TECHNICAL REPORT

ISO/TR 25102

First edition
2008-02-15

Intelligent transport systems — System architecture — “Use Case” pro-forma template

Systèmes intelligents de transport — Architecture des systèmes — Gabarit pro forma de «cas d'usage»

Reference number
ISO/TR 25102:2008(E)

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

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.

In exceptional circumstances, when a technical committee has collected data of a different kind from that which is normally published as an International Standard ("state of the art", for example), it may decide by a simple majority vote of its participating members to publish a Technical Report. A Technical Report is entirely informative in nature and does not have to be reviewed until the data it provides are considered to be no longer valid or useful.

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/TR 25102 was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

Introduction

The objective of this Technical Report is to propose a pro-forma template for “Use Cases” for intelligent transport systems (ITS), and provide guidance on how the template should be used.

While this Technical Report provides a pro forma template, the elements may be augmented or omitted as applicable. The Technical Report provides guidance to develop use cases and is a guide rather than a prescription to be followed without variation.

A “Use Case Model” is simply a term to describe, and in many cases define, a user's view of interactions with (and within) the system. Use cases show how entities interact, and are usually presented as structured text or diagrammatically.

“Use Cases” are a means to define requirements for a system in terms of the primary users (known as actors) that interact with the system and the scenarios or activities that are performed by the system in response to stimuli from the actors or from other system entities. Each “Use Case” has a starting state and conditions, a series of activity steps that together comprise a scenario, and a finishing state and conditions. There may be more than one scenario in a “Use Case”. The “Use Case” should also include exceptional cases with alternate outcomes.

In many situations, including some International Standards, there has been more attention paid to the definition of “Actors”, “Use Cases” and the relationships between them, rather than the detail of each “Use Case”, especially the explanatory text that goes with the “Use Case”.

The identification of “Use Cases” is most frequently associated with use case model diagrams using the “Unified Modelling Language” (UML)[4]. In this Technical Report, for consistency, this methodology is used throughout. However, “Use Cases” can be elaborated and developed for any system methodology and are as appropriate for process oriented methodology as object oriented methodology, and indeed there is no requirement to use any technical architecture methodology at all. A “Use Case” can often be elaborated simply with pen and paper.

The benefits of applying use cases to the development of ITS include the following:

— Common, standardized approach available for the first segment of software system development, namely requirements elicitation and definition;

— Requirements are related to each other informally, thus providing some assurance of compatibility and consistency.

Intelligent transport systems — System architecture — "Use Case" pro-forma template

1 Scope

This Technical Report discusses the application of "Use Cases" for requirements and related aspects of a software-intensive system such as an intelligent transport system (ITS).

The scope of this Technical Report is to provide a pro-forma template for the consistent consideration and development of "Use Cases" within ITS International Standards and associated deliverables.

NOTE This Technical Report provides a pro forma template; the elements may be augmented or omitted as applicable. It provides guidance to develop use cases and is a guide rather than a prescription to be followed without variation.

2 Terms and definitions

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

2.1
architecture
set of concepts and rules describing the inter-relationship between entities in the entire system, independent of the hardware and software environment. [Mil4]; architecture is described through a series of views that may be at varying levels of generality/specificity, abstraction/concretion, totality/component, and so on

2.1.1
system architecture
framework for ITS deployments; it is a single, high-level description of the major elements or objects and the interconnections among them

NOTE It provides the framework around which the interfaces, specifications and detailed system designs can be defined. An architecture is not a product design, nor a detailed specification for physical deployment. [15]

2.2
business rule
rigorous statement of policy, sometimes expressed in the format IF…THEN…ELSE…, that must be followed when the stated conditions are satisfied

2.3
condition
state of an entity or a set of state variables; also the qualification of a contract or agreement

2.4
exception
departure from normal or correct operation

2.5
model
representation of an entity from which the important elements have been abstracted by removing unimportant detail while at the same time retaining the interrelationship between the key elements of the whole

NOTE A model can be made more or less abstract by the successive suppression of detail such that the concepts and relationships come into enhanced focus and become more readily understood. However, the process can be taken too far when the simplification has exceeded the threshold where a necessary understanding is possible. Thus, the process of modelling is one of going only far enough to achieve the optimum understanding and insight – and no further.

