



中华人民共和国国家标准

GB/T 16975.1—2000
idt ISO/IEC 13712-1:1995

信息技术 远程操作 第1部分：概念、模型和记法

Information technology—Remote Operations
Part 1: Concepts, model and notation



2000-01-03 发布



C200106151

2000-08-01 实施

国家质量技术监督局 发布

前 言

本标准等同采用国际标准 ISO/IEC 13712-1:1995《信息技术 远程操作 第1部分:概念、模型和记法》、ISO/IEC 13712-1:1995/Amd. 1:1996 以及 ISO/IEC 13712-1:1995/技术更正 1。

GB/T 16975 在《信息技术 远程操作》总标题下,包括以下 3 个部分:

第 1 部分(即 GB/T 16975.1):概念、模型和记法

第 2 部分(即 GB/T 16975.2):OSI 实现 远程操作服务元素(ROSE)服务定义

第 3 部分(即 GB/T 16975.3):OSI 实现 远程操作服务元素(ROSE)协议规范

本标准的附录 A 是标准的附录,附录 B、附录 C 和附录 D 是提示的附录。

本标准由中华人民共和国信息产业部提出。

本标准由中国电子技术标准化研究所归口。

本标准起草单位:中国电子技术标准化研究所。

本标准主要起草人:王宝艾、过介堃。

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)通过国际组织建立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 建立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。

国际标准 ISO/IEC 13712-1 是由信息技术联合技术委员会 SC21:开放系统互连、数据管理和开放分布式处理与 ITU-T 合作制定的。这个文本也以 ITU-TX.880 公布。

ISO/IEC 13712 在《信息技术 远程操作》总标题下,包括以下 3 个部分:

- 第 1 部分:概念、模型和记法
- 第 2 部分:OSI 实现 远程操作服务元素(ROSE)服务定义
- 第 3 部分:OSI 实现 远程操作服务元素(ROSE)协议规范

附录 A 构成 ISO/IEC 13712-1 的一部分,而附录 B、附录 C 和附录 D 仅提供参考信息。

信息技术 远端操作 引言

远端操作(ROS)是客体间交互式通信的范例。因此它可用于分布式应用的设计和规范。所涉及的基本交互(作用)是由一个客体(调用者)调用某一操作,由另一个客体(执行者)执行该操作,可能还跟有返回给调用者的操作效果报告。

ROS 的概念是抽象的,可以用多种方法实现它。例如,使用 ROS 概念进行交互(作用)的客体可以由软件界面或 OSI 网络来隔开。

本标准描述了 ROS 概念和模型。它使用 ASN.1 来规定对应于 ROS 基本概念的信息客体类,例如操作和差错。这又提供了记法,以便设计者能规定这些类的特殊实例,例如特殊操作和差错。

本标准提供了一组类属的 PDU,这些 PDU 可用于相互远离的客体间 ROS 概念的实现。这些 PDU 可用于 ROS 的 OSI 实现,这些 PDU 在本系列标准的伙伴标准中规定。

本标准还对基于 ROS 应用的设计者提供了通用实用程序的一些定义。

目 次

前言 III

ISO/IEC 前言 IV

引言 V

1 范围 1

2 引用标准 1

3 定义 1

3.1 OSI 参考模型定义 1

3.2 ASN.1 定义 2

3.3 ROS 定义 2

4 缩略语 3

5 约定 3

6 ROS 模型 3

7 ROS 实现 5

8 ROS 概念 5

8.1 引言 5

8.2 操作 6

8.3 差错 7

8.4 操作包 8

8.5 连接包 8

8.6 联系约定 9

8.7 ROS 客体类 10

8.8 代码 11

8.9 优先级 11

9 类属 ROS 协议 11

9.1 引言 11

9.2 ROS 11

9.3 调用 12

9.4 返回结果 13

9.5 返回差错 14

9.6 拒绝 15

9.7 拒绝问题 17

9.8 调用标识符 17

9.9 非调用标识符 17

9.10 差错 18

9.11 联结 18

9.12 断联 18



10 实用定义	18
10.1 引言	18
10.2 空联结	18
10.3 空断联	19
10.4 拒绝	19
10.5 空操作	19
10.6 检查	19
10.7 确认	19
10.8 检查和确认	20
10.9 取消	20
10.10 取消失败	20
10.11 已取消	21
10.12 正向	21
10.13 反向	21
10.14 消费者执行	21
10.15 供应者执行	21
10.16 全部操作	22
10.17 重编码	22
10.18 转换	22
10.19 组合	23
10.20 ROS 单一抽象语法	23
10.21 ROS 消费者抽象语法	23
10.22 ROS 供应者抽象语法	23
附录 A(标准的附录) ASN.1 模块	24
附录 B(提示的附录) 记法使用指南	32
附录 C(提示的附录) ROS 宏的变换	37
附录 D(提示的附录) 客体标识符值的分配	39

