

NORME  
INTERNATIONALE  
INTERNATIONAL  
STANDARD

CEI  
IEC  
1131-3

First edition  
1993-03

---

---

**Programmable controllers –**

**Part 3:  
Programming languages**

*Automates programmables –*

*Partie 3:  
Langages de programmation*

IEC 61131-3:1993

<https://standards.iteh.ai/en/standards/iec/317e030-cf62-4ffb-9a84-9403172c2cd6/iec-61131-3-1993>



Reference number  
IEC 1131-3: 1993(E)

## Validité de la présente publication

Le contenu technique des publications de la CEI est constamment revu par la CEI afin qu'il reflète l'état actuel de la technique.

Des renseignements relatifs à la date de reconfirmation de la publication sont disponibles auprès du Bureau Central de la CEI.

Les renseignements relatifs à ces révisions, à l'établissement des éditions révisées et aux amendements peuvent être obtenus auprès des Comités nationaux de la CEI et dans les documents ci-dessous:

- **Bulletin de la CEI**
- **Annuaire de la CEI**  
Publié annuellement
- **Catalogue des publications de la CEI**  
Publié annuellement et mis à jour régulièrement

## Terminologie

En ce qui concerne la terminologie générale, le lecteur se reportera à la CEI 50: *Vocabulaire Electrotechnique International* (VEI), qui se présente sous forme de chapitres séparés traitant chacun d'un sujet défini. Des détails complets sur le VEI peuvent être obtenus sur demande. Voir également le dictionnaire multilingue de la CEI.

Les termes et définitions figurant dans la présente publication ont été soit tirés du VEI, soit spécifiquement approuvés aux fins de cette publication.

## Symboles graphiques et littéraux

Pour les symboles graphiques, les symboles littéraux et les signes d'usage général approuvés par la CEI, le lecteur consultera:

- la CEI 27: *Symboles littéraux à utiliser en électrotechnique;*
- la CEI 417: *Symboles graphiques utilisables sur le matériel. Index, relevé et compilation des feuilles individuelles;*
- la CEI 617: *Symboles graphiques pour schémas;*

et pour les appareils électromédicaux,

- la CEI 878: *Symboles graphiques pour équipements électriques en pratique médicale.*

Les symboles et signes contenus dans la présente publication ont été soit tirés de la CEI 27, de la CEI 417, de la CEI 617 et/ou de la CEI 878, soit spécifiquement approuvés aux fins de cette publication.

## Publications de la CEI établies par le même comité d'études

L'attention du lecteur est attirée sur les listes figurant à la fin de cette publication, qui énumèrent les publications de la CEI préparées par le comité d'études qui a établi la présente publication.

## Validity of this publication

The technical content of IEC publications is kept under constant review by the IEC, thus ensuring that the content reflects current technology.

Information relating to the date of the reconfirmation of the publication is available from the IEC Central Office.

Information on the revision work, the issue of revised editions and amendments may be obtained from IEC National Committees and from the following IEC sources:

- **IEC Bulletin**
- **IEC Yearbook**  
Published yearly
- **Catalogue of IEC publications**  
Published yearly with regular updates

## Terminology

For general terminology, readers are referred to IEC 50: *International Electrotechnical Vocabulary (IEV)*, which is issued in the form of separate chapters each dealing with a specific field. Full details of the IEV will be supplied on request. See also the IEC Multilingual Dictionary.

The terms and definitions contained in the present publication have either been taken from the IEV or have been specifically approved for the purpose of this publication.

## Graphical and letter symbols

For graphical symbols, and letter symbols and signs approved by the IEC for general use, readers are referred to publications:

- IEC 27: *Letter symbols to be used in electrical technology;*
- IEC 417: *Graphical symbols for use on equipment. Index, survey and compilation of the single sheets;*
- IEC 617: *Graphical symbols for diagrams;*

and for medical electrical equipment,

- IEC 878: *Graphical symbols for electromedical equipment in medical practice.*

The symbols and signs contained in the present publication have either been taken from IEC 27, IEC 417, IEC 617 and/or IEC 878, or have been specifically approved for the purpose of this publication.

## IEC publications prepared by the same technical committee

The attention of readers is drawn to the end pages of this publication which list the IEC publications issued by the technical committee which has prepared the present publication.

