**2018-07-18****(10) Proposal Stage****WD****2018****Vocabulary****1****14812****Technical Report**  Date: 2018-07-18

**ISO/TR 14812 (WD)**

ISO TC 204/WG 1

Secretariat: ANSI

**Intelligent transport systems — Vocabulary**

WD stage

**Warning for WDs and CDs**

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this draft are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

© ISO 2018, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO’s member body in the country of the requester.

ISO copyright office

Ch. de Blandonnet 8 • CP 401

CH-1214 Vernier, Geneva, Switzerland

Tel.  + 41 22 749 01 11

Fax  + 41 22 749 09 47

copyright@iso.org

www.iso.org

Contents

[Foreword iv](#_Toc519697446)

[Introduction v](#_Toc519697447)

[1 Scope 1](#_Toc519697448)

[2 Normative references 1](#_Toc519697449)

[3 Terms and definitions 1](#_Toc519697450)

[3.1 System Model 1](#_Toc519697451)

[3.2 Service Model 3](#_Toc519697452)

[3.2.1 Generic Service Model 3](#_Toc519697453)

[3.2.2 ITS Service Model 5](#_Toc519697454)

[3.2.3 ITS station service model 6](#_Toc519697455)

[3.3 Service Implementations 7](#_Toc519697456)

[3.3.1 ITS service implementation model 7](#_Toc519697457)

[3.3.2 ITS station service implementation model 8](#_Toc519697458)

[3.4 Architectural model 8](#_Toc519697459)

[3.4.1 Conceptual model of an architecture description 9](#_Toc519697460)

[3.4.2 Architecture types 11](#_Toc519697461)

[3.4.3 Physical view of an ITS architecture 13](#_Toc519697462)

[3.4.4 Realization of physical architecture 14](#_Toc519697463)

[3.4.5 Implementations 18](#_Toc519697464)

[3.5 Data concepts 19](#_Toc519697465)

[3.6 Vehicle Types 20](#_Toc519697466)

[3.7 Specifications 20](#_Toc519697467)

[Annex A (informative) Annex title e.g. Example of a figure and a table 21](#_Toc519697468)

[Bibliography 22](#_Toc519697469)

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see [www.iso.org/directives](http://www.iso.org/directives)).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see [www.iso.org/patents](http://www.iso.org/patents)).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see the following URL: [www.iso.org/iso/foreword.html](http://www.iso.org/iso/foreword.html).

The committee responsible for this document is Technical Committee ISO/TC *204*, Intelligent transport systems.

This is the first edition of this document.

Introduction

This document establishes the preferred vocabulary within ISO/TC 204. Standards developed by this committee are encouraged to copy the definitions found in this document as they have been formulated in accordance with major ISO standards such as ISO 704 *Terminology Work – Principles and Methds* and are based on a consistent concept model*.*

Other standards groups and organisations are also encouraged to adopt the terminology in this document to promote better understanding of terms among ITS professionals worldwide. The terms and definitions contained within this document can be searched online at ISO’s Online Browsing Platform available at https://www.iso.org/obp.

**Intelligent transport systems — Vocabulary**

# Scope

This document is intended to formally document the vocabulary used and developed by the entire intelligent transport systems (ITS) community. The terms and definitions are developed in an open environment according to the principles defined in Annex A with specific versions formally adopted through updates to this ISO document.

It is recognized that the contents of this document are not exhaustive, and that terminology evolves over time. ISO/TC 204 standards are encouraged to adopt the definitions included within this document; however, it is recognized that each document may need to define additional terms to meet its own needs. Annex B provides the best practices for defining terms in other documents. The process to suggest changes to the contents of this document is provided in Annex C.

In most cases, the definitions provided within this document are suitable for general application throughout ITS. In those circumstances where the definition only applies within a restricted context, the context is indicated at the beginning of the definition.

EDITOR’S NOTE 1: Text marked “EDITOR’S NOTE” in this document will be removed in the final TR.

EDITOR’S NOTE 2: To facilitate the review of this Technical Report by others, the attached Microsoft Excel file maps each of the terms defined in this document to similar terms defined in existing ISO/TC 204 documents, along with the definitions for those terms and the WGs responsible for developing and maintaining those documents.

# Normative references

There are no normative references in this document.

# Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at <http://www.iso.org/obp>

— IEC Electropedia: available at http://www.electropedia.org/

### 3.1 System Model

3.1.1

system

combination of interacting elements organized to achieve one or more stated purposes

Note 1 to entry: A second note

Note 2 to entry: This is a sample note

[Source: ISO 24774:2010]

3.1.2

transport system

**system** of infrastructure elements and optionally **vehicles** that jointly are designed to move entities from an **origin** to a **destination** Note 1 to entry: Transport systems are not necessarily limited to the transport of physical entities; the Internet could be considered a transport system of electronic entities.

3.1.3

surface transport system

[*transport system*](https://d.docs.live.net/b36e87b7c6bb83d7/Standards/ISO/TC204%20WG1/14812%20-%20Vocabulary%20(TR)/v1/WD/$element:/%7b8648D088-383A-40a1-9D19-B3E02A7060DF%7d) designed to move physical entities across the surface of the earth Note 1 to entry: A surface transport system might include tunnels, bridges, and similar elements.

3.1.4

technology system

**system** that is comprised of information, communication, sensor, and control technologies

[Source: local]

3.1.5

intelligent transport system

ITS

intelligent transportation system

[*technology system*](https://d.docs.live.net/b36e87b7c6bb83d7/Standards/ISO/TC204%20WG1/14812%20-%20Vocabulary%20(TR)/v1/WD/$element:/%7b03252B24-F743-49ac-87D9-5187F13128C7%7d) that is designed to benefit a [*surface transport system*](https://d.docs.live.net/b36e87b7c6bb83d7/Standards/ISO/TC204%20WG1/14812%20-%20Vocabulary%20(TR)/v1/WD/$element:/%7b261629D5-8D7B-439b-8DAE-5905D2EEF4D5%7d)

Note 1 to entry: Benefits potentially include, but are not limited to, increased safety, sustainability, efficiency, and comfort.

Note 2 to entry: The long form (i.e., intelligent transport system) is often used when the noun is used as a subject, whereas the abbreviation (i.e., ITS) is often used to modify another noun (e.g., Intelligent transport systems provide ITS services.).

Note 3 to entry: There is not complete agreement on the precise limits of ITS. Currently, the term is almost exclusively applied to ground-based travel of goods and people over significant distances. The term is viewed as including ferry systems, which often form an integ

3.1.6

cooperative ITS

C-ITS

subset of [*ITS*](https://d.docs.live.net/b36e87b7c6bb83d7/Standards/ISO/TC204%20WG1/14812%20-%20Vocabulary%20(TR)/v1/WD/$element:/%7b95C4A864-65B7-4046-8231-06670EF95C77%7d) where information is shared among *physical objects* based on mutual security agreements with a credentialing system

Note 1 to entry: Most current-day computer systems rely upon an agreement between the owners or users of the two systems involved in an exchange. For example, there is typically bi-lateral agreement between a provider and a user. C-ITS instead relies upon a system of third-party agreements. In other words, a third party defines the rules for operating within a domain and creates a trust network. Providers and users establish agreements with the third party agreeing to abide by the rules. Providers and users then directly interact based on the rules of the trust network without requiring any direct relationship between the two.

Note 2 to entry: C-ITS does not preclude two parties from having a separate bi-lateral agreement, but any services requiring a bi-lateral agreement would not technically be considered "C-ITS". For example, two emergency vehicles may have a bi-lateral agreement to share incident information containing sensitive data; such a service would not be considered C-ITS, since it is under a bi-lateral agreement. However, the same two vehicles may share vehicle location and motion information for collision avoidance purposes; this would be considered C-ITS since the agreement to share that information is under the broader use case that uses a third-party agreement.

## System Model

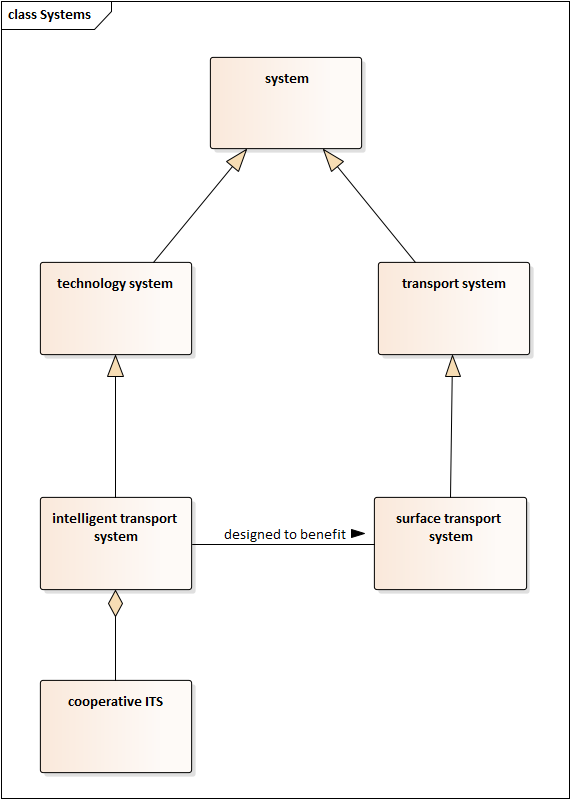


Figure 1: General terms

3.1.1

system

combination of interacting elements organized to achieve one or more stated purposes

[ISO 24774:2010]

3.1.2

transport system

**system** of infrastructure elements and optionally container entities that jointly are designed to move entities from an **origin** to a **destination**

Note 1 to entry: Transport systems are not necessarily limited to the transport of physical entities; the Internet could be considered a transport system of electronic entities.

3.1.3

surface transport system

**transport system** designed to move physical entities across the surface of the earth

Note 1 to entry: A surface transport system includes tunnels, bridges, and similar elements.

3.1.4

technology system

**system** that is comprised of information, communication, sensor, and/or control technologies

3.1.5

intelligent transport system

ITS

intelligent transportation system

**technology system** that is designed to benefit a **surface transport system**

Note 1 to entry: Benefits potentially include, but are not limited to, increased safety, sustainability, efficiency, and comfort.

