TECHNICAL REPORT

ISO/TR 26999

First edition
2012-10-15

Intelligent transport systems — Systems architecture — Use of process-oriented methodology in ITS International Standards and other deliverables

Systèmes intelligents de transport — Architecture de systèmes — Emploi d'une méthodologie orientée processus dans les Normes internationales ITS et autres produits livrables

Reference number
ISO/TR 26999:2012(E)

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is entirely informative in nature and does not have to be reviewed until the data it provides are considered to be no longer valid or useful.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 26999 was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

Introduction

The objective of this Technical Report (TR) is to provide guidance on the use of the process-oriented method (POM), also known as data flow modelling, in the development of intelligent transport systems (ITS) International Standards and other deliverables, and in the design and implementation of ITS systems. In particular, it is intended to be used as the basis for the development of high-level system architectures for ITS. These architectures are tools to aid ITS implementations, and a mechanism to identify and promote the creation and use of standards.

The advantages of applying POM to the development of high-level system architectures for ITS include the following:

— POM is easily understood, particularly by non-technical people (e.g. decision-makers) who are often the intended audience for high-level system architectures;

— POM enables a coherent description to be built up from multiple user views;

— training and tool support is available, particularly in Europe and the USA;

— the data descriptions produced by POM are capable of manipulation by a metadata registry for ITS;

— the results of creating a POM system architecture can be easily transferred into requests for quotations (RFQs), expressions of interest (EOIs), tenders and other similar documents;

— the results of POM system architectures can be translated into UML for use by software developers;

— POM is applicable to both hardware and software and does not, therefore, pre-suppose the form in which its functionality will be implemented.

The disadvantages of using POM include the following:

— POM has a bad image, e.g. it is old-fashioned, and is usually not now included in the training of systems analysts and designers;

— parts of a POM system architecture might require conversion to UML before it will be accepted by most software developers.

There are some risks in using POM, but the benefits of its ability to be easily understood by the usual initial audience for high-level system architectures can often help with the initial promotion of ITS implementations. This TR is intended to provide guidance to stakeholders who are considering the use of POM for ITS.

Intelligent transport systems — Systems architecture — Use of process-oriented methodology in ITS International Standards and other deliverables

1 Scope

The scope of this Technical Report is the use of the so-called process-oriented method (POM) in International Standards, Technical Specifications, Technical Reports and related documents.

This Technical Report discusses the use of POM in the development of high-level system architectures for intelligent transport systems (ITS). It is based on the results of the work of the FRAME-S project and the FRAME Forum. Much of the text from Clause 2 through to the end of the document is therefore reproduced by kind permission of the European Commission and the FRAME Forum.

2 Terms and definitions

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

2.1
actor
sub-element of a terminator

NOTE It is mainly used to enable a particular variant of a terminator to be differentiated from other variants, e.g. to differentiate a public transport vehicle driver from any other type of driver, or all drivers.

2.2
architectural model
model that contributes to the content of an architectural view

2.3
architecture (generic definition)
set of concepts and rules for a system that describes the inter-relationship between entities in the entire system, independent of the hardware and software environment

NOTE Architecture is described through a series of viewpoints that might be at varying levels of generality/specificity, abstraction/conception, totality/component, and so on. See also “communications viewpoint”, “functional viewpoint”, “organizational viewpoint” and “physical viewpoint” definitions below.

2.4
architectural viewpoint
representation of a system from the perspective of an identified set of architecture-related concerns

2.5
architectural description
collection of information items used to describe an architecture

2.6
aspiration
expression of what a stakeholder wants the ITS implementation to provide, usually written in the language of the stakeholder and thus possibly having little or no formal structure

NOTE There could be many aspirations for each ITS implementation, depending on its scope and the number of stakeholders that are involved.

2.7
communications viewpoint
one of several architectural viewpoints of the system of interest, showing the links between the building blocks in the physical viewpoint that will enable them to communicate with each other, and including details of expected data throughputs and any other constraints that will affect the eventual choices of communications hardware and software

2.8
functional viewpoint
one of several architectural viewpoints of the system of interest, showing the functionality that will be needed to fulfil the requirements expressed in the user needs, this functionality being shown as a series of functions and data stores plus the data flows between them and the data flows between the functions and the terminators

2.9
ITS architecture
specific form of a system architecture for use as a tool in the initial stages of an ITS implementation

2.10
model
representation of an entity from which the important elements have been abstracted by removing certain detail while at the same time retaining the interrelationship between the key elements of the whole

NOTE 1 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 can be achieved. Thus the process of modelling is one of going only far enough to achieve the optimum understanding and insight — and no further.

