INTERNATIONAL STANDARD

ISO 14813-5

First edition
2010-07-01

Intelligent transport systems — Reference model architecture(s) for the ITS sector —

Part 5: Requirements for architecture description in ITS standards

Systèmes intelligents de transport (ITS) — Architecture(s) de modèle de référence pour le secteur ITS —

Partie 5: Exigences pour la description d'architecture dans les normes ITS

Reference number
ISO 14813-5:2010(E)

Contents

Page

Foreword ... iv
Introduction ... v
1 Scope ... 1
2 Conformance ... 1
3 Normative references ... 1
4 Terms and definitions ... 2
5 Requirements ... 2
5.1 General requirements ... 2
5.1.1 Architecture description ... 2
5.1.2 Service description ... 2
5.1.3 Architecture description elements ... 2
5.2 Further guidance ... 3
5.3 ITS architecture elements ... 3
5.3.1 General requirements ... 3
5.3.2 Conceptual architecture ... 4
5.3.3 Logical architecture ... 5
5.3.4 Physical architecture (optional) ... 5
5.3.5 Communications architecture (optional) ... 5
5.3.6 Organisational architecture (optional) ... 5
5.4 Object-oriented architecture ... 5
5.4.1 General ... 5
5.4.2 Specific requirements for object-oriented description ... 6
5.4.3 Relationship to ITS service domains, service groups and services ... 6
5.4.4 Control behaviour ... 6
5.4.5 Multiple viewpoints ... 6
5.5 Process-oriented architecture ... 6
5.5.1 General ... 6
5.5.2 Specific requirements for process-oriented methodology description ... 6
5.6 Application architecture/deployment (implementation) design ... 8
5.7 Layout of architecture description in ITS standards and other deliverables ... 8
5.7.1 Description method ... 8
5.7.2 Usage of terms ... 8
5.7.3 Plain language ... 9
5.7.4 Deployment design ... 9
5.7.5 Use of annexes ... 9
5.7.6 Relationships with other standards ... 9
Annex A (informative) Standards for specific architecture methodologies ... 10
Annex B (normative) Glossary of ITS architecture terms, abbreviated terms and numeric notation ... 11
Bibliography ... 30

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

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.

ISO 14813-5 was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

This first edition of ISO 14813-5 cancels and replaces ISO/TR 14813-5:1999, of which it constitutes a technical revision.

ISO 14813 consists of the following parts, under the general title *Intelligent transport systems — Reference model architecture(s) for the ITS sector*:

— *Part 1: ITS service domains, service groups and services*

— *Part 2: Core TICS reference architecture* [Technical Report]

— *Part 3: Example elaboration* [Technical Report]

— *Part 4: Reference model tutorial* [Technical Report]

— *Part 5: Requirements for architecture description in ITS standards*

— *Part 6: Data presentation in ASN.1*

Introduction

"Architecture" can be defined as "*Design; the way components fit together*"[1]. Architecture is implicit in any construction, be it of a physical entity (such as a building), an operational entity (such as a company or organisation), a system entity (such as a software system) or a business entity (such as a commercial business operation).

While it may be stated that every entity has an architecture, the particular architecture may be an explicit construction as a result of a deliberate design process or the implicit result of an unplanned series of events, or sometimes the combination of both.

In physical construction, it is generally recognised that a deliberate design process will produce a better and more efficient building that one where a group of individuals have collected whatever materials happened to be nearby in order to create a shelter.

Intelligent transport systems (ITS) are systems deployed in transportation environments to improve both the driving experience and the safety and security of drivers, passengers and pedestrians. ITS can also assist in the labour, energy, environmental and cost efficiency of transportation systems. It is a feature of most ITS that their architecture involves the collection, use and exchange of information/data within and between software systems which affect or control the behaviour of physical equipment, providing a service to the actors involved in, or interacting with, the transport sector.

In order to maximise the efficiency of co-existing ITS, to obtain compatibility and/or interoperability and to eliminate contention, the systems need to co-exist and operate within a known and supportive architectural framework.

The ITS sector is still emerging and developing and is still close to the start of its evolution and application. The technology is developing and changing rapidly and ITS services have generally to make provision not only for its interaction with other services, but with migration from one technology generation to later iterations.

This part of ISO 14813 is designed to ensure that, in order to obtain maximum interoperability, efficiency and migration capability, architecture is an explicit process in the development of, and specifications defined within, ITS standards.

"Architecture" is used in an informal manner to mean a variety of different concepts and, in formal architecture design, there are differing methodologies and opinions as to their suitability for use in ITS system and standards design. This has limited effective communication in the ITS sector by causing uncertainty as to the meaning of the word when it is used in one context or another. A second function of this part of ISO 14813 is to provide consistent terminology to be used in describing architectural aspects of ITS standards and provide a consistent form for ITS architecture description in standards in the ITS sector.

An ITS architecture is a framework for ITS deployments. It is a high-level description of the major elements and the interconnections among them. It provides the framework around which the interfaces, specifications and detailed ITS designs can be defined. An ITS architecture is not a product design or a detailed specification for physical deployment and is not specific to any one location. "Systems architecture" is perhaps the closest general term, but this is sometimes too specific to include the conceptual aspects included in the term "ITS Architecture" and often also implies a location-specific solution. The purpose of an ITS architecture is to maximise efficiency, interoperability and multimodality of multiple interacting ITS in a complex and developing sector.

---

1) *Interoperability Clearinghouse Glossary of Terms,* http://www.ichnet.org/glossary.htm)

This part of ISO 14813 does not give preference to any one methodology for architecture development and description. It requires only that the consideration of architecture be an explicit process that takes into account the interrelationships and interoperability of ITS, and that an architecture description be provided within ITS standards.

This part of ISO 14813 requires that the architecture aspects of ITS standards be described explicitly in each and every ITS standard, and that all such standards be related to the (one or more) ITS service domains, service groups and services set out in ISO 14813-1 that they are designed to enable or support.

Intelligent transport systems — Reference model architecture(s) for the ITS sector —

Part 5: Requirements for architecture description in ITS standards

1 Scope

This part of ISO 14813 gives requirements for the description and documentation of the architecture of intelligent transport systems (ITS) in standards dealing with ITS. It also defines the terms to be used when documenting or referencing aspects of architecture description in those standards (see Annex B).

Although the use of contemporary systems engineering practices is assumed by this part of ISO 14813, it does not define such practices.

NOTE Guidance on the use of the unified modelling language (UML) in ITS architectures can be found in ISO/TR 17452 and ISO/TR 24529. Guidance on the use of the process-orientated methodology in ITS architectures can be found in ISO/TR 26999.