2.6
primary actor
principal entity interacting with the system being described to achieve an objective

NOTE An *actor* defines a coherent set of roles that users of an entity can play when interacting with the entity. An actor may be considered to play a separate role with regard to each “Use Case” with which it communicates. An Actor has a Name and may communicate with a set of Use Cases. An Actor may also have a set of Interfaces, each describing how other elements may communicate with the Actor. An Actor may have generalization relationships to other Actors.

2.7
relation
nature of how two entities affect each other including dependencies

2.8
requirement
statement of user need, typically expressed in a single-sentence form to assist with later verification of compliance

2.9
scenario
sequence of steps that are taken to change the state from that before the scenario to that immediately after the scenario

2.10
template
framework that may be used repeatedly to meet requirements that are similar to some extent

2.11
trigger
event that starts the process in a scenario

2.12
use case
series of interactions between an outside entity and the system, which ends by providing business value

NOTE A “Use Case” is a sequence of actions that an actor (usually a person, but perhaps an external entity, such as another system) performs within a system to achieve a particular goal [16].

2.13
user
actor that derives benefit from the normal end state of the “Use Case”.

3 Abbreviated terms

3.1
ITS
intelligent transport system

3.2
TICS
transport information and control systems

NOTE TICS is a term that has been largely superseded by ITS.

3.3
TR
technical report

3.4
UML
unified modelling language

3.5
WG 1
ISO/TC 204 Working Group 1, *Architecture*

4 Background

4.1 TC 204 Working Group WG 1

This Technical Report arose from work by ISO/TC 204/WG 1 on the elaboration of ITS architecture in the ISO 14813 series of International Standards. It had become apparent that the application of use cases was arbitrary and inconsistent and therefore required more explicit guidance.

4.2 Statement of requirements

There have been many proposed methods for the statement and management of requirements, the most common being a tabular collection of single verifiable statements. The problem then arises that with large numbers of requirements, the relationships between them become less and less clear.

Various strategies are employed effectively to organize (or manage) the requirements base and among these “Use Cases” have become increasingly popular because they can be understood readily by all stakeholders of a complex system. This is the most important motivation for this Technical Report to be developed – “Use Cases” are an important means to achieve consensus of all stakeholders.

4.3 Use cases

The first question to be answered is: What is a “Use Case”? Why use this term to define a very simple process that is essential to the creation of software or any other business system?

A “Use Case Model” is simply a term to describe, and in many cases define, a user's view of interactions with (and within) the system. Use cases show how entities interact, and are usually presented as structured text or diagrammatically.

The “Use Case” is just a method to facilitate information exchange. The commodity being acquired is the business knowledge of the stakeholder. This knowledge is transferred to a programmer/architect who then uses their expertise to develop a system or software to enable a service or system to function. In order to create quality software an explanation of how the system is to function must be clearly understood by all involved. Therefore, a “Use Case” stands as a mutually understood and accepted representation of how a user, commonly referred to as an actor, interacts with a desired or existing system.

Importantly, the purpose of describing use cases is not to fully specify the exact nature of what its subject will contain and how it is to be built. Instead, use cases define goals and purpose: the problems we are trying to solve. Establishing these goals lays the foundation for the scope that will follow. Additionally:

— If we simply consider the roles played by the actors, and the goals of those actors, the use-case model can very rapidly emerge.

— Use-case diagrams can distil a complex project into a more easily comprehensible picture.

— A well-constructed use-case model can be understood by all the stakeholders in a project: developers, managers and clients. It's a powerful aid to collaborative development.

— Use cases ensure that scope is under control from the outset. The identification of use cases and their dependencies makes it easy to distinguish between core goals that must be satisfied and subsidiary enhancements that may be postponed.

— Scoping in this manner allows for better planning and prioritization.

— It's an *implementation-neutral* picture of a project. No assumptions about tools and technologies are made, nor should they be.

— Use case methodology is transportable. No special tools are required — sticky notes, a whiteboard, pencil and paper, or your favourite graphics application can all be used to document your vision.

