



中华人民共和国国家标准

GB/T 17175.1—1997
idt ISO/IEC 10165-1:1993

信息技术 开放系统互连 管理信息结构 第1部分：管理信息模型

Information technology—Open Systems Interconnection—
Structure of management information—
Part 1: Management Information Model



1997-12-25 发布



C9900581

1998-08-01 实施

国家技术监督局 发布

前 言

本标准等同采用 ISO/IEC 10165-1:1993《信息技术 开放系统互连 管理信息结构 第1部分：管理信息模型》，以及 ISO/IEC 10165-1:1993/Cor. 1:1994《信息技术 开放系统互连 管理信息结构 第1部分：管理信息模型 技术修改1》。

GB/T 17175 在《信息技术 开放系统互连 管理信息结构》的总标题下，目前包括下列三部分：

第1部分：管理信息模型

第2部分：管理信息定义

第4部分：被管客体的定义指南

本标准无论在技术内容上，还是在编排格式上均与国际标准保持一致。

本标准由中华人民共和国电子工业部提出。

本标准由电子工业部标准化研究所归口。

本标准起草单位：电子工业部标准化研究所。

本标准主要起草人：冯惠、黄家英、玉宝艾。



ISO/IEC 前言

ISO(国际标准化组织)和 IEC(国际电工委员会)是世界性的标准化专门机构。国家成员体(它们都是 ISO 或 IEC 的成员国)通过国际组织创建的各个技术委员会参与制定针对特定技术范围的国际标准。ISO 和 IEC 的各技术委员会在共同感兴趣的领域内进行合作。与 ISO 和 IEC 有联系的其他官方和非官方国际组织也可参与国际标准的制定工作。

对于信息技术,ISO 和 IEC 创建了一个联合技术委员会,即 ISO/IEC JTC1。由联合技术委员会提出的国际标准草案需分发给国家成员体进行表决。发布一项国际标准,至少需要 75%的参与表决的国家成员体投票赞成。

ISO/IEC 10165-1 是由 ISO/IEC JTC1 信息技术联合技术委员会与 CCITT 合作制定的,这个文本也以 CCITT X.720 发布。

ISO/IEC 10165 在《信息技术 开放系统互连 管理信息结构》的总标题下,包括下列部分:

第 1 部分:管理信息模型

第 2 部分:管理信息定义

第 4 部分:被管客体的定义指南

第 5 部分:类属管理信息

第 6 部分:与管理信息相关的实现一致性声明形式表的要求和指南

目次

前言 II

ISO/IEC 前言 IV

引言 V

1 范围 1

2 引用标准 1

3 定义 2

3.1 基本参考模型定义 2

3.2 管理框架定义 2

3.3 系统管理概述定义 2

3.4 公共管理信息服务定义 2

3.5 抽象语法记法一 定义 2

3.6 被管客体的定义指南的定义 2

3.7 安全体系结构定义 2

3.8 补充定义 3

4 缩略语 5

5 信息模型 5

5.1 使用面向对象设计的被管客体概念 6

5.2 兼容性和互操作性 8

5.3 系统管理操作 11

5.4 筛选器 18

5.5 通知 19

6 可容度和命名原则 19

6.1 可容度 19

6.2 命名树 20

6.3 名字结构 21

7 顶部属性 22

中华人民共和国国家标准

信息技术 开放系统互连 管理信息结构 第1部分:管理信息模型

GB/T 17175.1—1997
idt ISO/IEC 10165-1:1993

Information technology—Open Systems Interconnection— Structure of management information— Part 1: Management Information Model

1 范围

本标准是用于 OSI 管理信息服务(MIS)的系列标准之一。它定义了对应于 GB/T 17142 系统管理概述中所引入的系统管理模型信息方面的被管客体及其属性的信息模型,因此提供了开发其他系统管理标准所需的模型化概念。它还定义了被管客体的命名原则和属性。

本标准定义了系统管理信息的逻辑结构。按照 GB/T 9387.4 和 GB/T 17142,利用被管客体及其属性,以及在其上能执行的管理操作和能发出的通知,管理信息可被结构化。开放系统中的被管客体集合、连同它们的属性一起构成该开放系统的管理信息库(MIB)。

本标准定义了信息模型中被管客体的概念,并且规定了被管客体的命名原则及其属性,以致它们可以通过管理协议被标识和访问。本标准还描述了被管客体类的概念和被管客体之间的关系以及被管客体类可能要考虑的继承、专门化、同质异性和可容度。

本标准适用于被管客体及其属性的所有定义,进而用于系统管理。

注:尽管本标准适用于系统管理,当定义层管理时也可以使用本标准。

2 引用标准

下列标准所包含的条文,通过在本标准中引用而构成为本标准的条文。本标准出版时,所示版本均为有效。所有标准都会被修订,使用本标准的各方应探讨使用下列标准最新版本的可能性。