Note 2 to entry: There is not complete agreement on the precise limits of ITS. Currently, the term is almost exclusively applied to ground-based travel of goods and people over significant distances. The term is viewed as including ferry systems, which often form an integral part of a local surface transport system; it is less clear if it includes long-distance sea-fairing ships. ITS is also generally applied to transport systems that cover a considerable distance (e.g., control systems for factory conveyance technologies are not often referred to as “ITS”). Some have suggested that air travel, which is arguably a transport system designed to move physical entities between points on the surface of the earth, should be included in the scope of the term, but this perspective is not universally accepted. It is expected that the exact limits of the term will be further refined as the technology matures.

Note 3 to entry: The long form (i.e., “intelligent transport system”) is often used when the noun is used as a subject, whereas the abbreviation (i.e., “ITS”) is often used to modify another noun (e.g., “Intelligent transport systems provide ITS services.”).

3.1.6

cooperative ITS

C-ITS

subset of **ITS** where information is shared among **ITS stations** based on an agreement with a tertiary party rather than any direct contractual relationship with each other

Note 1 to entry: Most current-day computer systems rely upon an agreement between the owners or users of the two systems involved in an exchange. For example, there is typically bi-lateral agreement between a provider and a user. C-ITS instead relies upon a system of third-party agreements. In other words, a third party defines the rules for operating within a domain and creates a trust network. Providers and users establish agreements with the third party agreeing to abide by the rules. Providers and users then directly interact based on the rules of the trust network without requiring any direct relationship between the two.

Note 2 to entry: C-ITS does not preclude two parties from having a separate bi-lateral agreement, but any services requiring a bi-lateral agreement would not technically be considered “C-ITS”. For example, two emergency vehicles may have a bi-lateral agreement to share incident information containing sensitive data; such a service would not be considered C-ITS, since it is under a bi-lateral agreement. However, the same two vehicles may share vehicle location and motion information for collision avoidance purposes; this would be considered C-ITS since the agreement to share that information is under the broader use case that uses a third-party agreement.

## Service model

### Generic service model

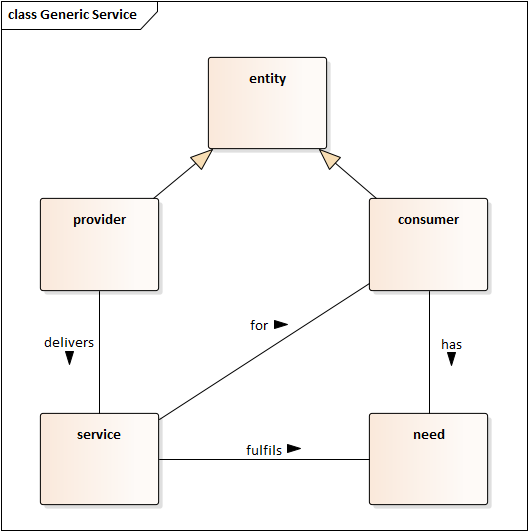


Figure 2: Model for generic services

3.2.1.1

entity

concrete or abstract thing that exists, did exist, or might exist, including associations among these things

EXAMPLE: A person, object, event, idea, process, etc.

[Source: ISO/TR 17185-1]

3.2.1.2

service provider

entity that delivers one or more **services**

3.2.1.3

service

performance of one or more tasks or provision of one or more facilities to enable one or more tasks for a **consumer** to fulfil a **need**

3.2.1.4

consumer

user

entity that has a **need** to be fulfilled

3.2.1.5

need

factor or condition necessary to achieved desired results within a specified context of use

[Adapted from ISO 25060:2010(en), 2.25]

### ITS service model

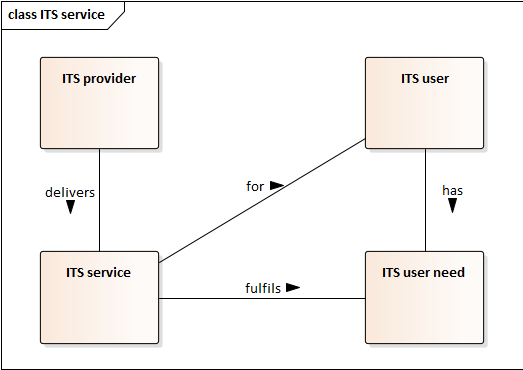


Figure 3: ITS service model

3.2.2.1

service provider

**<ITS> entity** that delivers one or more **ITS services**

3.2.2.2

service

**<ITS>** performance of one or more tasks that fulfils an **ITS user need** for an **ITS user**

3.2.2.3

user

**<ITS> entity** that has an **ITS user need** to be fulfilled

3.2.2.4

user need

**<ITS> need** of an entity external to the intelligent transport system for a **surface transport system** benefit that can be met with the use of a **technology system**

### ITS station service model

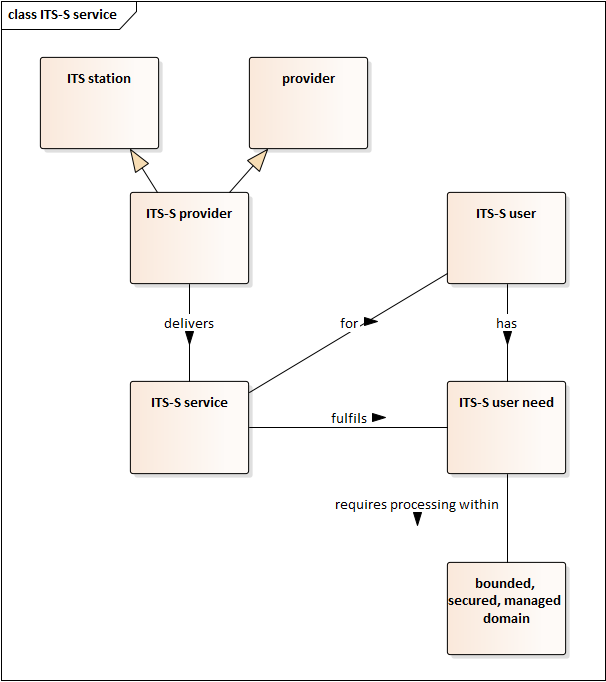


Figure 4: ITS station service model

3.2.3.1

service provider

<ITS-S> **ITS station** that delivers one or more **ITS-S services**

3.2.3.2

service

<ITS-S>performance of one or more tasksthat fulfils an **ITS-S user need** for an **ITS-S user**

Note 1 to entry An ITS-S service is also an ITS service if the consumer is external to the intelligent transport system.

3.2.3.3

user

<ITS-S> **entity** that has an **ITS-S user need**

3.2.3.4

user need

<ITS-S> **need** for processing within a **bounded, secure, managed domain**

3.2.3.5

bounded, secure, managed domain

controlled processing environment that adheres to a minimum set of management and security principles and procedures so as to establish a level of trust between itself and other similar ITS stations with which it might communicate

### ITS-S communication service model



Figure 5: ITS-S communication service model

3.2.4.1

communication process

<ITS-S> **entity** that delivers one or more **ITS-S communication services**

3.2.4.2

communication service

<ITS-S>performance of one or more tasksthat fulfils an **ITS-S communication need** for an **ITS-S communication user**

3.2.4.3

communication user

<ITS-S> **entity** that has an **ITS-S communication need**

3.2.4.4

communication need

<ITS-S> *need* for communication functionality that connects an *ITS-S* to other nodes

## Service Implementations

### ITS service implementation model

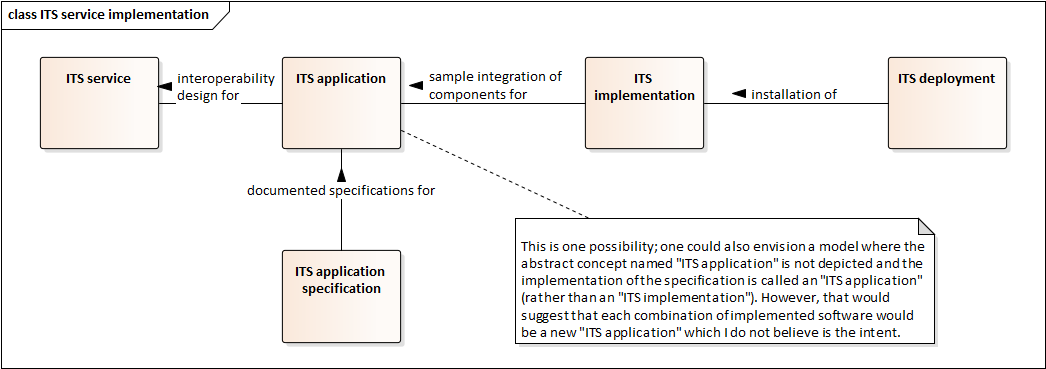


Figure 6: ITS service implementation model

3.3.1.1

interoperability design

characteristics necessary to fully define how various **system elements** and **terminators** interoperate to provide a **service**

Note 1 to entry: Characteristics often include but are not limited to functional, performance, and interface requirements.

3.3.1.1

ITS application

**interoperability design** for an **ITS service**

3.3.1.2

ITS application specification

one or more documents that fully detail the **ITS application**

3.3.1.3

ITS implementation

integration of each **system element** and **terminator** necessary to implement one or more **ITS applications**

Note 1 to entry: An ITS application typically requires multiple components (e.g., a ITS-S acting as a user and another ITS-S acting as a provider). An ITS implementation includes a sample of each component necessary for the service but often does not represent a complete deployment.

Note 2 to entry: An ITS implementation is typically used for a laboratory or other experimental environment prior to a full-scale deployment.

3.3.1.4

ITS deployment

installation capable of implementing one or more **ITS applications**

Note 1 to entry: An ITS deployment typically refers to the support, central and roadside ITS stations coupled with a tacit acknowledgement of the mobile ITS stations that will communicate with the support, central, or roadside ITS stations.