Use-case driven development is a mindset, as much as it is a technique. By emphasizing the actors and what they wish to achieve, project teams can advance with greater confidence and clarity.

A good method is to describe a purpose (which is the requirements). The “Use Case” contents are written in consistent prose. It may have plurality where several scenarios can be described per use case. The typical use case structure is semi-formal.

“Use Cases” have become widely adopted due to the recognition of their simplicity and effective means to express activity in interconnected systems and components. This has also been recognized in the development of the “Unified Modelling Language” (UML) that has singled out the “Use Case” as the most appropriate means to summarize requirements in a dynamic system context. Nevertheless, use cases can and do exist separately from UML.

There are so many interpretations of how to structure a “Use Case” that it can get confusing. It is best to realize that there is no “right” way, only the best way for that particular situation. The job of a “Use Case” is to ascertain that best way and maintain focus on the essential business need.

It is a good idea to employ use cases when developing or changing software; this practice can keep everyone in the same ballpark. Part of the difficulty in software development is making it clear what the “customer” really needs in order to get on with the business of their business.

The purpose of a “Use Case Template” is to provide a consistently rigorous approach in order to provide quality to the result. While the form of the template remains consistent, not all fields need to be populated. Empty rows may be deleted. If appropriate, additional rows may be added to the template.

4.4 “Unified Modeling Language” (UML)

UML is the subject of standardization (ISO/IEC 19501) and this does provide a brief description of a “Use Case” as well as the normative requirements for “Use Case” model. In clause 4.11, ISO/IEC 19501 says: *“The purpose of a “Use Case” is to define a piece of behaviour of an entity without revealing the internal structure...”(p126) “Each use case specifies a service the entity provides to its users; ...”(ibid)*

“A use case can be described in plain text, ...” (p127)

and this Technical Report is intended to demonstrate how to elaborate use cases so that public and business policy, planning and management can understand and agree on them (without need for any UML notation, or knowledge of the detail of UML).

Then, having achieved that level of broad agreement, it is a natural step to elaborate the use case model (by specialization, <<includes>> and <<extends>>) to maintain consistency and traceability.

ISO/IEC 19501 states "*...use cases can be used to specify subsystems and classes."(ibid)* and goes on to say: *"The usage of use cases at all levels imply not only a uniform way of specification of functionality at all levels, but also a powerful technique for tracing requirements at the system package level down to operations of classes."*(ibid).

This Technical Report provides a practical basis for such provisions to be achieved, and is achieved through effective application of "Use Cases" and Actors in a "Use Case" Model, and the detailed specification of each "Use Case" following the template described below.

A primary benefit of this approach will be the standardization of a use case pattern in textual form that is suitable for a wide-ranging audience who is unfamiliar with UML, but who can gain the benefit of drafting, reviewing and approving "Use Cases" as a powerful design artefact suitable for all stakeholders.

The most important thing about use cases is the text.

NOTE There have been cases in industry recently where the UML "Use Case" diagrams have been over-engineered, with all sorts of stereotyped relationships between use cases that are quite often simply wrong. This kind of thing puts people off UML in general, which is regrettable. UML a useful technique but a use case diagram is merely a convenient pictorial index into the essential textual descriptions.

4.5 The bottom line value of "Use Cases"

The bottom line is that "Use Cases" are effective and widely used; they therefore require some degree of uniformity and consistency to assist in their use. This TR is intended to provide guidance to help to achieve such uniformity.

Use cases focus on the users of the system, not the system itself, thus the *real* system needs are brought to light early on. Since a "Use Case" consists mainly of narrative text, it is easily understandable by all stakeholders, including customers, users and executives, not just developers and testers. By including all the stakeholders during the early planning stages of a project, you bring in people who best understand the problems at hand, promote a sense of buy-in from end users, and eliminate surprises when the system is deployed.

Each "Use Case" describes one way the system is used, but one of the big benefits of use case modelling is that it also describes *all of the things that might go wrong*. Identifying exceptions to a successful scenario early in the project saves a lot of time by finding subtle requirements.

