

PD ISO/PAS 19450:2015



BSI Standards Publication

Automation systems and integration — Object-Process Methodology

bsi.

...making excellence a habit.™

National foreword

This Published Document is the UK implementation of ISO/PAS 19450:2015.

The UK participation in its preparation was entrusted to Technical Committee AMT/5, Industrial architectures and integration frameworks.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2015. Published by BSI Standards Limited 2015

ISBN 978 0 580 87861 9

ICS 25.040.01

Compliance with a British Standard cannot confer immunity from legal obligations.

This Published Document was published under the authority of the Standards Policy and Strategy Committee on 31 December 2015.

Amendments issued since publication

Date	Text affected
------	---------------

**PUBLICLY
AVAILABLE
SPECIFICATION**

**ISO/PAS
19450**

First edition
2015-12-15

**Automation systems and integration —
Object-Process Methodology**

*Systèmes d'automatisation et intégration — Object-Process
Methodology*



Reference number
ISO/PAS 19450:2015(E)



COPYRIGHT PROTECTED DOCUMENT

© ISO 2015, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

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

Contents

	Page
Foreword	vi
Introduction	vii
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Symbols	8
5 Conformance	10
6 OPM principles and concepts	10
6.1 OPM modelling principles.....	10
6.1.1 Modelling as a purpose-serving activity.....	10
6.1.2 Unification of function, structure, and behaviour.....	11
6.1.3 Identifying functional value.....	11
6.1.4 Function versus behaviour.....	11
6.1.5 System boundary setting.....	12
6.1.6 Clarity and completeness trade-off.....	12
6.2 OPM Fundamental concepts.....	12
6.2.1 Bimodal representation.....	12
6.2.2 OPM modelling elements.....	12
6.2.3 OPM things: objects and processes.....	13
6.2.4 OPM links: procedural and structural.....	13
6.2.5 OPM context management.....	14
6.2.6 OPM model implementation.....	14
7 OPM thing syntax and semantics	15
7.1 Objects.....	15
7.1.1 Description.....	15
7.1.2 Representation.....	15
7.2 Processes.....	15
7.2.1 Description.....	15
7.2.2 Representation.....	16
7.3 OPM things.....	16
7.3.1 OPM thing defined.....	16
7.3.2 Object-process test.....	16
7.3.3 OPM thing generic properties.....	17
7.3.4 Default values of thing generic properties.....	18
7.3.5 Object states.....	18
8 OPM link syntax and semantics overview	20
8.1 Procedural link overview.....	20
8.1.1 Kinds of procedural links.....	20
8.1.2 Procedural link uniqueness OPM principle.....	20
8.1.3 State-specified procedural links.....	20
8.2 Operational semantics and flow of execution control.....	20
8.2.1 The Event-Condition-Action control mechanism.....	20
8.2.2 Preprocess object set and postprocess object set.....	21
8.2.3 Skip semantics of condition versus wait semantics of non-condition links.....	21
9 Procedural links	22
9.1 Transforming links.....	22
9.1.1 Kinds of transforming links.....	22
9.1.2 Consumption link.....	22
9.1.3 Result link.....	23
9.1.4 Effect link.....	23
9.1.5 Basic transforming links summary.....	23