GB 9387—88 信息处理系统 开放系统互连 基本参考模型(idt ISO 7498:1984)

GB/T 9387.2—1995 信息处理系统 开放系统互连 基本参考模型 第2部分:安全体系结构(idt ISO/IEC 7498-2:1989)

GB/T 9387.3—1995 信息处理系统 开放系统互连 基本参考模型 第3部分:命名和编址(idt ISO/IEC 7498-3:1989)

GB/T 9387.4—1996 信息处理系统 开放系统互连 基本参考模型 第4部分:管理框架(idt ISO/IEC 7498-4:1989)

GB/T 16262—1996 信息技术 开放系统互连 抽象语法记法—(ASN.1)规范(idt ISO/IEC 8824:1990)

GB/T 16264.2—1996 信息技术 开放系统互连 目录 第2部分:模型(idt ISO/IEC 9594-2:1990)

GB/T 16264.3—1996 信息技术 开放系统互连 目录 第3部分:抽象服务定义(idt ISO/IEC

9594-3:1990)

GB/T 16644—1996 信息技术 开放系统互连 公共管理信息服务定义(idt ISO/IEC 9595:1991)

GB/T 16645.1—1996 信息技术 开放系统互连 公共管理信息协议 第1部分:规范(idt ISO/IEC 9596-1:1991)

GB/T 17142—1997 信息技术 开放系统互连 系统管理综述(idt ISO/IEC 10040:1992)

GB/T 17143.5—1997 信息技术 开放系统互连 系统管理 第5部分:事件报告管理功能(idt ISO/IEC 10164-5:1993)

GB/T 17175.4—1997 信息技术 开放系统互连 管理信息结构 第4部分:被管客体的定义指南(idt ISO/IEC 10165-4:1992)

3 定义

本标准采用下列定义。

3.1 基本参考模型定义

本标准采用 GB 9387(OSI 基本参考模型)中定义的下列术语:

- a) 开放系统 open system;
- b) 系统管理 systems management;
- c) (N)实体 (N)-entity;
- d) (N)层 (N)-layer;
- e) (N)协议 (N)-protocol。

3.2 管理框架定义

本标准采用 GB 9387.4 中定义的下列术语:

- a) 管理信息库 management information base;
- b) 被管客体 managed object。

3.3 系统管理概述定义

本标准采用 GB/T 17142 中定义的下列术语:

- a) 代理 agent;
- b) 管理者 manager;
- c) 通知 notification;
- d) 被管客体类 managed object class;
- e) (系统管理)操作 (systems management)operation。

3.4 公共管理信息服务定义

本标准采用 GB/T 16644 中定义的下列术语:

- a) 属性 attribute;
- b) 有值集合的属性 set-valued attribute。

3.5 抽象语法记法一定义

本标准采用 GB/T 16262 中定义的下列术语:

类型 type。

3.6 被管客体的定义指南的定义

本标准采用 GB/T 17175.4 中定义的下列术语:

模板 template。

3.7 安全体系结构定义

本标准采用 GB 9387.2 中定义的下列术语:

a) 访问控制 access control;

b) 安全策略 security policy.

3.8 补充定义

3.8.1 动作 action

被管客体上的操作,其语义定义为被管客体类定义的一部分。

3.8.2 实际类 actual class

被管客体是实例的那个被管客体类,不同于该被管客体的同质异性类。

3.8.3 (被管客体的)同质异性类 allomorphic class(of a managed object)