Finally, once a use case model has been developed, it can be used to drive many other aspects of software development, including project planning (cost, complexity and timing estimates), object models, test case definitions, and user documentation.

The crucial benefit of use cases is the way they encourage a directed method of considering project requirements. From the very beginning, we are designing a product by concentrating upon the needs and wants of those who will use it.

The better our understanding of a particular use case, the easier, more focussed, and more appropriate will be the work that follows. Use cases are the *context* that allows us to easily picture where, within a project's life, a particular element will fit, thus promoting clearer decision-making throughout design and development.

There needs to be a business case for any International Standard or associated deliverable. Without a viable business case, a standard will lie unused in the library. The “Use Case” is a means to explain and support a business model to its potential sponsors/clients. In the context of system architecture, most of the deliverables from ISO/TC 204/WG 1 refer to or imply a business case, and in practice will benefit from the development of use cases. The following WG 1 deliverables will, in implementation, require or will benefit from, the development of a “Use Case”:

— ISO 14813 (all parts), *Intelligent transport systems — Reference model architecture(s) for the ITS sector*

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

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

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

— ISO/TR 24098, *Intelligent transport systems — System architecture, taxonomy and terminology — Procedures for developing ITS deployment plans utilizing ITS system architecture*

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

— 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*

— ISO/TR 25100, *Intelligent transport systems — Systems architecture — Harmonization of ITS data concepts*

— ISO/TR 28682, *A joint APEC-ISO study of progress to develop and deploy ITS Standards*

This also applies to almost all of the deliverables of other working groups of ISO/TC 204.

5 Use case elements

5.1 General

The following describes the sections proposed for use in a typical text-based “Use Case”. However, the elements may be augmented or omitted as applicable. When an element is used, it is strongly recommended that it be used in the manner described below.

If the required element differs from that stated, it is recommended that a new and differentiated element is created with a unique element title that does not overload any title already stated below.

5.2 Normal use cases description

A normal “Use Case” includes a *Title*, a *Primary Actor* field, a *Participants* field, a *Goal* field, a *Precondition*, a *Postcondition*, a sequence of *Steps*, a set of *Any Extensions*, and a set of *Extension Points*. The description of these elements is as follows.

A use case description can be seen as a two-part description with:

— a static part that includes the use case “Title”, “Primary Actor”, “Participants”, “Goal”, “Precondition” and “Postcondition” fields, and

— a dynamic or procedural part that consists of the use case “Steps”.

5.3 “Use Case” template

This pro-forma template has been created by amalgamation of several recommended templates together with practical experience of the drafting committee.

5.4 “Use Case” title name

The name of the “Use Case” is normally expressed as a short verb phrase, e.g. “Monitor Pedestrian Traffic”, and should be chosen from the viewpoint of the user.

Title: a label that uniquely identifies the “Use Case” within the use case model.

5.5 “Use Case” description

A short abstract of the intent and scope of the “Use Case”.

5.6 “Use Case” scope

A concise specification of what is included in the “Use Case” (and by implication what is excluded).

Scope: specifies what system is being considered black box under design (e.g. whole business, system, sub-system, function).

5.7 “Use Case” level

Is the “Use Case” level of detail. Use cases are typically at the level of an individual user goal, but they can also be higher level (expressing a more abstract business goal), or lower level (expressing a sub-task of a user goal).

5.8 Target system release

Where applicable, the target system release to which this “Use Case” applies.

5.9 “Use Case” level of generality or abstraction

Use cases may be described at different levels of abstraction or of extent. They may be generalized and broad or defined with greater precision and specification. (as in geographical or other physical extent such as stages in a supply chain) or both. This concept can be difficult to understand, but in essence is quite simple (see Figure 1).

