TM Forum Technical Report

Connector Model

TR293

Maturity Level: Beta	**Team Approved Date: 11-Apr-2023**
Release Status: Pre-production	**Approval Status: Member Evaluated**
Version 2.0.0	**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

Executive Summary

The connector model introduces vocabulary that enables the model federation for non-RDF based models which could be defined by TM Forum itself (e.g., SID models) or by other SDOs.

Introduction

This document describes the base vocabulary for connector model which can be used to create intent extension models that are specific to different SDOs and are applicable for the relevant domains or use cases. The corresponding intent extension models are defined in another suite of models.

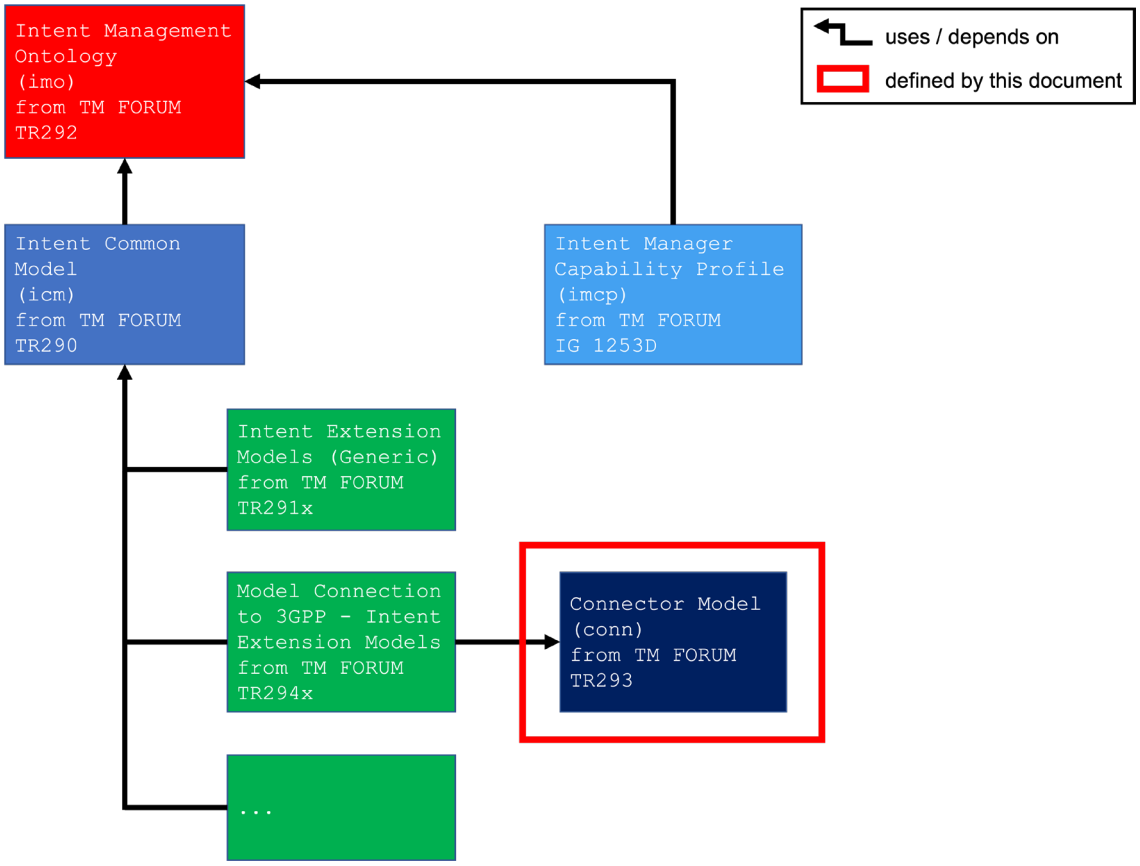

Figure 0.1: Relationship of Connector Model with other models from TM Forum

1. Motivation and Background

Model federation is one of the key benefits of RDF based modeling. This helps to create a new RDF model by using the artifacts that are defined in an existing RDF model. The concept of intent modeling through model federation is introduced in IG1253. However, this only specifies the federation of models that are RDF based, i.e., they can be referred to by simply using the IRI of the model. On the other hand, there exist a plenty of models that are non-RDF based and defined by different SDOs, e.g., SID models defined by TM Forum, models defined in technical specifications of 3GPP, and so on. For these non-RDF based models, the federation is not straightforward since no IRIs exist for such models. In order to refer to a property or KPI that is defined in these models, the proposed idea is to use connector model which is defined in this document.

