INTERNATIONAL STANDARD

ISO/IEC 15909-1

Second edition
2019-08

Systems and software engineering — High-level Petri nets —

Part 1: Concepts, definitions and graphical notation

Ingénierie du logiciel et des systèmes — Réseaux de Petri de haut niveau —
Partie 1: Concepts, définitions et notation graphique

Reference number
ISO/IEC 15909-1:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 7, *Software and systems engineering*.

This second edition cancels and replaces the first edition (ISO/IEC 15909-1:2004), which has been technically revised.

The main change compared to the previous edition is as follows:

— a complete redrafting of the concepts and definitions of Petri nets and Petri net types in a simpler, modular and incremental way.

A list of all parts in the ISO/IEC 15909 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

The ISO/IEC 15909 series is concerned with defining a modeling language and its transfer format, known as high-level Petri nets. This document is Part 1 of the ISO/IEC 15909 series. This document provides the mathematical definition of high-level Petri nets, called the semantic model, its execution semantics, the graphical form of the technique and its mapping to the semantic model. This document also introduces some common notational conventions for the graphical form of high-level Petri nets.

Petri nets have been used to describe a wide range of systems since their invention in 1962. The technique is mathematically defined and can thus be used to provide unambiguous specifications and descriptions of applications. It is also an executable technique, allowing specification prototypes to be developed to test ideas at the earliest and cheapest opportunity. Specifications written in the technique can be subjected to analysis methods to prove properties about the specifications, before implementation commences, thus saving on testing and maintenance time and providing a high level of quality assurance.

A problem with Petri nets is the explosion of the number of elements in their graphical form when they are used to describe complex systems. High-level Petri nets were developed to overcome this problem by introducing higher-level concepts, such as the use of complex structured data as tokens, and using algebraic expressions to annotate net elements. The use of "high-level" to describe these Petri nets is analogous to the use of "high-level" in high-level programming languages (as opposed to assembly languages), and is the usual term used in the Petri net community. Two of the early forms of high-level nets that this document builds on are predicate-transition nets and coloured Petri nets, first introduced in 1979 and developed during the 1980s. It also uses some of the notions developed for algebraic Petri nets, first introduced in the mid-1980s. It is believed that this document captures the spirit of these earlier developments (see Bibliography).

The technique has multiple uses. For example, it can be used directly to specify systems or to define the semantics of other less formal languages. It can also serve to integrate techniques currently used independently such as state-transition diagrams and data flow diagrams. The technique is particularly suited to parallel and distributed systems development as it supports concurrency. The technique is able to specify systems at a level that is independent of the choice of implementation (i.e. by software, hardware (electronic and/or mechanical) or humans or a combination). This document may be cited in contracts for the development of systems (particularly distributed systems) or used by application developers or Petri net tool vendors or users.

This document provides an abstract mathematical syntax and a formal semantics for the technique. Conformance to the document is possible at several levels. The level of conformance depends on the class of high-level net chosen.

This document is Part 1 of the ISO/IEC 15909 series. It describes definitions, semantics, execution and graphical notations for high-level Petri nets. A transfer format for the high-level Petri nets is the subject of Part 2, while Part 3 addresses techniques for enrichments, extensions and structuring mechanisms.

Reliable software development requires powerful mathematical models and tools. The usability of Petri nets has been proven for non-trivial industrial applications.

This document is written as a reference for systems analysts, designers, developers, maintainers and procurers, and for Petri net tool designers and developers.

This document defines high-level Petri nets showing common concepts for Petri nets first, and then describing several typical types of Petri nets, such as place/transition nets, symmetric nets, and Petri nets with time. Each of the Petri net types is described with its definition, semantics, and execution. Their graphical notations are provided in Annex B.

Systems and software engineering — High-level Petri nets —

Part 1:
Concepts, definitions and graphical notation

1 Scope

This document defines a Petri net modeling language or technique, called high-level Petri nets, including its syntax and semantics. It provides a reference definition that can be used both within and between organizations, to ensure a common understanding of the technique and of the specifications written using the technique. This document also facilitates the development and interoperability of Petri net computer support tools.

This document is applicable to a wide variety of concurrent discrete event systems and in particular distributed systems. Generic fields of application include:

— requirements analysis;

— development of specifications, designs and test suites;

— descriptions of existing systems prior to re-engineering;

— modeling business and software processes;

— providing the semantics for concurrent languages;

— simulation of systems to increase confidence;

— formal analysis of the behavior of systems;

— and development of Petri net support tools.