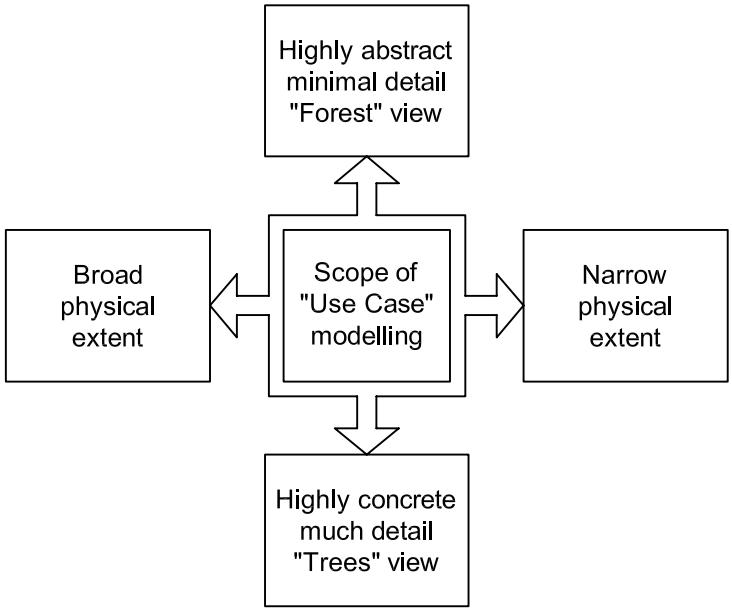

Figure 1 — “Use Case” range of scope

The advantage of a generalized view is that it can easily be comprehended by a non-expert. The analogy in Figure 1 equates this view to that of looking at a forest from a distant viewpoint. However, it may not be precise enough for a system to be elaborated. A more detailed view will be required to enable a system analyst to take the generalized view to a specification. The analogy here would be looking at the trees in the forest rather than the forest as a whole.

5.10 "Use Case" author/primary actor

The primary author of the "Use Case" with their contact details.

Primary Actor: the actor that initiates the use case.

5.11 "Use Case" stakeholders/participants

An explicit listing of the stakeholders with a primary interest in this "Use Case". This list shall include the stakeholders, which have a primary relationship with this "Use Case". The approver of the "Use Case" should be one of the stakeholders.

Participants: other actors participating in the use case.

5.12 "Use Case" goal

The goal or intended primary successful outcome for this "Use Case".

Goal: a statement of the primary actor expectation at the successful completion of the use case.

5.13 "Use Case" references to requirements

A "Use Case" is a powerful means to capture requirements in a form where their interrelationships are visible and may be considered in combination (rather than in isolation).

5.14 "Use Case" assumptions

Any assumptions that are implicit in the "Use Case" should be stated explicitly so they may be reviewed and agreed by stakeholders.

5.15 "Use Case" technology restrictions

Any restrictions in the achievement of the "Use Case" that arise from choice of technology.

5.16 Relationship to other "Use Cases"

This is the textual form of the "Use Case" model and should be expressed in tabular form (as well as the usual UML diagram if required).

NOTE Some tools are available that will support this process.

5.17 Actors associated with "Use Case"

Primary "Actors" should be identified. An "Actor" is a role that an entity enacts (e.g. driver, passenger, manager, etc.). The same entity may perform several roles and hence appear as several actors [e.g. a driver of a vehicle may, from a slightly different perspective, also be considered (modelled) as a vehicle occupant (see Figure 2)].

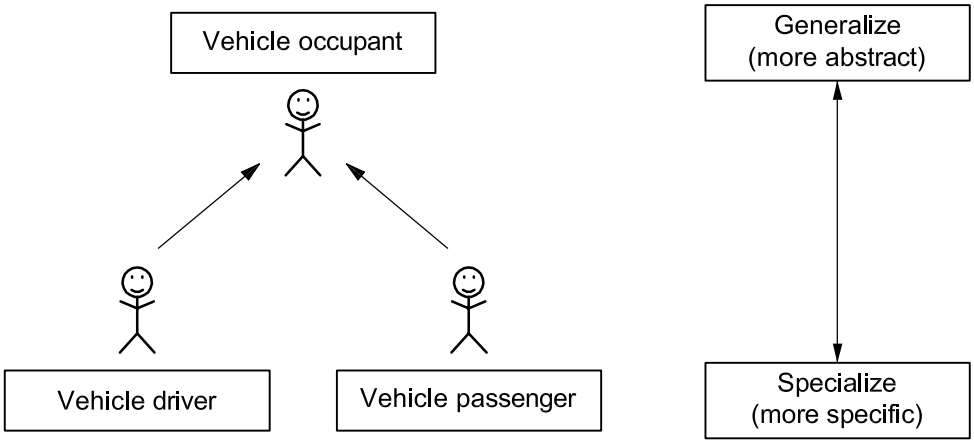

Figure 2 — Actors and roles: Degrees of abstraction-specification

5.18 "Use Case" triggers

Theses are the events that give rise to the start of the scenario.