NOTE 2 A model is a way of representing something, other then in its natural state (see “Models of ITS” documents at the web site given in Reference [2] in the Bibliography).

2.11
organizational viewpoint
one of several architectural viewpoints of the system of interest, showing how the building blocks from the physical viewpoint (or the functional viewpoint) can be allocated to the different types of organization (or organizations themselves, if known) that will be involved with the ITS implementation

2.12
physical viewpoint
one of several architectural viewpoints of the system of interest, showing how the functionality from the functional viewpoint can be allocated to different physical locations and combined into different building blocks

2.13
stakeholder
entity that is involved in some way with the ITS implementation

2.14
stakeholder need
formal expression, using “shall” language, to define what the stakeholders expect the ITS implementation to provide, and from which the functional viewpoint is created, also known as “user need”

2.15
system architecture
single, high-level, description of the major elements or objects of a system plus the inter-connections between them

2.16
system of interest
another name used for the system, applied to whatever will be included in the ITS implementation that is created out of the ITS architecture

2.17
terminator
entity that is external to the system but with which the system communicates either to obtain inputs or to which it can send outputs

NOTE 1 Terminators may be split up into actors if necessary.

NOTE 2 In most ITS architectures, the terminators may be the same in both the functional and the physical viewpoints.

NOTE 3 In the US National ITS Architecture, a terminator defines the boundary of the system of interest. Each terminator may represent the people, systems and general environment that interface to ITS. The interfaces between terminators and the sub-systems and processes within the National ITS Architecture are defined, but no functional requirements are allocated to terminators. The logical and physical architecture views of the National ITS Architecture both have exactly the same set of terminators.

2.18
unified modelling language
UML
object-oriented modelling language specified in ISO/IEC 19501

2.19
user need
another name for "stakeholder need"

3 Symbols and abbreviated terms

3.1
E-FRAME
extended framework architecture made for Europe

3.2
FRAME
framework architecture made for Europe

3.3
ITS
intelligent transport system

3.4
KAREN
keystone architecture required for European networks

3.5
POM
process-oriented method

4 Background

4.1 TC 204, working group 1 (WG1)

This Technical Report arose to complement work done by WG 1 on the elaboration of Technical Report ISO/TR 24529 which covers the use of UML in ITS International Standards and other deliverables. WG 1 has never mandated the sole use of UML above other architecture methodologies, and this Technical Report is intended to describe the use of an alternative methodology.

POM should not be seen as a rival to UML, and the two should be seen as complementary rather than competitors. Each methodology should be seen as being most suitable for (parts of) architectures that are created at particular stages in ITS implementations. In general, POM is best suited to high-level architectures that are to be used by decision-makers and others to refine their original concepts for ITS implementation and explore alternatives, without the need for detailed knowledge of architecture modelling techniques. UML is seen by some as more suited to detailed lower-level architectures from which software is to be designed, coded and tested before being included in the implementation.

4.2 Systems and architectures

The topic of system architecture is surrounded by shibboleths and silver bullets which imply that "if you are not using my preferred method of describing them, then you are wrong". The basic flaw in this argument is that system architectures are used for a number of different purposes, and so it is not surprising that different modelling techniques are required. Thus the relevant phrase should be "horses for courses" rather than "one size fits all". Let us consider the two words "system" and "architecture in turn.

Systems of interest are often described in a hierarchical manner and, when this is done, each lower stage contains greater detail than the one above it. Thus each higher stage provides the requirements for the one below it, which is a "design" that conforms to those requirements [1]. In the world of ITS, we can identify at least seven levels of requirements and design, each with their own set of characteristics and needs, as follows [note that (a) some of the terms used are described below and (b) the bottom two bullets are at the same level]:

— The top-level requirements are stakeholder aspirations.

— These are interpreted into a structured design by the system architect in the form of stakeholder needs.

— The stakeholder needs are then used as requirements by the system architect who creates a high-level design — the ITS architecture.

— ITS engineers then use (part of) the ITS architecture to create requirements for procurement.

— The system engineer in the supplier's organization uses these requirements for procurement to create a detailed design for the final equipment.

— Software and hardware engineers use parts of this detailed design to produce requirements for the software and hardware, respectively.

— A programmer uses the software requirements to produce a program, which is a form of design.

— A hardware engineer uses the hardware requirements to produce a design for the hardware.

Consequently, a modelling technique that was originally created to aid, say, the development of (low-level) software sub-systems might not be suitable for describing the higher levels of the system of interest, which include features from other engineering disciplines. Also, the intended audiences for the highest levels of abstraction might not always be from an engineering background, and will therefore need something that is easy and intuitive to understand and that contains few "hidden" semantics.