NORME  
INTERNATIONALE  
INTERNATIONAL  
STANDARD

CEI  
IEC  
1131-3

First edition  
1993-03

---

---

**Programmable controllers —**

**Part 3:  
Programming languages**

*Automates programmables —*

*Partie 3:  
Langages de programmation*

© CEI 1993 Droits de reproduction réservés — Copyright — all rights reserved

Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'éditeur.

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 the publisher.

Bureau Central de la Commission Electrotechnique Internationale 3, rue de Varembe Genève, Suisse

---

---



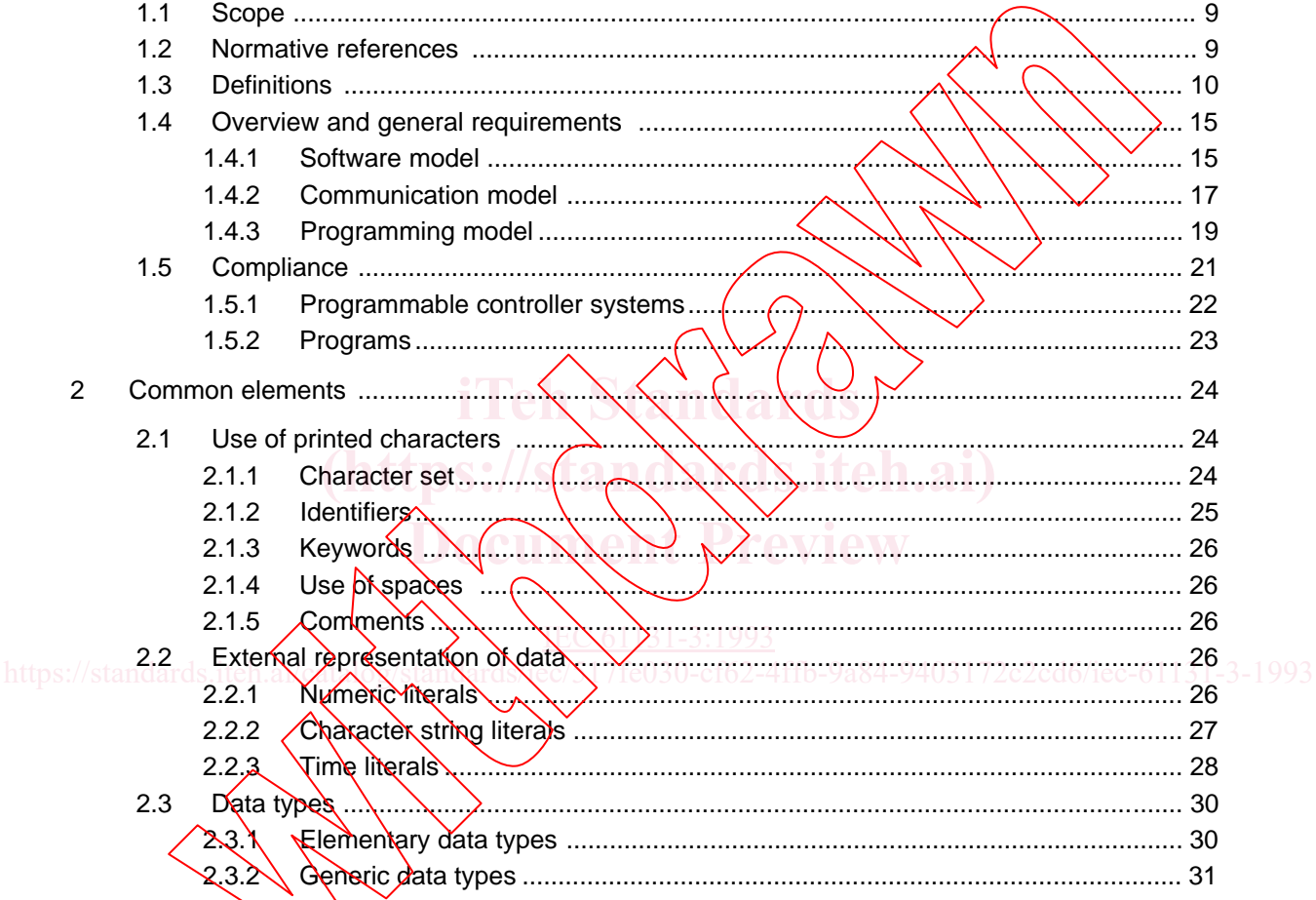
Commission Electrotechnique Internationale  
International Electrotechnical Commission  
Международная Электротехническая Комиссия