5.19 "Use Case" pre-conditions

The pre-existing conditions that pertain in order for the "Use Case" to be feasible.

Precondition: a condition that must hold before an instance of the "Use Case" can be executed.

5.20 "Use Case" scenarios

A "Use Case" will comprise one or more concurrent or alternative scenarios.

5.20.1 Scenario (S)

The single thread of activity within a scenario comprising a series of consecutive steps.

5.20.1.1 Step (S.t)

One of many steps within a single scenario.

Steps: a sequence of steps. Each step references a use case step operation. A use case step operation may be a "concept operation instance", a "branching statement" or a "reference to an included use case".

NOTE A step may also include a set of "*step extensions*" each defining an alternative behaviour following the step.

5.21 "Use Case" expected outcomes

The expected or desired normal outcome of the "Use Case" – when all things go as intended.

5.22 "Use Case" acceptance criteria

The criteria that must be met for the "Use Case" outcomes to be generally accepted as successful. The acceptance criteria should be developed with consultation of all stakeholders and their consensus is very desirable.

5.23 “Use Case” exceptions

Exceptions are undesirable or unintended outcomes arising from variations in one or more scenarios.

NOTE The point of departure where the abnormal event occurs and departs from the normal sequence of events in a scenario should be indicated as clearly identifiable discrete step(s) within the scenario description.

5.24 “Use Case” post-conditions

The detailed conditions that are expected to pertain after completion of all scenarios. This set of conditions must all be satisfied.

Postcondition: a condition that must be true at the end of a “successful” execution of an instance of the use case.

NOTE If the post-conditions are not fully satisfied, the “Use Case” still has more possible steps to complete and is therefore by definition incomplete.

5.25 “Use Case” extensions

UML and general practice allows for further stages to follow a “Use Case” and these can be usefully provided by extensions.

A “Use Case Extension” is:

— *Any extensions:* a set of “*step extensions*” that applies to all the steps in the use case.

— *Extension points:* a set of locations in the use case steps where additional behaviours defined in extension use cases might be inserted.

5.26 “Use Case” inclusions

A discrete “Use Case” may form part of a more extensive “Use Case” that has already been developed. This can be shown in a “Use Case” model as an inclusion.

A use case *inclusion statement* refers to an included use case. The meaning of a use case inclusion is that the steps of the included “Use Case” replace the inclusion statement in the base use case. The remaining steps of the base “Use Case” are normally executed after the included steps.

The various forms of user interfaces used by the actors should be described, with special attention to avoid overlooking the less obvious interfaces.

5.27 “Use Case” business rules

There may be important business rules required by stakeholders and these should be stated explicitly.

5.28 “Use Case” verification approach

When the system is implemented and the time comes to verify that the “Use Case” has been satisfied (and the acceptance criteria have been met), there are several different approaches that may be used for verification.

The description of the functional, behavioural, and structural system essentials form central components of any serious requirements specification. In the UML, for example, functional requirements are expressed by use cases, their behaviour is modelled by e.g. activity graphs, and structural requirements are captured by class models.

At the highest level, use case verification starts with a detailed inspection of the formal parts of the use case model (e.g. pre- and post-conditions). Then the (refined) class model is checked to detect incomplete, inconsistent, and/or ambiguous specifications and/or missing items. Finally, each «*interaction*» specification is examined concerning compliance with its class scope and the specification of the root operation.

When these detailed inspections have been completed, the use case model is verified. The class model is now consistent and unambiguous.

5.29 "Use Case" test cases

Where test cases are available and identified for this "Use Case", it should be recorded here.

5.30 "Use Case" version

Each "Use Case" will have a life cycle and shall be versioned and subjected to formal change management procedures.