中华人民共和国国家标准

信息技术 远程操作 第1部分:概念、模型和记法

GB/T 16975.1—2000
idt ISO/IEC 13712-1:1995

Information technology—Remote Operations

Part 1: Concepts, model and notation

1 范围

本标准规定了远程操作服务(ROS),该服务使用抽象语法记法(ASN.1)来定义与ROS基本概念相对应的信息客体类,又提供了允许应用设计者规定这些类的特殊实例的记法。

本标准还提供了一批定义,以规定使用ROS概念进行通信的客体间的类属协议。这些定义用于本系列标准的伙伴标准以提供ROS的OSI实现中使用的协议数据单元、服务原语和应用背景定义。

本标准还对基于ROS应用的设计者提供了通用实用程序的一些定义。

本标准没有规定一致性要求。

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/T 9387.1—1998 信息技术 开放系统互连 基本参考模型 第1部分:基本模型
(idt ISO/IEC 7498-1:1994)

GB/T 16975.2—1997 信息技术 远程操作 第2部分:OSI实现 远程操作服务元素(ROSE)
服务定义(idt ISO/IEC 13712-2:1995)

GB/T 16975.3—1997 信息技术 远程操作 第3部分:OSI实现 远程操作服务元素(ROSE)
协议规范(idt ISO/IEC 13712-3:1995)

ISO/IEC 8824-1:1995 信息技术 抽象语法记法一(ASN.1):基本记法规范

ISO/IEC 8824-2:1995 信息技术 抽象语法记法一(ASN.1):信息客体规范

ISO/IEC 8824-3:1995 信息技术 抽象语法记法一(ASN.1):约束规范

ISO/IEC 8824-4:1995 信息技术 抽象语法记法一(ASN.1):ASN.1规范参数化

ISO/IEC 9072-1:1989 信息处理系统 文本通信 远程操作 第1部分:模型、记法和服务定义
(eqv CCITT 建议 X.219:1988)

ISO/IEC 9072-2:1989 信息处理系统 文本通信 远程操作 第2部分:协议规范
(eqv CCITT 建议 X.229:1988)

CCITT 建议 X.407:1988 消息处理系统:抽象服务定义约定

3 定义

3.1 OSI参考模型定义

本标准采用GB/T 9387.1中定义的下列术语:

- a) 抽象语法 abstract syntax;
- b) 协议数据单元 protocol data unit;
- c) 服务质量 quality of service.

3.2 ASN.1 定义

本标准采用 ISO/IEC 8824-1 中定义的下列术语:

- a) (数据)类型 (data) type;
- b) (数据)值 (data) value.

本标准采用 ISO/IEC 8824-2 中定义的下列术语:

- a) 字段 field;
- b) (信息)客体 (information) object;
- c) (信息)客体类 (information) object class;
- d) (信息)客体集 (information) object set.

本标准采用 ISO/IEC 8824-3 中定义的下列术语:

- a) 约束 constraint;
- b) 异常值 exception value.

本标准采用 ISO/IEC 8824-4 中定义的下列术语:

- a) 参数化的 parameterized.