2 Conformance

There are no specific conformance tests specified within or associated with this part of ISO 14183.

Developers of standards claiming conformance with this part of ISO 14183 are, however, required to describe the architecture of their system in their deliverables, or to make reference to other standards or publicly available documents that provide such description. The level of detail or the methodology used for such description is not specified and is left to the discretion of the standards developers.

Implementers of ITS cannot, of course, be required to make such provision, but are advised to do so in their plans and tender documents. This part of ISO 14813 is therefore also designed as a consistent reference for ITS system designers.

3 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 8824 (all parts), *Information technology — Abstract Syntax Notation One (ASN.1)*[2)]

ISO/IEC 8825 (all parts), *Information technology — ASN.1 encoding rules*[2)]

2) ASN.1 standards are divided into the abstract syntax notation one (ASN.1) specifications and the ASN.1 encoding rules. ISO/IEC 8824-1 to ISO/IEC 8824-4 and ISO/IEC 8825-1 to ISO/IEC 8825-4 correspond to ITU-T Recommendations X.680, X.681, X.682 and X.683, and X.690, X.691, X.692 and X.693, respectively. See http://www.itu.int/ITU-T/studygroups/com17/languages/.

ISO/IEC 9834-1, *Information technology — Open Systems Interconnection — Procedures for the operation of OSI Registration Authorities: General procedures and top arcs of the International Object Identifier tree — Part 1*

ISO 14813-1, *Intelligent transport systems — Reference model architecture(s) for the ITS sector — Part 1: ITS service domains, service groups and services*

ISO/TR 14813-6, *Intelligent transport systems — Reference model architecture(s) for the ITS sector — Part 6: Data presentation in ASN.1*

ISO/IEC 19501:2005, *Information technology — Open Distributed Processing — Unified Modeling Language (UML) Version 1.4.2*

4 Terms and definitions

For the purposes of this document, the terms and definitions given in Annex B and the following apply.

4.1
ITS architecture
non-specific system design for a family of functionally different intelligent transport systems (ITS), interconnected to operate in consort and harmony

NOTE 1 An ITS architecture can be described from different viewpoints, and from multiple viewpoints by conceptual, logical and/or physical representations.

NOTE 2 An ITS architecture is not specific to any single location.

5 Requirements

5.1 General requirements

5.1.1 Architecture description

All ITS standards shall provide an *architecture description*, by inclusion in the standard or by reference to other relevant standards. The architecture description shall include information detailing the vision and mission to be achieved by applying the standard, together with a description of the architectural aspects of the standard, detailed in one of the forms specified in this part of ISO 14813. Such information may appear in either the scope or requirements clauses, as considered appropriate by the authors.

5.1.2 Service description

All architecture descriptions shall either start with, or be clearly related to, one or more of the ITS service domains, service groups and services in accordance with ISO 14813-1.

5.1.3 Architecture description elements

It is important that all ITS standards be able to be compared for inter-relationships and, consequently, this part of ISO 14813 needs to be applied to the architecture description elements of all ITS standards.

The requirements for architecture description elements are as follows.

a) **Architecture scope**

The scope of the architecture shall be described by reference to the ITS service domains, service groups and services defined in ISO 14813-1.

In implementing an application based on an ITS standard, implementers will normally also need to ensure that they are designing consistently with their organisation's enterprise architecture. Enterprise architectures do not normally form part of ITS standards.

b) **ITS system descriptions/definition**

Where schematics are included, they shall either be simply understood high-level use-case diagrams or shall be expressed in the form of process or object models.

c) **Protocol descriptions/definition**

In most cases, protocol descriptions will not be required in an architecture standard. If protocol descriptions are required, they shall be written in a widely accepted and standardised formal description language (e.g. SDL, XML, ASN.1).

d) **Data description/definition**

Data concepts, where defined, and as required by ISO 14813-6, shall be described using ASN.1 in accordance with ISO 14813-6, ISO\IEC 8824 and, where appropriate, ISO\IEC 8825.

This requirement is aimed at maximising interoperability and reuse of data. Actual data may be defined and applied using other formats, such as SDL or XML, but shall be described as an SDL, XML module within a formal ASN.1 data definition. ISO 14813-6 provides examples of how to achieve this.

There is no requirement that architecture description be elaborated in levels of detail that require data definition, and indeed this will not normally be done. However, where data definition is made, it shall be done in a way that it is consistent with the requirements of ISO 14813-6.

Where a sector already uses an existing standard notation (e.g. EDIFACT, X12), such use is permissible so long as the message content, structure and transaction elements are clearly and separably defined. Usually, this will be by reference to another standard (e.g. EDIFACT Board standard). Where attribute limitations (e.g. range of numerical values) are appropriate within the ITS standard — such as where a time/size limited air interface transaction is involved — such attribute limitations shall be specified.

5.2 Further guidance

Further guidance and assistance in the description and elaboration of systems architecture is to be found in the other parts of ISO 14813, as well as in ISO/TR 17452, ISO 19501, ISO/TR 24529, ISO 24531 and ISO/TR 26999 (see the Bibliography).

5.3 ITS architecture elements

5.3.1 General requirements

Systems architecture identifies the major actors, use cases, interfaces and components and provides a basis for understanding all their inter-relationships and interactions.

All ITS standards shall provide a description of the aspects of their architecture, either by process-oriented or object-oriented analysis, with the depth of such description varying according to the relevance to the deliverable. Where a limited standard (e.g. one determining protocols alone) is prepared, a simple statement determining its relevance in the overall architecture shall suffice.

As the name implies, an *object-oriented* architecture starts with the objects or entities that are involved in the system, characterises them and defines their relationships with other objects, entities, actors or exit interfaces, etc. A *process-oriented* methodology, conversely, starts with the processes and works towards the object/entities/actors/exit interfaces. Both approaches have their advantages and disadvantages and it is not within the remit of this part of ISO 14813 to require or prefer either.

There are some aspects and viewpoints that should, however, be considered regardless of the methodology, and at least a summary explanation of these aspects should be given in the architecture portion of any ITS standard. The documentation for each of these architecture aspects is specified below.

Architectures can, and should, be viewed from different perspectives according to the views of those parties (both human and machine) likely to be involved. An ITS architecture shall consider the following architecture aspects:

a) conceptual (or reference) architecture, starting from, and related to, one or more specific ITS service domains, service groups and services (see ISO 14813-1);

b) logical (sometimes called “functional”) architecture;

c) physical architecture;

d) communications architecture;

e) organisational architecture.

