TECHNICAL REPORT

ISO/TR 14813-2

First edition
2000-12-15

Transport information and control systems — Reference model architecture(s) for the TICS sector —

Part 2:
Core TICS reference architecture

Systèmes de commande et d'information des transports — Architecture(s) du modèle de référence du secteur TICS —

Partie 2: Architecture de référence du noyau des TICS

Reference number
ISO/TR 14813-2:2000(E)

PDF disclaimer

This PDF file may contain embedded typefaces. In accordance with Adobe's licensing policy, this file may be printed or viewed but shall not be edited unless the typefaces which are embedded are licensed to and installed on the computer performing the editing. In downloading this file, parties accept therein the responsibility of not infringing Adobe's licensing policy. The ISO Central Secretariat accepts no liability in this area.

Adobe is a trademark of Adobe Systems Incorporated.

Details of the software products used to create this PDF file can be found in the General Info relative to the file; the PDF-creation parameters were optimized for printing. Every care has been taken to ensure that the file is suitable for use by ISO member bodies. In the unlikely event that a problem relating to it is found, please inform the Central Secretariat at the address given below.

Contents

List of figures

List of tables

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 main task of technical committees is to prepare International Standards, but in exceptional circumstances a technical committee may propose the publication of a Technical Report of one of the following types:

— type 1, when the required support cannot be obtained for the publication of an International Standard, despite repeated efforts;

— type 2, when the subject is still under technical development or where for any other reason there is the future but not immediate possibility of an agreement on an International Standard;

— type 3, when a technical committee has collected data of different kind from that which is normally published as an International Standard ("state of the art", for example).

Technical Reports of types 1 and 2 are subject to review within three years of publication, to decide whether they can be transformed into International Standards. Technical Reports of type 3 do not necessarily have to be reviewed until the data they provide are considered to be no longer valid or useful.

Technical Reports are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

Attention is drawn to the possibility that some of the elements of this part of ISO/TR 14813 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

ISO/TR 14813-2, which is a Technical Report of type 2, was prepared by Technical Committee ISO/TC 204, *Transport information and control systems*.

This document is being issued in the Technical Report (type 2) series of publications (according to subclause G.3.2.2 of Part 1 of the ISO/IEC Directives, 1995) as a “prospective standard for provisional application” in the field of transport information and control systems because there is an urgent need for guidance on how standards in this field should be used to meet an identified need.

This document is not to be regarded as an “International Standard”. It is proposed for provisional application so that information and experience of its use in practice may be gathered. Comments on the content of this document should be sent to the ISO Central Secretariat.

A review of this Technical Report (type 2) will be carried out not later than three years after its publication with the options of: extension for another three years; conversion into an International Standard; or withdrawal.

ISO/TR 14813 consists of the following parts, under the general title *Transport information and control systems — Reference model architecture(s) for the TICS sector:*

— *Part 1: TICS fundamental services.* This document presents the definition of 32 TICS fundamental services that are the informational products or services or applications areas provided to a TICS user.

— *Part 2: Core TICS reference architecture.* This document describes an abstract object-oriented system architecture based on the TICS fundamental services.

— *Part 3: Example elaboration.* This document refines the core TICS reference architecture (part 2) with some emphasis on traffic management.

— *Part 4: Reference model tutorial.* This document describes the basic terms, graphical representations and modelling views exploited in the object-oriented definition of the architecture development of parts 2 and 3.

— *Part 5: Requirements for architecture description in TICS standards.* This document describes the terminology and form to be used when documenting or referencing aspects of architecture description in TICS standards.

— *Part 6: Data presentation in ASN.1.* This document establishes the use of ASN.1 as the normal syntax notation to be used in standards for the TICS sector and a common message form for such ASN.1 based data elements.

Annex A of this part of ISO/TR 14813 is for information only.

Introduction

TC204/WG1 is a working group whose prime objectives are to provide services to ISO TC204 and its working groups. A specific mission of WG1 is to:

"Provide ISO TC204, its working Groups, related bodies and those involved in the TICS sector, with a reference model of Conceptual Reference Architecture(s) that show the structure and inter-relationships of the sector ..."

It is expected that there may well be more than one single TICS Architecture approach to be considered and documented and that existing architecture approaches will have previously-produced documentation developed according to disparate standards and conventions.