---

---

# CONTENTS

	Page
FOREWORD .....	7
Clause	
1 General .....	9
1.1 Scope .....	9
1.2 Normative references .....	9
1.3 Definitions .....	10
1.4 Overview and general requirements .....	15
1.4.1 Software model .....	15
1.4.2 Communication model .....	17
1.4.3 Programming model .....	19
1.5 Compliance .....	21
1.5.1 Programmable controller systems .....	22
1.5.2 Programs .....	23
2 Common elements .....	24
2.1 Use of printed characters .....	24
2.1.1 Character set .....	24
2.1.2 Identifiers .....	25
2.1.3 Keywords .....	26
2.1.4 Use of spaces .....	26
2.1.5 Comments .....	26
2.2 External representation of data .....	26
2.2.1 Numeric literals .....	26
2.2.2 Character string literals .....	27
2.2.3 Time literals .....	28
2.3 Data types .....	30
2.3.1 Elementary data types .....	30
2.3.2 Generic data types .....	31
2.3.3 Derived data types .....	32
2.4 Variables .....	36
2.4.1 Representation .....	36
2.4.2 Initialization .....	38
2.4.3 Declaration .....	39
2.5 Program organization units .....	44
2.5.1 Functions .....	44
2.5.2 Function blocks .....	61
2.5.3 Programs .....	76



Clause	Page
2.6 Sequential Function Chart (SFC) elements .....	77
2.6.1 General .....	77
2.6.2 Steps .....	77
2.6.3 Transition .....	79
2.6.4 Actions .....	83
2.6.5 Rules of evolution .....	92
2.6.6 Compatibility of SFC elements .....	101
2.6.7 Compliance requirements .....	101
2.7 Configuration elements .....	102
2.7.1 Configurations, resources, and access paths .....	103
2.7.2 Tasks .....	105
3 Textual languages .....	115
3.1 Common elements .....	115
3.2 Language IL (Instruction List) .....	115
3.2.1 Instructions .....	115
3.2.2 Operators, modifiers and operands .....	116
3.2.3 Functions and function blocks .....	117
3.3 Language ST (Structured Text) .....	118
3.3.1 Expressions .....	118
3.3.2 Statements .....	121
4 Graphic languages .....	124
4.1 Common elements .....	124
4.1.1 Representation of lines and blocks .....	124
4.1.2 Direction of flow in networks .....	125
4.1.3 Evaluation of networks .....	126
4.1.4 Execution control elements .....	128
4.2 Language LD (Ladder Diagram) .....	130
4.2.1 Power rails .....	130
4.2.2 Link elements and states .....	130
4.2.3 Contacts .....	131
4.2.4 Coils .....	131
4.2.5 Functions and function blocks .....	131
4.2.6 Order of network evaluation .....	131
4.3 Language FBD (Function Block Diagram) .....	134
4.3.1 General .....	134
4.3.2 Combination of elements .....	134
4.3.3 Order of network evaluation .....	134
 Annexes	
A Specification method for textual languages .....	135
A.1 Syntax .....	135
A.2 Semantics .....	137

Annexes (concluded)

B	Formal specifications of language elements .....	138
B.0	Programming model .....	138
B.1	Common elements .....	138
B.2	Language IL (Instruction List) .....	148
B.3	Language ST (Structured Text) .....	149
C	Delimiters and keywords .....	151
D	Implementation-dependent parameters .....	154
E	Error conditions .....	156
F	Examples .....	157
F.1	Function WEIGH .....	157
F.2	Function block CMD_MONITOR .....	158
F.3	Function block FWD_REV_MON .....	161
F.4	Function block STACK_INT .....	166
F.5	Function block MIX_2_BRIX .....	171
F.6	Analog signal processing .....	174
F.7	Program GRAVEL .....	183
F.8	Program AGV .....	191
G	Index .....	195
H	Software compliance testing .....	207