This part of ISO 14813 recognises that in general practice there could be several other viewpoints needed to fully comprehend an architectural model, and this option is available to those providing architecture description in ITS standards. What matters most is that the composite description satisfies all user and interface requirements, all non-functional requirements and that it provides a rigorous basis not only for the initial design, but also for the ongoing development of the system as it evolves and interacts in new ways with its environment.

Architectures may be described and defined in many different ways. Differing descriptive formats and notations may be used in these descriptions, but the notation that is being adopted most rapidly and widely is the unified modelling language (UML) (see ISO/IEC 19501). However, many forms of process-oriented methodology have been in use for a long period of time and are well-known and understood by many users.

This part of ISO 14813 strongly recommends, but does not require, that architecture descriptions use overview *use-case* diagrams/descriptions, with pictorial elaboration, rather than system-modelling software, to explain the context of the standard at an early stage of the architecture description in order to enable non-architecture experts to understand the scope and context of the architecture description.

The depth at which architecture considerations need to be defined in a standard, or indeed in a specification or terms of reference (TOR), will vary from a simple statement of one or two paragraphs or a figure or table, to a fully detailed specification that can be used as the basis for detailed software development. This part of ISO 14813 leaves the depth of coverage to the judgement of the standard, system or TOR developer; however, the following subclauses detail the perspectives where some description is required in an ITS standard.

5.3.2 Conceptual architecture

5.3.2.1 The clause or section of the ITS standard on conceptual architecture shall provide an overall operational description of an ITS system, incorporating operational concepts and user requirements, together with its known inter-relationships with other systems. The description of the conceptual architecture shall always use one or more of the ITS service domains, service groups and services as its starting point; or its scope shall be related clearly to assisting the fulfilment of one or more specified ITS service domains, service groups and services.

5.3.2.2 The overview shall be described by means of a vision/mission statement describing the objective result of applying the standard. The method by which they are to be achieved shall be explained.

5.3.2.3 This shall be accompanied by a simple use-case diagram/description, hierarchy chart or network diagram (e.g. reference model) and/or an overview description dealing with the overall system concepts and relationships and reference points only and, where appropriate, summarising the business case.

5.3.3 Logical architecture

A clause or section shall describe the nature of the system based on the information, control or functions and shall describe the interrelations of these aspects; the logical architecture is independent of any hardware focus or software.

A logical architecture can be described either from an object- or process-oriented perspective. Either methodology may be used at the discretion of the working group preparing the standard.

5.3.4 Physical architecture (optional)

Following the explanation of the logical architecture, this description, where provided, shall allocate, in generic terms, the logical architecture to physical entities (but not in relation to the deployment of equipment).

NOTE A physical architecture, while it describes physical configurations in system terms, is not specific to any particular location.

5.3.5 Communications architecture (optional)

Communications architecture, where provided, shall offer a high-level description of the media and medium standards and protocols used to support and communicate through the system.

5.3.6 Organisational architecture (optional)

Organisational architecture, where provided, shall identify how the organisation's(s') specific requirements are to be met. The development process shall include recognition of dependencies and boundaries for functions.

5.4 Object-oriented architecture

5.4.1 General

This clause or section in an ITS standard is relevant where object-oriented architecture description/system design is employed.

UML is being used increasingly worldwide to describe the static and dynamic logical design and behaviour of complex systems because of its *de facto* universality in describing software intensive systems in a manner that is understandable across cultures, companies and customs.

UML provides for user requirements, when encapsulated in use cases, to be related to other requirements in other use cases and also to other UML artefacts [when suitable computer-aided software engineering (CASE) tools are employed]. However, it is not necessary that use cases be expressed in UML in order for them to be meaningful and unambiguous (this is discussed in greater detail in ISO/TR 24529 and ISO/TR 17452).

UML is also useful in documenting data models as described in ISO 14817.

All architecture description is heavily dependent on the concept of abstraction. In object-oriented architecture definition, the use of abstraction is particularly important as an approach to design at every level, but especially at the architectural design level.

5.4.2 Specific requirements for object-oriented description

If an object-oriented approach is used to describe the logical architecture, it shall provide an integrated description working from the identified classes, their attributes, methods and messages and associations. In order to achieve consistency of approach and enable cross-referencing between deliverables, the object analysis symbols used shall be from the UML, as specified in ISO/IEC 19501, and shall, as far as practicable, use the UML "views" of the model that are deemed appropriate. There is, however, no requirement that UML views always be used.

5.4.3 Relationship to ITS service domains, service groups and services

In an object-oriented analysis of the architectural aspects of an ITS standard, the highest-level (abstract) "classes" described shall always be related to the provision of one or more specified ITS service domains, service groups and services, as defined in ISO 14813-1.

5.4.4 Control behaviour

Control behaviour describes changes of ITS architecture elements from one state to another.

5.4.5 Multiple viewpoints

The need for multiple viewpoints in architecture models has been widely recognised and an object-oriented architecture shall describe multiple views. The appropriateness of particular viewpoints is left to the discretion of the developer of the deliverable.

5.5 Process-oriented architecture

5.5.1 General

A process-oriented (functional) decomposition of the logical architecture is represented by functional, control and information architectures. If a process-oriented approach is used, the requirements defined in 5.5.2 shall be provided.

There are three basic types of process-oriented methodology for ITS architecture:

a) framework;

b) defined;

c) specific.

The principal differences between these approaches depends on

— the viewpoints created,

— the "outputs" produced, and

— how these viewpoints and outputs are used.

5.5.2 Specific requirements for process-oriented methodology description

5.5.2.1 Context

The following clauses or sections are relevant where process-oriented architecture is employed.

5.5.2.2 Framework ITS architecture

A framework ITS architecture provides the most general and flexible approach and shows the functionality needed for a set of stakeholder aspirations.

A framework ITS architecture shall comprise identification of stakeholder needs and a functional viewpoint.

It may, at the discretion of the developers of the deliverable, also include guidance documents for creation of other outputs.

A framework architecture shall provide a high-level defined ITS architecture which provides a description of the way in which ITS is to be deployed and shall describe at least the following:

— stakeholder needs;

— functional viewpoint;

— physical viewpoint;

— communications viewpoint.

A framework architecture should strive to enable close control of outputs, particularly physical and communications viewpoint contents. Several defined architectures can usually be created from one framework architecture.

The use of architecture tools is not required in order to develop a framework architecture; however, the complexity and interdependencies of most ITS mean that it is usually highly desirable and often a necessity.

A framework architecture shall be unambiguous: clear and with only one possible reasonable interpretation.

A framework architecture shall be technology-independent and shall identify the data that flows between the functions and the “things” that interact with the identified functions.