9.2	Enabling links.....	24
9.2.1	Kinds of enabling links	24
9.2.2	Agent and Agent Link.....	24
9.2.3	Instrument and Instrument Link.....	25
9.2.4	Basic enabling links summary.....	26
9.3	State-specified transforming links.....	26
9.3.1	State-specified consumption link.....	26
9.3.2	State-specified result link.....	27
9.3.3	State-specified effect links.....	28
9.3.4	State-specified transforming links summary.....	30
9.4	State-specified enabling links.....	31
9.4.1	State-specified agent link.....	31
9.4.2	State-specified instrument link.....	32
9.4.3	State-specified enabling links summary.....	32
9.5	Control links.....	33
9.5.1	Kinds of control links.....	33
9.5.2	Event links.....	34
9.5.3	Condition links.....	40
9.5.4	Exception links.....	47
10	Structural links.....	48
10.1	Kinds of structural links.....	48
10.2	Tagged structural link.....	48
10.2.1	Unidirectional tagged structural link.....	48
10.2.2	Unidirectional null-tagged structural link.....	49
10.2.3	Bidirectional tagged structural link.....	49
10.2.4	Reciprocal tagged structural link.....	49
10.3	Fundamental structural relations.....	50
10.3.1	Kinds of fundamental structural relations.....	50
10.3.2	Aggregation-participation relation link.....	51
10.3.3	Exhibition-characterization link.....	52
10.3.4	Generalization-specialization and inheritance.....	55
10.3.5	Classification-instantiation link.....	58
10.3.6	Fundamental structural relation link and tagged structural link summary.....	61
10.4	State-specified structural relations and links.....	62
10.4.1	State-specified characterization relation link.....	62
10.4.2	State-specified tagged structural relations.....	63
11	Relationship cardinalities.....	67
11.1	Object multiplicity in structural and procedural links.....	67
11.2	Object multiplicity expressions and constraints.....	69
11.3	Attribute value and multiplicity constraints.....	71
12	Logical operators: AND, XOR, and OR.....	71
12.1	Logical AND procedural links.....	71
12.2	Logical XOR and OR procedural links.....	73
12.3	Diverging and converging XOR and OR links.....	74
12.4	State-specified XOR and OR link fans.....	76
12.5	Control-modified link fans.....	77
12.6	State-specified control-modified link fans.....	77
12.7	Link probabilities and probabilistic link fans.....	79
13	Execution path and path labels.....	81
14	Context management with OPM.....	83
14.1	Completing the SD.....	83
14.2	Achieving model comprehension.....	83
14.2.1	OPM refinement-abstraction mechanisms.....	83
14.2.2	Control (operational) semantics within an in-zoomed process context.....	87
14.2.3	OPM fact consistency principle.....	98
14.2.4	Abstraction ambiguity resolution for procedural links.....	99

Annex A (normative) OPL formal syntax in EBNF	102
Annex B (informative) Guidance for OPM	121
Annex C (informative) Modelling OPM using OPM	124
Annex D (informative) OPM dynamics and simulation	157
Bibliography	163

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The 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 ISO documents 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 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).

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

For an explanation on the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: [Foreword - Supplementary information](#)

The committee responsible for this document is Technical Committee ISO/TC 184, *Automation systems and integration*, Subcommittee SC 5, *Interoperability, integration, and architectures for enterprise systems and automation applications*.

Introduction

Object-Process Methodology (OPM) is a compact conceptual approach, language, and methodology for modelling and knowledge representation of automation systems. The application of OPM ranges from simple assemblies of elemental components to complex, multidisciplinary, dynamic systems. OPM is suitable for implementation and support by tools using information and computer technology. This Publicly Available Specification specifies both the language and methodology aspects of OPM in order to establish a common basis for system architects, designers, and OPM-compliant tool developers to model all kinds of systems.

OPM provides two semantically equivalent modalities of representation for the same model: graphical and textual. A set of hierarchically structured, interrelated Object-Process Diagrams (OPDs) constitutes the graphical model, and a set of automatically generated sentences in a subset of the English language constitutes the textual model expressed in the Object-Process Language (OPL). In a graphical-visual model, each OPD consists of OPM elements, depicted as graphic symbols, sometimes with label annotation. The OPD syntax specifies the consistent and correct ways to manage the arrangement of those graphically elements. Using OPL, OPM generates the corresponding textual model for each OPD in a manner that retains the constraints of the graphical model. Since the syntax and semantics of OPL are a subset of English natural language, domain experts easily understand the textual model.

OPM notation supports the conceptual modelling of systems with formal syntax and semantics. This formality serves as the basis for model-based systems engineering in general, including systems architecting, engineering, development, life cycle support, communication, and evolution. Furthermore, the domain-independent nature of OPM opens system modelling to the entire scientific, commercial and industrial community for developing, investigating and analysing manufacturing and other industrial and business systems inside their specific application domains; thereby enabling companies to merge and provide for interoperability of different skills and competencies into a common intuitive yet formal framework.