The connector model helps to define all the parameters which identify the non-RDF based model (e.g., name, description, publisher etc.). Furthermore, it can also be specified which type of artifact is to be used from the model, e.g., a KPI defined in the model or a managed object that is stored in another system managed by the operator. The connector model also helps the SDOs to create new intent extension models that are domain and use case specific. For example, 3GPP can create an intent extension model for specifying the RAN intent, and use the vocabulary of connector model in order to refer to the KPIs defined in its specifications.

A similar concept of connecting models is proposed by W3C in Simple Knowledge Organization System (SKOS), i.e., a common data model for sharing and linking knowledge organization systems that can be expressed as machine-readable data. SKOS defines a knowledge organization system as a concept scheme that comprises a set of concepts. The SKOS data model includes semantic relation properties (e.g., broader, narrower, related etc.) that are used to link concepts within a concept scheme. On the other hand, for linking SKOS concepts between different concept schemes, mapping properties are defined which include broadMatch, narrowMatch, relatedMatch, closeMatch, exactMatch. The idea of mapping relationships between concepts existing in different concept schemes or models in SKOS is similar to what is proposed in the connector model defined in this document, however it is a broad concept and the model is missing concrete usage examples that depict its effectiveness. On the other hand, the connector model is much more detailed and explanatory in terms of how different models can be connected and used for creating new RDF models. Furthermore, it specifies how the definition of model elements (e.g., KPIs) and managed objects can be done differently depending on if there already exist some instances in a managed system which can be directly referenced in the new RDF model.

2. Notation and Namespaces

The connector model is defined in a namespace under the TM Forum domain. It depends on the following models and uses the respective namespaces:

Model	Prefix	Namespace	Published by
Connector Model	conn	http://tio.models.tmforum.org/tio/v1.0.0/ConnectorModel/	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
XML Schema	xsd	http://www.w3.org/2001/XMLSchema#	W3C

The proposed prefix label for connector model is "conn".

Resource Description Framework (RDF) and Resource Description Framework Schema (RDFS) provide the base model stack for defining connector model. For example, classes in the connector model are derived from the definition of "Class" in RDFS, and properties in the connector model are derived from the definition of "Property" in RDF.

The connector model uses "string" datatype that is defined in XML Schema.

3. Connector Types

The connector model defines four different types of connectors which are described below:

3.1. Model Connectors

Model connectors describe the details about non-RDF based models, e.g., name, description, publisher etc.

3.2. Model Element Connectors

Model element connectors describe the artifacts (e.g., properties, parameters, KPIs etc.) that are defined in non-RDF based models. When providing the details about a model element connector, the corresponding model connector of the relevant model also needs to be specified in order to describe which model defines the corresponding artifact.

3.3. Managed System Connectors

Managed system connectors describe the managed systems that managing entities (e.g., operators) use for storing their managed objects. A managed system can be an inventory, database, or any other system used for storing managed objects, and can be specified using a URI for example. When creating an RDF model, the managed system connectors are only used if there is a need to refer to already existing managed objects that are stored in other managed systems.

3.4. Managed Object Connectors

The managed object connectors describe the managed objects that are stored in managed systems. When providing the details about a managed object connector, the corresponding managed system connector of the relevant managed system also needs to be specified in order to describe which managed system contains the corresponding managed object.

It should be noted that the managed object connectors are only used when new instances of the artifacts (defined in a certain non-RDF based model) do not need to be created, but instead the instances of corresponding artifacts are already existing in other managed systems, and they just need to be referred in an RDF model.