5.5.2.3 Context diagram

A context diagram shall be provided to show the relationship between a system and those parts (called “terminators”) of its environment with which it interacts. The context diagram shall identify the system boundary using terminators. A terminator can be a person, an organisation or another system and shall have a description and identify what is expected of it as a terminator.

5.5.2.4 Data flow diagrams

Data flow diagrams shall be provided and shall comprise

— functions, which “do” things within the system to fulfil user needs,

— data stores, which contain data that is used by one or more functions,

— data flows, which identify the transfer of data between functions, between data stores and functions, and which transfer data between functions and terminators, and

— trigger data flows, which are sent by one function to activate another function.

5.5.2.5 Functional decomposition

Functional decomposition shall be provided. The functional areas for decomposition shall be related and referenced to the ITS service domains, service groups and services (see ISO 14813-1). Functions can only exist in one location and will show the resulting data flows.

Each function that comprises lower-level functions has its own data flow diagram (DFD). This shall also provide a description of the functions which gives a summary of what the function does and links to diagrams containing that function, and shall identify data flows, which shall be named, entering and leaving the function and a list of user needs that the function is designed to satisfy. Terminators shall be named and the top-level terminator data flows identified.

NOTE Physical data flows typically comprise the functional data flows that pass data from

— subsystem to subsystem,

— module to module,

— module to subsystem,

— subsystem to terminator,

— module to terminator.

Where appropriate, diagrams shall be used to support descriptions.

5.5.2.6 Communications viewpoint

An analysis of the physical data flows shall be provided to identify the characteristics of the physical links that will carry the data.

EXAMPLE Types of data to be transferred, physical mode of data transfer, security requirements, data transfer capacity required.

5.6 Application architecture/deployment (implementation) design

The specific design for a deployment describes the actual equipment deployment, in one or more specific locations, designed to achieve the application architectures. The deployment architecture is not considered appropriate for standardisation and shall not be included in an ITS standard. Where considered useful to assist understanding of the deliverable, an informative annex may show an example of a deployment design that uses the architecture defined in the deliverable.

5.7 Layout of architecture description in ITS standards and other deliverables

5.7.1 Description method

So long as the definition and description requirements are met in explicit clauses of ITS deliverables, their authors shall have the freedom to describe the aspects of their system's architecture in the manner that best describes their architecture to the lay person.

5.7.2 Usage of terms

The usage in ITS deliverables of the terms given in Annex B is not a requirement of this part of ISO 14813. However, when those terms are used, the definition of the terms shall be in accordance with Annex B. In order to avoid confusion, an architecture description given in an ITS deliverable shall not use standardised terminology in any way other than as defined in this part of ISO 14813.

Other terms and their definitions may also be used in an architecture definition; however, where used, they shall be defined in the standard and shall be explicit; alternatively, reference to a public source of the term and definition shall be provided.

5.7.3 Plain language

The deliverable describing architecture shall otherwise use plain language for architecture description and shall, as far as possible, avoid the use of jargon.

5.7.4 Deployment design

Deployment (implementation) design shall not form a part of the architecture description, nor indeed any part of the normative clause of a standard for the ITS sector. Where the inclusion of an example of an implementation design is considered to assist understanding of the architecture description in an ITS standard, it shall be given in an informative annex and it shall be clearly stated that the annex provides only an informative example. There shall be no inference, either directly or indirectly, that the example is normative or offers a preferred deployment.

5.7.5 Use of annexes

Where it is considered appropriate by the authors of an ITS standard, and where it is considered essential in order to provide interoperability, technical solutions (but not location-specific deployment designs) may be provided as informative annexes to the standard. In such cases, the main body of the standard shall clearly determine and define all normative requirements to be met and any informative annex shall simply provide example solutions on how to meet these requirements.

NOTE There is always an opportunity for additional informative annexes to be added at revision of a standard.

5.7.6 Relationships with other standards

Where known and appropriate, other relationships and inter-relationships with other known existing or planned ITS standards shall be described at the appropriate place in the architecture description of an ITS standard; nevertheless, such references shall not place or imply any limitations whatsoever on the scope or use of other standards.

Annex A
(informative)

Standards for specific architecture methodologies

This part of ISO 14813 provides requirements and guidance at a *non-technique-specific* level. For guidance on specific architecture methodology in an ITS context the reader is referred to

— ISO/TR 24529,

— ISO/TR 17452, and

— ISO/TR 26999.

See References [13], [11] and [16].

Annex B
(normative)

Glossary of ITS architecture terms, abbreviated terms and numeric notation

The usage in an ITS standard of the actual terms, abbreviated terms and numeric notation that are defined in this annex is not a requirement of this part of ISO 14813; nor are all of the terms defined herein. However, where used, their definitions shall be those given in this annex.

Other terms may also be used, but their definitions shall be provided in the standard in which they are used and shall be explicit; alternatively, reference to a public source of the term and definition shall be provided.

B.1 Terms and definitions

B.1.1
abstraction
essential characteristics of an entity that distinguish it from all other kinds of entities

NOTE 1 Adapted from ISO/IEC 19501.

NOTE 2 An abstraction defines a boundary relative to the perspective of the viewer.

NOTE 3 Abstraction is perhaps the most powerful tool available to a software engineer. Abstraction aims at reducing detail, making the thing that has been subjected to abstraction simpler to handle. Abstractions usually build on lower-level abstractions, leading to a layered, hierarchical design. (Szyperski, 1998)

B.1.2
action
specification of an executable statement that implements a procedure

[ISO/IEC 19501:2005]

B.1.3
action sequence
series of actions in a predetermined sequence

B.1.4
activation
execution of an action

[ISO/IEC 19501:2005]

B.1.5
actor
coherent set of roles that users play when interacting within use cases

B.1.6
aggregate class
class that can be divided into subclasses

B.1.7
aggregation
special form of association that specifies a whole-part relationship between the aggregate (whole) and a component part

[ISO/IEC 19501:2005]

B.1.8
analysis
part of the software development process whose primary purpose is to formulate a model of a domain

[ISO/IEC 19501:2005]

NOTE Analysis focuses on what to do, design focuses on how to do it. Contrast: **design** (B.1.47).

B.1.9
application architecture
set of functions combined to form a high-level system design

B.1.10
architecture
organisational structure and associated behaviour of a system

[ISO/IEC 19501:2005]

NOTE An architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling parts. Parts that interact through interfaces include classes, components and subsystems.

B.1.11
architecture element
definable element of a system, which forms part of a component or system but which does not necessarily have independent operational functionality

B.1.12
artefact
physical piece of information that is used in or produced by a software development process

NOTE An artefact may constitute the implementation of a deployable component.