不同于被管客体实际类的一种类,使用同质异性可将被管客体作为这种类来管理。

3.8.4 同质异性 allomorphism

被管客体的一种能力,该被管客体是给定类的实例,它作为一个或多个其他被管客体类的实例来管理。

3.8.5 属性组 attribute group

为便于访问,已经赋予单个标识符的一组属性。

3.8.6 属性标识符 attribute identifier

用来将被管客体类属性同所有其他属性区分开的标识符。

3.8.7 属性类型 attribute type

特定种类的属性的已命名的定义,包括它的语法(类型)和语义的定义。一个属性是一个属性类型的实例。

3.8.8 属性值断言 attribute value assertion

可以是真也可以是假的一个语句,与属性值有关。

3.8.9 属性值集合 attribute value set

值的集合,其成员是属性的有效值。

3.8.10 行为 behaviour

被管客体、名字联编、属性、通知和与实际资源交互作用的动作,这些资源模型化并彼此交互作用的方法。

3.8.11 特性 characteristic

被管客体类定义的元素;它可以是属性定义、属性组定义、通知定义、行为定义、参数定义或包定义。

3.8.12 条件包 conditional package

如果其被管客体类定义所给定的条件被满足,在该给定的被管客体中呈现的包。

3.8.13 可容度 containment

被管客体的结构化关系,其中被管客体的存在依赖于包含的被管客体的存在。

3.8.14 可辨别名 distinguished name

由客体的 RDNs 序列及其各自的上级客体所形成的客体名字。

3.8.15 封装 encapsulation

被管客体与它的属性及行为之间的关系,它表示只有通过被管客体上的管理操作或者它所发出的通知才可以观察到属性和行为的性质。

3.8.16 继承 inheritance

由子类从上级类获得属性、通知、操作和行为的概念机制。

3.8.17 继承分级 inheritance hierarchy

被管客体类的分级排列,其中分级是在类专门化的基础上组织的。

3.8.18 初值被管客体 initial value managed object

用作源的被管客体,用于导出另一被管客体的初值。

3.8.19 例示 instantiation

按照被管客体类定义创建一被管客体的过程。

3.8.20 被管客体边界 managed object boundary

概念上的定位,这里低层资源的情况对管理来说是可视的,它定界了被管客体的定义范围。

3.8.21 必备包 mandatory package

在给定被管客体类的所有实例中必须呈现的包。

3.8.22 多继承 multiple inheritance

允许子类从一个以上的上级类获得属性、通知、操作和行为的概念机制。

3.8.23 名字联编 name binding

客体类间的关系,其指明一个已标识类的客体可以是另一个已命名类的客体的上级。名字联编定义也包含有关关系的其他信息,可定义成也能应用于上级或下级类或两者的子类。

3.8.24 命名模式 naming schema

名字联编的汇集。

3.8.25 命名树 naming tree

客体的分级排列,其中分级是在名字联编关系的基础上组织的。用来命名另一被管客体的客体在分级上高于被命名的客体。将命名的客体称之为被命名客体的上级,被命名的客体称之为下级。

3.8.26 包 package

属性、通知、操作和/或行为的汇集,这个汇集被处理为被管客体类规范中的单个模块。当在被管客体类定义中引用时,包可以被规定为必备的或有条件的。

3.8.27 参数 parameter

类型的值,它具有相关语义,并且与客体标识符和其他信息相关,这里类型的值可以在协议中登载。

3.8.28 许可的值集合 permitted value set

属性值集合,它包含已规定属性类型的尾性允许采用的所有值。

3.8.29 相关可辨别名 relative distinguished name

特定属性具有特定值的属性值断言,而该特定值可用来把所有那些直接下级中的某一个客体给一给定客体。它作为客体可辨别名的成分使用。

3.8.30 要求的值集合 required value set

属性值集合,它包含已规定属性类型的属性允许采用的所有值。

3.8.31 专门化 specialization

通过继承和增加新的特性,从一个或多个现存被管客体类派生新的被管客体类的技术。

3.8.32 子类 subclass

通过专门化由另一类别所派生的类。

3.8.33 上级类 superclass

通过专门化用于派生另一类的类。

3.8.34 上级客体 superior object

见 3.8.25。

3.8.35 下级客体 subordinate object

见 3.8.25。

3.8.36 非例示的被管客体类 uninstantiable managed object class

在开放系统内,通过系统管理操作或通过本地操作都不希望例示的类。

注:下列术语:

——属性;

——属性值断言;

——相关可辨别名；

——可辨别名。

也用于 GB/T 16264 中,并且在类似意义上考虑用于此,以反映目录模型和管理信息模型之间的相似性。然而,这些术语在两个模型中的用法在细节上还是不同的。

4 缩略语

AVA 属性值断言

CMIP 公共管理信息协议

CMIS 公共管理信息服务

GDMO 被管客体的定义指南

Id 标识符

IVMO 初值被管客体

MIB 管理信息库

MIS 管理信息服务

RDN 相关可辨别名

SMI 管理信息结构

5 信息模型

信息模型的目的是对通过系统管理协议外部运送的管理信息给出结构并且模型化相关资源(例如 X.25 协议机)的管理方面。信息模型涉及被管客体。被管客体是用于管理的抽象化的数据处理和数据通信资源(例如协议状态机、连接和调制解调器)。资源存在独立于它们被管理的需要。资源和作为抽象资源的被管客体间存在的关系不用一般的方法模型化,即,抽象的明确性质和作用于资源上的管理操作的特定效果必须作为被管客体类规范的一部分来规定。

管理可视的被管客体和用于管理目的而表示的资源之间的区别是这样描述的,即在被管客体的边界处,属性、操作和通知是管理可视的;另一方面,由被管客体表示的资源的内部功能是管理不可视的。被管客体边界的概念不隐含实现,但提供了两种定义之间的体系结构上的区别,一种是由被管客体类定义者(例如层组)开发的定义,该定义在边界处内侧,另一种是系统管理其余部分的定义和有关标准,而这些在边界处外侧。