Tables

1	Character set features .....	25
2	Identifier features .....	25
3	Comment feature .....	26
4	Numeric literals .....	27
5	Character string literal feature .....	28
6	Two-character combinations in character strings .....	28
7	Duration literal features .....	29
8	Date and time of day literals .....	29
9	Examples of date and time of day literals .....	30
10	Elementary data types .....	31
11	Hierarchy of generic data types .....	32
12	Data type declaration features .....	34
13	Default initial values .....	34
14	Data type initial value declaration features .....	35
15	Location and size prefix features for directly represented variables .....	37
16	Variable declaration keywords .....	40
17	Variable type assignment features .....	41
18	Variable initial value assignment features .....	43
19	Graphical negation of Boolean signals .....	45
20	Use of EN input and ENO output .....	47
21	Typed and overloaded functions .....	49
22	Type conversion function features .....	52
23	Standard functions of one numeric variable .....	53
24	Standard arithmetic functions .....	54
25	Standard bit shift functions .....	55

Clause	Page
26	Standard bitwise Boolean functions ..... 56
27	Standard selection functions ..... 57
28	Standard comparison functions ..... 58
29	Standard character string functions ..... 59
30	Functions of time data types ..... 60
31	Functions of enumerated data type ..... 60
32	Examples of function block I/O parameter usage ..... 62
33	Function block declaration features ..... 65
34	Standard bistable function blocks ..... 71
35	Standard edge detection function blocks ..... 72
36	Standard counter function blocks ..... 73
37	Standard timer function blocks ..... 74
38	Standard timer function blocks – timing diagrams ..... 75
39	Program declaration features ..... 76
40	Step features ..... 79
41	Transitions and transition conditions ..... 81
42	Declaration of actions ..... 84
43	Step/action association ..... 86
44	Action block features ..... 87
45	Action qualifiers ..... 88
46	Sequence evolution ..... 94
47	Compatible SFC features ..... 101
48	SFC minimal compliance requirements ..... 101
49	Configuration and resource declaration features ..... 104
50	Task features ..... 108
51	Examples of instruction fields ..... 115
52	Instruction List (IL) operators ..... 117
53	Function block invocation features for IL language ..... 118
54	Standard function block input operators for IL language ..... 118
55	Operators of the ST language ..... 120
56	ST language statements ..... 121
57	Representation of lines and blocks ..... 126
58	Graphic execution control elements ..... 129
59	Power rails ..... 130
60	Link elements ..... 131
61	Contacts ..... 132
62	Coils ..... 133

## Figures

1	Software model ..... 16
2	Communication model ..... 18
3	Combination of programmable controller language elements ..... 21
4	Examples of function usage ..... 44
5	Use of formal parameter names ..... 46
6	Examples of function declarations ..... 48
7	Examples of explicit type conversion with overloaded functions ..... 50
8	Examples of explicit type conversion with typed functions ..... 52
9	Function block instantiation example ..... 62
10	Examples of function block declarations ..... 64
11	Graphical use of function block names as variables ..... 66

Clause	Page
12 Examples of use of input/output variables .....	69
13 Semaphore usage example .....	70
14 ACTION_CONTROL function block – External interface .....	89
15 ACTION_CONTROL function block body .....	90
16 Action control example .....	91
17 SFC evolution rules .....	98
18 SFC errors .....	99
19 Configuration example .....	102
20 Examples of CONFIGURATION and RESSOURCE declaration features .....	105
21 Synchronization of function blocks .....	112
22 EXIT statement example .....	123
23 Feedback path example .....	128
24 Boolean OR examples .....	134

Withhold

iTech Standards  
(<https://standards.iteh.ai>)  
Document Preview

IEC 61131-3:1993

<https://standards.iteh.ai/catalog/standards/iec/317fe030-cf62-4ffb-9a84-9403172c2cd6/iec-61131-3-1993>



## INTERNATIONAL ELECTROTECHNICAL COMMISSION

**PROGRAMMABLE CONTROLLERS –****Part 3: Programming languages**

## FOREWORD

- 1) The formal decisions or agreements of the IEC on technical matters, prepared by technical committees on which all the National Committees having a special interest therein are represented, express, as nearly as possible, an international consensus of opinion on the subjects dealt with.
- 2) They have the form of recommendations for international use and they are accepted by the National Committees in that sense.
- 3) In order to promote international unification, the IEC expresses the wish that all National Committees should adopt the text of the IEC recommendation for their national rules in so far as national conditions will permit. Any divergence between the IEC recommendation and the corresponding national rules should, as far as possible, be clearly indicated in the latter.

