Information technology — Object Management Group Unified Modeling Language (OMG UML)

Part 1: Infrastructure
This British Standard is the UK implementation of ISO/IEC 19505-1:2012.

The UK participation in its preparation was entrusted to Technical Committee IST/15, Software and systems engineering.

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 2012

Published by BSI Standards Limited 2012

ISBN 978 0 580 65509 8

ICS 35.060

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

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 30 November 2012.

Amendments issued since publication

<table>
<thead>
<tr>
<th>Amd. No.</th>
<th>Date</th>
<th>Text affected</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Information technology — Object Management Group Unified Modeling Language (OMG UML) —

Part 1: Infrastructure
# Table of Contents

1. **Scope** .......................................................... 1

2. **Conformance** .................................................. 1
   2.1 General ......................................................... 1
   2.2 Language Units ............................................... 2
   2.3 Compliance Levels ........................................... 2
   2.4 Meaning and Types of Compliance ......................... 3
   2.5 Compliance Level Contents .................................. 5

3. **Normative References** ....................................... 5

4. **Terms and Definitions** ....................................... 6

5. **Notational Conventions** ..................................... 6

6. **Additional Information** ..................................... 6
   6.1 Architectural Alignment and MDA Support ................ 6
   6.2 How to Proceed ............................................... 6
      6.2.1 Diagram format ........................................... 7

7. **Language Architecture** ..................................... 13
   7.1 General ....................................................... 13
   7.2 Design Principles ........................................... 13
   7.3 Infrastructure Architecture ................................ 13
   7.4 Core ......................................................... 14
   7.5 Profiles ...................................................... 16
   7.6 Architectural Alignment between UML and MOF ........... 16
   7.7 Superstructure Architecture ............................... 17
   7.8 Reusing Infrastructure ..................................... 18
   7.9 The Kernel Package ......................................... 18
   7.10 Metamodel Layering ....................................... 18
   7.11 The Four-layer Metamodel Hierarchy .................... 19
BS ISO/IEC 19505-1:2012
ISO/IEC 19505-1:2012(E)

7.12 Metamodelling ............................................................. 19
7.13 An Example of the Four-level Metamodel Hierarchy .......... 20

8. Language Formalism ......................................................... 23
  8.1 General ........................................................................... 23
  8.2 Levels of Formalism ........................................................ 23
  8.3 Package Specification Structure ....................................... 24
      8.3.1 Class Descriptions .................................................... 24
      8.3.2 Diagrams .................................................................. 24
      8.3.3 Instance Model ......................................................... 24
  8.4 Class Specification Structure .......................................... 24
      8.4.1 Description ............................................................... 25
      8.4.2 Attributes ................................................................. 25
      8.4.3 Associations ............................................................. 25
      8.4.4 Constraints ............................................................... 25
      8.4.5 Additional Operations (optional) ............................... 25
      8.4.6 Semantics ................................................................. 25
      8.4.7 Semantic Variation Points (optional) ......................... 25
      8.4.8 Notation ................................................................... 26
      8.4.9 Presentation Options (optional) ............................... 26
      8.4.10 Style Guidelines (optional) ................................. 26
      8.4.11 Examples (optional) .............................................. 26
      8.4.12 Rationale (optional) ............................................. 26
      8.4.13 Changes from UML 1.4 .......................................... 26
  8.5 Use of a Constraint Language ......................................... 26
  8.6 Use of Natural Language ................................................ 27
  8.7 Conventions and Typography ......................................... 27