被管客体类作为包的汇集来定义,其中每个包又作为属性、操作、通知和相关行为的汇集来定义。根据显式陈述的条件,这些包或者是必备的或者是有条件的。被管客体是被管客体类的实例。

为了使被管客体类的规范及其相关特性文件化,可使用一组模板。用于系统管理的模板在 GB/T 17143.5 中规定。

按照模板的规定,被管客体类的定义由下列内容组成:

——继承分级中的被管客体类的位置;

——属性、操作、通知和行为的必备包的汇集;

——属性、操作、通知和行为以及根据每个包将呈现的条件汇在一起的条件包的汇集;

——包结构内部,

——在被管客体边界处的可视属性;

——可适用于被管客体的操作;

——由被管客体所展示的行为;

——可由被管客体发出的通知。

在这些有关事项中,对于给定被管客体类的实例和用于命名的属性(见第 6 章)一起,其他模板规定了可能的上级客体。

由被管客体类所表示的资源的其他方面对于系统管理是不可视的。

按照一组规则,被管客体可被例示。按照模板的方法,这些规则如何规定类规范在创建被管客体中被实现。这些规则是:

a) 在所有的必备包和所有的满足条件的条件性包中,被管客体应支持已规定的所有属性、管理操作、行为和通知;

b) 按照适当模板规定了名字联编,并且被管客体被例示时,则被管客体应支持名字联编。如果不支持的名字联编被请求,则例示将失效。

每个被管客体是包含所有被管客体的一类实例,并且是所有被管客体共享相同的定义。可辨别名用于无二义性地命名每个被管客体。

从管理的观点来看,如果被管客体有一可辨别名(如 6.3.2 中定义的),并且支持为它的类定义的操作和通知,则该被管客体是存在的。否则从管理的观点来看,它是不存在的,即使物理对方是存在的。

5.1 使用面向对象设计的被管客体概念

在系统管理标准的表述中,当需要,而增加新的被管客体类和功能是已标识的。因此系统管理的设计要求采用某种途径使这些标准能按模块化的风格进行标准化,并提供协议和规程的扩充。信息模型使用面向对象的设计原则,因为该原则提供了上述能力并提供部分规范的重用性。

在信息模型中,面向对象的设计适用于管理活动涉及的开放系统在协议交换时所看到的管理信息规范。但不必适用于系统实现。

面向对象的设计通过客体定义来表征,这里,客体是物理或逻辑事情的抽象。

注:当在比 OSI 管理更广泛的上下文中引用客体时,术语“客体”用于本文件。术语“被管客体”用来引用为管理的目的而表示资源的客体。

5.1.1 封装

面向对象设计的某一方面就是封装。封装确保客体的完整性得到保护。这种情况要求要执行的所有操作是通过向客体发送“报文”来完成。即,被管客体的内部操作在客体的边界处是不可视的,除非属性、操作或通知被定义用来外露该信息。被管客体类的定义规定了什么操作可以被执行,什么一致性约束被要求用来维护被管客体的完整性。

5.1.2 被管客体类及其特性

共享相同定义的被管客体是相同被管客体类的实例。给定类的不同实例将要共享该类的必备包中定义的属性、操作、通知和行为,并且将共享条件包中定义的属性、操作、通知和行为,从而这些实例满足与这些包相关的条件。

5.1.2.1 包

包是特性即属性、通知、操作和/或行为的汇集,它是被管客体类定义的完整模块。当包在被管客体定义中被引用时,它规定为必备的或有条件的。必备包必须呈现在给定被管客体类的所有实例中。条件包是应呈现在被管客体中的包,对于被管客体而言,被管客体类定义中,与那个包相关的显式条件为 TRUE。相同特性可以呈现在一个以上的包中。包呈现的条件或者与被管客体所模型化的低层资源能力相关,或者与被管系统所支持的管理功能存在或不存在相关。在 OSI 标准被管客体(例如,运输层协议机)的情形中,这些包将模型化已规定为相关规范一部分的一些选项。

包有下列性质:

- 仅有一个给定包的实例能在被管客体中存在;
- 由于仅有包的一个实例能在任何被管客体中存在,则名字联编无须向包赋值;
- 被管客体一旦已封装,则属性、操作、通知和行为成为被管客体的完整部分,并仅当作为该被管客体一部分时是可访问的;
- 若没有被管客体封装了包,则该包决不能被例示;
- 在作为被管客体的同时,包必须被例示,包的例示在较后时刻是不允许的;

f) 在作为被管客体的同时,包必须被删除,包的删除在较早时刻是不允许的;