OPM facilitates a common view of the system under construction, test, integration, and daily maintenance, providing for working in a multidisciplinary environment. Moreover, using OPM, companies can improve their overall, big-picture view of the system's functionality, flexibility in assignment of personnel to tasks, and managing exceptions and error recovery. System specification is extensible for any necessary detail, encompassing the functional, structural and behavioural aspects of a system.

One particular application of OPM is in the drafting and authoring of technical standards. OPM helps sketch the implementation of a standard and identify weaknesses in the standard to reduce, thereby significantly improving the quality of successive drafts. With OPM, even as the model-based text of a system expands to include more details, the underlying model keeps maintaining its high degree of formality and consistency.

This Publicly Available Specification provides a baseline for system architects and designers, who can use it to model systems concisely and effectively. OPM tool vendors can utilise the PAS as a formal standard specification for creating software tools to enhance conceptual modelling.

This Publicly Available Specification provides a presentation of the normative text that follows the Extended Backus-Naur Form (EBNF) specification of the language syntax. All elements are presented in [Clauses 6 to 13](#) with only minimal reference to methodological aspects, [Clause 14](#) presents the context management mechanisms related to in-zooming and unfolding.

This specification utilizes several conventions for the presentation of OPM. Specifically, Arial bold font in text and Arial bold italic font in figure captions, table captions and headings distinguish label names for OPM objects, processes, states, and link tags. OPL reserved words are in Arial regular font with commas and periods in Arial bold font. Most figures contain both a graphic image, the OPD portion, and a textual equivalent, the OPL portion. Because this is a language specification, the precise use of term definitions is essential and several terms in common use have particular meaning when using OPM. Clause B.6 explains other conventions for the use of OPM.

Annex A presents the formal syntax for OPL, in EBNF form.

Annex B presents conventions and patterns commonly used in OPM applications.

Annex C presents aspects of OPM as OPM models.

Annex D summarizes the dynamic and simulation capabilities of OPM.

The International Organization for Standardization (ISO) draws attention to the fact that it is claimed that compliance with this document may involve the use of a patent concerning OPM as a modelling system given in [Clauses 6](#) to [14](#).

ISO takes no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured the ISO that he/she is willing to negotiate licences either free of charge or under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO. Information may be obtained from:

Prof. Dov Dori

Technion Israel Institute of Technology

Technion City

Haifa 32000, Israel

dori@ie.technion.ac.il

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. ISO shall not be held responsible for identifying any or all such patent rights.

ISO (www.iso.org/patents) and IEC (<http://patents.iec.ch>) maintain on-line databases of patents relevant to their standards. Users are encouraged to consult the databases for the most up to date information concerning patents.

Automation systems and integration — Object-Process Methodology

1 Scope

This Publicly Available Specification specifies Object-Process Methodology (OPM) with detail sufficient for enabling practitioners to utilise the concepts, semantics, and syntax of Object-Process Methodology as a modelling paradigm and language for producing conceptual models at various extents of detail, and for enabling tool vendors to provide application modelling products to aid those practitioners.

While this Publicly Available Specification presents some examples for the use of Object-Process Methodology to improve clarity, it does not attempt to provide a complete reference for all the possible applications of Object-Process Methodology.

2 Normative references

There are no normative references.

3 Terms and definitions

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

3.1

abstraction

decreasing the extent of detail and system model *completeness* (3.8) in order to achieve better comprehension

3.2

affectee

transformee (3.78) that is affected by a *process* (3.58) occurrence, i.e. its *state* (3.69) changes

Note 1 to entry: An affectee can only be a *stateful object* (3.66). A *stateless object* (3.67) can only be created or consumed, but not affected.

3.3

agent

enabler (3.17) that is a human or a group of humans

3.4

attribute

object (3.39) that characterizes a *thing* (3.76) other than itself

3.5

behaviour

transformation (3.77) of *objects* (3.39) resulting from the execution of an *Object-Process Methodology* (3.43) model comprising a collection of *things* (3.76) and *links* (3.36) to objects in the model

3.6

beneficiary

<system> *stakeholder* (3.65) who gains functional *value* (3.82) from the system's *operation* (3.46)