### ITS station service implementation model

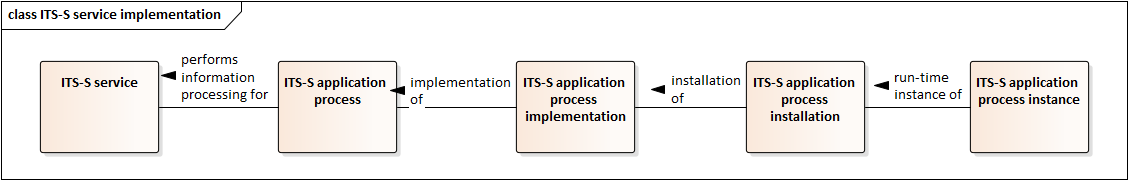


Figure 7: ITS station service implementation model

3.3.1.5

ITS-S application process

**element** in an **ITS station** that performs information processing for one or more **ITS-S services**

EDITOR’S NOTE 1: It is unclear if this term is intended to describe the 1) the logical concept of a process that fulfils a role (e.g., “word processor”), 2) a specific implementation of the logical concept (e.g., “MS Word”), 3) a specific deployment of that implementation (e.g., “MS Word with license XXX”), or a specific run-time instance of the implementation. I believe our standards will need to be clear about these distinctions, especially as we get into security, provisioning, and deployment issues.

3.3.1.6

ITS-S application process implementation

implementation of an **ITS-S application process**

Note 1 to entry: A unique implementation is typically associated with an implementation name, a version, and release number.

3.3.1.7

ITS-S application process installation

installation of an **ITS-S application process implementation**

Note 1 to entry: A unique installation is typically associated with a specific license code or serial number.

3.3.1.8

ITS-S application process instance

run-time instance of an **ITS-S application process installation**

## Architectural model

### Conceptual model of an architecture description

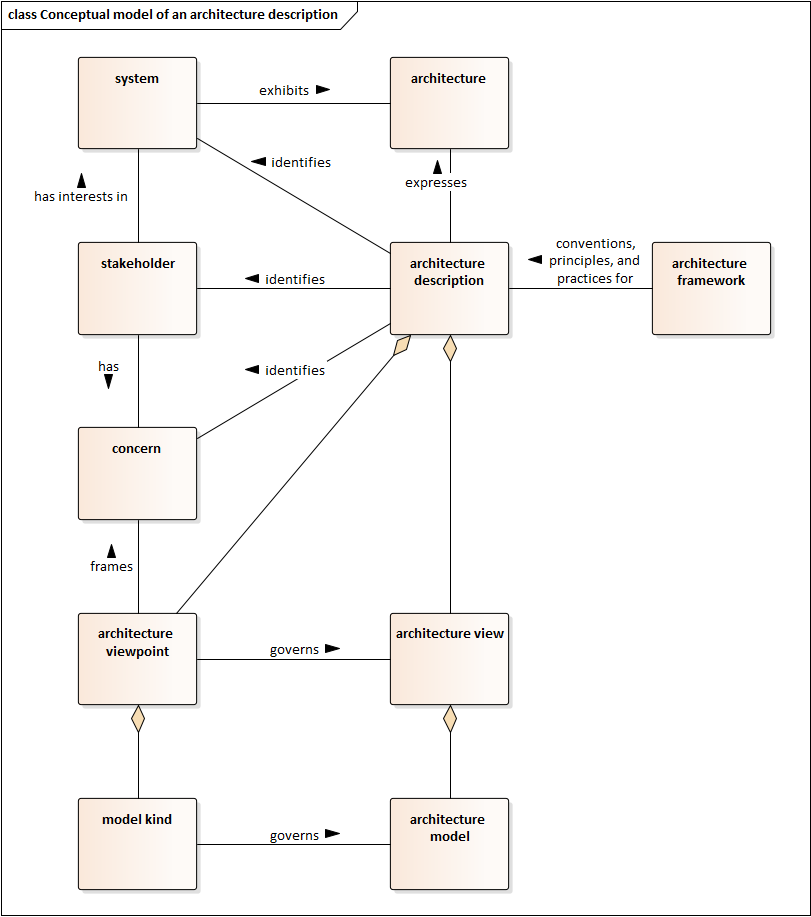


Figure 8: Conceptual model of an architecture description

3.4.1.1

architecture

<system> fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution

[ISO 42010:2011, 3.2]

3.4.1.2

architecture description

work product used to express an architecture

[ISO 42010:2011, 3.3]

3.4.1.3

architecture framework

conventions, principles and practices for the description of architectures established within a specific domain of application and/or community of stakeholders

EXAMPLE 1 Generalised Enterprise Reference Architecture and Methodologies (GERAM) [ISO 15704] is an architecture framework.

EXAMPLE 2 Reference Model of Open Distributed Processing (RM-ODP) [ISO/IEC 10746] is an architecture framework.

[ISO 42010:2011, 3.4]

3.4.1.4

architecture model

work product representing one or more architecture views and expressed in a format governed by a model kind

3.4.1.4

architecture view

work product expressing the architecture of a system from the perspective of specific system concerns

[ISO 42010:2011, 3.5]

3.4.1.5

architecture viewpoint

work product establishing the conventions for the construction, interpretation and use of architecture views to frame specific system concerns

[ISO 42010:2011, 3.6]

3.4.1.6

concern

<system> interest in a system relevant to one or more of its stakeholders

NOTE A concern pertains to any influence on a system in its environment, including developmental, technological, business, operational, organizational, political, economic, legal, regulatory, ecological and social influences.

[ISO 42010:2011, 3.7]

3.4.1.7

model kind

conventions for a type of modelling

NOTE Examples of model kinds include data flow diagrams, class diagrams, Petri nets, balance sheets, organization charts and state transition models.

[ISO 42010:2011, 3.9]

3.4.1.8

stakeholder

<system> individual, team, organization, or classes thereof, having an interest in a system

[ISO 42010:2011, 3.10]

3.4.1.9

element

<system> component member of an architecture

### Architecture types

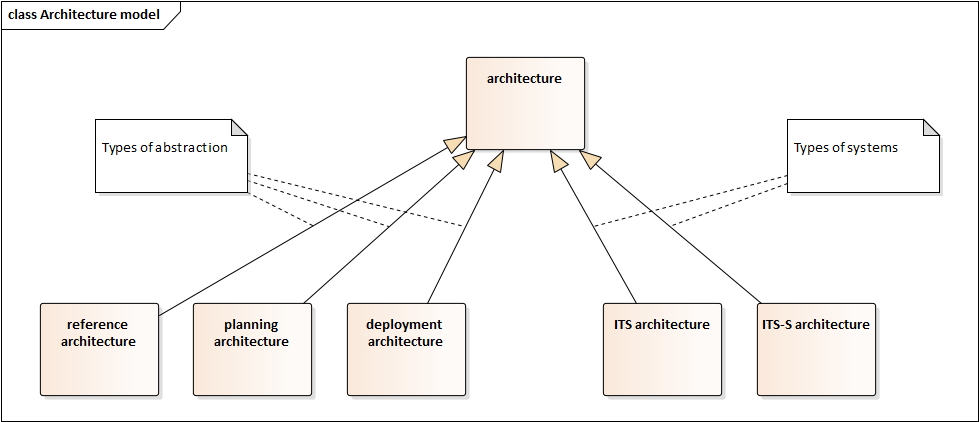


Figure 9: Architecture model

3.4.2.1

reference architecture

template solution for an architecture for a particular domain [14]

Note 1 to entry: Interface standards are based on a reference architecture, which should be explicitly described.

3.4.2.2

planning architecture

architecture that provides a long-term vision of system elements that may be deployed and managed by different projects and/or entities

Note 1 to entry: Some countries use the term “regional architecture”, but in international standards, the term “regional” is avoided due to its multiple meanings.

3.4.2.3

deployment architecture

architecture that provides a vision of a specific deployment of a system

3.4.2.4

ITS architecture

architecture for one or more ITS services

Note 1 to entry: An ITS architecture might be a reference, planning, or deployment architecture.

Note 2 to entry: The Harmonised Architecture Reference for Technical Standards (HARTS; http://htg7.org) is an example of an ITS reference architecture.

3.4.2.5

ITS-S architecture

reference architecture for ITS components as defined in ISO 21217

Note 1 to entry: The ITS-S architecture is the preferred architecture for ITS components.

### Physical view of an ITS architecture

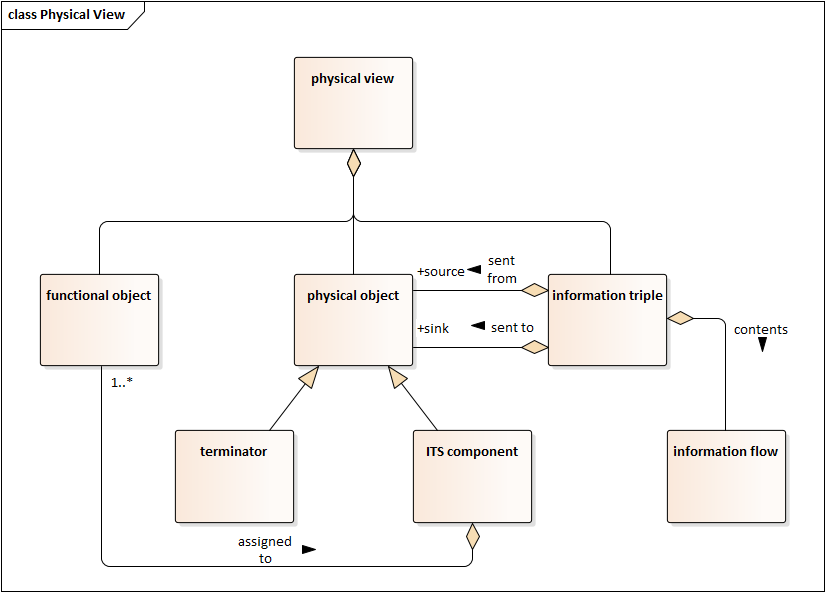


Figure 10: Physical view model

3.4.3.1

functional object

group of similar system processes from the functional view of the architecture that are jointly assigned to an ITS component in the physical view.

3.4.3.2

information flow

information that is exchanged between physical objects

3.4.3.3

information triple

information flow from a physical object acting as an information source and sent to another physical object acting as an information sink

3.4.3.4

ITS component

physical object that has been assigned one or more ITS functional objects

Note 1 to entry: Physical objects are ITS components if they are an integral part of the system; otherwise they are terminators.