[ISO/IEC 19501:2005]

B.1.13
ASN.1
abstract syntax notation one

[ISO/IEC 8824-1:2002]

B.1.14
ASN.1 type
data type (or type for short) that represents in a formalised way a class of information (e.g. numerical, textual, still image or video information)

[ISO/IEC 8824-1:2002]

B.1.15
associated ASN.1 type
ASN.1 type which is used to represent a non-ASN.1 type in an ASN.1 module

[ISO/IEC 8824-1:2002]

B.1.16
association
semantic relationship between two or more classifiers that specifies connections among their instances

[ISO/IEC 19501:2005]

B.1.17
attribute
abstraction of properties and values belonging to, or characteristic of, an entity

B.1.18
basic encoding rules
BER
standardised determination of data encoding conforming to ASN.1, as defined ISO/IEC 8825, in accordance with ISO/IEC 8824

NOTE Alternative forms of encoding include the packed encoding rules (PER).

B.1.19
behaviour
observable effects of an operation or event, including its results

B.1.20
binary association
association between two classes

[ISO/IEC 19501:2005]

B.1.21
boolean
enumeration whose values are true and false

[ISO/IEC 19501:2005]

NOTE A field of mathematical logic developed in the mid-19th century by the English mathematician George Boole which allows a database searcher to combine concepts in a keyword search using three commands [subsequently extended to four commands, also known as “operators”: TRUE/YES, FALSE/NO, AND, OR (different instantiations may use other similes)].

B.1.22
boolean expression
expression that evaluates to a boolean value

[ISO/IEC 19501:2005]

B.1.23
cardinality
number of elements in a set or group (considered as a property of that grouping)

[wordnet.princeton.edu]

B.1.24
child
specialization of another element, the parent, in a generalization relationship

[ISO/IEC 19501:2005]

NOTE See **subclass** (B.1.153), **subtype** (B.1.155). Contrast: **parent** (B.1.112).

B.1.25
class
description of a set of objects that share the same attributes, operations, methods, behaviour, relationships and semantics

B.1.26
classifier
mechanism that describes behavioural and structural features

[ISO/IEC 19501:2005]

NOTE Classifiers include interfaces, classes, datatypes and components.

B.1.27
classification
assignment of an object to a classifier

[ISO/IEC 19501:2005]

B.1.28
class diagram
diagram that shows a collection of declarative (static) model elements, such as classes, types and their contents and relationships

[ISO/IEC 19501:2005]

B.1.29
collaboration
interactions between entities

B.1.30
compatibility
ability of any (sub)system to interact with another (sub)system according to a set of pre-defined rules in the form of an interface specification and protocol definition

B.1.31
component
modular, deployable, and replaceable part of a system that encapsulates implementation and exposes a set of interfaces

[ISO/IEC 19501:2005]

B.1.32
concurrency
occurrence of two or more activities during the same time interval

[ISO/IEC 19501:2005]

B.1.33
communications architecture
description of the media and medium standards and protocols used to support and communicate through the system

B.1.34
conceptual architecture
high-level abstract architecture that defines overall principles and objectives of a system or entity and which provides an overall description of a system incorporating operational concepts and user requirements, together with its known inter-relationships with other systems

NOTE A conceptual architecture is often expressed/supported by means of vision/mission statements, a simple hierarchy chart or network diagram (e.g. reference model), dealing with only the overall concepts and relationships and reference points. A conceptual architecture is not specific to any location.

B.1.35
constraint
semantic condition or restriction

[ISO/IEC 19501:2005]

B.1.36
control architecture
description of the control behaviour of ITS architecture elements to effect change from one state (condition) to another (state transition management)

NOTE A control architecture is not specific to any location.

B.1.37
data construct
group of one or more data elements used to represent state or information

B.1.38
data element
one or a group of data primitives

[ISO/IEC 8824-1:2002]

B.1.39
data model
relationships between the pieces of data (rather than the particulars of individual records); data modelling depends on abstraction

B.1.40
data primitive
data element that cannot be further subdivided meaningfully within the context of ASN.1

[ISO/IEC 8824-1:2002]

B.1.41
data type
named set of values

[ISO/IEC 8824-1:2002]

B.1.42
datatype
descriptor of set of values that lack identity and whose operations do not have side-effects

NOTE Datatypes include primitive pre-defined types and user-definable types. Pre-defined types include numbers, string and time. User-definable types include enumerations.

B.1.43
dependency
relationship between two modelling elements, in which a change to one modelling element (the independent element) will affect the other modelling element (the dependent element)

[ISO/IEC 19501:2005]

B.1.44
implementation
deployment design
physical specification for an instantiation of a single actual physical manifestation/implementation of a system or entity to achieve the application architecture objectives

NOTE Deployment design is not considered appropriate for standardisation. Deployment (implementation) design is sometimes incorrectly referred to as a "physical architecture". However, deployment (application) design is specific in location and often also in time, whereas an ITS architecture is not.

B.1.45
deployment of ITS
actual ITS equipment, software, facilities, devices, etc. at a specific location and at a specific time

B.1.46
derived element
model element that can be computed from another element, but that is shown for clarity or that is included for design purposes even though it adds no semantic information

[ISO/IEC 19501:2005]

B.1.47
design
how parts or constituents are related to an organised whole, providing specification for the structure, organization, appearance, etc. of a system or entity

B.1.48
diagram
graphical presentation of a collection of model elements, most often rendered as a connected graph of relationships with other model elements

B.1.49
domain
class of all entities of similar group and common characteristic

B.1.50
electronic data interchange
EDI
passing of a message, or series of messages, between computers and/or between different software systems

EXAMPLE EDIFACT.

NOTE Within the context of ITS, an EDI message is normally compatible with the form specified in ISO/IEC 9897.

B.1.51
electronic data transfer
EDT
passing of data sets comprising an entire message from one computer to another or from one software system to another

B.1.52
electronic data interchange for administration, commerce and transport
EDIFACT
specific message format for the sector in question as specified in ISO 9735

B.1.53
element
atomic constituent of a model

B.1.54
entry action
action executed upon entering a state in a state machine regardless of the transition taken to reach that state

B.1.55
enumeration
list of named values used as the range of a particular attribute type

EXAMPLE RGBColor ={red, green, blue}.

[ISO/IEC 19501:2005]

B.1.56
event
specification of a significant occurrence that has a location in time and space

NOTE In the context of state diagrams, an event is an occurrence that can trigger a transition.

[ISO/IEC 19501:2005]

B.1.57
expression
string that evaluates to a value of a particular type

EXAMPLE (7 + 5 * 3), which evaluates to a value of type number.