This document can be applied to the design of a broad range of systems and processes, including aerospace, air traffic control, avionics, banking, biological and chemical processes, business processes, communication protocols, computer hardware architectures, control systems, databases, defense command and control systems, distributed computing, electronic commerce, fault-tolerant systems, games, hospital procedures, information systems, Internet protocols and applications, legal processes, logistics, manufacturing systems, metabolic processes, music, nuclear power systems, operating systems, transport systems (including railway control), security systems, telecommunications and workflows.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 15909-2, *Systems and software engineering — High-level Petri nets — Part 2: Transfer format*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 15909-2 and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1
arc
directed edge of a net which may connect a *place* (3.28) to a *transition* (3.37) or a transition to a place, normally represented by an arrow

3.2
arc annotation
expression that may involve constants, variables and *operators* (3.21) used to annotate an *arc* (3.1) of a net

Note 1 to entry: The expression shall evaluate to a *multiset* (3.17) over the type of the arc's associated *place* (3.28).

3.3
basis set
set of objects used to create a *multiset* (3.17)

3.4
carrier
set of a *many-sorted algebra* (3.14)

3.5
concurrent enabling
<transition modes> state, for a *multiset* (3.17) of *transition* (3.37) modes, when all the involved *input places* (3.13) contain enough tokens to satisfy the sum of all of the demands imposed on them by each *input arc* (3.12) *annotation* (3.2) evaluated for each transition mode in the multiset

3.6
concurrency
property of a system in which events may occur independently of each other, and hence are not ordered

Note 1 to entry: See also step and *concurrent enabling* (3.5).

3.7
declaration
set of statements which define the sets, constants, *parameter* (3.24) values, typed variables and functions required for defining the annotations on a *high-level Petri net* (3.9)

3.8
enabling
<a transition> state of a *transition* (3.37) in a particular mode and net *marking* (3.15) when the following conditions are met: the marking of each *input place* (3.13) of the transition satisfies the demand imposed on it by its *arc annotation* (3.2) evaluated for the particular transition mode; the demand is satisfied when the place's marking contains (at least) the *multiset* (3.17) of tokens indicated by the evaluated arc annotation; the determination of transition modes guarantees that the Transition Condition is satisfied

Note 1 to entry: See *concurrent enabling* (3.5).

3.9
high-level Petri net
high-level net
HLPN
algebraic structure comprising: a set of *places* (3.28); a set of *transitions* (3.37); a set of types; a function associating a type to each place, and a set of modes (a type) to each transition; a pre-function imposing token demands (*multisets* (3.17) of tokens) on places for each transition mode; a post-function determining output tokens (multisets of tokens) for places for each transition mode; and an *initial marking* (3.10)

3.10
initial marking
<a net> set of initial making of places (3.11) given with the *high-level net* (3.9) definition

3.11
initial marking of place
special *marking* (3.15) of a *place* (3.28), defined with the net

3.12
input arc
<a transition>arc directed from a *place* (3.28) to the *transition* (3.37)

3.13
input place
<a transition> *place* (3.28) connected to the *transition* (3.37) by an *input arc* (3.12)

3.14
many-sorted algebra
mathematical structure comprising a set of sets and a set of functions taking these sets as domains and co-domains

3.15
marking
<a net> set of the *place* (3.28) markings for all places of the net.

3.16
marking of a place
multiset (3.17) of tokens associated with ('residing in') the *place* (3.28)

3.17
multiset
bag
collection of items where repetition of items is allowed

3.18
multiplicity
natural number (i.e. non-negative integer) which describes the number of repetitions of an item in a *multiset* (3.17)

3.19
multiset cardinality
cardinality of a multiset
sum of the multiplicities of each of the members of the *multiset* (3.17)

3.20
node
vertex of a net graph, i.e. either a *place* (3.28) or a *transition* (3.37)

3.21
operator
symbol representing the name of a function

3.22
output arc
<a transition> arc directed from the *transition* (3.37) to a *place* (3.28)

3.23
output place
<a transition> *place* (3.28) connected to the *transition* (3.37) by an *output arc* (3.22)

3.24
parameter
symbol that can take a range of values defined by a set

Note 1 to entry: It is defined as a constant in the *signature* (3.32).

3.25
Petri net
net
algebraic structure with two sets, one called *places* (3.28) and the other called *transitions* (3.37), together with their associated relations and functions, and named after their inventor, Carl Adam Petri

3.26
Petri net with priorities
Petri net (3.25) having priorities which can be used for selecting the enabled *transitions* (3.37) according to the given priority scheme

Note 1 to entry: The firing rule is the same as a Petri net without priorities.