g) 操作总是在被管客体上执行,而不在包上执行。

由于被管客体类的被管客体并非全部包含该被管客体类定义的所有允许的条件包。由被管客体支持的已登记的包在被管客体的包属性(见第7章)中进行标识。

5.1.2.2 属性

被管客体有属性。属性有相关值,该值可以展现结构,即它可由元素的集合或元素序列组成。属性值断言(AVA)是一语句,就属性值而言,该断言可以为真也可以为假。

属性值可被观察(在被管客体边界处)。属性值能够确定或反映被管客体的行为。通过对被管客体发送读(得到)或写(替代)值的请求,属性的值可被观察或修改。定义附加的操作用于有值集合的属性;这些属性是其值为元素集合的属性,每个元素具有相同的数据类型。关于属性操作被定义为根据包含属性的被管客体而不是直接根据属性来执行。被管客体有能力对属性值强制实施约束以保证内部的一致性。被管客体类的定义可以在各个属性的值之间规定约束。对一特定属性所执行的操作在被管客体类的定义中加以规定。

属性或在必备包中或在条件性包中被定义。因此,被定义为必备包一部分的属性可呈现在被管客体类的所有实例中,另一方面,那些被定义为条件包一部分的属性可呈现在那些满足与包相关的条件的实例中。

5.1.2.2.1 属性值集合

属性的语法是 ASN.1 类型,其描述了属性值的实例如何在协议中被登载。该语法对于属性是固有的并且对属性的所有用法保持恒定。

在被管客体类规范内,属性的性质根据允许的值集合和要求的值集合进一步定义。允许的值和要求的值集合规定了对属性值的限制。

要求的值集合规定了要求属性能够获得的所有值。如果未要求特定值,则该集合是空的。

被管客体必须能由要求的值集合中规定的任何一个值来替代属性的值,并受行为或其他限制,诸如访问控制的约束。

允许的值集合规定了允许属性获得的可能值。

被管客体在响应该被管客体读属性值的操作请求时,不应返回允许的值集合之外的属性值。被管客体应拒绝修改允许的值集合之外的属性值的请求。

允许的值集合应是语法值的子集,而要求的值集合应是允许的值集合的子集,这里,同一性在两种情形下都是允许的。

5.1.2.2.2 有值集合的属性

有值集合的属性是一属性,其值是给定类型成员的无序集合。集合的大小是可变的,并且集合可以是空的。有值集合的属性的定义部分对于集合的基数是允许的和要求的值。此外,对于所有的属性类型,操作都是能用的,用于有值集合属性所定义的操作允许向有值集合的属性增加元素或从其移走元素。

5.1.2.3 属性组

属性组提供了引用包含属性集合的被管客体内的属性集合的手段。定义了属性组的两种类型:固定的和可扩充的。不管扩充是不是可能,都是属性组的定义部分。

固定属性组是一属性组,其属性集合被定义为初始属性组定义的一部分,并且其属性集合不能用任何方式改变。对于固定属性组,所有的属性都是属性组的一部分,在同一包中所有属性应定义为属性组。

可扩充的属性组是这样的属性组,即属性可以增加作为专门化结果。对于可扩充的属性组,为每一个扩充规定的属性或者作为属性组在相同的条件包中被定义,或者在必备包中被定义。

在被管客体类定义中规定组成属性组的各个属性。属性组没有其自身的值。只有在属性组上允许的操作是不要求规定值的。

在属性组上允许的操作被解释为引用该属性组所包含的每个单独属性上所对应的操作。该操作以

无特定次序应用于属性。

被管客体类可以有一个以上的属性组。各个属性可以包含在一个以上的属性组中。

5.1.2.4 行为

被管客体类定义的一部分是行为。

行为可以定义：

- a) 属性、操作和通知的语义；
- b) 对管理操作的响应在被管客体上被调用；
- c) 发出通知的环境；
- d) 特定属性值之间的依赖性，这种依赖性必须用一种方式来表达，该方式考虑条件性包可能存在或不存在；
- e) 对参与被管客体的关系影响；
- f) 关于属性的一致性约束；
- g) 当操作和通知能假设为有效含义时，标识条件的前提条件；
- h) 标识管理操作处理或通知发出的结果的后置条件；
- i) 作用于被管客体的整个生命期的不变量和用于被管客体的操作条件为真的描述条件；
- j) 被管客体的同步性质。

本标准定义了一组用来定义被管客体行为各方面的模板。

5.1.3 专门化和继承

一个被管客体类可以由另一个被管客体类通过它作为其他被管客体类的扩充而被专门化。这样的扩充是通过定义包含一个或多个下列内容的进一步的包来完成的：

- 新的管理操作；
- 新的属性；
- 新的通知；
- 新的行为；
- 扩充原始被管客体类的特性。

给定被管客体类的能力可以扩充的方法在 5.2.2 中详细规定。

由另一被管客体类专门化的被管客体类称为那个类(上级类)的子类。称为顶部的一個被管客体类在类分级中被指定为最终上级类。顶部是一不可例示的被管客体类。

子类继承上级类的操作、属性、通知、包和行为。本标准仅允许特性的严格继承，即按 5.2.2 中定义的规则，子类的每个实例与其上级类兼容。通过删除上级类的任何特性而专门化是不允许的。