9. Core::Abstractions .............................................................. 31
  9.1 BehavioralFeatures Package .......................................... 33
      9.1.1 BehavioralFeature .................................................. 33
  9.2 Parameter ...................................................................... 34
  9.3 Changeabilities Package ................................................. 35
      9.3.1 StructuralFeature (as specialized) ......................... 36
  9.4 Classifiers Package ........................................................ 36
      9.4.1 Classifier .............................................................. 37
      9.4.2 Feature ................................................................. 38
  9.5 Comments Package ........................................................ 39
      9.5.1 Comment .............................................................. 39
      9.5.2 Element ............................................................... 40
<table>
<thead>
<tr>
<th>Section</th>
<th>Package Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.6</td>
<td>Constraints Package</td>
<td>41</td>
</tr>
<tr>
<td>9.6.1</td>
<td>Constraint</td>
<td>42</td>
</tr>
<tr>
<td>9.6.2</td>
<td>Namespace (as specialized)</td>
<td>45</td>
</tr>
<tr>
<td>9.7</td>
<td>Elements Package</td>
<td>46</td>
</tr>
<tr>
<td>9.7.1</td>
<td>Element</td>
<td>46</td>
</tr>
<tr>
<td>9.8</td>
<td>Expressions Package</td>
<td>47</td>
</tr>
<tr>
<td>9.8.1</td>
<td>Expression</td>
<td>47</td>
</tr>
<tr>
<td>9.8.2</td>
<td>OpaqueExpression</td>
<td>48</td>
</tr>
<tr>
<td>9.8.3</td>
<td>ValueSpecification</td>
<td>49</td>
</tr>
<tr>
<td>9.9</td>
<td>Generalizations Package</td>
<td>51</td>
</tr>
<tr>
<td>9.9.1</td>
<td>Classifier (as specialized)</td>
<td>52</td>
</tr>
<tr>
<td>9.9.2</td>
<td>Generalization</td>
<td>53</td>
</tr>
<tr>
<td>9.10</td>
<td>Instances Package</td>
<td>54</td>
</tr>
<tr>
<td>9.10.1</td>
<td>InstanceSpecification</td>
<td>55</td>
</tr>
<tr>
<td>9.10.2</td>
<td>InstanceValue</td>
<td>58</td>
</tr>
<tr>
<td>9.10.3</td>
<td>Slot</td>
<td>59</td>
</tr>
<tr>
<td>9.11</td>
<td>Literals Package</td>
<td>60</td>
</tr>
<tr>
<td>9.11.1</td>
<td>LiteralBoolean</td>
<td>60</td>
</tr>
<tr>
<td>9.11.2</td>
<td>LiteralInteger</td>
<td>61</td>
</tr>
<tr>
<td>9.11.3</td>
<td>LiteralNull</td>
<td>62</td>
</tr>
<tr>
<td>9.11.4</td>
<td>LiteralReal</td>
<td>63</td>
</tr>
<tr>
<td>9.11.5</td>
<td>LiteralSpecification</td>
<td>64</td>
</tr>
<tr>
<td>9.11.6</td>
<td>LiteralString</td>
<td>64</td>
</tr>
<tr>
<td>9.11.7</td>
<td>LiteralUnlimitedNatural</td>
<td>65</td>
</tr>
<tr>
<td>9.12</td>
<td>Multiplicities Package</td>
<td>66</td>
</tr>
<tr>
<td>9.12.1</td>
<td>MultiplicityElement</td>
<td>67</td>
</tr>
<tr>
<td>9.13</td>
<td>MultiplicityExpressions Package</td>
<td>70</td>
</tr>
<tr>
<td>9.13.1</td>
<td>MultiplicityElement (specialized)</td>
<td>71</td>
</tr>
<tr>
<td>9.14</td>
<td>Namespaces Package</td>
<td>73</td>
</tr>
<tr>
<td>9.14.1</td>
<td>NamedElement</td>
<td>73</td>
</tr>
<tr>
<td>9.14.2</td>
<td>Namespace</td>
<td>75</td>
</tr>
<tr>
<td>9.15</td>
<td>Ownships Package</td>
<td>76</td>
</tr>
<tr>
<td>9.15.1</td>
<td>Element (as specialized)</td>
<td>77</td>
</tr>
<tr>
<td>9.16</td>
<td>Redefinitions Package</td>
<td>78</td>
</tr>
<tr>
<td>9.16.1</td>
<td>RedefinableElement</td>
<td>79</td>
</tr>
<tr>
<td>9.17</td>
<td>Relationships Package</td>
<td>81</td>
</tr>
<tr>
<td>9.17.1</td>
<td>DirectedRelationship</td>
<td>81</td>
</tr>
<tr>
<td>9.17.2</td>
<td>Relationship</td>
<td>82</td>
</tr>
<tr>
<td>9.18</td>
<td>StructuralFeatures Package</td>
<td>83</td>
</tr>
</tbody>
</table>
11.3.1 Expression ................................................................. 110
11.3.2 OpaqueExpression ..................................................... 111
11.3.3 ValueSpecification ..................................................... 111