3.4.3.5

physical object

element within the physical view of an ITS architecture that represents an physical entity that interacts with other physical objects in the provision of ITS services

3.4.3.6

terminator

physical object that represents an external user of ITS

### Realization of physical architecture

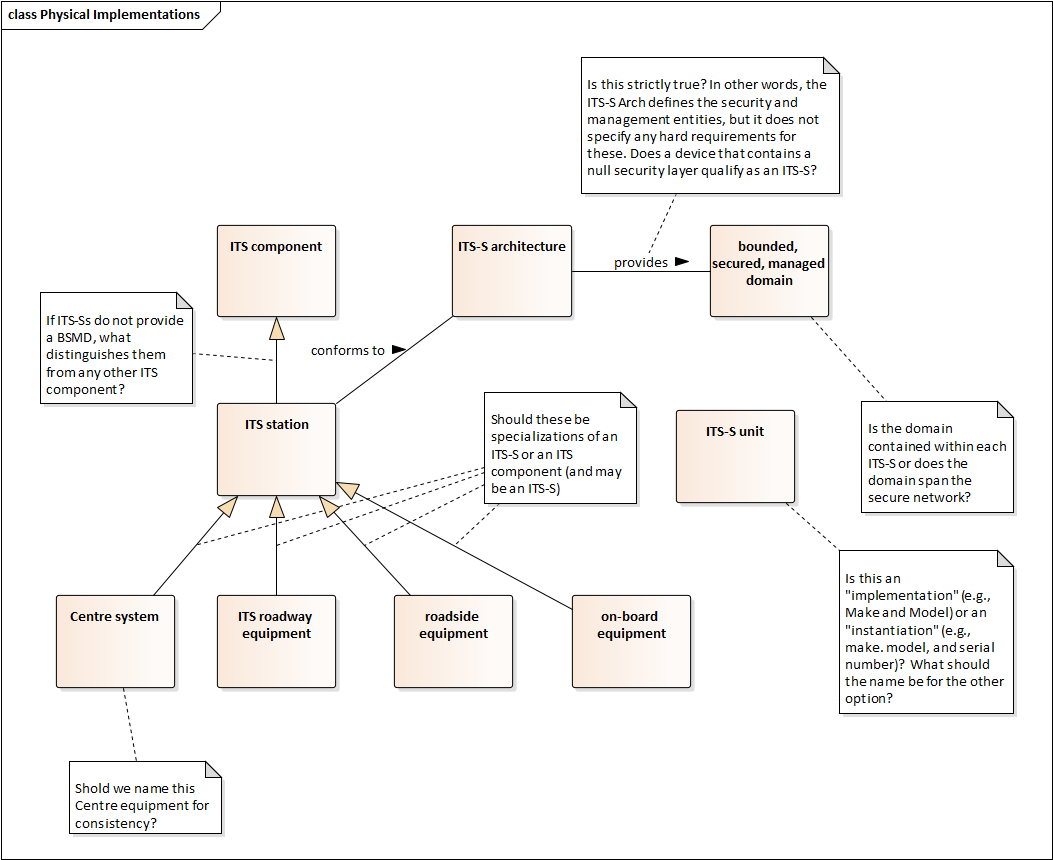


Figure 11: Physical implementation model

3.4.4.1

ITS station

ITS component that provides a bounded secured managed domain and conforms to the ITS station architecture

Note 1 to entry: The ITS station architecture is defined in ISO 21217.

EDITOR’S NOTE 1 There appears to be ambiguity (and perhaps disagreement within the industry) as to whether a “bounded secured managed domain” is the logical domain within each ITS-S or if it represents the ITS-S network. The proposed definition uses the former concept.

EDITOR’S NOTE 2 The proposed definition assumes that an ITS-S does not cease being an ITS-S simply because it is disconnected from any network; therefore “in a communications network” is an invalid constraint.

EDITOR’S NOTE 3 As can be seen in a few examples, the phrase “network, comprised of applications…” is ambiguous as it is unclear if the ITS-S is comprised of the items or if the network is comprised of the items.

EDITOR’S NOTE 4 The proposed definition assumes that ITS-S are not required to have a wireless network interface. It is my understanding that any ITS device that conformed to the ITS-S architecture (which is almost a null requirement) and provides a BSMD (which is ill-defined) would be considered an ITS-S. Thus, at least at present, I see the term ITS-S to really mean any ITS physical entity that can claim to be secure (and it would be just a claim since there are no measurable requirements). And in fact, some definitions even make the BSMD optional – though I think this is wildly unwise.

EDITOR’S NOTE 5 Logically, an ITS-S seldom (if ever) provides ITS services because an ITS-S is a component of an ITS network; the ITS service is provided by the ITS as a whole. The ITS-S logically should provide ITS-S services, or more specifically, the ITS-S is comprised of the ITS-S architectural components (i.e., applications entity, facilities layer, etc), which all provide ITS-S services.

3.4.4.3

Centre

3.4.4.4

ITS roadway equipment

3.4.4.5

roadside equipment

3.4.4.6

on-board equipment

3.4.4.7

ITS unit

3.4.4.8

ITS-S unit

3.4.4.9

ITS-S communication unit

3.4.4.10

ITS Roadway unit

3.4.4.11

Roadside unit

3.4.4.12

Roadside communication unit

3.4.4.13

On-board unit

## Data concepts

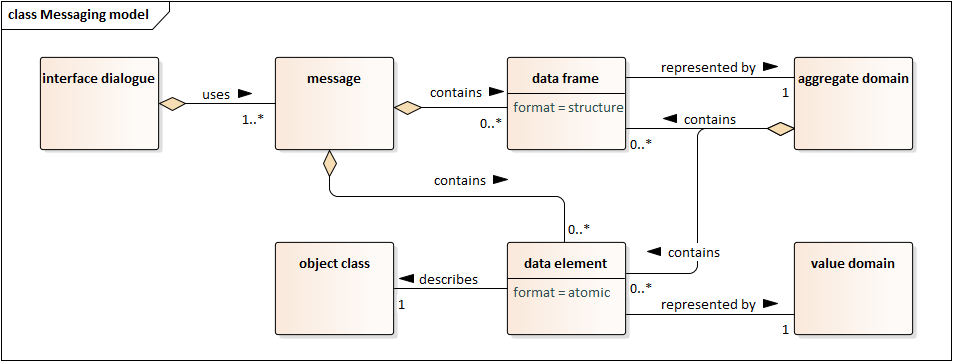


Figure 12: Messaging model

3.5.1

data concept

item that may be stored in a data dictionary that refers to an abstraction or thing in the natural world that can be identified with explicit boundaries and meaning and whose properties and behaviour all follow the same rules

Note 1 to entry: Data concepts can be classified into the following types: object class, value domain, data element, aggregate domain, data frame, message, interface dialogue, dictionary document, or module

[ISO 14817-1]

3.5.2

interface dialogue

dialogue

**data concept** that defines bi-directional communication sequence between two parties in accordance with predetermined protocols

[ISO 14817-1]

3.5.3

message

**data concept** that is a grouping of **data elements**, **data frames**, or **data elements** and **data frames** that is used to convey a complete set of information

[ISO 14817-1]

3.5.4

data frame

**data concept** represented by a specific **aggregate domain** and that describes information of interest through a useful grouping of more atomic properties about one or more **object classes**.

Note 1 to entry: The grouping may be a set, sequence, or a choice.

[ISO 14817-1]

3.5.5

aggregate domain

**data concept** that defines a grouping of **data elements** and/or **data frames**

[ISO 14817-1]

3.5.6

data element

**data concept** represented by a specific **value domain** and that describes a single atomic property about an **object class**.

Note 1 to entry: A data element is composed of an object class, a property of the represented object class and a value domain.

[ISO 14817-1]

3.5.7

value domain

**data concept** that defines a set of permissible values

[ISO 14817-1]

3.5.8

object class

description of a set of objects that share the same properties, relationships, and semantics

Note 1 to entry: Adapted from ISO/IEC 11179-1:2013; an object class is conceptually similar to an ISO/IEC 11179 object, but it does not include operations or methods and ISO/IEC 11179 "attributes" are called "properties" in this International Standard.

## Location model

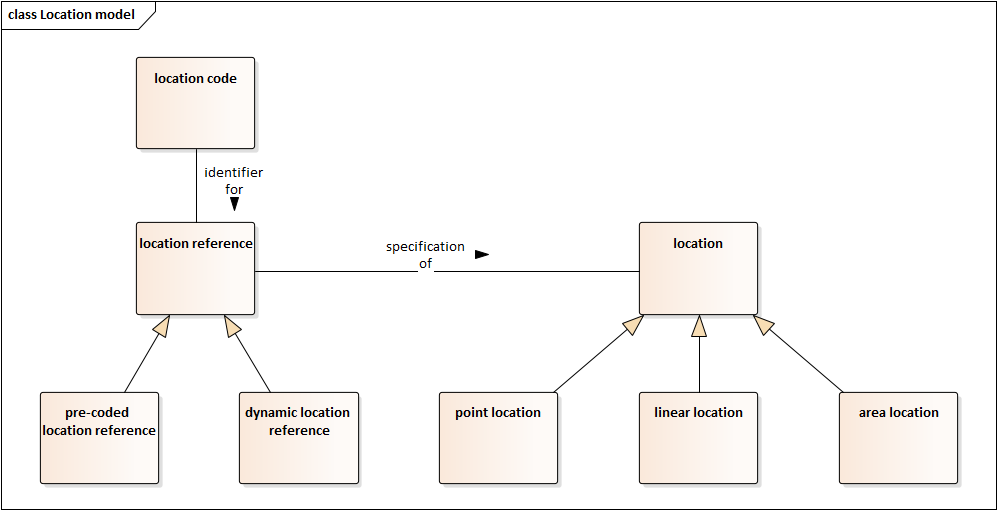


Figure 13: Location model

3.6.1

location

position on the surface of the earth

3.6.2

point location

**location** that has a zero-dimensional character

[ISO 17572-1:2015(en), 2.1.33]

3.6.3

linear location

**location** that has a one-dimensional character

[ISO 17572-1:2015(en), 2.1.19]

3.6.4

area location

**location** that has a two-dimensional character

3.6.5

location reference

specification of a **location** according to a specific set of rules

3.6.6

location code

identifier assigned to a location reference

Note 1 to entry: A location code is often a numeric or alphanumeric identifier, but could be a textual name, symbol, or any other type of identifier.