多继承是子类从一个以上的上级类中被专门化的能力。子类从一个以上的上级类中继承操作、属性、通知、包和行为。

当一类从多个上级类多继承同一特性时，则那个类被定义为似乎仅从单个上级类继承那一特性。专门化应不在子类定义中引入矛盾。

当它可能适用于管理时，图 1 示出了一个继承分级的例子。

5.2 兼容性和互操作性

5.2.1 需求

管理和被管系统间存在互操作性的需求。或当被管系统被增强时，或当一个或多个被管客体定义被扩充时，也存在维护互操作性的需求。

下列内容是用于给定被管客体的系统管理的特定互操作性需求：

- a) 一个系统管理相当了解给定被管客体的被管客体类定义的另一系统必须是可能的；
- b) 一个系统管理较少了解给定被管客体的被管客体类定义的另一系统必须是可能的；
- c) 在可行范围内，一系统管理较多了解给定被管客体的被管客体类定义的一系统必须是可能的。

特别是,如果不要要求扩充能力的一种需求,管理必须就象被管系统没有扩充能力那样尽可能有效。

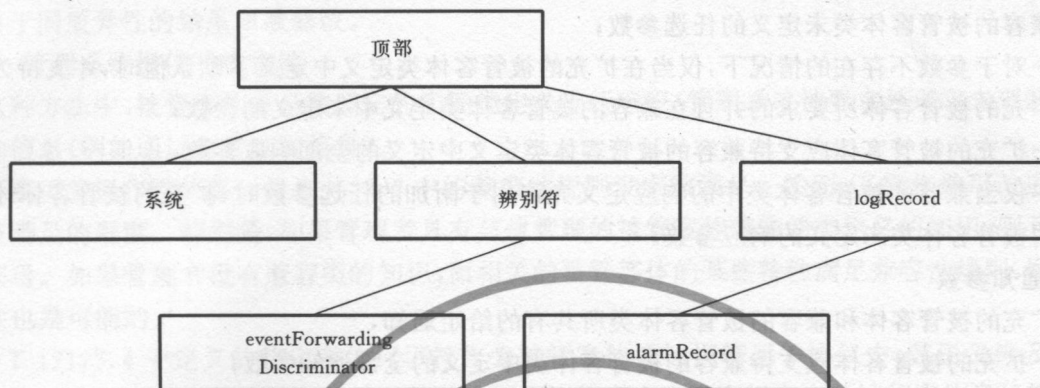


图1 继承分级的例子

5.2.2 兼容性规则

本条定义了一组规则,它保证一个被管客体类实例的一被管客体(称为扩充的被管客体)与第二个被管客体类(称为兼容的被管客体类)的定义兼容。这两个被管客体类不必通过继承相关联。

定义这些规则用于如下两个目的:

- 用于严格继承的定义(见 5.1.3);
- 用于互操作性方法。

5.2.2.1 附加特性

扩充的被管客体应包含在相同条件下例示的兼容被管客体类的实例中呈现的属性、属性组、管理操作和通知。附加属性、属性组、管理操作和通知也可包含在扩充的被管客体中。

如果上述规则适宜于这些必备包中的所有特性,则扩充的被管客体中例示的必备包和兼容的被管客体类定义中定义的那些必备包不必相关联。

扩充的被管客体应包含为兼容的被管客体类而定义的所有条件包,其呈现的条件适宜于扩充的被管客体。

5.2.2.2 包条件

兼容的被管客体类中的条件包呈现的条件为真,在这种情况下,扩充的被管客体中相同的条件包的条件也应被满足。该规则允许兼容的被管客体类中的条件包在扩充的被管客体中为必备的。

5.2.2.3 关于属性值的约束

关于扩充的被管客体 and 兼容的被管客体类共有的属性采取的值存在约束。关于每个这样的属性的一般条件是:兼容的被管客体类定义中定义的要求的值集合是由扩充的被管客体支持的值集合的子集,它依次是为兼容的被管客体类而定义的允许的值集合的子集(在两种情况下允许相等)。因此,扩充的被管客体支持兼容的被管客体类所承认的全部值,而不支持该类中不允许的值。