4. Vocabulary Specification

This section describes all the classes and properties that are specified in the connector model.

4.1. Classes

The connector model contains the following classes:

Class:	**conn:ModelConnector**
Definition:	An instance of class `conn:ModelConnector` specifies the details about a non-RDF based model, e.g., name, description, publisher etc.
Instance of:	`rdfs:Class`

Class:	**conn:ModelElementConnector**
Definition:	An instance of class `conn:ModelElementConnector` specifies the artifacts (e.g., properties, parameters, KPIs etc.) that are defined in a non-RDF based model.
Instance of:	`rdfs:Class`

Class:	**conn:ManagedSystemConnector**
Definition:	An instance of class `conn:ManagedSystemConnector` specifies a managed system that a managing entity (e.g., operator) uses to store its managed objects.
Instance of:	`rdfs:Class`

Class:	**conn:ManagedObjectConnector**
Definition:	An instance of class `conn:ManagedObjectConnector` specifies a managed object that is stored in a managed system.
Instance of:	`rdfs:Class`

4.2. Properties

The connector model contains the following properties:

Property:	`conn:modelPublisher`
Definition:	This property provides information about the SDO that published a model, e.g., 3GPP
Instance of:	`rdf:Property`

Property:	conn:modelPublisher
Domain:	conn:ModelConnector
Range:	xsd:string

Property:	conn:modelLabel
Definition:	This property provides the label for identifying a model defined by the SDO, e.g., TS 28.312
Instance of:	rdf:Property
Domain:	conn:ModelConnector
Range:	xsd:string

Property:	conn:modelVersion
Definition:	This property describes the version of a model defined by the SDO, e.g., v17.1.1
Instance of:	rdf:Property
Domain:	conn:ModelConnector
Range:	xsd:string

Property:	conn:modelDescription
Definition:	This property provides the description about a model defined by the SDO
Instance of:	rdf:Property
Domain:	conn:ModelConnector
Range:	xsd:string

Property:	conn:modelElementLabel
Definition:	This property provides a label for identifying an artifact in the model, e.g., PLMNId
Instance of:	rdf:Property
Domain:	conn:ModelElementConnector
Range:	xsd:string

Property:	conn:inModel
Definition:	This property describes the model in which a particular artifact is defined by the SDO
Instance of:	rdf:Property
Domain:	conn:ModelElementConnector
Range:	conn:ModelConnector

Property:	conn:inManagedSystem
Definition:	This property describes the managed system in which a particular managed object is stored

Property:	conn:inManagedSystem
Instance of:	rdf:Property
Domain:	conn:ManagedObjectConnector
Range:	conn:ManagedSystemConnector

Property:	conn:managedSystemDomain
Definition:	This property describes the domain under which a managed system is residing
Instance of:	rdf:Property
Domain:	conn:ManagedSystemConnector
Range:	xsd:string

Property:	conn:managedSystemSubnetwork
Definition:	This property describes the subnetwork of the domain under which a managed system is residing
Instance of:	rdf:Property
Domain:	conn:ManagedSystemConnector
Range:	xsd:string

Property:	conn:managedSystemId
Definition:	This property describes the identifier of a managed system. A managed system can be uniquely identified by a combination of properties conn:managedSystemDomain, conn:managedSystemSubnetwork and conn:managedSystemId
Instance of:	rdf:Property
Domain:	conn:ManagedSystemConnector
Range:	xsd:string

Property:	conn:managedSystemUri
Definition:	This property describes the URI of a managed system. Instead of providing the domain, subnetwork, and identifier of a managed system, a URI pointing to the managed system can be directly specified.
Instance of:	rdf:Property
Domain:	conn:ManagedSystemConnector
Range:	xsd:string

Property:	conn:managedObjectId
Definition:	This property describes the identifier of a managed object which can be found in a particular managed system. The information about the corresponding managed system (in which one can look up for the managed object) needs to be provided using the property conn:inManagedSystem
Instance of:	rdf:Property
Domain:	conn:ManagedObjectConnector
Range:	xsd:string

5. Model Usage and Examples

This section provides examples of how to use the connector model. The examples are provided using the "Terse RDF Triple Language" (TURTLE), i.e., a commonly used format for RDF models.