Architectures describe the fixed parts of the structure of the system of interest and its sub-systems. They range from a high-level structure of a system of interest that all users need to understand to some degree, to the low-level structure of a (complex) piece of software which only needs to be understood by its developers and maintainers. In fact, during the development of a large and complex system, architectures might be used at a number of different levels, each containing different levels of detail [1].

The situations described above are extended when a system architecture is not used to describe a single system of interest, but a class of systems from which any particular deployment might take some, but not all, of the components. This is called a system framework architecture.

This Technical Report describes a process for the first four stages stated above, using a sequence of models that have been developed specifically to aid in the creation of large complex systems of interest [1] and is proven in use. It is also the process and models that underlie the European ITS Framework Architecture [2], which has already formed the basis of a number of regional, national and trans-national ITS architectures, as well as architectures for individual projects.

4.3 ITS architecture development approaches

Several approaches to architecture development have been tried during the last 20-30 years. In essence, all of them are trying to capture the mental model that system designers create when asked to design a system. As these systems have become more and more complex, they have often needed to be partitioned into sub-systems, or for the required functionality to be provided through the creation of several separate systems with what is hoped to be fully agreed communications between them.

Once communications have been agreed between systems, it usually means that they are said to be “interoperable”. This is required to enable systems to deliver the same services across national and other geographical boundaries and to enable individual systems, sub-systems and components to be replaced without adversely affecting system operation or performance. The probability of systems communicating with each other successfully can be improved through the use of system architectures, as these can be used to ensure that communicating systems are not based on conflicting assumptions.

ITS has not escaped this move towards the need for high degrees of complexity and interoperability. Indeed, ITS, by their very nature, are highly complex and, in many cases, interdependent. This has resulted in the adoption of formalized approaches to the development of ITS system architectures, or “ITS architectures” as they are usually called.

At first the ITS architecture development approaches were manual, but in time and as a result of the added complexity already mentioned, they have become computer based, with some levels of computer assistance also being introduced. At the moment, there are two main approaches in use: the object-oriented (OO) approach, usually using UML, and the process-oriented method (POM), also known as data flow modelling. Both have their advocates and detractors and both have computer-based tools to aid in their use.

Proponents of UML will quite rightly claim their differences and in some circumstances advantages over POM, and it is not the objective of this Technical Report to claim preference for POM, but simply to say that in many cases it is a suitable technique and to consider the ways of using it most effectively.

5 The process-oriented model

5.1 General

Before beginning to understand the process-oriented model, is important to understand its objectives and the limitations of the claims that are being made for it. They are as follows:

a) It is an effective model to describe a high-level ITS framework architecture, and the ITS architectures that are derived from it.

b) It is anticipated that it will be used in the early part of a hierarchy of requirements and designs. In particular, it is anticipated that (parts of) a resulting ITS architecture will be used to produce component and infrastructure specifications for suppliers to design and build (using their own in-house methods and modelling techniques).

c) An ITS framework architecture might not be complete. It might be necessary to add items to create a defined ITS architecture to satisfy its stakeholders.

d) The model is technology-independent, i.e. the decision as to how to implement the components and infrastructure is left to the suppliers (subject to any restrictions imposed by the purchaser).

The model comprises the following elements:

- Stakeholder aspirations
- Stakeholder needs
- Functional viewpoint
  - Context diagram
  - Data flow diagrams
  - Control model
- Physical viewpoint
  - Context diagram (same as that for the functional viewpoint)
  - Sub-system specifications
  - Module specifications (optional — depending upon system complexity)
- Communications viewpoint

5.2 Stakeholder aspirations

These are unstructured statements that express the problems that need to be addressed and the desires of the various stakeholders. Ideally, they should be written by the stakeholders, but in practice they normally need guidance from their ITS architects. A stakeholder is any person that affects, or is affected by, either directly or indirectly, the system under consideration. They can be categorized into four generic groups, as follows:

- *Want ITS* — These are the problem owners. Their problems may be concerned with traffic and/or transport, and they might be authorities that need to improve the transport environment of their political masters, network owners or operators, or travellers who wish to improve their travelling experience.
- *Make ITS* — These are solution providers. They comprise the component manufacturers and the system integrators.
- *Use ITS* — These are the travellers, system operators and service providers that will come into direct contact with the systems and use them to solve their problems.
- *Rule ITS* — These are the authorities that provide the legislative framework within which the solutions will be created and used. They also include the creators of standards.

Some stakeholders, such as service providers, can be part of more than one category.

Aspirations are written in the stakeholders' own words. Thus they are likely to be unstructured, as can be seen in the examples given in Figure 1.