对于非可读属性,仅仅对要求的值集合的条件是相关的。相反地,对于非可写属性,仅仅对允许的值集合的条件是相关的。

注:兼容性不保证扩充的被管客体将使用兼容的被管客体类中规定的初始的和默认的值。

5.2.2.4 属性组的约束

扩充的被管客体中的可扩充的属性组应包含按照兼容的被管客体类中相同属性组的定义要求呈现的全部属性,以及按照用于包括那些属性的包的条件所要求呈现的全部属性。

5.2.2.5 关于动作和通知参数呈现的约束

下列条件应适用于动作和通知参数。

a) 动作参数

对兼容的被管客体类和扩充的被管客体两者所共有的给定动作,

- 扩充的被管客体应支持兼容的被管客体类中定义的全部动作参数;

——仅当兼容的被管客体类定义中的动作定义允许用于附加的任选参数时,扩充的被管客体才应支持对兼容的被管客体类未定义的任选参数;

——对于参数不存在的情况下,仅当在扩充的被管客体类定义中定义了默认值时,才支持为了执行动作由扩充的被管客体所要求的并且在兼容的被管客体类定义中未定义参数;

——扩充的被管客体应支持兼容的被管客体类定义中定义的全部响应参数;

——仅当兼容的被管客体类中的响应定义允许用于附加的任选参数时,扩充的被管客体才应支持对兼容的被管客体类未定义的响应参数。

b) 通知参数

对扩充的被管客体 and 兼容的被管客体类所共有的给定通知,

——扩充的被管客体应支持兼容的被管客体类中定义的全部通知参数;

——仅当兼容类定义中通知的定义允许用于附加的参数时,扩充的被管客体才应支持兼容的被管客体类定义中未定义参数。

5.2.2.6 扩充行为定义

行为定义的扩充规则是:扩充的被管客体类的行为定义与兼容的被管客体类的行为不相矛盾。

部分保证这样的矛盾不会发生,下列规则用于扩充的被管客体的行为:

——扩充的被管客体应包含兼容的被管客体类中的全部不变量;

——扩充的被管客体中的先决条件应为兼容的被管客体类中定义的先决条件的合取组合,并且任何新的先决条件适用于扩充的被管客体;

——扩充的被管客体中的后置条件应为兼容的被管客体类中定义的后置条件的析取组合,并且任何新的后置条件适用于扩充的被管客体。

5.2.3 提供互操作性的方法

保证互操作性的两种方法在 5.2.3.1 和 5.2.3.2 中描述。这些方法不论在代理系统或者在管理系统提供附加能力方面都有根本差别。

5.2.3.1 代理系统提供的互操作性

同质异性是被管客体的能力,当这种能力代理系统提供时,该被管客体是作为一个或多个其他被管客体类的成员而受管理的给定类的实例。

5.2.3.1.1 对于被管客体实例的同质异性

同质异性是被管客体的性质。如果被管客体与 5.2.2 中定义的被管客体类兼容,则支持同质异性的被管客体可以就像它是另一被管客体类的实例那样而加以管理。这些可被管理的被管客体类称为它的同质异性类。

给定被管客体的同质异性类可以是继承分级中的被管客体的类的上级类之一。然而,这不是对同质异性的要求。

5.2.3.1.2 对于操作的同质异性类的确定

支持同质异性的被管客体可以作为它的同质异性类中的任一个而加以管理。在被管客体的边界,对于被管客体来说,该被管客体类被要求是同质异性并且是可用的。

被管客体按照下列一般原则进行响应:

——对操作在同质异性类中的有效性进行检验;

——按照实际类的行为来执行操作;

——按照 5.3 中的同质异性行为的描述来生成响应。

5.2.3.1.3 对于通知的同质异性类的确定

当支持同质异性的被管客体发出通知时,它使得用于定义通知的同质异性类的集合连同该通知在被管客体边界处是可用的。如果通知被选择用于传输,则代理系统或确定是同质异性类,或确定是实际类,这都包含在事件报告中传给管理者。

事件报告的信息内容将精确地是用于发出通知的被管客体而在被管客体类中定义的那个信息,即它不能由于同质异性的结果而被修改。

5.2.3.2 管理系统提供互操作性

在这种方法中,被管客体总是按照它的实际类定义进行响应。管理系统被要求处理它未理解的或期望的附加信息(例如通过忽略这种信息)。

为了获得互操作性的某一级别,5.2.2中的兼容性规则没有被满足。然而,互操作性可改进到兼容性规则被满足的程度。特别是,如果管理者具有与他管理的被管客体兼容的类定义的知识,则互操作性很容易获得。如果管理者没有兼容类的知识,而相关的被管客体的某些特性满足兼容性规则,则有限的互操作性也是可能的。

GB/T 17175.4中定义的特定客体标识符作为被管客体类标识符用于协议中,其语义涉及相关被管客体的实际被管客体类。这种客体标识符的使用允许管理者在没有规定实际被管客体类的情况下请求执行管理操作。