3.3 ROS 定义

本标准采用下列定义。

3.3.1 自变量 argument

调用一个操作附带的数据值。

3.3.2 联系 association

一对客体间的关系,对于操作的调用和执行起关联的作用。

3.3.3 联系约定 association contract

彼此可能有联系的一对通信客体的作用的规范。

3.3.4 不对称 asymmetrical

描述操作包(或联系约定)的双方所能执行的操作集是不同的。

3.3.5 连接包 connection package

在其间动态地建立和释放联系的过程中,一对通信客体的作用的规范。

3.3.6 约定 contract

为了规定共同的行为,对一个或多个客体要求的集合。

3.3.7 差错 error

操作不成功执行的报告。

3.3.8 等幂的 idempotent

一种操作特征,即在不改变执行者状态的情况下,可重复调用该操作。

3.3.9 链接操作 linked operation

一种操作,该操作在另一个操作的执行期间由后者的执行者调用且又是后者的调用者试图执行的一种操作。

3.3.10 客体 object

系统(也可能是系统中一个独立部分)的模型,该模型通过其初始状态及其行为来表征,这些行为是通过严格定义的接口处的外部交互(作用)而引起的。

3.3.11 操作 operation

一个客体(调用者)能够向另一个客体(执行者)请求的功能。

3.3.12 操作包 operation package

用于规定一对通信客体的作用的有关操作的集合,其中的每个操作能够被这对客体中的一个或两个调用,并且能够被对方执行。

3.3.13 (差错的)参数 parameter (of an error)

差错报告可附带的数据值。

3.3.14 结果 result

操作成功执行的报告可附带的数据值。

3.3.15 ROS 客体 ROS-object

一种客体,它与其他客体的交互(作用)是用 ROS 概念来描述的。

3.3.16 对称 symmetrical

描述操作包(或联系约定)的双方所能执行的操作集是相同的。

3.3.17 同步 synchronous

一种操作特征,即该操作一经调用,其调用者就只有在报告了效果之后才能调用另外一个(具有相同的预期执行者的)同步操作。

4 缩略语

本标准采用下列缩略语:

ASN.1	抽象语法记法一
PDU	协议数据单元
QOS	服务质量
RO(或 ROS)	远程操作

5 约定

本标准使用 ASN.1 来定义:

a) 对应于 ROS 概念的信息客体类——如此还提供了 ROS 应用的设计者能用以规定这些类特殊实例的记法。

b) 那些类的特定信息客体。

c) 类属 ROS 协议的 PDU。

d) 在这些定义中所需的数据类型。

这些定义的大部分被参数化了,因此,为了完善它们,其用户必须给出实际的参数。

6 ROS 模型

远程操作(ROS)是客体间交互通信的范例。使用 ROS 描述和规定进行交互(作用)的客体就是 ROS 客体。所涉及的基本交互(作用)是由一个 ROS 客体(调用者)调用操作,而由另一个 ROS 客体(执行者)执行其操作。

操作执行完成后(成功或不成功),可导致执行者对调用者返回一个执行效果的报告。这些如图 1 所示。

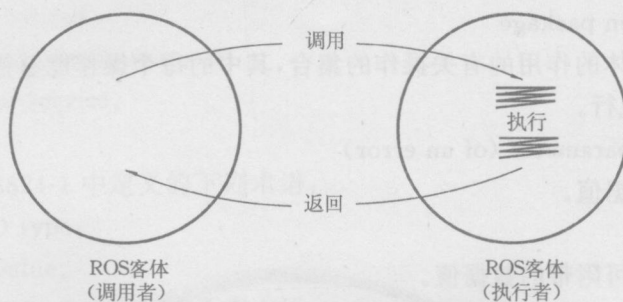


图1 操作的调用、执行和返回

操作成功完成的报告是**结果(result)**；操作不成功完成的报告是**差错(error)**。

在执行操作期间,执行者可以调用旨在由原来操作的调用者执行的**链接操作(linked operations)**。

为了正确地互工作,必须让调用者和执行者双方都了解操作的某些特点。这些特点包括:

是否要返回报告。若要,则返回哪一种;

值的类型。如果有的话,是随着操作的调用还是操作的返回来运送;

哪些操作。如果有的话,能与它链接;

为了把该操作与可能被调用的其他操作区分开来而使用的代码值。

根据称为**操作包(operation packages)**的相关操作集来定义某一**ROS 客体类(ROS-object class)**的(成对的)ROS 客体的互工作能力。包可以是对称的(**symmetrical**),在这种情况下,用单一操作集来定义,它可由这一对 ROS 客体中的每一个客体来调用。另一方面,包可以是不对称的(**asymmetrical**),在这种情况下,用两个操作集来定义,即可由这一对 ROS 客体中的某一客体调用的那些操作和可由另一个客体调用的那些操作来定义。为了定义非对称包,这两个客体可分别(任意地)标记为**消费者(consumer)**和**供应者(supplier)**。

注1:通常这些标记是任意的,一般是进行直观的分配,在这样一对客体中,一个提供服务,而另一个享用服务。

为了对操作的调用和执行起上下文的作用,一对 ROS 客体间必须存在联系。通过**联系约定(association contract)**来管理每一个这样的联系。通过(集中地)决定联系可调用的操作的包集合来规定约定。如果约定规范包括一个或多个非对称包,则约定本身是不对称的。为了规定不对称联系约定,相互建立联系的两个 ROS 客体可被标记为**发起者(initiator)**和**响应者(responder)**。

