **individuals**	**Description**
`icl:InitialComplianceLatencyShort`	The time specified about initial compliance target is too short. The intent handler rejects, because it does not expect that it can transition the system into a compliant state within the specified time
`icl:UpdateComplianceLatencyShort`	The time specified about update compliance target is too short. The intent handler rejects, because it does not expect that it can transition the

imo:RejectionReason individuals	Description
	system into a compliant state after an update within the specified time
icl:RecoveryLatencyShort	The time specified about recovery compliance target is too short. The intent handler rejects, because it does not expect that it can transition the system into a compliant state after it got degraded, within the specified time

2.2.3. Properties

Property:	icl:initialComplianceLatency
Definition:	Defines the time it is allowed to take until the intent becomes first compliant after it got received by the handler.
Instance of:	rdf:Property
Domain:	icm:RequirementDefiner
Range:	icl:ComplianceLatency

Property:	icl:updateComplianceLatency
Definition:	Defines the time it is allowed to take until the intent becomes compliant after it an update was received.
Instance of:	rdf:Property
Domain:	icm:RequirementDefiner
Range:	icl:ComplianceLatency

Property:	icl:recoveryComplianceLatency
Definition:	Defines the time it is allowed to take until the intent becomes first compliant after it got received by the handler.
Instance of:	rdf:Property
Domain:	icm:RequirementDefiner
Range:	icl:ComplianceLatency

2.3. Model usage and examples

2.3.1. Specifying time budget for recovery from degradation

The following example intent shows how to define a target time budget for recovery from degradation

Example 1: Time for recovery

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix icl:
<http://tio.models.tmforum.org/tio/v1.0.0/IntentComplianceLatency/> .
@prefix icm:
<http://tio.models.tmforum.org/tio/v2.0.0/IntentCommonModel/> .
@prefix imo:
<http://tio.models.tmforum.org/tio/v1.0.0/IntentManagementOntology/> .
@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 cat:  <http://www.operator.com/Catalog/> .
@prefix :
<http://www.operator.org/IntentNamespace/intent20220322_12345/> .

# --- Intent --------------------
:ExampleIntentXYZ
  a icm:Intent ;
  icl:RecoveryComplianceLatency [ t:numericDuration "2"^^xsd:integer ;
                                  t:temporalUnit t:unitMinute ;
                                ] ;
  icm:hasExpectation :E1, :E2, :E_reporting ;
.
# --- Targets --------------------
:T1_service a icm:Target .

# --- Expectations --------------------
:E1
  a icm:DeliveryExpectation ;
  icm:target :T1_service ;
  icm:hasParameter :P1 ;
.
:E2
  a icm:PropertyExpectation ;
  icm:target :T1_service ;
  icm:atLeast P2 ;
.
:E_reporting
  a icm:ReportingExpectation ;
  icm:target :ExampleIntentXYZ ;
  icm:anyOf ( P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 ) ;
.
# --- Parameters --------------------
:P1
  a icm:DeliveryParameter ;
  icm:targetType cat:ExampleService ;
.
:P2
  a icm:PropertyParameter ;
  icm:metric met:ThroughputPerUser ;
  icm:value "10.0"^^xsd:decimal ;
  icm:hasUnit [ icm:unit "MBit/s"^^xsd:string ] ;
.
:P3 icm:event icm:intentRejected .
:P4 icm:event icm:intentAccepted .
:P5 icm:event icm:intentComplies .
:P6 icm:event icm:intentDegrades .
:P7 icm:event icm:handlingEnded .
:P8 icm:event icm:updateRejected .
:P9 icm:event icm:updateFinished .
:P10 icm:event icl:InitialComplianceLatencyExpired .
:P11 icm:event icl:UpdateComplianceLatencyExpired .
:P12 icm:event icl:RecoveryComplianceLatencyExpired .
```

In this example intent owner asks the intent handler that it should recover from degradation within two minutes after the degradation has happened.

Furthermore, the intent owner expects an intent report if the compliance latency for initial compliance, compliance after update or compliance after degradation has expired

2.3.2. Setting default recovery time budget for an intent handler

This example demonstrates how to use an intent for defining default compliance latencies of an intent handler

Example 2: Default time budget

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix icl:
<http://tio.models.tmforum.org/tio/v1.0.0/IntentComplianceLatency/> .
@prefix icm:
<http://tio.models.tmforum.org/tio/v2.0.0/IntentCommonModel/> .
@prefix imo:
<http://tio.models.tmforum.org/tio/v1.0.0/IntentManagementOntology/> .
@prefix t:    <http://www.w3.org/2006/time#> .
@prefix xsd:  <http://www.w3.org/2001/XMLSchema#> .
@prefix :
<http://www.operator.org/IntentNamespace/intent20220322_12345/> .

# --- Intent --------------------
:ExampleIntentXYZ
  a icm:Intent ;
  icm:hasExpectation :E1, :E_report ;
.
# --- Targets --------------------
:T1_imf a :IntentManagerABC001 ; .

# --- Expectations --------------------
:E1
  a icm:PropertyExpectation
  icm:target :T1_imf ;
  icm:hasParameter :P1 ;
.
:E_reporting
  a icm:ReportingExpectation ;
  icm:target :ExampleIntentXYZ ;
  icm:anyOf ( P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 ) ;
.
# --- Parameters --------------------
:P1
  a icm:PropertyParam
  icl:InitialComplianceLatency  [ t:numericDuration "5"^^xsd:integer ;
                                  t:temporalUnit t:unitMinute ;
                                ] ;
  icl:UpdateComplianceLatency   [ t:numericDuration "1"^^xsd:integer ;
                                  t:temporalUnit t:unitMinute ;
                                ] ;
  icl:RecoveryComplianceLatency [ t:numericDuration "2"^^xsd:integer ;
                                  t:temporalUnit t:unitMinute ;
                                ] ;
.
:P3 icm:event icm:intentRejected .
:P4 icm:event icm:intentAccepted .
:P5 icm:event icm:intentComplies .
:P6 icm:event icm:intentDegrades .
:P7 icm:event icm:handlingEnded .
:P8 icm:event icm:updateRejected .
:P9 icm:event icm:updateFinished .
:P10 icm:event icl:InitialComplianceLatencyExpired .
:P11 icm:event icl:UpdateComplianceLatencyExpired .
:P12 icm:event icl:RecoveryComplianceLatencyExpired .
```

In this example a property expectation is used with a target referring to an individual intent manager. So the stated requirement is about the operation of the intent manager. This use of intent can set default operational requirements. An intent like this typically has an owner associated with setting operational policies. This can for example be a frontend system through which technical personnel of the network operator maintains and configures the autonomous network system.

In the property expectation sets now goals for compliance latency. As this is a goal for the intent manager, it is therefore applicable to all actions across other intents the intent manager is taking. If any action to reach compliance in any intent handled by this intent manager fails to meet these goals, this intent is degraded and the respective event is issued. In this example there is also intent reporting specified. This means the intent handler will report that this intent has degraded.

3. 1. Administrative Appendix

3.1. Document History

3.1.1. 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

3.1.2. 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

3.2. Acknowledgments

3.2.1. Guide Lead & Author

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

3.2.2. 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

3.2.3. 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