11.4 Classes Diagram .......................................................... 112
  11.4.1 Association ............................................................... 113
  11.4.2 Class .............................................................. 120
  11.4.3 Classifier ............................................................... 123
  11.4.4 Operation ............................................................... 126
  11.4.5 Property ............................................................... 126

11.5 Classifiers Diagram ......................................................... 131
  11.5.1 Classifier ............................................................... 132
  11.5.2 Feature ............................................................... 133
  11.5.3 MultiplicityElement ................................................... 134
  11.5.4 RedefinableElement ................................................. 134
  11.5.5 StructuralFeature ..................................................... 135
  11.5.6 Type ............................................................... 136
  11.5.7 TypedElement ........................................................ 137

11.6 Constraints Diagram ....................................................... 137
  11.6.1 Constraint .............................................................. 138
  11.6.2 Namespace ............................................................. 139

11.7 DataTypes Diagram ......................................................... 139
  11.7.1 DataType ............................................................. 140
  11.7.2 Enumeration ......................................................... 141
  11.7.3 EnumerationLiteral ................................................ 143
  11.7.4 Operation .............................................................. 144
  11.7.5 PrimitiveType ......................................................... 144
  11.7.6 Property .............................................................. 145

11.8 Namespaces Diagram ..................................................... 146
  11.8.1 ElementImport ......................................................... 146
  11.8.2 NamedElement ....................................................... 149
  11.8.3 Namespace ............................................................. 150
  11.8.4 PackageableElement ............................................... 151
  11.8.5 PackageImport ....................................................... 152

11.9 Operations Diagram ....................................................... 153
  11.9.1 BehavioralFeature .................................................... 154
  11.9.2 Operation ............................................................. 156
  11.9.3 Parameter ............................................................. 159
  11.9.4 ParameterDirectionKind .................................................. 160

11.10 Packages Diagram ......................................................... 161
  11.10.1 Type ................................................................. 161
  11.10.2 Package .............................................................. 162
  11.10.3 PackageMerge ........................................................ 165

12. Core::Profiles ............................................................... 175
12.1 General ................................................................. 175

12.2 Profiles package ...................................................... 177
  12.2.1 Class (from Profiles) .................................................. 178
  12.2.2 Extension (from Profiles) ........................................... 179
  12.2.3 ExtensionEnd (from Profiles) ................................. 182
  12.2.4 Image (from Profiles) ............................................... 183
  12.2.5 Package (from Profiles) .......................................... 184
  12.2.6 PackageableElement (from Profiles) ...................... 186
  12.2.7 Profile (from Profiles) ............................................. 186
  12.2.8 ProfileApplication (from Profiles) ....................... 193
  12.2.9 Stereotype (from Profiles) ...................................... 194

13. PrimitiveTypes ......................................................... 203
  13.1 General ............................................................... 203
  13.2 PrimitiveTypes Package ........................................... 203
    13.2.1 Boolean ....................................................... 203
    13.2.2 Integer ......................................................... 204
    13.2.3 Real ............................................................ 205
    13.2.4 String .......................................................... 206
    13.2.5 UnlimitedNatural ............................................ 207

Subpart III - Annexes ................................................. 209

  Annex A: XMI Serialization and Schema .......................... 211
  Annex B: Support for Model Driven Architecture ............ 213
  Annex C: UML XMI Documents ..................................... 215

INDEX ............................................................................... 217
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.

Attention is drawn to the possibility that some of the elements of this part of ISO/IEC 19505 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

This International Standard was prepared by Technical Committee ISO/IEC/TC JTC1, Information technology, in collaboration with the Object Management Group (OMG), following the submission and processing as a Publicly Available Specification (PAS) of the OMG Unified Modeling Language (UML) specification.