可通过“脱机”手段来建立和释放联系。另一种办法是动态地建立和释放联系。本标准中描述的选择方案是分别通过调用和执行专用的**联结(bind)**和**断联(unbind)**操作来动态地建立和释放联系。后一种类型的联系约定包括**连接包(connection package)**,其中包括所使用的具体的联结和断联操作。

注2:也可以用其他标准中描述的其他方法进行联系的建立和释放。

联系需要客体间的关系,即对于某一联系约定项目,关系对应于客体的共同协定的实体。

注3:本规范没有涉及建立或终止这些关系的方法。

在上述内容中,在操作中包括的客体才有可能是调用者和执行者。但是,一般情况下,操作的调用者和执行者并不直接地互相连接,而是通过可运送调用和返回的某一媒体来进行连接。展开的视图如图2所示。

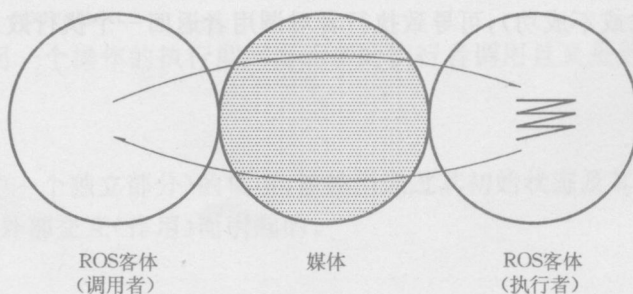


图2 展开的视图

在调用和返回的运送过程中以及在联系的建立、释放和维持时,媒体可能引起延迟,并且引起失败或不准确的可能性。它也可引起联系及其操作的安全受到威胁的可能性。这些内容(与其他因素一起)可描述为服务质量(QOS)。

联系约定目前能看作有三个部分,第三个部分是媒体。在约定期间媒体的职责是满足 QOS 要求。

注 4: 今后,目标和最低可接受 QOS 要求可能形成操作、操作包规范的一部分,并且直接形成联系约定本身的一部分。

7 ROS 实现

为了在 ROS 客体之间运送调用和返回结果,ROS 的实现(realization)包括定义适当的媒体。例如,这样的媒体可包含:

a) 信息传递或过程调用能力以使操作的调用者和执行者能在单个计算机系统中以独立开发的软件模块来实现;

b) 通信能力以使操作的调用者和执行者能在独立的计算机系统中实现。

实现可以是通用的,在这种情况下,它能用来支持任一联系约定。另一些实现是专用的,它仅适用于某些特殊约定。

图 3 描述了利用通信手段实现 ROS 的方法,这种方法可能具有很广泛的用途。

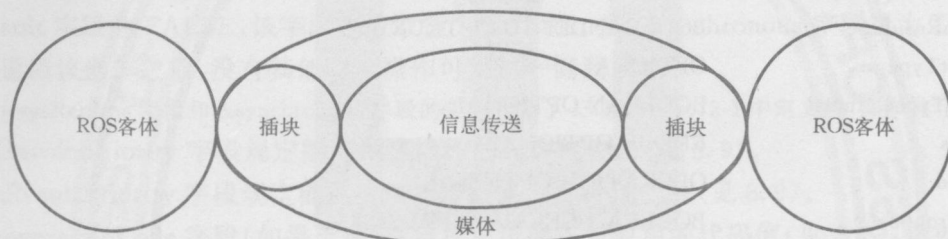


图 3 ROS 的通信实现方法

在这种方法中,媒体由若干插块(stub)客体和信息传送客体组成,每一个 ROS 客体对应一个插块客体。与每个 ROS 客体相关的插块客体似乎起着相应 ROS 客体的作用。但是,它实际上并不调用或执行任何操作,它只不过是相应地将调用和返回转换成 PDU,反之亦然。通过信息传送客体在插块间交换这些 PDU。

因此,为了调用一个操作,调用者调用相关插块的操作,它形成一个描述该调用的 PDU。插块利用信息传送能力将 PDU 发送给另一个插块。后一个插块解释该 PDU,然后调用与执行者相关的 ROS 客体的适当操作。当操作执行完时,执行者向其相关的插块运送任一个返回,该插块形成一个描述返回的 PDU。然后该插块利用信息传送能力将该 PDU 运送给另一个插块。后一个插块解释该 PDU 并且把返回报告给调用者。