3.6.7

pre-coded location reference

location reference using a unique identifier that is agreed upon in both sender and receiver system to select a location from a set of pre-coded locations

[ISO 17572-1:2015(en), 2.1.35]

3.6.8

dynamic location reference

location reference generated on-the-fly based on geographic properties in a digital map database

[ISO 17572-1:2015(en), 2.1.11]

## Vehicle Types

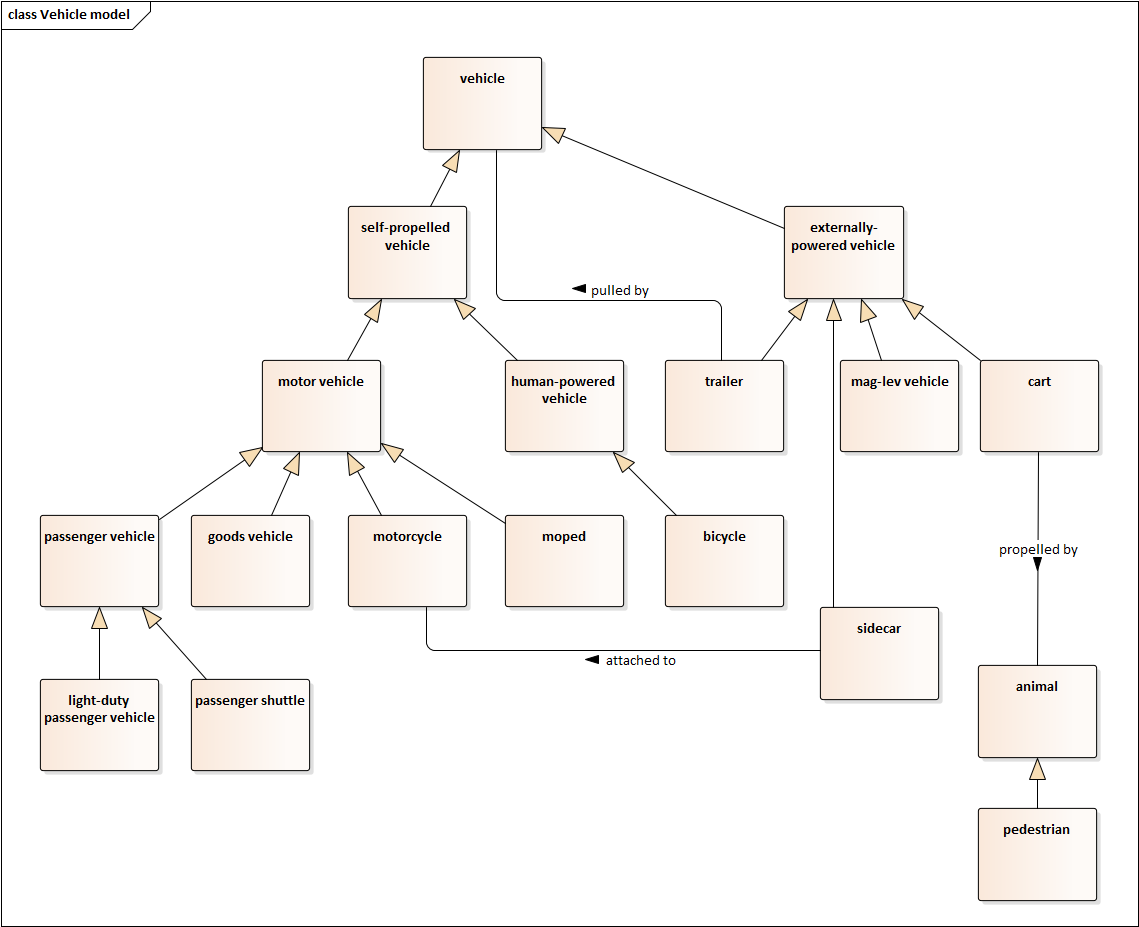


Figure 14: Vehicle model

3.6.1

vehicle

entity used to transport people or physical goods

3.6.2

self-propelled vehicle

vehicle propelled by an on-board source

Note 1 to entry: Contrast to externally propelled vehicle.

3.6.3

motor vehicle

vehicle propelled by an on-board motor, having at least two wheels, and designed to be used on a road network

3.6.4

passenger vehicle

motor vehicle with four or more wheels designed primarily for the carriage of (a) person(s)

3.6.5

light-duty passenger vehicle

passenger vehicle with not more than eight seating positions in addition to the driver’s seating position (if any) and not designed for standing passengers

3.6.6

passenger shuttle

passenger vehicle designed for the carriage of more than eight passengers, whether seated or standing, in addition to the driver (if any)

3.6.7

goods vehicle

motor vehicle with four or more wheels designed primarily for the carriage of goods per UNECE WP.29/1045

Note 1 to entry: UNECE WP.29/1045 provides a formula to disambiguate between passenger vehicles (Category 1 vehicles) and goods vehicles (Category 2 vehicles).

3.6.8

motorcycle

motor vehicle with 2 or 3 wheels and a maximum design speed exceeding 50 km/h

3.6.9

moped

motor vehicle with 2 or 3 wheels and a maximum design speed not exceeding 50 km/h

3.6.10

human-powered vehicle

vehicle propelled by the muscular energy of one or more on-board person(s)

3.6.11

bicycle

human-powered vehicle with two wheels

3.6.12

externally-propelled vehicle

vehicle propelled by an external entity

Note 1 to entry: The external entity is often a self-propelled vehicle, but may be any external source, including and not limited to an animal (e.g., an ox pulling a cart), a cable (e.g., San Francisco Trolley cars), or magnets (e.g., a maglev train).

3.6.13

trailer

externally-propelled vehicle pulled by another vehicle

EDITOR’S NOTE: Should this include self-propelled vehicles being towed?

3.6.14

sidecar

externally-propelled vehicle attached to a motorcycle

3.6.15

cart

externally-propelled vehicle propelled by an animal (which might be a person)

## Specifications

3.8.1

scenario

<use case> a description of the sequence of events from the user’s perspective to perform a task in a specified context

[ISO/IEC 25062:2006(en), A.17]

3.8.2

use case

description of the behavioural requirements of a system and its interaction with a user

[ISO/IEC/IEEE 26515:2011(en), 4.15]

1. (informative)  
     
   Purpose and methodology for ITS vocabulary
   1. Purpose

ISO/TC 204 and other ITS standards development organisations have produced a large number of published documents, each defining its own terms. In most cases, identical terms in different documents describe the same concept, but often with slight variations in the text of the definition.

The ISO 14812 project was initiated to:

1. Promote a more consistent set of terminology and definitions that can be used throughout ITS discussions and thereby enable better understanding,
2. Improve the quality of the definitions by conforming to the terminological principles defined in ISO 704:2009, and
3. Graphically represent relationships between concepts to better promote understanding of concepts as defined in ISO 24156-1:2014.

ISO/TC 204 does not enforce consistency or require revisions to definitions in existing standards. The authority for the appropriate terminology and definitions in each standard is the Working Group (WG) producing the source standard. Revision of definitions in source standards is accomplished only through approval of subsequent revisions of the standards, addenda, or corrigenda as produced by the appropriate WG.

In keeping with established resolutions and policy, editors of new and updated ISO/TC 204 standards are to consult ISO 14812 in drafting new and revised terms and definitions. Editors either refer to the terms as defined in the vocabulary or recommend new definitions and terms as required.

* 1. Methodology

Developing a set of harmonized definitions for terms already defined in existing standards can be a challenge. Even if there is general agreement on the underlying concept being described, it may be difficult for multiple groups to agree on a single definition unless everyone agrees to a clear set of principles at the start of the effort.

Luckily, terminology experts within ISO have standardized the principles that should be followed when developing a vocabulary for ISO standards. These are defined in ISO 704:2009 *Terminology work – Principles and methods*. Interested parties should become familiar with this document, but a summary of the suggested process is provided below.

1. Select the field for which a vocabulary is to be defined.

In the case of this document, the field is Intelligent transport systems.

1. Analyse the intension and extension of each concept

In other words, analyse the characteristics that jointly distinguish each concept from other concepts (i.e., the intension) and the set of objects that are represented by the concept (i.e., the extension). By considering these both simultaneously, one can refine the characteristics that the extension objects share in common.

EXAMPLE A “moped” has the following characteristics (intension):

* type of entity
* used to transport a person and optionally goods
* designed to be used on a road network
* propelled by an on-board motor
* has a maximum design speed not exceeding 50 km/h
* has 2 or 3 wheels

EXAMPLE The following images provide sample extensions of “moped”.

*From left to right: Swedish moped class I[15], Norsjö Shopper 3-wheel moped red[16], and Moped in Myrtos[17]*

Figure 15: Three example mopeds

1. Determine the relationships to other concepts

Identify how each concept relates to other concepts within the domain. Typical relationships include:

* Generic relations (i.e., when the characteristics of one concept includes all of the characteristics of another concept plus at least one additional characteristic)

EXAMPLE The “moped” concept has a generic relationship with the “vehicle” concept, where “vehicle” is the superordinate concept and “moped” is the subordinate concept.

* Partitive relations (i.e., when once concept forms a part of another concept)

EXAMPLE The “wheel” concept has a partitive relationship with the “moped” concept, where “moped” is the superordinate concept and “wheel” is the subordinate concept.

* Associative relations (i.e., when one concept has some other relationship to another concept)

EXAMPLE The “road network” concept has an associative relationship with the “moped” concept, where the “moped” is “designed to be used on” the “road network”.

1. Illustrate these relationships with a diagram

Once the relationships have been determined, they should be documented with the use of diagrams. While various diagrammatic techniques are allowed, this document uses UML to depict the relationships according to the conventions defined in ISO 24156-1:2014 *Graphic notations for concept modelling in terminology work and its relationship with UML – Part 1: Guidelines for using UML notation in terminology work*. The rules in this document are summarised in A.3.

EXAMPLE The following figure depicts the relationships among concepts according to the rules of ISO 24156-1. The example demonstrates how many of the characteristics of “moped” are inherited from more generalised concepts.

