



中华人民共和国国家标准

GB/T 28174.3—2011

统一建模语言(UML) 第3部分:对象约束语言(OCL)

Unified modeling language(UML)—
Part 3: Object constraint language(OCL)

2011-12-30 发布

2012-06-01 实施

中华人民共和国国家质量监督检验检疫总局
中国国家标准化管理委员会 发布

目 次

前言	III
引言	IV
1 范围	1
2 规范性引用文件	1
3 对象约束语言描述	1
4 抽象句法	23
5 具体句法	45
6 采用 UML 描述的语义	75
7 OCL 标准库	108
8 UML 模型中 OCL 表达式的使用	127
9 OCL、UML 与 MOF 三种元模型的衔接	136

前 言

GB/T 28174《统一建模语言(UML)》分为4个部分:

- 第1部分:基础结构;
- 第2部分:上层结构;
- 第3部分:对象约束语言(OCL);
- 第4部分:图交换。

本部分为GB/T 28174的第3部分。

本部分按照GB/T 1.1—2009给出的规则起草。

本部分参考面向对象工作组(OMG)的《统一建模语言:对象约束语言(OCL)》2.0版。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本部分由全国信息技术标准化技术委员会(SAC/TC 28)提出并归口。

本部分起草单位:广东省广业信息产业集团有限公司、广东万维博通信息技术有限公司、镇江金钛软件有限公司、北京大学、中国电子技术标准化研究所。

本部分主要起草人:许立勇、周伟强、唐泽欢、江善东、黄孝和、杨三宝、丁力、吴炯祥、邓海强、胡红林、高健。

引 言

统一建模语言(UML)是一种可视化规约语言,用于定义和构造计算机信息系统的制品,并将其文档化。它是一种通用建模语言,可以和所有主流的面向对象和面向构件的方法一起使用,并适用于所有的应用领域和实现平台(如:CORBA、J2EE、.NET 等)。

0.1 统一建模语言不同版本之间的关系

由于 UML 的技术较新,所以该国际标准历经多次的版本演化,下面是 UML 在 OMG 的演化过程:

1997	UML1.1
1998	UML1.2
1999	UML1.3
2001	UML1.4
2003	UML2.0

GB/T 28174 的本部分正文中的 UML 均指 UML2.0 统一建模语言和 GB/T 28174。

0.2 关于对读者的建议

需要了解语言中的元模型构造物,利用这些构造物进行元模型扩展或者是构造新的建模语言的用户可阅读基础结构部分(GB/T 28174.1)。

应用系统建模用户和建模工具制造方都需阅读上层结构部分(GB/T 28174.2)。但要注意,该部分的内容是交叉引用的,可不按目次顺序阅读。

对于要精确地对模型进行约束的应用系统建模用户或要支持对象约束语言的建模工具制造方,需阅读对象约束语言部分(GB/T 28174.3)。

支持在不同的软件工具间平滑且无缝地交换文档的建模工具制造方,需阅读图交互部分。

统一建模语言(UML)

第3部分:对象约束语言(OCL)

1 范围

GB/T 28174 的本部分规定了用于对各类软件系统进行可视化、详述、构造和文档化的统一建模语言。本语言也可用于对其他领域进行建模。

本部分是一种形式语言,它适用于描述 UML 模型上的表达式。这些表达式以规范的方式,规定了如何描述系统在建模过程中应成立的不变条件以及对模型中对象的查询。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件,仅注日期的版本适用于本文件。凡是不注日期的引用文件,其最新版本(包括所有的修改单)适用于本文件。

GB/T 28174.1 统一建模语言(UML) 第1部分:基础结构

GB/T 28174.2 统一建模语言(UML) 第2部分:上层结构

3 对象约束语言描述

3.1 OCL 的作用

UML 图(例如类图)通常不够精细,无法提供与规范有关的一切侧面。这其中就缺少描述模型中关于对象的添加约束。这些约束常常采用自然语言描述。而实践表明,这样做经常造成歧义。为了写出无歧义的约束,已经开发出几种新的所谓“形式语言”。传统上的形式语言,缺点是仅适合于有相当数学背景的人员,普通业务或系统建模者难以使用。

OCL 即为填补这一空白而研制出来。它是一种保留了易读易写特点的形式语言。它已在 IBM 的保险分部作为一种业务建模语言开发出来,根植于 Syntropy 方法。

OCL 是一种纯粹的规约语言,因而保证了 OCL 表达式不带副作用。当对一 OCL 表达式求值时,只是返回一个值。它不能改动模型中的任何事物。这意味着,系统的状态绝不因 OCL 表达式的求值而改变,纵然能用 OCL 表达式去规定一个状态的改变(例如,在后置条件中)。

OCL 不是编程语言,因此,不可能以 OCL 写出程序逻辑或流控制。在 OCL 之内,不能启用进程或激活非查询操作。OCL 本是一种建模语言,因此 OCL 表达式按定义不能直接执行。

OCL 是一种类型化语言,其中每一 OCL 表达式都有一类型。OCL 表达式要成为良构的,就应符合该语言的类型符合性规则。例如,不能拿 Integer 与 String 作比较。UML 模型内定义的每一类目(Classifier),都代表一种独特的 OCL 类型。此外,OCL 还包含一组补充的预定义类型(这在第 7 章“OCL 标准库”中描述)。

作为一种规约语言,所有实现问题都超出 OCL 的范围,不能以 OCL 表达。

对 OCL 表达式的求值是瞬时的。这就是说,模型中各对象的状态,在求值期间不能改变。

3.1.1 OCL 使用

OCL 能用于若干不同目的:

- a) 作为一种查询语言;
- b) 在类模型中规定对类和类型的不变式;
- c) 规定衍型(Stereotype)的类型不变式;