5.3 系统管理操作

定义了两种管理操作,一种管理操作可被发送到适用于其属性的被管客体,而另一种管理操作适用于作为整体的被管客体。这里定义的操作在被管客体边界本身处是可视的原始操作。仅当调用的管理系统具有执行该操作所需的访问权,并且一致性约束没有被违反,则在被管客体上执行的操作才是成功的。

这种一致性约束的例子是属性值之间必须维护的关系。一致性约束被规定为属性的行为定义或被管客体类定义的一部分。当执行一个操作,例如替代一属性值,可能违反一已定义的约束时,则这个操作不被执行且返回一个“处理失败”的指示。该差错指示可伴随由被管客体类定义者定义的“特定差错”参数。

操作的某些实例被证实,就是说它们要求向操作的调用者发送响应,以指示成功或失败;其他操作是未证实的,即它们要求不向调用者发送响应。就使用的证实而论,可以标识两种操作:一种操作,证实是内在的并且特定信息(如果有)要求作为结果的一部分(例如,获取属性值和创建操作);另一种操作,按照管理政策由管理系统可以任选地对证实进行选择(例如,替代属性值)。每种操作所属的种类在本章中根据各自的操作加以规定。

5.3.1 对管理信息的访问控制

根据安全策略,管理操作服从于访问控制。

5.3.2 管理操作的原子型同步

可以请求被管系统在带有原子型同步的几个被管客体上执行操作;即,使得或者所有操作应是成功,或者如果这是不可能的,则没有操作应被执行。原子型同步不适用于创建操作。当原子型同步有效时,由操作引起的中间状态在穿越其他管理操作时是不可视的。

为了原子型同步的目的,要求成功的定义用于每个管理操作。对于面向属性的操作,成功的定义是该操作在表中规定的全部属性都被成功地执行。对于删除操作,成功的定义是被管客体被删除。对于动作操作,成功的定义是该动作在无差错指示下被执行。

在特定的开放系统中,交叉客体同步的特定支持是该系统的本地事情。当对于特定被管客体的同步不被支持时,且当原子型同步被请求时,则该操作失败。

5.3.3 面向属性的操作

下列管理操作可被发送到适用于其属性的被管客体:

- 获取属性值;
- 替代属性值;
- 用默认值替代;
- 增加成员;

——移去成员。

5.3.3.1 全部面向属性操作的共有行为

本条定义全部面向属性的共有行为的方方面面。

适用于封装在被管客体中的属性的操作总是根据属性表操作的；即所有的属性根据操作请求进行操作，这些属性作为单个操作的一部分对被管客体来说认为是可使用的。

下列内容对于被管客体在确定面向属性操作是否被执行或如何被执行中是有用的：

——选择在确定哪些客体时所使用的属性标识符及相关比较操作符和值来执行操作（见 5.4）。

下列内容作为面向属性操作的结果在被管客体边界处是有用的：

——对那些依据能被操作时属性值的属性标识符及其相关值；

——对那些依据不能被操作时属性的差错指示。

下列差错指示是可辨别的：

——未知的属性标识符；

——请求的被管客体类不是实际的被管客体类或该被管客体的同质异性类之一；

——带有任选的“特定差错”参数的请求的处理失败。

注：对被管客体有用的或由被管客体使用的这种信息的手段不属于标准化。

对于在被管客体的属性上的管理操作性能的直接效果由管理操作来定义。例如，在属性上的替代操作的直接效果就是修改属性值。

在被管客体内的一个或多个属性上执行的管理操作能引起其他可觉察的变化，这些称为间接效果。间接效果是低层资源中相互关系的结果。下列间接效果能够发生：

——相同被管客体内的属性的修改；

——被管客体的行为的变化；

——相关被管客体中属性的修改；

——由目标被管客体中的一个或多个属性的修改引起的相关被管客体的行为变化。

上面描述的前两个间接效果是包含属于管理操作的属性的被管客体的行为结果。后两个间接效果是相关被管客体或关系定义的行为结果。

5.3.3.2 获取属性值

范围

该操作适用于封装在被管客体中属性，该客体的类定义允许根据属性获取属性值操作。

语义

读所请求的属性值表，或如果属性值表没有被提供，则读所有的属性值，并返回读的属性值同时指明不能读的属性值的差错。

如果在请求时提供的是空表（区别于无表），则应返回空属性表。

行为

该操作总是被证实的。

下列附加信息对被管客体在确定获取属性值操作是否被执行和如何被执行中是有用的：

——被读的属性值的属性或属性组标识符。

下列附加信息作为获取属性值操作的结果在被管客体边界处是有用的：

——能被读的那些属性的属性标识符及其值；

——不能被读的那些属性的差错指示。下列差错指示从所有面向属性的行为所共有的差错中是可辨别的：

——对于非可读的属性，属性值不可读。

注：对被管客体有用的或由被管客体使用的这些差错指示和属性值的手段不属于标准化。

同质异性行为