Example 1: Specifying a model connector

```
@prefix conn: <http://tio.models.tmforum.org/tio/v1.0.0/ConnectorModel/> .
@prefix  mod: <http://www.example.com/ModelNamespace#> .

mod:ModelConn_3GPP28312
  a conn:ModelConnector ;
  conn:modelPublisher "3GPP" ;
  conn:modelLabel "TS 28.312" ;
  conn:modelVersion "v17.1.1" ;
  conn:modelDescription "Intent driven management services for mobile
networks" ;
.
```

Example 1 shows how to specify a model connector. In this example, `mod:3GPP_28312` is defined as a model connector and its details are provided as well. This model is published by 3GPP, it is labelled with specification number TS 28.312, the version used is v17.1.1, and its description shows that it specifies intent driven management services for mobile networks.

Example 2: Specifying an attribute using model element connector

```
@prefix conn: <http://tio.models.tmforum.org/tio/v1.0.0/ConnectorModel/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix  rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix  mod: <http://www.example.com/ModelNamespace#> .

mod:RAT
  a rdfs:Class ;
  rdfs:subClassOf rdf:Container ;
  rdfs:subClassOf conn:ModelElementConnector ;
  conn:inModel mod:ModelConn_3GPP28312 ;
  conn:modelElementLabel "RAT" ;
.
```

Example 2 shows how to use model element connector for specifying an attribute whose instances can be created and used in an RDF model. In this example, a class named `mod:RAT` has been defined which is a subclass of `rdf:Container` meaning that it is a container with multiple elements. It is also a subclass of conn:ModelElementConnector meaning that it is not an ordinary class, but instead it defines an attribute specified in a non-RDF based model. The information about model connector (see Example 1) of the relevant model is provided using `conn:inModel`, and the attribute name is provided using `conn:modelElementLabel` which should exactly match the name specified in the actual model.

Example 3: Specifying a metric using model element connector

```
@prefix conn: <http://tio.models.tmforum.org/tio/v1.0.0/ConnectorModel/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix  icm:
<http://tio.models.tmforum.org/tio/v2.0.0/IntentCommonModel/> .
@prefix  mod: <http://www.example.com/ModelNamespace#> .

mod:LowSINRRatio
  a rdfs:Class ;
  rdfs:subClassOf icm:Quantity ;
  rdfs:subClassOf conn:ModelElementConnector ;
  conn:inModel mod:ModelConn_3GPP28312 ;
  conn:modelElementLabel "LowSINRRatio" ;
.
```

Example 3 shows how to use model element connector for specifying a metric to be used in an RDF model. In this example, a class named `mod:LowSINRRatio` has been defined which is a subclass of `icm:Quantity` meaning that it is a numerical value. It is also a subclass of `conn:ModelElementConnector` meaning that it is not an ordinary class, but instead it defines a metric specified in a non-RDF based model. The information about model connector (see Example 1) of the relevant model is provided using `conn:inModel`, and the metric name is provided using `conn:modelElementLabel` which should exactly match the name specified in the actual model.

Example 4: Specifying a managed system connector

```
@prefix conn: <http://tio.models.tmforum.org/tio/v1.0.0/ConnectorModel/> .
@prefix  msc: <http://www.example.com/ManagedSystemConnectorsNamespace#> .

# --- OPTION 1 --------------------
msc:ManagedSystem_Operator1
  a conn:ManagedSystemConnector ;
  conn:managedSystemDomain "operator1.com" ;
  conn:managedSystemSubnetwork "region1" ;
  conn:managedSystemId "id101-555" ;
.

# --- OPTION 2 --------------------
msc:ManagedSystem_Operator2
  a conn:ManagedSystemConnector ;
  conn:managedSystemUri "http://operator2.com/region2/id202-666/" ;
.
```

Example 4 shows two different options of how to specify a managed system connector. In option 1, `msc:ManagedSystem_Operator1` is defined as a managed system connector, and the domain, subnetwork, and identifier of the managed system are separately provided which help to uniquely identify the managed system. In option 2, another managed system connector is defined, i.e., `msc:ManagedSystem_Operator2`, but in this case, a URI pointing to the managed system is directly specified instead of separately providing the domain, subnetwork, and identifier of the managed system.