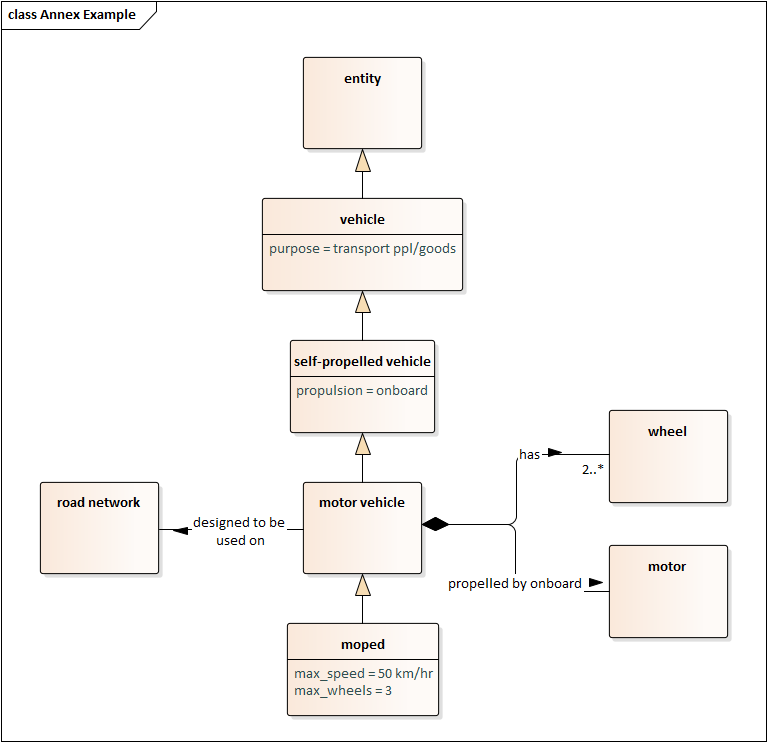


Figure 16: Example concept model diagram

*The figure indicates that a* ***moped*** *is a type of* ***motor vehicle*** *with the characteristics that it has a maximum speed not greater than 50 km/hr and no more than 3* ***wheels****. A* ***motor vehicle*** *is a type of* ***self-propelled vehicle*** *that is propelled by an onboard* ***motor****, has at least two* ***wheels****, and is designed to be used on the* ***road network****. A* ***self-propelled vehicle*** *is a type of* ***vehicle*** *that has an onboard propulsion source. A* ***vehicle*** *is a type of* ***entity*** *for the purpose of transporting people and/or goods.*

1. Formulate definitions based on relationships

The definition of each concept should be directly derived from the defined relationships. The definition should not restate characteristics inherited by other concepts; it should merely reference the more general term. The definition should not provide supplemental information as might appear in an encyclopaedic entry. If there is a need to provide such information, it should be provided in a note or an example immediately after the definition or provided as a part of the main body of the document. The intent of the definitions is that they should be worded such that they can be inserted in place of the term wherever the term occurs in the document (i.e., they are not complete sentences and they do not start with an article (e.g., they do not start with “a” or “an”)).

EXAMPLE motor vehicle with 2 or 3 wheels and a maximum design speed not exceeding 50 km/h

1. Assign a designation to each concept

In other words, the term is only given its final name (i.e., designation) once the relationships and definitions have been determined. In practice, every concept is given a name as soon as we begin to formulate it, but this name should be considered temporary until we have completed the analysis. In this case of this document, the vast majority of concepts defined have previously been included and named in other documents. As a general rule, terms already in use should retain their name, unless there are significant reasons to change. For example, if an existing term is confusing with another concept or if different groups use different terms for the same concept, the community will have to weigh the benefits of retaining usage of the term versus changing to a more descriptive term.

EXAMPLE moped

* 1. Overview of the concept model diagrams

Concept models can be depicted using a variety of different notations. One of the most common notations to use is the Unified Modelling Language (UML). UML is a graphical notation originally standardised by the Object Management Group in 1997 and later adopted by ISO in 2005 as ISO 19501. It was designed primarily for describing software systems, but it was also found to be useful in documenting concept models. In 2008, ISO published ISO 24156 that defined how to use a subset of UML to represent the results of a terminological concept analysis. This standard was last updated by ISO 24156-1 in 2014.

The primary aspect of UML used in modeling concepts is the UML class diagram. Each rectangle on a diagram represents a distinct concept and is given a name. Typically, UML requires the name to be a single word (i.e., no spaces) with an initial uppercase character; but in concept modelling, the names should be the term names (i.e., typically lowercase with spaces between words).

EXAMPLE Figure 15 depicts 8 concepts: entity, vehicle, self-propelled vehicle, motor vehicle, road network, wheel, motor, and moped.

The rectangles may be shown with a single compartment or divided into two compartments by a horizontal line. If only one compartment is present, it will depict the name of the concept. If two compartments are present, the upper compartment depicts the name of the concept and the lower compartment describes the characteristics of the concept by setting named attributes to specific values.

EXAMPLE In Figure 15, the concept named “entity” has a single compartment.

EXAMPLE In Figure 15, the concept named “moped” has two compartments with the lower compartment showing two attributes: max\_speed and max\_wheels.

Concept relationships are shown by lines connecting the concepts. Lines with a filled-in arrowhead indicate that the concept without the arrowhead is a specialized subordinate concept to the superordinate concept near the arrowhead. In this case, all examples (extensions) of the subordinate concept are also included in the extension of the superordinate concept.

EXAMPLE In Figure 15, a “moped” is defined to be a subordinate concept to the more general “motor vehicle” concept. This means that all mopeds are also motor vehicles and the characteristics of a motor vehicle also applies to a moped.

Lines with a diamond on the end indicate a partitive relationship where the concept near the diamond is a composite or aggregation and the concept on the other end is a part of the larger concept.

EXAMPLE In Figure 15, motor vehicle has a partitive relationship to both wheel and motor. In each case, the motor vehicle is the composite/aggregate and the wheel and motor are the components of the larger whole.

Lines without a diamond or arrowhead indicate associative relationships.

EXAMPLE In Figure 15, motor vehicle has a associative relationship to road network.

Associative relationships in concept models should (and partitive relationships may) have a name that describe the relationship further. The name is often adorned with a filled-in triangle to indicate the directionality of the named relation.

EXAMPLE In Figure 15, the associative relationship between “motor vehicle” and “road network” has the name “designed to be used on”. The name is adorned with a filled-in triangle to indicate that the motor vehicle is designed to be used on the road network (i.e., the road network is not “designed to be used on” the motor vehicle).

Finally, the ends of partitive and associative relationships can be adorned with numbers to indicate any required numerical constraints.

EXAMPLE In Figure 15, a motor vehicle is required to have 2 or more (i.e., “2..\*”) wheels.

* 1. Concept models and data models
     1. General principles

The use of UML to model concepts in terminological work is very similar to the task of using UML to define data concepts in systems engineering. In fact, it only makes sense that the resulting models should be entirely consistent with one another, even though the specifics will vary.

Terminological work is intended to define the meaning of terms in natural (human) language. Data concepts are intended to define the meaning of data when exchanged between computer systems – or really, between the programmers of the computer systems. In general, it is desirable for each concept to have a single designation (i.e., name) and definition used within both terminology (human terms) and systems engineering (computer terms).

* + 1. Practical considerations

In practice, differences between the models are often required, but care should be taken to ensure that the models are entirely consistent with each other. Differences might include:

* Variations to conform to specific language rules,

EXAMPLE A data model might require class names to use UpperCamelCase. As such, the term “vehicle” becomes the data concept “Vehicle”, the term “motor vehicle” becomes the data concept “MotorVehicle”, etc.

* Variations to improve systems engineering design efficiencies, and

EXAMPLE In order to promote efficiency, it might be advantageous to develop a data model without classes for “Entity”, “Vehicle”, and “SelfPropelledVehicle” as long as all of the associated functionality is implemented in the “MotorVehicle” class.

EXAMPLE In the above example, it is strongly recommended that the implemented class be called “MotorVehicle” rather than “Vehicle” to promote consistent terminology and understanding as well as providing a data model that can be maintained in a consistent fashion over time. If the class were to be named “Vehicle”, it may confuse some users and result in inappropriate usage of the class.

* Variations to represent data elements versus classes. Within terminology, all concepts are shown as rectangular boxes; a literal translation to a data model would mean each term would be defined as its own class with a value field – this would be a highly inefficient way to implement code. Within data models, the rectangular boxes depict classes and the second compartment is used to define attributes, each of which can be assigned a value at runtime. The class and all of its attributes should have definitions that are consistent with the defined concept model.

EXAMPLE A data model might include a “numberOfWheels” attribute for the “MotorVehicle” class rather than associating the “MotorVehicle” with multiple “Wheel” objects.

EXAMPLE A data model might include a “motorFuelType” attribute for the “MotorVehicle” class rather than associating the “MotorVehicle” class with a “Motor” class and then associating the “Motor” class with a “MotorFuelType” class which would then have a value.

* + 1. Additional specifications

Within systems engineering, a data concept must have additional specifications, which should be defined in separate meta-data fields. For example, ISO 14817-1 defines a number of meta-attributes that should be defined for each type of data concept. The meta-attributes include a “descriptive name” as well as a “definition”. Within the limitations as described above, these should be consistent with the “designation” and “definitions” defined in the concept model. However, ISO 14817-1 defines an array of other meta-attributes, such as “data concept version”, “data type”, “format”, etc. These provide additional details that are often required or useful for implementing systems but are less important for terminology work. In fact, within data models, the same base concept might be implemented with multiple variants. In such cases, the name of the data concept needs to indicate the variance.

EXAMPLE The term “vehicle speed” might be defined to be “the instantaneous **speed** of a **vehicle**”, where “speed” and “vehicle” are defined separately. Within a data model, the concept “vehicle” would likely be portrayed as a “Vehicle” class and “vehicle speed” would be portrayed as the attribute “speed” of the “Vehicle” and would receive the name “Vehicle.speed” using ISO 14817-1 conventions. A data element goes one step further and defines how the data is presented. The preferred solution for such a data element is to use the same definition as used by the term (i.e., “the instantaneous **speed** of a **vehicle**”) and to use the other meta-attributes to further specify how the information is presented, such as:

* + Descriptive name: Vehicle.speed:measure-kph
  + Data type: INTEGER
  + Unit of measure: km/hr
  1. Maintaining consistency between terminological model and data model

Over time, the terminological concept models and data models will need to be maintained. As one is changed, the other will need to be updated as well. While the need is known, the exact process has not yet been defined, but will likely become the subject of another ISO/TC 204 standard or technical report.