This part of International Standard IEC 1131 has been prepared by sub-committee 65B: Devices, of IEC technical committee 65: Industrial-process measurement and control.

The text of this standard is based on the following documents:

DIS	Report on Voting
65B(CO)85	65B(CO)87

Full information on the voting for the approval of this standard can be found in the Voting Report indicated in the above table.

IEC 1131 will consist of the following parts, of which this is the third under the general title: Programmable controllers.

- Part 1: 1992, General information.
- Part 2: 1992, Equipment requirements and tests.
- Part 3: 1993, Programming languages.
- Part 4, User guidelines (*under consideration*).
- Part 5, Messaging service specification (*under consideration*).

Annexes A, B, C, D and E form an integral part of this part of IEC 1131.  
Annexes F, G and H are for information only.

A type 2 technical report (TR) will provide "pre-standardization" guidance for the implementation and application of the programming language defined in this part of IEC 1131, including such issues as operating system/program interaction and requirements for programming support environments.

Withdrawing

iTech Standards  
(<https://standards.iteh.ai>)  
Document Preview

IEC 61131-3:1993

<https://standards.iteh.ai/catalog/standards/iec/517fe030-cf62-4ffb-9a84-9403172c2cd6/iec-61131-3-1993>

## PROGRAMMABLE CONTROLLERS –

### Part 3: Programming languages

#### 1 General

##### 1.1 Scope

This part of IEC 1131 applies to the printed and displayed representation, using characters of the ISO/IEC 646 character set, of the programming languages to be used for Programmable Controllers as defined in Part 1 of IEC 1131. Graphic and semigraphic representation of the language elements which are defined in this part is allowed, but is not defined in this part.

The functions of program entry, testing, monitoring, operating system, etc., are specified in Part 1 of IEC 1131.

##### 1.2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of IEC 1131. At the time of publication, the editions indicated were valid. All normative documents are subject to revision, and parties to agreements based on this part of IEC 1131 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

IEC 50: *International Electrotechnical Vocabulary (IEV)*

IEC 559: 1989, *Binary floating-point arithmetic for microprocessor systems*

IEC 617-12: 1991, *Graphical symbols for diagrams – Part 12: Binary logic elements*

IEC 617-13: 1978, *Graphical symbols for diagrams – Part 13: Analogue elements*

IEC 848: 1988, *Preparation of function charts for control systems*

ISO/AFNOR: 1989, *Dictionary of computer science, ISBN 2-12-486911-6*

ISO/IEC 646: 1991, *Information technology – ISO 7-bit coded character set for information processing interchange*

ISO 8601: 1988, *Data elements and interchange formats – Information interchange – Representations of dates and times*

ISO 7185: 1990, *Information technology – Programming languages – Pascal*

ISO 7498: 1984, *Information processing systems – Open systems interconnection – Basic reference model*

### 1.3 Definitions

For the purposes of this part of IEC 1131, the following definitions apply. Definitions applying to all parts of IEC 1131 are given in part 1.

#### NOTES

- 1 Terms defined in this subclause are italicized where they appear in the bodies of definitions.
- 2 The notation "(ISO)" following a definition indicates that the definition is taken from the *ISO/AFNOR Dictionary of computer science*.
- 3 The *ISO/AFNOR Dictionary of computer science* and the *International Electrotechnical Vocabulary* should be consulted for terms not defined in this standard.

1.3.1 **absolute time:** The combination of time of day and date information.

1.3.2 **access path:** The association of a symbolic name with a variable for the purpose of open communication.

1.3.3 **action:** A Boolean variable, or a collection of operations to be performed, together with an associated control structure, as specified in 2.6.4.

1.3.4 **action block:** A graphical language element which utilizes a Boolean input variable to determine the value of a Boolean output variable or the enabling condition for an *action*, according to a predetermined control structure as defined in 2.6.4.5.