It is also implicit in the work being undertaken by WG1, that working group members will require a clear, well-structured understanding of the work of the following participant groups:

- Other TC 204 Working Groups
- CEN TC 278 Working Groups
- Japanese initiatives
- European Road Transport and Traffic Telematics programs
- US Intelligent Transportation Systems program
- Australian initiatives
- Canadian Initiatives

Full documentation of all possible architectural approaches is obviously not feasible given the high level of resources required to carry this out. Indeed full documentation and description of all possible approaches is undesirable as an item for Standardisation.

A defined and consistent approach is however required to facilitate the specification of architecture requirements to enable a clear view to be developed and presented of the work of each participant group This document is one of a set of WG1 documents intended to respond to stated WG1 objectives regarding the production of a TICS Reference Architecture.

In order to document an architecture, graphical and textual components of a model are required. WG1 has adopted a methodology based on the Unified Modelling Language (UML) for documenting the TICS Reference Architecture. A tutorial on the UML is provided in Part 4. UML is a visual modelling language for building object-oriented and component-based systems. A commercially available Computer Aided Software Engineering (CASE) tool has been used by WG1 to document the Architecture. While the tool is a commercial product, UML is open and non-proprietary.

Transport information and control systems — Reference model architecture(s) for the TICS sector — Part 2: Core TICS reference architecture

1 Scope

The architecture of an information and control system merges hardware and software considerations into a coordinated and integrated system view. The system architecture is a high level abstraction, or model, of the system. A system architecture should embrace both today's applications and the applications that are expected in the future. Architecture begins with the definition of the conceptual services (e.g. Part 1 - TICS fundamental services). There are several identifiable stages of system architecture development:

— 1. Reference architecture

— 2. Logical architecture

— 3. Physical architecture

A reference architecture is the first of all architectures. It is a concise generic framework which guides the development of more concrete system architectures. It is large enough that distinct concepts are not merged out of necessity and small enough that it does not become unwieldy.

A most significant example of a reference architecture in information systems is the Reference Model of Open Systems Interconnection (often called the seven layer model) developed by ISO in the 1970's. This model has underpinned the development of all modern computer networks, allowing services such as global networking, of which the prime example is the Internet, to become a reality.

A reference architecture is generic and non-prescriptive and captures the concepts of the system. A logical architecture elaborates the conceptual behaviour, and in so doing it provides more detail about the modularity. A physical architecture is reached when the actual distribution of the system modules is defined, thus leading to important implications for communications.

There is no firm demarcation between a reference architecture and a logical architecture. Thus the essence of behaviour and modularity is present in a reference architecture. The TICS Reference Architecture developed by WG1 shows important inter-relationships that arise in the provision of the services of the sector. However the TICS Reference Architecture is more abstract than, for example, the logical architecture of the US National Architecture.

It is envisioned that the TICS Reference Architecture will be used by the TC204 Working Groups to develop their own logical and physical architectures in a cohesive manner.

Some TICS Fundamental Services are already well developed by the industry, while others are less mature. Therefore the TICS Reference Architecture does not have a uniform granularity across all services. This characteristic is a direct result of the fore mentioned requirement that architecture embrace the applications that are intended in the future. This suggests one of the ways in which the architecture will undergo change in the future.

Architectures may present only static characteristics or both static and dynamic characteristics. Dynamic characteristics may be seen as belonging solely to the design/implementation stages of system development. However by including dynamic characteristics at the reference architecture stage one gains important insights into the static architecture. Thus two orthogonal views of architecture are presented:

- 1. static relationship view (class diagram)
- 2. dynamic interactive view (sequence diagram)

This part of ISO/TR 14813 develops a core reference architecture. The static scope is determined by deriving the system boundary and the use cases from an analysis of the TICS fundamental services (part 1 of ISO/TR 14813).

The Core Reference Architecture is a reference for the development of national architectures.

Part 3 of ISO/TR14813 elaborates the core reference architecture by refinement of two orthogonal views. The elaboration calls upon domain expertise that would be provided by other TC204 Working Groups in the development of ISO standards or by national groups developing national architectures and standards.

The core reference architecture is described in clauses 5 to 8. Clause 5 introduces the architecture at a highly abstract level. Clause 6 defines all the actors. Clause 7 derives all the use case from the TICS fundamental services and develops eight use case diagrams. Clause 8 defines an abstract collection of classes and develops a set of sequence diagrams, one per use case diagram.

Readers should refer to Part 4 of ISO/TR 14813 (Tutorial) for an introduction to the modelling views used in this part and the methodology applied. The methodology is repeated in Annex A.