1. (informative)  
     
   Usage examples
   1. Incorporating terms into other documents
      1. Including terms contained in ISO 14812 (this document)

Other ISO/TC 204 documents are strongly encouraged to adopt the terms and definitions defined within this document. Ensuring that all ISO/TC 204 standards use a consistent terminology will promote understanding of the information contained within standards and improve interoperability of systems as the domains of different working groups merge into a common Internet of Things (IoT) environment. However, at present, this document, ISO 14812, is a technical report; therefore, standards are not allowed to make normative references to the definitions contained herein. Instead, other documents will need to duplicate the definitions contained within this document until such a time that this content is standardized.

The terms and definitions should be referenced in a manner that is consistent with the rules defined in the ISO Directives. In short, this means that ISO/TC 204 standards should copy the terms and definitions needed by their standard and paste them into a “Terms and definitions” section of their standard (typically Section 3). The terms should be renumbered to reflect a sequential order within the new standard and should cite this document, ISO 14812, as the source; if any changes are made, they should be noted within the reference.

EXAMPLE The following shows an example of how another document should reference the terms and definitions contained within this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at <http://www.iso.org/obp>

— IEC Electropedia: available at http://www.electropedia.org/

3.1

ITS application

**interoperability design** for an **ITS service**

[SOURCE: ISO/TR 14812:2019, 3.3.1.1]

3.2

ITS application specification

one or more documents that fully detail the **ITS application**

Note 1 to entry: This is a sample note.

[SOURCE: ISO/TR 14812:2019, 3.3.1.2, modified – Note 1 to entry has been added]

3.3

ITS implementation

integration of each **system element** and **terminator** necessary to implement one or more **ITS applications**

Note 1 to entry: An ITS application typically requires multiple components (e.g., a ITS-S acting as a user and another ITS-S acting as a provider). An ITS implementation includes a sample of each component necessary for the service but often does not represent a complete deployment.

[SOURCE: ISO/TR 14812:2019, 3.3.1.2, modified – Note 2 to entry has been removed]

If the ISO 14812 definition cites an external source, the importing document should cite the same external source.

In general, it is recommended that other documents do not copy the UML diagrams contained within this document. It is believed that adding these UML diagrams to other standards would likely detract from the content of those standards. Those interested in the full model of relationships can always reference to ISO 14812 directly.

* + 1. Defining terms not contained in ISO 14812

This document is intended to provide the definitions of terms as used within intelligent transport system discussions, however, this scope is limited by:

1. The fact that this document is the initial version of a document that reflects an on-going work-in-progress. At the initiation of this project, ISO/TC 204 had already defined roughly 2,000 terms, many with inconsistencies. Rather than trying to reach TC-wide consensus on all of these terms for the first version, it was determined that it would be better to focus on the most widely used terms.
2. Some of the terms defined might be specific to the scope of a single standard and defining them within a TC-wide vocabulary may not be useful.
3. As the industry evolves, new terms are likely to emerge

The existence of this document should not be construed to prohibit the definition of additional terms; every document should define the terms that it needs to unambiguously express its intent. However, when developing such terms, the respective working group is encouraged to work with ISO/TC 204/ WG 1 so that the definition can be developed in a consistent manner with all of the other terms.

* + 1. Updating an existing standard
       1. General

Most existing standards already define their own terms. It is in the interest of the industry that all ITS standards use consistent definitions; as such those updating existing standards are encouraged to revise their documents to use terminology that is consistent with this document, ISO 14812. Each existing term will fall into at least one of the following categories as discussed below:

1. The existing term and definition are consistent with the ISO 14812 content
2. The existing term and definition are largely consistent with the ISO 14812 content, but some nuance is missing
3. The existing term is defined in ISO 14812, but describes a different concept
4. The existing concept is included within ISO 14812, but is assigned to a different term
5. Neither the existing term or concept are included within the current version of ISO 14812
   * + 1. Existing term and definition are consistent with ISO 14812

In a few cases, you may find that an existing term and definition are *identical* to what is contained in ISO 14812; however, this will likely be exceptional as the ISO 14812 definitions have been developed according to the rules defined in ISO 704 and therefore are based on the conceptual terminology model that was defined as a part of the development of ISO 14812.

However, just because the exact text of the definition is different does not mean that the term is describing a different concept. In general, the terms defined in ISO 14812 are intended to be completely consistent with their meanings prior to the development of ISO 14812; the new definitions were developed to improve comprehension by creating an ITS-wide terminological model that relates terms to one another, as recommended by terminological experts. Thus, in most cases, it is expected that existing terms will be consistent with the definitions in ISO 14812.

In this case, the existing standard should simply be updated with the definition contained within ISO 14812.

* + - 1. ISO 14812 is missing a nuance (see Annex C)

Any time a definition changes, one has to be concerned about introducing subtle differences in meaning. If a working group believes that ISO 14812 has introduced such a change, the working group should work with ISO/TC 204/WG 1 according to the procedures in Annex C to resolve the issue. The intent of the resolution process is to develop a consistent vocabulary for the whole of the intelligent transport system domain while also ensuring the necessary precision needed by the various standards.

* + - 1. Same term used to describe a different concept

In some cases, the same term may be used to describe different concepts between ISO 14812 and an existing document. In most such cases, the development of the ISO 14812 definition was likely done in coordination with the working group responsible for maintaining the existing standard, in which case the resolution for the inconsistency should already be known (and might be achieved by ISO 14812 assigning a different term to the existing concept as described by B.1.3.5). Otherwise, the editor will need to determine if it is appropriate to continue using the existing term or if a change might be more appropriate.

In general, it is desirable for each term to only have a single meaning; but in practice, multiple meanings for the same term is not uncommon. Multiple meanings are acceptable as long as the context provides adequate disambiguation among the multiple meanings and each meaning is clearly labeled to be applicable to a specific domain. Where the context might be unclear, the domain name can be used as a modifier to clarify the intent.

EXAMPLE The physical object representing a **user** of an **ITS service** is called a **terminator**.

The above sentence is accurate and sufficiently unambiguous based on the definitions included in this document. If the word “ITS” was removed in front of “service”, the sentence (at least when read by itself) would imply a generic service and the sentence would no longer necessarily be true as the word “terminator” is only applicable to the physical view of the system architecture. In theory, one could add the term “ITS” in front of the term “user” as well (e.g., “The physical object representing an ITS user of an ITS service is called a terminator.”), but this would make the sentence rather redundant and would detract from the information that the sentence is trying to convey.

However, multiple meanings should always be avoided for terms where similar concepts are likely to be confused with one another. In these cases, the working group responsible for maintaining the existing standard will need to work with ISO/TC 204/ WG1 to determine the appropriate resolution.

The resolution will often require the editor of the document to perform a search for every occurrence of the existing term and to make edits as needed so that the updated standard is consistent with the TC 204 vocabulary.

* + - 1. Same concept assigned to a different term

In some cases, ISO 14812 has chosen to use a different term to describe concepts already defined. This might be done to resolve existing differences between existing standards or because the terms themselves are not fully descriptive of the concept being defined.

The resolution will often require the editor of the document to perform a search for every occurrence of the existing term and to make edits as needed so that the updated standard is consistent with the TC 204 vocabulary.

* + - 1. Neither concept nor term defined in ISO 14812

As mentioned above, the current version of ISO 14812 does not claim to define all terms applicable to ITS. Any document is allowed to extend its vocabulary as needed by defining its own terms; however, ISO/TC 204/ WG 1 encourages working groups defining new terms to work with them to ensure the terms fit into the broader ITS terminology concept model.

* 1. Using terms within the text of the document
     1. General

When using a defined term, editors are encouraged to highlight the fact that the term has a formal definition by presenting the term in boldface. This convention allows users to recognize that the term has a formal definition.

ISO Directives also allow terms to be followed by the clause number where the term is defined. This is mainly useful in documents where terms are not defined in alphabetical order and where no index is provided. However, extensive cross-references in a document complicate document maintenance. Either the cross-references have to be maintained manually, which quickly becomes challenging, or the cross-references use special codes, which can create other problems, especially in large documents. Thus, the preferred method withi TC 204 is to simply highlight defined terms using **boldface**.

* + 1. Disambiguation

Within ITS, some terms are used as the designation of more than one concept (e.g., “service”). When this occurs, each concept is provided with its own terminological entry and the definition is preceded by the domain to which the definition applies within brackets.

A document that needs to precisely reference such a concept when the domain is not obvious can disambiguate among multiple concepts by adding the domain as a modifying term.

EXAMPLE If the term “service” is too ambiguous, it can be clarified by adding a domain, which produces a term such as “ITS service”.

* + 1. Data dictionaries

Documents may reference both terms and data concepts within the text of the document. The definition of both should be consistent with one another as described within A.4; however, the data concept will be more specific and often include representational form information. When wishing to reference a data concept, editors are encouraged to use a fixed with font whereas terminology should be shown in the normal font with a **boldface**.

1. (informative)  
     
   Procedures for maintaining the vocabulary
   1. Scope of maintenance procedures

This Annex describes how additions, revisions, and withdrawals of terms and definitions in ISO TR 14812, *Intelligent transport systems – Vocabulary* will be managed. Extensive or complex revisions of the entire document are not necessarily within scope of this Annex.

* 1. Maintenance environment

All definitions are to be developed according to the principles of ISO 704. Definitions of general concepts that apply beyond the scope of ISO/TC 204 might import definitions from other well-established sources, typically ISO standards from other technical committees. Any such definitions explicitly cite the source of the definition.

All other terms are defined according to their relationships to other defined terms as documented within a concept model. The concept model is documented in UML according to the conventions contained in ISO 24156.

The model is maintained using three key software products:

* Enterprise Architect[[1]](#footnote-2) (EA) by Sparx,
* Github (https://github.com/isotc204/14812/projects), and
* LemonTree by LieberLieber.

The UML file is created and stored within an EA file. Each subsection within this document is generally represented by a separate package within the EA file. Definitions, notes, examples, and all other information for each term is stored within the “notes” field of the concept within the EA file.

The EA file is version controlled using a Github. Branches and forks are created to manage enhancements to the current file while allowing easy access to the approved version. When an expert or group proposes changes, reviewers can use LemonTree[[2]](#footnote-3) to identify differences between the proposal and the base version of the model (or any other version). Users with appropriate rights are able to merge selected changes into a new master file. The exact maintenance process is described below.