This International Standard is related to:


This International Standard consists of the following parts, under the general title Information technology - Open distributed processing - UML specification:

- Part 1: Infrastructure
- Part 2: Superstructure

Apart from this Foreword, the text of this International Standard is identical with that for the OMG specification for UML, v2.4.1, Part 1.
Introduction

The rapid growth of distributed processing has led to a need for a coordinating framework for this standardization and ITU-T Recommendations X.901-904 | ISO/IEC 10746, the Reference Model of Open Distributed Processing (RM-ODP) provides such a framework. It defines an architecture within which support of distribution, interoperability, and portability can be integrated.

RM-ODP Part 2 (ISO/IEC 10746-2) defines the foundational concepts and modeling framework for describing distributed systems. The scopes and objectives of the RM-ODP Part 2 and the UML, while related, are not the same and, in a number of cases, the RM-ODP Part 2 and the UML specification use the same term for concepts that are related but not identical (e.g., interface). Nevertheless, a specification using the Part 2 modeling concepts can be expressed using UML with appropriate extensions (using stereotypes, tags, and constraints).

RM-ODP Part 3 (ISO/IEC 10746-3) specifies a generic architecture of open distributed systems, expressed using the foundational concepts and framework defined in Part 2. Given the relation between UML as a modeling language and Part 2 of the RM ODP standard, it is easy to show that UML is suitable as a notation for the individual viewpoint specifications defined by the RM-ODP.

The Unified Modeling Language (UML) is a general-purpose modeling language with a semantic specification, a graphical notation, an interchange format, and a repository query interface. It is designed for use in object-oriented software applications, including those based on technologies recommended by the Object Management Group (OMG). As such, it serves a variety of purposes including, but not limited to, the following:

- a means for communicating requirements and design intent,
- a basis for implementation (including automated code generation),
- a reverse engineering and documentation facility.

As an international standard, the various components of UML provide a common foundation for model and metadata interchange:

- between software development tools,
- between software developers, and
- between repositories and other object management facilities.

The existence of such a standard facilitates the communication between standardized UML environments and other environments.

While not limited to this context, the UML standard is closely related to work on the standardization of Open Distributed Processing (ODP).
1 Scope

This International Standard defines the Unified Modeling Language (UML), revision 2. The objective of UML is to provide system architects, software engineers, and software developers with tools for analysis, design, and implementation of software-based systems as well as for modeling business and similar processes.

The initial versions of UML (UML 1) originated with three leading object-oriented methods (Booch, OMT, and OOSE), and incorporated a number of best practices from modeling language design, object-oriented programming, and architectural description languages. Relative to UML 1, this revision of UML has been enhanced with significantly more precise definitions of its abstract syntax rules and semantics, a more modular language structure, and a greatly improved capability for modeling large-scale systems.

One of the primary goals of UML is to advance the state of the industry by enabling object visual modeling tool interoperability. However, to enable meaningful exchange of model information between tools, agreement on semantics and notation is required. UML meets the following requirements:

- A formal definition of a common MOF-based metamodel that specifies the abstract syntax of the UML. The abstract syntax defines the set of UML modeling concepts, their attributes and their relationships, as well as the rules for combining these concepts to construct partial or complete UML models.
- A detailed explanation of the semantics of each UML modeling concept. The semantics define, in a technology-independent manner, how the UML concepts are to be realized by computers.
- A specification of the human-readable notation elements for representing the individual UML modeling concepts as well as rules for combining them into a variety of different diagram types corresponding to different aspects of modeled systems.
- A detailed definition of ways in which UML tools can be made compliant with this International Standard. This is supported (in a separate specification) with an XML-based specification of corresponding model interchange formats (XMI) that must be realized by compliant tools.

2 Conformance

2.1 General

UML is a language with a very broad scope that covers a large and diverse set of application domains. Not all of its modeling capabilities are necessarily useful in all domains or applications. This suggests that the language should be structured modularly, with the ability to select only those parts of the language that are of direct interest. On the other hand, an excess of this type of flexibility increases the likelihood that two different UML tools will be supporting different subsets of the language, leading to interchange problems between them. Consequently, the definition of compliance for UML requires a balance to be drawn between modularity and ease of interchange.