2 Normative references

The following normative documents contain provisions that, through reference in this text, constitute provisions of this part of ISO/TR 14813. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO/TR 14813 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO/TR 14813-1:1999, *Transport information and control systems – Reference model architecture(s) for the TICS sector – Part 1:TICS fundamental services.*

ISO/TR 14813-3:2000, *Transport information and control systems – Reference model architecture(s) for the TICS sector – Part 3: Example elaboration.*

ISO/TR 14813-4:2000, *Transport information and control systems – Reference model architecture(s) for the TICS sector – Part 4: Reference model tutorial.*

3 Terms and definitions

For the purposes of this part of ISO/TR 14813, the following semantic definitions apply.

3.1
The **Unified Modeling Language** (UML) is the industry-standard language for specifying, visualising, constructing, and documenting the artefacts of software systems. It simplifies the complex process of software design, making a "blueprint" for construction.[1]

3.2
A **use case** is a coherent unit of functionality provided by a system or class as manifested by sequences of messages exchanged among the system and one or more outside interactors (called actors) together with actions performed by the system.

[1] http://www.rational.com/uml/

3.3
An **actor** is a role of an object or objects outside of a system that interacts directly with it as part of a coherent work unit (a use case). An Actor element characterises the role played by an outside object; one physical object may play several roles and therefore be modelled by several actors.

3.4
Use case diagrams show elements from the use case model. The use case model represents functionality of a system or a class as manifested to external interactors with the system.

There are several standard relationships among use cases or between actors and use cases.

Communicates - The participation of an actor in a use case. This is the only relationship between actors and use cases.

Uses - A uses relationship from use case A to use case B indicates that an instance of the use case A will also include the behavior as specified by B.

To reinforce the externality of actors a system boundary separates the actor symbols from the use case symbols.

3.5
A **package** is a grouping of model elements. Packages themselves may be nested within other packages. A package may contain both subordinate packages and ordinary model elements. The entire system description can be thought of as a single high-level *system* package with everything else in it. All kinds of UML model elements and diagrams can be organised into packages.

Packages own model elements and model fragments and are the basis for configuration control. Each model element can be directly owned by a single package, so the package hierarchy is a strict tree.

Packages can reference other packages so the usage network is a graph. Relationships drawn between package symbols denote relationships between at least some of the elements in the packages. A dependency relationship implies one or more dependencies among the model elements, in which a change in the targeted element may require a change in the source element.

3.6
A **class** is the descriptor for a set of objects with similar structure, behavior, and relationships. UML provides notation for declaring classes and specifying their properties, as well as using classes in various ways. Classes are declared in class diagrams and used in most other diagrams. UML provides a graphical notation for declaring and using classes, as well as a textual notation for referencing classes within the descriptions of other model elements.

3.7
A **class diagram** is a graph of Classifier elements connected by their various static relationships. (Note that a ‘‘class’’ diagram may also contain interfaces, packages, relationships, and even instances, such as objects and links. Perhaps a better name would be ‘‘static structural diagram’’ but ‘‘class diagram’’ is shorter and well established.)

3.8
An **object** represents a particular instance of a class. It has identity and attribute values. The same notation also represents a role within a collaboration because roles have instance-like characteristics.

3.9
An **operation** is a service that an instance of the class may be requested to perform. It has a name and a list of arguments.

3.10
A binary **association** is an association among exactly two classes (including the possibility of a reflexive association from a class to itself).

3.11
Generalization is the taxonomic relationship between a more general element and a more specific element that is fully consistent with the first element and that adds additional information. It is used for classes, packages, use cases, and other elements.

3.12
A **sequence diagram** represents an **Interaction**, which is a set of messages exchanged among objects within a collaboration to effect a desired operation or result. A sequence diagram shows an interaction arranged in time sequence. In particular, it shows the objects participating in the interaction by their ''lifelines'' and the messages that they exchange arranged in time sequence. It does not show the associations among the objects.

In addition to the UML definitions cited above, the methodology described in Part 4 and Annex A uses some additional semantics.

3.13
The **system boundary** depicted in a sequence diagram maps to the same entity in a use case diagram. Thus any interaction emanating or terminating in the system boundary involves an actor.

In the methodology classes are invented for one of three purposes: information, control and interface.

An information class defines objects, which will store data relevant to the operation of the system and the actors and maintain that data with database like services.

A control class defines objects whose primary purpose is to implement the functions of the system.

An interface class defines objects that perform the data presentation and application interfaces for the actors.

3.14
The **architecture boundary** divides the interface classes from those classes, which form the actual architecture, namely the control classes and the information classes.

In the sequence diagrams developed in later clauses there is often an implicit interaction across the system boundary involving an actor. This may be implied whenever a message is initiated or terminated at an interface class.