3.27
Petri net with time
Petri net (3.25) having timing constraints associated with the *nodes* (3.20) or arcs

Note 1 to entry: These constraints affect the *enabling* (3.8) and firing rules.

3.28
place
node (3.20) of a net, usually represented by an ellipse

3.29
place/transition net
P/T net
Petri net (3.25) comprising a net with positive integers associated with arcs and an *initial marking* (3.10) function which associates a natural number of simple tokens ('black dots') with *places* (3.28)

3.30
reachability graph
directed graph where the *nodes* (3.20) correspond to *reachable markings* (3.31) and the edges to *transition* (3.37) occurrences

3.31
reachable marking
marking (3.15) of the net that can be reached from the *initial marking* (3.10) by the occurrence of *transitions* (3.37)

3.32
signature
mathematical structure comprising a set of *sorts* (3.33) and a set of *operators* (3.21)

3.33
sort
symbol representing the name of a set

3.34
symmetric net
high-level Petri net (3.9) where types are finite and only built-in operations are allowed

3.35
term
expression comprising constants, variables and *operators* (3.21) built from a *signature* (3.32) and a set of sorted variables

3.36
time Petri net
Petri net (3.25) with time where timing constraints are associated with *transitions* (3.37)

3.37
transition
node (3.20) of a net, usually represented by a rectangle

4 Conformance

4.1 General

There are different levels of conformance to this document. The level of conformance is defined according to the net type.

4.2 Mandatory conformance: common concepts for Petri nets

Prior to claim conformance to any type of Petri net, defined in this document, an implementation shall demonstrate that it has the semantics and the execution mechanisms defined in Clause 5.

4.3 Place/Transition nets

To claim P/T net Level conformance to this document an implementation shall demonstrate that it has the semantics and the execution defined in Clause 6, by providing a mapping from the implementation to the semantic model and the execution.

4.4 Symmetric nets

To claim Symmetric net Level conformance to this document an implementation shall demonstrate that it has the semantics and the execution defined in Clause 7, by providing a mapping from the implementation to the semantic model and the execution.

4.5 High-level Petri nets

4.5.1 Level 1

To claim high-level Petri net Level 1 conformance to this document an implementation shall demonstrate that it has the semantics and the execution defined in Clause 7 (Symmetric nets), by providing a mapping from the implementation to the semantic model and the execution.

4.5.2 Level 2

To claim high-level Petri net Level 2 conformance to this document an implementation shall have satisfied the requirements of high-level Petri net Level 1 conformance, and in addition shall comply with the definition and semantics defined in Clause 8.

4.6 Petri nets with priorities

To claim Petri nets with priorities conformance to this document an implementation shall demonstrate that it has the semantics, the execution, and the prioritized Petri net model defined in Clause 9, by providing a mapping from the implementation to the semantic model and the execution.

4.7 Petri nets with time — Level 1

To claim Petri nets with time Level 1 conformance to this document an implementation shall demonstrate that it has the semantics, the execution, and the time Petri net model defined in Clause 10, by providing a mapping from the implementation to the semantic model and the execution.

IMPORTANT — This document only defines a single type of Petri nets with time among the many variants available. It corresponds to the semantics defined by Merlin in 1974, which is the most commonly used. Other types of Petri nets with time can still be defined using tool-specific information.

5 Common concepts for Petri nets

5.1 General

All common concepts for Petri nets in this Clause, and Petri net types in subsequent Clauses 6 to 10, shall be defined using the mathematical conventions in Annex A.

5.2 Definition

5.2.1 Concept 1 (net)

A net is defined by a triple $N = <P, T, F>$, where:

— P is a set of elements called places,

— T is a set of elements called transitions, disjoint from P,

— F is a flow relation $F \subseteq (P \times T) \cup (T \times P)$.

If P and T are finite, then N is said to be finite.

5.2.2 Concept 2 (marking of a net)

A marking of a net N is a mapping $M : P \rightarrow MS$, where MS is a multiset (see A.5). A place p is said to be marked by a marking M if $M(p)$ is not an empty multiset.

5.2.3 Concept 3 (marked net)

A marked net is a tuple $N = <P, T, F, M>$, where $<P, T, F>$ is a net and M a marking of this net.

By convention, the initial marking of a marked net N is noted M_0.

By convention, the set of possible marked nets derived from N is noted N.

5.2.4 Concept 4 (precondition of a transition)

A place $p \in P$ is a precondition of the transition $t \in T$ iff $(p,t) \in F$.

5.2.5 Concept 5 (postcondition of a transition)

A place $p \in P$ is a postcondition of the transition $t \in T$ iff $(t,p) \in F$.