B.1.58
feature
property, such as an operation or attribute, which is encapsulated within a classifier, such as an interface, class or datatype

[ISO/IEC 19501:2005]

B.1.59
framework
stereotyped package that contains model elements which specify a reusable architecture for all or part of a system

NOTE Frameworks typically include classes, patterns or templates. When frameworks are specialised for an application domain, they are sometimes referred to as application frameworks.

B.1.60
functional architecture
aspect of the logical, process-oriented decomposition of an overall ITS architecture (process-oriented logical architecture) that provides an arrangement of functions and their subfunctions and interfaces (internal and external) and that defines the execution sequencing, conditions for control or data flow and performance requirements to satisfy the requirements baseline

NOTE 1 Adapted from IEEE 1220.

NOTE 2 A functional architecture is not specific to any location.

B.1.61
generalization
taxonomic relationship between a more general element and a more specific element where the more specific element is fully consistent with the more general element and contains additional information

[ISO/IEC 19501:2005]

NOTE An instance of the more specific element may be used where the more general element is allowed. See **inheritance** (B.1.67).

B.1.62
information architecture
with respect to process-oriented decomposition, the information architecture defines the entities and relationships of information (data model) and principal data constructs including the structural design of shared information elements, the organisation of data sites, web sites etc.

NOTE 1 Process-oriented decomposition of information, which refers to data modelling and the analysis and design of the information in the system, concentrating on entities and their interdependencies, comes very close to object-oriented architecture, the principal difference being that, in process decomposition, it is the result of logical analysis, whereas in object-oriented architecture, it is the starting point for logical analysis.

NOTE 2 The information architecture is not specific to any location.

B.1.63
information object
instance of some information object class, being composed of a set of fields which conform to the field specification of the class

[ISO/IEC 8824-2:2002]

B.1.64
information object class
set of fields, forming a template for the definition of a potentially unbounded collection of information objects, the instances of the class

[ISO/IEC 8824-2:2002]

B.1.65
institutional architecture
architecture based on political or administration infrastructure partitioning and its division of responsibilities (rather than functions)

B.1.66
interoperability
ability of systems to provide services to and accept services from other systems and to use the services so exchanged to enable them to operate effectively together

B.1.67
inheritance
mechanism by which more specific elements incorporate structure and behaviour of more general elements related by behaviour

[ISO/IEC 19501:2005]

B.1.68
instance
entity that has unique identity, a set of operations that can be applied to it, and state that stores the effects of the operations

[ISO/IEC 19501:2005]

B.1.69
instantiation
single actual physical manifestation/implementation of a system or entity

B.1.70
interaction
response experience in which both actor and reactor are engaged in a mutually affecting experience

[Nathan]

B.1.71
interaction diagram
diagrams that describe object interactions

B.1.72
interface
named set of operations that characterise the behaviour between two or more parties

B.1.73
ITS component
subsystem of ITS system, assembly or other major element of a system which does not have, by itself, independent operational functionality

B.1.74
ITS system
set of interrelated components which interact with one another in an organised fashion to provide independent operational functionality

NOTE Adapted from the IEEE definition of a system (that definition, qualified by the “independent operational functionality” phase).

B.1.75
ITS unit
minimum component capable of independent functionality

B.1.76
layer
organisation of classifiers or packages at the same level of abstraction

[ISO/IEC 19501:2005]

B.1.77
link
semantic connection among a tuple of objects

NOTE It is an instance of association.

[ISO/IEC 19501:2005]

B.1.78
logical architecture
description of the nature of a system, based on the information, control or functions, and of the interrelations of these aspects

NOTE 1 The logical architecture is independent of any hardware focus or software.

NOTE 2 A logical architecture can be described either from the perspective of an **object-oriented** (B.1.102) or **process-oriented** (B.1.115) **methodology**. In an object-oriented perspective, a logical architecture elaborates the conceptual behaviour and, in so doing, defines some detail of the objects. In a process-oriented perspective, logical architecture determines the nature of the system as being based on information and functions, and describes the inter-relationships of these aspects.

NOTE 3 The logical architecture is independent of any hardware or software approach.

B.1.79
maintainability
ability of a device to be maintained at, or restored to, specified conditions within a given period of time

B.1.80
message
⟨general⟩ object of communication

NOTE Depending on the context, the term may apply to both the information content and its actual presentation.

B.1.81
message
⟨software⟩ set of data grouped together for transmission that is ordered according to the rules of a given protocol suite, such that it is intelligible to the sending and receiving software

[San Diego State University]

NOTE In software systems, there is usually the expectation that activity will ensue as a result of the receipt of the message.

B.1.82
metaclass
class whose instances are classes

[ISO/IEC 19501:2005]

B.1.83
metadata
high-level data that describes the characteristics of other data, such as the content, quality, condition and other characteristics of the data

B.1.84
metamodel
model that explains a set of related models by defining the language for expressing such models

[ISO/IEC 19501:2005]

B.1.85
methodology
specific set of means or procedures used in attaining an end

B.1.86
method
means of implementation of an operation

B.1.87
mission statement
statement of what objective results are to be provided by the standard and how it is intended to achieve the vision

NOTE See **vision statement** (B.1.166).

B.1.88
model
abstraction of a physical system with a certain purpose

B.1.89
model aspect
dimension of modelling that characterises a particular quality of the metamodel

[ISO/IEC 19501:2005]

B.1.90
model elaboration
process of generating a repository type from a published model

NOTE It includes the generation of interfaces and implementations, which allows repositories to be instantiated and populated based on, and in compliance with, the model elaborated.

[ISO/IEC 19501:2005]

B.1.91
model element
element that is a characterised abstraction drawn from the system being modelled

[ISO/IEC 19501:2005]

B.1.92
module
any of a number of distinct but interrelated units from which a program may be built up or into which a complex activity may be analysed

[OED]

B.1.93
method
means of implementation of an operation

B.1.94
module identifier
instance of an object identifier type which relates to an associated module

B.1.95
multiple inheritance
semantic variation of generalisation in which a type may have more than one supertype

[ISO/IEC 19501:2005]

B.1.96
name
word or combination of words constituting the individual designation by which an entity is identified

[OED]

B.1.97
namespace
part of the model in which the names may be defined and used

NOTE Within a namespace, each name has a unique meaning. See **name** (B.1.96).

B.1.98
node
junction point or termination point where actions or results may be expected

NOTE In software, it is often a classifier that represents a run-time computational resource, which generally has at least a memory and often a processing capability. Run-time objects and components may reside on nodes.