4 Symbols and abbreviated terms

4.1 Use Case diagram

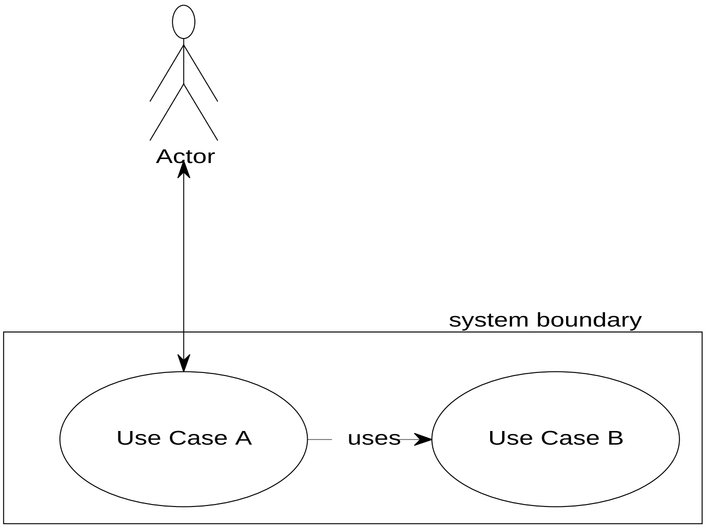

Figure 1 — Use Case diagram

4.2 Package Diagram

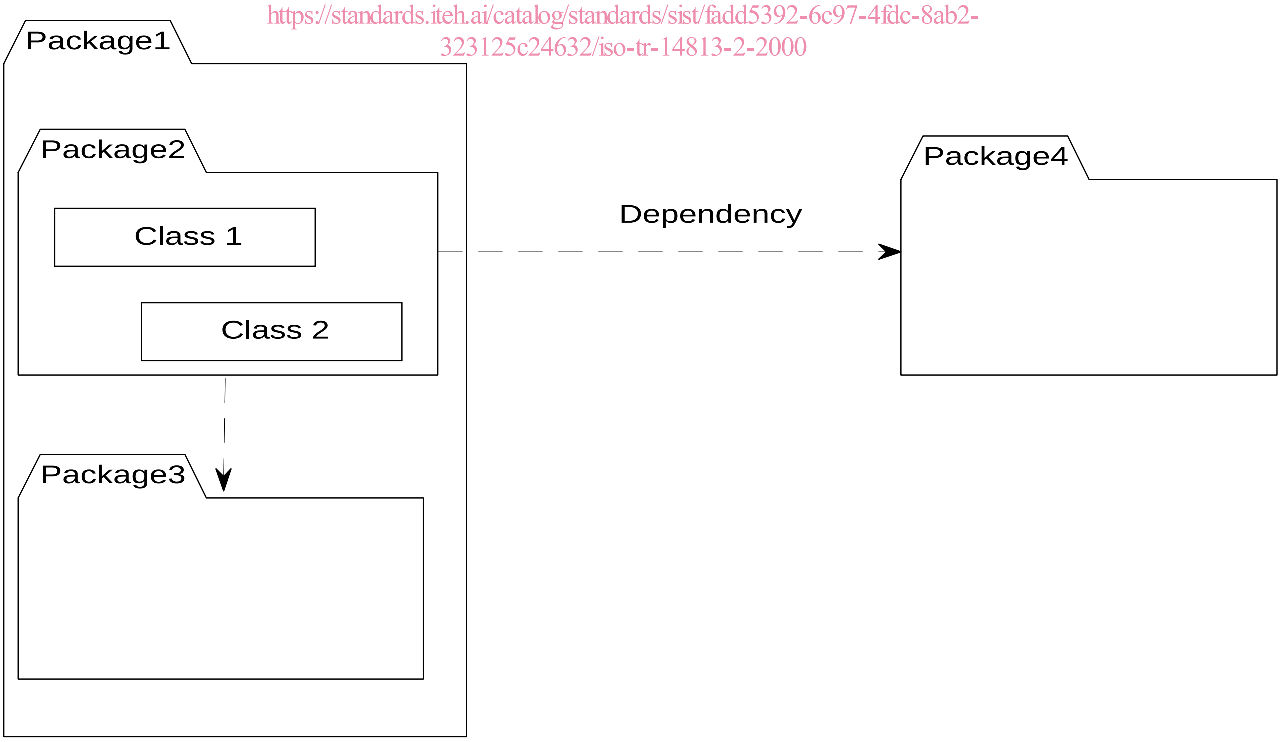

Figure 2 — Package Diagram