Example 5: Creating an instance of a class with/without managed object connector

```
@prefix conn: <http://tio.models.tmforum.org/tio/v1.0.0/ConnectorModel/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix  mod: <http://www.example.com/ModelNamespace#> .
@prefix  msc: <http://www.example.com/ManagedSystemConnectorsNamespace#> .
@prefix  int: <http://www.example.com/IntentNamespace#> .

# --- OPTION 1 (without managed object connector) --------------------
int:RAT1
  a mod:RAT ;       # mod:RAT is defined as a subclass of
conn:ModelElementConnector and rdf:Container
  rdfs:member "UTRAN" ;
  rdfs:member "EUTRAN" ;
  rdfs:member "NR" ;
.

# --- OPTION 2 (with managed object connector) --------------------
int:RAT2
  a mod:RAT ;       # mod:RAT is defined as a subclass of
conn:ModelElementConnector and rdf:Container
  rdfs:subClassOf conn:ManagedObjectConnector ;
  conn:inManagedSystem msc:ManagedSystem_Operator1 ;
  conn:managedObjectId "id123" ;
.
```

Example 5 shows how to create an instance of a class which has been defined as a subclass of `conn:ModelElementConnector` meaning that it refers to an artifact defined in a non-RDF based model. There are two possibilities in this scenario. One possibility is that an instance needs to be created, and its value is also defined locally in the RDF model. The other possibility is that an instance only needs to refer to a managed object (that resides in a certain managed system), and its value is defined in the managed system instead of defining it locally in the RDF model, in which case `conn:ManagedObjectConnector` needs to be used. In Example 5, two options depict the above-mentioned possibilities. In option 1, `int:RAT1` is an instance created from the class `mod:RAT` which is a subclass of `conn:ModelElementConnector` (see Example 2). In this case, the value of `int:RAT1` is defined locally, i.e., it is a container with three entries. In option 2, `int:RAT2` is an instance that is also created from the class `mod:RAT`, but its value is not defined locally. Instead, `int:RAT2` is defined as a subclass of `conn:ManagedObjectConnector` meaning that it refers to a managed object stored in another managed system where its value is also defined. The information about managed system connector (see Example 4) of the relevant managed system is provided using `conn:inManagedSystem`, and the managed object identifier is provided using `conn:managedObjectId` which can be used to look up the managed object in the corresponding managed system.

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-Jan-2023	Alan Pope	Initial Release
2.0.0	11-Apr-2023	Alan Pope	Final edits prior to publication

6.1.2. Release History

Release Status	Date Modified	Modified by:	Description of changes
Pre-production	31-Jan-2023	Alan Pope	Initial Release
Preproduction	17-Mar-2023	Adrienne Walcott	Updated to Member Evaluated status
Pre-production	11-Apr-2023	Alan Pope	Updated to version 2.0.0
Pre-production	15-May-2023	Adrienne Walcott	Updated to Member Evaluated status

6.2. Acknowledgments

6.2.1. Guide Lead & Author

Member	Title	Company
Muhammad Rehan Raza	Senior Researcher	Ericsson

6.2.2. Main Contributors

Member	Title	Company
Muhammad Rehan Raza	Senior Researcher	Ericsson
Jörg Niemöller	Expert of Analytics and Customer Experience	Ericsson
Kevin McDonnell	Senior Director, Intelligent Automation	Huawei
Yuval Stein	AVP Technologies	TEOCO
Leonid Mokrushin	Principal Researcher	Ericsson

6.2.3. Additional Inputs

Member	Title	Company
Aaron Boasman-Patel	Vice President, AI, Labs and Innovation	TM Forum
Dave Milham	Chief Architect	TM Forum
Alan Pope	Collaboration Manager	TM Forum