1.3.5 **aggregate:** A structured collection of data objects, forming a *data type*. (ISO)

1.3.6 **argument:** Synonymous with *input parameter* or *output parameter*.

1.3.7 **array:** An *aggregate* that consists of data objects, with identical attributes, each of which may be uniquely referenced by *subscripting*. (ISO)

1.3.8 **assignment:** A mechanism to give a value to a variable or to an *aggregate*. (ISO)

1.3.9 **based number:** A number represented in a specified base other than ten.

1.3.10 **bistable function block:** A *function block* with two stable states controlled by one or more inputs.

1.3.11 **bit string:** A data element consisting of one or more bits.

1.3.12 **body:** That portion of a *program organization unit* which specifies the operations to be performed on the declared *operands* of the program organization unit when its execution is *invoked*.

1.3.13 **call:** A language construct for *invoking* the execution of a *function* or *function block*.

1.3.14 **character string:** An *aggregate* that consists of an ordered sequence of characters.

1.3.15 **comment:** A language construct for the inclusion of text in a program and having no impact on the execution of the program. (ISO)

1.3.16 **compile:** To translate a *program organization unit* or *data type* specification into its machine language equivalent or an intermediate form.

1.3.17 **configuration:** A language element corresponding to a *programmable controller system* as defined in IEC 1131-1.

1.3.18 **counter function block:** A *function block* which accumulates a value for the number of changes sensed at one or more specified *inputs*.

1.3.19 **data type:** A set of values together with a set of permitted operations. (ISO)

1.3.20 **date and time:** The date within the year and the time of day, represented according to ISO 8601.

1.3.21 **declaration:** The mechanism for establishing the definition of a *language element*. A declaration normally involves attaching an identifier to the language element, and allocating attributes such as *data types* and algorithms to it.

1.3.22 **delimiter:** A character or combination of characters used to separate program *language elements*.

1.3.23 **direct representation:** A means of representing a variable in a programmable controller program from which a manufacturer-specified correspondence to a physical or *logical location* may be determined directly.

1.3.24 **double word:** A data element containing 32 bits.

1.3.25 **evaluation:** The process of establishing a value for an expression or a *function*, or for the *outputs* of a network or *function block*, during program execution.

1.3.26 **execution control element:** A *language element* which controls the flow of program execution.

1.3.27 **falling edge:** The change from 1 to 0 of a Boolean variable.

1.3.28 **function:** A *program organization unit* which, when executed, yields exactly one data element (which may be multi-valued, e.g., an *array* or *structure*), and whose *invocation* can be used in textual languages as an *operand* in an expression.

1.3.29 **function block instance (function block):** An *instance* of a *function block type*.

1.3.30 **function block type:** A programmable controller programming *language element* consisting of: (i) the definition of a data structure partitioned into input, output, and internal variables; and (ii) a set of operations to be performed upon the elements of the data structure when an *instance* of the function block type is *invoked*.

1.3.31 **function block diagram:** One or more networks of graphically represented *functions*, *function blocks*, data elements, *labels*, and connective elements.

1.3.32 **generic data type:** A *data type* which represents more than one type of data, as specified in 2.3.2.

1.3.33 **global scope:** Scope of a declaration applying to all program organization units within a *resource* or *configuration*.

1.3.34 **global variable:** A variable whose *scope* is *global*.

1.3.35 **hierarchical addressing:** The *direct representation* of a data element as a member of a physical or logical hierarchy, e.g., a point within a module which is contained in a rack, which in turn is contained in a cubicle, etc.

1.3.36 **identifier:** A combination of letters, numbers, and underline characters, as specified in 2.1.2, which begins with a letter or underline and which names a *language element*.

1.3.37 **initial value:** The value assigned to a variable at system start-up.

1.3.38 **input parameter (input):** A parameter which is used to supply an argument to a *program organization unit*.

1.3.39 **instance:** An individual, named copy of the data structure associated with a *function block type* or *program type*, which persists from one *invocation* of the associated operations to the next.

1.3.40 **instance name:** An *identifier* associated with a specific *instance*.

1.3.41 **instantiation:** The creation of an *instance*.