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»

iTeh STANDARD PREVIEW
(standards.iteh.ai)

ISO/TR 25102:2008
https://standards.iteh.ai/catalog/standards/sist/930b3848-4d84-4203-9915-7959ae8c484e/iso-tr-25102-2008

Reference number
ISO/TR 25102:2008(E)

© ISO 2008

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.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/TR 25102:2008
https://standards.iteh.ai/catalog/standards/sist/930b3848-4d84-4203-9915-7959ae8c484e/iso-tr-25102-2008

COPYRIGHT PROTECTED DOCUMENT

© ISO 2008

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

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 ... v
Introduction ... vi
1 Scope ... 1
2 Terms and definitions ... 1
3 Abbreviated terms ... 2
4 Background ... 3
4.1 TC 204 Working Group WG 1 ... 3
4.2 Statement of requirements ... 3
4.3 Use cases ... 3
4.4 “Unified Modeling Language” (UML) ... 4
4.5 The bottom line value of “Use Cases” ... 5
5 Use case elements ... 6
5.1 General ... 6
5.2 Normal use cases description ... 6
5.3 “Use Case” template ... 7
5.4 “Use Case” title name ... 7
5.5 “Use Case” description ... 7
5.6 “Use Case” scope ... 7
5.7 “Use Case” level ... 7
5.8 Target system release ... 7
5.9 “Use Case” level of generality or abstraction ... 7
5.10 “Use Case” author/primary actor ... 8
5.11 “Use Case” stakeholders/participants ... 8
5.12 “Use Case” goal ... 8
5.13 “Use Case” references to requirements ... 8
5.14 “Use Case” assumptions ... 8
5.15 “Use Case” technology restrictions ... 8
5.16 Relationship to other “Use Cases” ... 8
5.17 Actors associated with “Use Case” ... 8
5.18 “Use Case” triggers ... 9
5.19 “Use Case” pre-conditions ... 9
5.20 “Use Case” scenarios ... 9
5.21 “Use Case” expected outcomes ... 9
5.22 “Use Case” acceptance criteria ... 9
5.23 “Use Case” exceptions ... 10
5.24 “Use Case” post-conditions ... 10
5.25 “Use Case” extensions ... 10
5.26 “Use Case” inclusions ... 10
5.27 “Use Case” business rules ... 10
5.28 “Use Case” verification approach ... 10
5.29 “Use Case” test cases ... 11
5.30 “Use Case” version ... 11
5.31 “Use Case” modification history ... 11
5.32 “Use Case” approvals ... 11
5.33 “Use Case” application notes ... 11
5.34 “Use Case” open issues ... 11
6 Suggested “Use Case Template” ... 11
6.1 “Use Case Template” rationale ... 11
6.2 Form of “Use Case Template” ... 11

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/TR 25102:2008
https://standards.iteh.ai/catalog/standards/sist/930b3848-4d84-4203-9915-7959ae8e484e/iso-tr-25102-2008

6.3 Example “Use Case Template” 11
Bibliography 13

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/TR 25102:2008
https://standards.iteh.ai/catalog/standards/sist/930b3848-4d84-4203-9915-7959ae8c484e/iso-tr-25102-2008

© ISO 2008 – All rights reserved

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.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

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*.

ISO/TR 25102:2008
https://standards.iteh.ai/catalog/standards/sist/930b3848-4d84-4203-9915-7959ae8c484e/iso-tr-25102-2008

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”.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/TR 25102:2008
https://standards.iteh.ai/catalog/standards/sist/930b3848-4d84-4203-9915-7959ae8c484a/iso-tr-25102-2008

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.

© ISO 2008 – All rights reserved

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

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/TR 25102:2008
https://standards.iteh.ai/catalog/standards/sist/930b3848-4d84-4203-9915-7959ae8c484e/iso-tr-25102-2008

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

iTeh STANDARD PREVIEW
(standards.iteh.ai)

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

ISO/TR 25102:2008
https://standards.iteh.ai/catalog/standards/sist/930b3848-4d84-4203-9915-7959ae8c484e/iso-tr-25102-2008

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

© ISO 2008 – All rights reserved

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.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/TR 25102:2008
https://standards.iteh.ai/catalog/standards/sist/930b3848-4d84-4203-9915-7959ae8c484e/iso-tr-25102-2008

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.

iTeh STANDARD PREVIEW
(standards.iteh.ai)
ISO/TR 25102:2008
https://standards.iteh.ai/catalog/standards/sist/930b3848-4d84-4203-901d-7959ae8e484e/iso-tr-25102-2008

“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)

© ISO 2008 – All rights reserved

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.

iTeh STANDARD PREVIEW
(standards.iteh.ai)

4.5 The bottom line value of "Use Cases"

ISO/TR 25102:2008
https://standards.iteh.ai/catalog/standards/sist/930b9848-4d84-4205-9915-7959ae8c484e/iso-tr-25102-2008

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.

© ISO 2008 – All rights reserved