5.31 "Use Case" modification history

The "Use Case" modification history is the historical record of changes, summarized or referenced.

5.32 "Use Case" approvals

The formal approvals for each "Use Case" shall be recorded and a reference to the record shall be shown in the "Use Case".

5.33 "Use Case" application notes

As experience is gained with a "Use Case", this section can be used to record application notes.

5.34 "Use Case" open issues

As issues arise with each "Use Case", it is recommended to make a record of them within the "Use Case" itself, in order to ensure the record tracks with changes to the "Use Case" itself. If this becomes cumbersome, then it may be achieved by reference to an issues register.

6 Suggested "Use Case Template"

6.1 "Use Case Template" rationale

The development of a "Use Case" pro-forma template is supported by practitioners and in literature, and assists with the maintenance of a rigorous and quality approach to the development of a use case.

6.2 Form of "Use Case Template"

A textural form of "Use Case" may or may not be based on UML definition. However, where it is based on ISO/IEC 19501, which states that use cases may be used at any hierarchical level, it may be expressed in textual form.

6.3 Example "Use Case Template"

This Technical Report proposes the use of the template provided in Table 1, or a similar construct.

Table 1 — Example "Use Case Template"

USE CASE TEMPLATE/OUTLINE TABLE OF CONTENTS	
Use Case Name	
"Use Case" Description	
"Use Case" Scope	
"Use Case" level	
Target System Release	
Generality/Abstraction level	
"Use Case" Author/ Primary actor	Name: Phone: Email: @
"Use Case" Stakeholders	
"Use Case" Goal	
"Use Case" Requirements Reference	
"Use Case" Assumptions	
"Use Case" Technology Restrictions	
Relationships to other "Use Case(s)"	
Actors associated with this "Use Case"	
"Use Case" Triggers	
"Use Case" Pre-conditions	
"Use Case" Scenario #Scenario.	REPEAT AS NECESSARY TO COVER ALL SCENARIOS
"Use Case" Step #Scenario.#Step	REPEAT AS NECESSARY TO COVER ALL SCENARIOS
"Use Case" Expected Outcomes	
"Use Case" Acceptance Criteria	
"Use Case" Exceptions	
"Use Case" Post-conditions	
"Use Case" Extensions	
"Use Case" Inclusions	
"Use Case" Business Rules	
"Use Case" Verification Approach	
"Use Case" Test Cases	
"Use Case" Version	
"Use Case" Modification History	
Template Version	1.0 ISO/TR 25102:2008
Template Modification history	
Approvals	
Application Notes:	
Open Issues	

Bibliography

[1] ISO 14813 (all parts), *Intelligent transport systems — Reference model architecture(s) for the ITS sector*

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

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

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

[5] ISO/TR 24098, *Intelligent transport systems — System architecture, taxonomy and terminology — Procedures for developing ITS deployment plans utilizing ITS system architecture*

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

[7] 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*

[8] ISO/TR 25100, *Intelligent transport systems — Systems architecture — Harmonization of ITS data concepts*

[9] ISO/TR 28682, *A joint APEC-ISO study of progress to develop and deploy ITS Standards*

[10] COCKBURN, A. *Writing effective use cases*, Addison Wesley. 2001

[11] CONSTANTINE, LARRY, L. *Beyond chaos. The expert edge in managing software development,* Addison Wesley. 2001

[12] GOLD-BERNSTEIN, B. and RUH, W. Enterprise integration, *The essential guide to integration solutions,* Addison-Wesley. 2005

[13] GOMAA, H. *Designing software product lines with UML,* Addison Wesley. 2005

[14] KULAK, D. and GUINEY, E. *Use cases, Requirements in context*, 2nd ed., Addison Wesley. 2004

[15] MILES, and CHEN, K. (Eds). *PIARC: ITS handbook* - 2nd ed., World road association (PIARC). 2004

[16] ROSENBERG, D. and KENDALL, S. *Use case driven object modelling with UML*, *A practical approach*, Addison Wesley. 2004

ICS 03.220.01; 35.240.60

Price based on 13 pages