B.1.99
non-ASN.1 type
type which is not conformant with ISO/IEC 8824

B.1.100
non-specific design
design, in generic terms, based on a requirement rather than an exact identification of equipment specification or manufacturer's identification

B.1.101
object
entity with a well-defined boundary and identity that encapsulates state and behaviour, where its state is represented by attributes and relationships, behaviour is represented by operations, methods and state machines

NOTE In software, an object is an instance of a class.

[ISO/IEC 19501:2005]

B.1.102
object-oriented methodology
methodology based on objects, classes and messages between objects

B.1.103
object diagram
diagram that encompasses objects and their relationships at a point in time

[ISO/IEC 19501:2005]

B.1.104
object identifier
value (distinguishable from all others) which is associated with an object

[ISO/IEC 8824-1:2002]

B.1.105
object identifier type
simple type whose distinguished values are the set of all object identifiers allocated in accordance with the rules of ISO/IEC 9834-1 and ISO 14813-6

[ISO/IEC 8824-1:2002]

NOTE ISO/IEC 8824-1 permits a wide range of authorities to independently associate object identifiers with objects.

B.1.106
operation
service that can be requested from an object to effect behaviour

[ISO/IEC 19501:2005]

B.1.107
organisational architecture
description which identifies how organisation-specific requirements will be met

NOTE The development process includes recognition of dependencies and boundaries for functions.

B.1.108
open systems interconnection model
OSI model
reference model developed by ISO to enable different or similar systems to dialogue with one another

NOTE This model constitutes a reference framework for describing data exchanges. It has seven layers. See Reference [1].

B.1.109
package
general-purpose mechanism for organising elements into groups

[ISO/IEC 19501:2005]

NOTE Packages may be nested within other packages.

B.1.110
parameter
specification of a variable that can be changed, passed, or returned

[ISO/IEC 19501:2005]

NOTE A parameter may typically include a name, type, and direction. Parameters are used for operations, messages, and events.

B.1.111
parameterised element
descriptor for a class with one or more unbound parameters

[ISO/IEC 19501:2005]

B.1.112
parent
generalisation of another element, the child, in a generalisation relationship

[ISO/IEC 19501:2005]

B.1.113
participate, verb
connection of a model element to a relationship or to a reified relationship

EXAMPLE A class participates in an association; an actor participates in a use case.

B.1.114
physical architecture
application architecture
allocation of the logical architecture to physical entities but not relating to the deployment of equipment

NOTE A physical architecture, while it describes physical configurations in system terms, is not specific to any particular location.

B.1.115
process-oriented methodology
methodology based on decomposition of the conceptual architecture into functional, control and information architectures

B.1.116
postcondition
constraint that must be true at the completion of an operation

[ISO/IEC 19501:2005]

B.1.117
precondition
constraint that must be true when an operation is invoked

[ISO/IEC 19501:2005]

B.1.118
primitive type
pre-defined basic datatype without any substructure, such as an integer or a string

[ISO/IEC 19501:2005]

B.1.119
process, noun
in an operating system, the heavyweight unit of concurrency and execution, and in the software development process, the steps and guidelines by which to develop a system

B.1.120
process, verb
execute an algorithm or otherwise handle something dynamically

[ISO/IEC 19501:2005]

B.1.121
profile
stereotyped package that contains model elements which have been customised for a specific domain or purpose using extension mechanisms such as stereotypes, tagged definitions and constraints

NOTE A profile may also specify model libraries on which it depends and the metamodel subset that it extends.

B.1.122
property
named value denoting a characteristic of an element that has semantic impact

B.1.123
physical system
instantiation of a system which can include hardware, software, hardware and people, that are organised to accomplish a specific purpose

NOTE A physical system can be described from one or more different viewpoints.

B.1.124
receive, verb
⟨message⟩ handling of a stimulus passed from a sender instance

[ISO/IEC 19501:2005]

B.1.125
receiver
object handling a stimulus passed from a sender object

[ISO/IEC 19501:2005]

B.1.126
reference point
identifier for a component of an architecture which indicates the component is to be elaborated in a subsequent architecture

B.1.127
registry
facility for recording and storing object models, interfaces and data concepts within an organised and managed environment

B.1.128
relationship
semantic connection among model elements

[ISO/IEC 19501:2005]

EXAMPLE Associations, generalisations.

B.1.129
reliability
ability of a device to perform its intended function under given conditions of use for a specified period of time (or number of cycles)

B.1.130
repository
physical facility for storing object models, interfaces, and implementations

[ISO/IEC 19501:2005]

B.1.131
requirement
something called for or demanded; a condition which must be complied with

[OED]

B.1.132
reuse
second or further use of a pre-existing artefact

B.1.133
role
named specific behaviour of an entity participating in a particular context

[ISO/IEC 19501:2005]

NOTE A role may be static (e.g. an association end) or dynamic (e.g. a collaboration role).

B.1.134
run time
period of time during which a computer program executes

[ISO/IEC 19501:2005]

B.1.135
scenario
specific sequence of actions that illustrates behaviours

[ISO/IEC 19501:2005]

NOTE A scenario may be used to illustrate an interaction or the execution of a use case instance.

B.1.136
send, verb
⟨message⟩ passing of a stimulus from a sender instance to a receiver instance

[ISO/IEC 19501:2005]

B.1.137
sender
object passing a stimulus to a receiver object

[ISO/IEC 19501:2005]

B.1.138
sequence diagram A
term used in object-oriented description for a diagram that shows object interactions arranged in time sequence, in particular the objects participating in the interaction and the sequence of messages exchanged

NOTE Unlike a collaboration diagram, a sequence diagram includes time sequences but does not include object relationships. A sequence diagram can exist in a generic form (describes all possible scenarios) and in an instance form (describes one actual scenario). Sequence diagrams and collaboration diagrams express similar information, but show it in different ways.

[ISO/IEC 19501:2005]

B.1.139
signal
specification of an asynchronous stimulus communicated between instances

NOTE Signals may have parameters.

B.1.140
signature
name and parameters assigned to identify a behavioural feature

NOTE A signature may include an optional returned parameter.

B.1.141
simple (data) types
types defined by directly specifying the set of its values

[ISO/IEC 8824-1:2002]

B.1.142
single inheritance
semantic variation of generalisation in which a type may have only one supertype

[ISO/IEC 19501:2005]

B.1.143
software
set(s) of instructions, also called program(s), that control the use of a computer

[ISO/IEC 19501:2005]

B.1.144
software module
software unit of storage and manipulation

NOTE Modules include source code modules, binary code modules, and executable code modules.