* 1. Maintenance process

The general maintenance process is depicted in Figure 16. The process starts (Step 1) with the current version of the EA file as stored in the development branch of the ISO/TC 204 Github site.

Individuals or groups (‘submitters’) wishing to suggest changes to the vocabulary should fork the current version of the development branch onto their own Github account. A fork copies the EA file onto the submitter’s own site where the submitter can edit the file without affecting the actual development copy. Figure 16 shows the creation of two such forks (Steps 2 and 3) where two different submitters make changes in parallel (Steps 4 and 5).

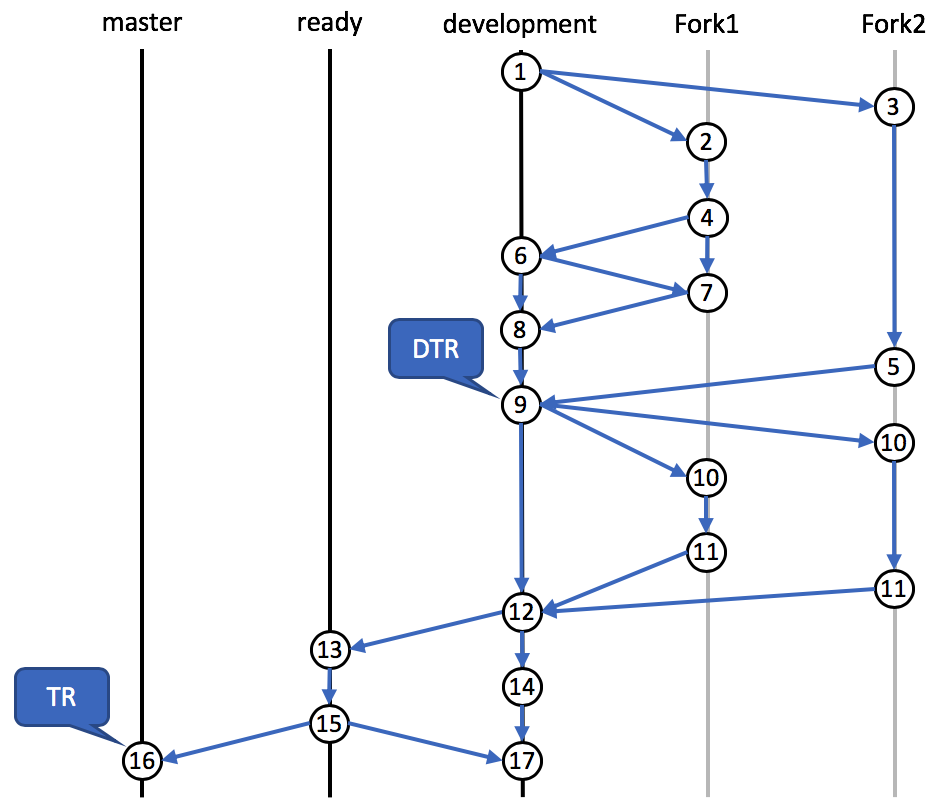


Figure 17: Maintenance process

Each fork may include an unlimited number of proposals to:

* Add one or more new terms (with definitions),
* Revise one or more terms and/or definitions, and/or
* Withdraw one or more obsolete terms and definitions (e.g., due to withdrawal of a standard).

When a submitter is done (or nearly done) with proposed edits, the submitter will use Github to issue a ‘pull request’ to the 14812 Vetting Team. The Vetting Team then uses LemonTree to identify differences between the current development branch and the proposal. LemonTree allows the Vetting Team to approve or reject each incremental change and thereby create a new development version (Steps 6 and 9). If the Vetting Team is unable to complete the review of a submission at a single meeting, the submitter is still able to update their proposal (Step 7) based on on-going discussions and the new file can be used in the next review by the Vetting Team to develop a new version of the development branch (Step 8).

Once the Vetting Team believes the development version is ready to be distributed as a Draft Technical Report (DTR), the Github version is tagged as such (Step 9) and the documentation is generated from the EA file using the scripts developed and maintained on the github site. Once generated, the documentation is formatted and submitted to ISO for a DTR ballot.

When sent to DTR ballot, outside entities are encouraged to review the file and they may make comments either by the process described above (Steps 10 and 11) or may make comments through the standard ISO procedure. The Vetting Team will consider this input and produce a new development version (Step 12). Once the file is considered final by the Vetting Team, the file will be stored in a “ready” branch (Step 13) and the generated documentation will be submitted to ISO for publication. This allows continued development of the document within the development branch (Step 14) while the ISO TR is finalized for publication. If the editorial process requires any changes to the EA file, those changes are made in the ready branch (Step 15). Once the TR is published, the ready branch version is merged back into the master branch (Step 16) and the development branch (Step 17).

The process then starts over (i.e., Step 17 becomes the next Step 1).

* 1. Formation of Validation Team

The ISO 14812 Validation Team is composed of interested experts of ISO/TC 204/ WG 1.

Bibliography

[1] ISO 704:2009, *Terminology work — Principles and methods*

[2] ISO 10241‑1:2011, *Terminological entries in standards — Part 1: General requirements and examples of presentation*

[3] ISO 14817-1

[4] ISO/TR 17185-1, *Intelligent transport systems — Public transport user information — Part 1: Standards framework for public information systems*

[5] ISO 17572-1:2015, *Intelligent transport systems (ITS) — Location referencing for geographic databases — Part 1: General requirements and conceptual model*

[6] ISO 21217:2014, *Intelligent transport systems — Communications access for land mobiles (CALM) — Architecture*

[7] ISO 24156‑1:2014, *Graphic notations for concept modelling in terminology work and its relationship with UML — Part 1: Guidelines for using UML notation in terminology work*

[8] ISO 24774:2010, *Systems and software engineering -- Life cycle management -- Guidelines for process description*

[9] ISO 25060:2010, *Systems and software engineering -- Systems and software product Quality Requirements and Evaluation (SQuaRE) -- Common Industry Format (CIF) for usability: General framework for usability-related information*

[10] ISO 25062:2006, *Software engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Common Industry Format (CIF) for usability test reports*

[11] ISO/IEC/IEEE 26515:2011, *Systems and software engineering -- Developing user documentation in an agile environment*

[12] ISO 42010:2011, *Systems and software engineering — Architecture description*

[13] UNECE TRANS/WP.29/1045, *Concerning the common definitions of vehicle categories, masses and dimensions (S.R. 1)*. 15 Sept 2005. Available from: <https://www.unece.org/fileadmin/DAM/trans/doc/2005/wp29/TRANS-WP29-1045e.pdf>

[14] Wikipedia the free encyclopedia – definition of reference architecture [Internet]. [cited 2018 Jul 17]; Available from: <http://en.wikipedia.org/wiki/reference_architecture>

[15] Wikimedia commons – File:Swedish moped class I.jpg [Internet]; photo taken by Bluescan sv.wiki on 2010 Oct 23 and made available under Creative Commons Attribution-Share Alike 3.0 Unported license. [cited 2018 Aug 3]; Available from: https://commons.wikimedia.org/wiki/File:Swedish\_moped\_class\_I.jpg

[16] Wikimedia commons – File:NorsjoShopperRed01.JPG [Internet]; photo taken by Fb35523 on 2011 May 11 and made available under GNU Free Documentation License. [cited 2018 Aug 3]; Available from: https://commons.wikimedia.org/wiki/File:NorsjoShopperRed01.JPG

[17] Wikimedia commons – File:Moped in Myrtos.JPG [Internet]; photo taken by [Mikel Holland-Moritz](https://commons.wikimedia.org/w/index.php?title=User:Mikelito.hm&action=edit&redlink=1) on 2015 Jul 16 and made available under Creative Commons Attribution-Share Alike 4.0 International license. [cited 2018 Aug 3]; Available from: https://commons.wikimedia.org/wiki/File:Moped\_in\_Myrtos.JPG

Index

A

aggregate domain 19

application

ITS 11

application specification

ITS 11

architecture 13

architecture description 13

architecture framework 13

architecture model 14

architecture view 14

architecture viewpoint 14

area location 21

B

bicycle 23

bounded, secure, managed domain 10

C

cart 23

C-ITS 7

concern 14

consumer 8

cooperative-ITS 7

D

data concept 19

data element 20

data frame 19

deployment

ITS 11

deployment architecture 15

dialogue 19

dynamic location reference 21

E

element 14

entity 8

externally-propelled vehicle 23

F

functional object 16

G

goods vehicle 23

H

human-powered vehicle 23

I

implementation

ITS 11

information flow 16

information triple 16

intelligent transport system 7

interface dialogue 19

interoperability design 11

ITS 7

ITS application 11

ITS application specification 11

ITS architecture 15

ITS component 16

ITS deployment 11

ITS implementation 11

ITS service 9

ITS service provider 9

ITS station 17

ITS user 9

ITS user need 9

ITS-S application process 12

ITS-S application process implementation 12

ITS-S application process installation 12

ITS-S application process instance 12

ITS-S architecture 15

ITS-S user need 10

L

light-duty passenger vehicle 23

linear location 21

location 20

location code 21

location reference 21

M

message 19

model kind 14

moped 23

motor vehicle 22

motorcycle 23

N

need 8

O

object class 20

P

passenger shuttle 23

passenger vehicle 22

physical object 16

planning architecture 15

point location 20

pre-coded location reference 21

R

reference architecture 15

S

scenario 24

self-propelled vehicle 22

service 8

ITS 9

ITS-S 10

service provider 8

ITS 9

ITS-S 10

sidecar 23

stakeholder 14

surface transport system 6

system 6

T

technology system 6

terminator 16

trailer 23

transport system 6

U

use case 24

user 8

ITS 9

ITS-S 10

user need

ITS 9

ITS-S 10

V

value domain 20

vehicle 22

1. Enterprise Architect is a software tool developed by Sparx, who has provided free licenses to experts of ISO/TC 204 for the development of ISO standards. [↑](#footnote-ref-2)
2. LemonTree is a software tool developed by LieberLieber, who has provided free licenses to experts of ISO/TC 204 for the development of ISO standards. [↑](#footnote-ref-3)