[ISO/IEC 19501:2005]

B.1.145
specification
declarative description of what something is or does, including the means of achieving its objectives

[ISO/IEC 19501:2005]

B.1.146
specified design
exact equipment specification or manufacturer's identification as installed in a deployment

B.1.147
state
condition or situation during the life of an object during which it satisfies some condition, performs some activity, or waits for some event

[ISO/IEC 19501:2005]

B.1.148
statechart diagram
diagram that shows a state machine

[ISO/IEC 19501:2005]

B.1.149
state machine
behaviour that specifies the sequences of states that an object or an interaction goes through during its life in response to events, together with its responses and actions

[ISO/IEC 19501:2005]

B.1.150
stereotype
modelling element that extends the semantics of the metamodel

[ISO/IEC 19501:2005]

NOTE Stereotypes must be based on certain existing types or classes in the metamodel. Stereotypes may extend the semantics, but not the structure of pre-existing types and classes.

B.1.151
stimulus
result of passing of information from one instance to another, such as raising a signal or invoking an operation

[ISO/IEC 19501:2005]

NOTE The receipt of a signal is normally considered an event.

B.1.152
string
sequence of text characters

[ISO/IEC 19501:2005]

B.1.153
subclass
specialisation of another class, the superclass, in a generalisation relationship

[ISO/IEC 19501:2005]

B.1.154
subsystem
grouping of model elements that represent a behavioural unit in a physical system

NOTE A subsystem offers interfaces and has operations. In addition, the model elements of a subsystem can be partitioned into specification and realization elements.

B.1.155
subtype
specialisation of another type, the supertype, in a generalisation relationship

[ISO/IEC 19501:2005]

B.1.156
superclass
generalisation of another class, the subclass, in a generalisation relationship

[ISO/IEC 19501:2005]

B.1.157
supertype
generalisation of another type, the subtype, in a generalisation relationship

[ISO/IEC 19501:2005]

B.1.158
thread (of control)
single path of execution through a program, a dynamic model, or some other representation of control flow

[ISO/IEC 19501:2005]

B.1.159
type
stereotyped class that specifies a domain of objects together with the operations applicable to the objects, without defining the physical implementation of those objects

[ISO/IEC 19501:2005]

NOTE A type may not contain any methods, maintain its own thread of control, or be nested. However, it may have attributes and associations.

B.1.160
use case
use-case class
specification of a sequence of actions, including variants, that a system (or other entity) can perform, interacting with actors of the system

[ISO/IEC 19501:2005]

B.1.161
use-case diagram
diagram that shows the relationships among actors and use cases within a system

[ISO/IEC 19501:2005]

B.1.162
use-case instance
performance of a sequence of actions being specified in a use case

[ISO/IEC 19501:2005]

B.1.163
use-case model
model that describes a system's functional requirements in terms of use cases

NOTE See **use case class** (B.1.160).

B.1.164
user
prime beneficiary of an ITS system

NOTE A user may be an end user (e.g. vehicle operator) or a system provider (e.g. traffic system manager) or other intermediary who uses the system.

B.1.165
view/viewpoint
projection of a model, seen from a given perspective or vantage point and which omits entities that are not relevant to this perspective

[ISO/IEC 19501:2005]

B.1.166
vision statement
statement summarising the overall concept and goals that the standard seeks to achieve

B.2 Abbreviated terms

BER basic encoding rules (of ASN.1, see ISO\IEC 8825)

DFD data flow diagram

ITS intelligent transport systems

OMG object management group

RTTT road transport and traffic telematics (former term for intelligent transport systems)

TICS transport information and control systems (former term for intelligent transport systems)

B.3 Numeric notations

The following conventions are applicable to ITS standards:

— decimal (“normal”) notation shall have no subscript, e.g. 127;

— hexadecimal numbers shall be noted by the subscript “16”, e.g. $7F_{16}$;

— binary numbers shall be noted by the subscript “2”, e.g. 01111111_2.

Bibliography

[1] ISO 7498-1, *Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model*

[2] ISO 8824-1:2008, *Information technology — Abstract Syntax Notation One (ASN.1): Specification of basic notation — Part 1*[3)]

[3] ISO/IEC 8824-2:2002, *Information technology — Abstract Syntax Notation One (ASN.1): Information object specification — Part 2*

[4] ISO 9735 (all parts), *Electronic data interchange for administration, commerce and transport (EDIFACT) — Application level syntax rules (Syntax version number: 4, Syntax release number: 1)*

[5] ISO/IEC 9897 (all parts), *Freight containers — Container equipment data exchange (CEDEX)*

[6] ISO/IEC 10746 (all parts), *Information technology — Open Distributed Processing*

[7] ISO/TR 14813-2, *Transport information and control systems — Reference model architecture(s) for the TICS sector — Part 2: Core TICS reference architecture*

[8] ISO/TR 14813-3, *Transport information and control systems — Reference model architecture(s) for the TICS sector — Part 3: Example elaboration*

[9] ISO/TR 14813-4, *Transport information and control systems — Reference model architecture(s) for the TICS sector — Part 4: Reference model tutorial*

[10] ISO 14817, *Transport information and control systems — Requirements for an ITS/TICS central Data Registry and ITS/TICS Data Dictionaries*

[11] ISO/TR 17452, *Intelligent transport systems — Using UML for defining and documenting ITS/TICS interfaces*

[12] ISO/TR 24097-1, *Intelligent transport systems — Using web services (machine-machine delivery) for ITS service delivery — Part 1: Realization of interoperable web services*

[13] ISO/TR 24529, *Intelligent transport systems — Systems architecture — Use of unified modelling language (UML) in ITS International Standards and deliverables*

[14] ISO 24531, *Intelligent transport systems — System architecture, taxonomy and terminology — Using XML in ITS standards, data registries and data dictionaries*

[15] ISO/TR 24532, *Intelligent transport systems — Systems architecture, taxonomy and terminology — Using CORBA (Common Object Request Broker Architecture) in ITS standards, data registries and data dictionaries*

[16] ITU Recommendation Z.100, *Specification and Description Language (SDL)*

[17] The Unified Modelling Language: http://www.uml.org[4)]

3) Revised by ISO/IEC 8824-1:2008.

4) Architectures can be described and defined in many different ways: for example, the RM-ODP (see ISO/IEC10746), or Putman, 2001, Hofmeister, 2000, or Clements, 2003, of the SEI (software engineering institute) of the USA.

[18] IEEE 1220, *Standard for Application and Management of the Systems Engineering Process — Description* [5)]

5) Institute of Electrical and Electronics Engineers, USA, standard.

ICS 03.220.01; 35.240.60

Price based on 31 pages