第 9 章定义了一批合适的 PDU。

在这种 ROS 实现中利用了各种信息传送能力。尤其重要的是 OSI 信息传送能力。本标准的伙伴标准 GB/T 16975.2 和 GB/T 16975.3 描述了许多这样的实现。

8 ROS 概念

8.1 引言

8.1.1 本章定义了对应于 ROS 基本概念的信息客体类,以便规定这些类的客体具有的特性。定义了下列信息客体类:

- OPERATION(描述操作);
- ERROR(描述差错);
- OPERATION-PACKAGE(描述操作包);

——CONNECTION-PACKAGE(描述连接包);

——CONTRACT(描述联系约定);

——ROS-OBJECT-CLASS(描述 ROS 客体类)。

8.1.2 使用 ASN.1 来定义信息客体类。如此提供的记法可用于 ROS 应用的设计者规定这些类的具体事例。鼓励设计者使用该规范方法,但不强制使用。如果使用其他的方法,最终的规范应包括或涉及如何获得有效使用所提供的记法的描述。

注:许多现有的规范使用 ASN.1 宏记法(它在 ROS 标准的先前版本中定义:见 ISO/IEC 9072-1)来规定有关 ROS 信息客体的操作、差错和其他类。附录 C 描述了如何将这些宏的使用转换成本标准所提供的记法。对于新的应用不应使用这些宏。

8.2 操作

8.2.1 操作是一个客体(调用者)能向另一个客体(执行者)请求的功能。规定信息客体类 OPERATION(所有操作均属于它)如下,各个字段在 8.2.2~8.2.13 中描述:

OPERATION::=CLASS		
{		
&ArgumentType	OPTIONAL,	
&argumentTypeOptional	BOOLEAN OPTIONAL,	
&returnResult	BOOLEAN DEFAULT TRUE,	
&ResultType	OPTIONAL,	
&resultTypeOptional	BOOLEAN OPTIONAL,	
&Errors	ERROR OPTIONAL,	
&Linked	OPERATION OPTIONAL,	
&synchronous	BOOLEAN DEFAULT FALSE,	
&idempotent	BOOLEAN DEFAULT FALSE,	
&alwaysReturns	BOOLEAN DEFAULT TRUE,	
&InvokePriority	Priority OPTIONAL,	
&ResultPriority	Priority OPTIONAL,	
&operationCode	Code UNIQUE OPTIONAL	
}		
WITH SYNTAX		
{		
[ARGUMENT	&ArgumentType	[OPTIONAL &argumentTypeOptional]]
[RESULT	&ResultType	[OPTIONAL &resultTypeOptional]]
[RETURN RESULT	&returnResult]	
[ERRORS	&Errors]	
[LINKED	&Linked]	
[SYNCHRONOUS	&synchronous]	
[IDEMPOTENT	&idempotent]	
[ALWAYS RESPONDS	&alwaysReturns]	
[INVOKE PRIORITY	&InvokePriority]	
[RESULT-PRIORITY	&ResultPriority]	
[CODE	&operationCode]	
}		

8.2.2 &ArgumentType 字段规定了操作的自变量的数据类型。如果在某一操作中省略该字段,则该操作无自变量值。

8.2.3 &argumentTypeOptional 字段规定操作自变量的数据类型是否可被省略,只有当

&ArgumentType 字段出现时,该字段才出现。如果该字段不存在或取值为 FALSE,在 Invoke{}PDU 中不能省略 &ArgumentType 的值(见 9.3)。

8.2.4 &returnResult 字段规定,如果该操作执行成功是否返回结果,若返回,则取值 TRUE,否则取值 FALSE。

8.2.5 &ResulType 字段规定操作的结果返回值的数据类型。如果省略该值,那么该操作就无结果值返回。如果 &returnResult 字段为 FALSE,应省略该字段。

8.2.6 &resultTypeOptional 字段规定是否可以省略作为执行操作结果的返回值的数据类型,只有当 &ResultType 字段出现时,该字段才出现。如果该字段不存在或取值 FALSE,在 ReturnResult{}PDU 中不能省略 &ResultType 的值(见 9.4)。

8.2.7 &Errors 字段规定一组差错,其中的任何一个都可返回来报告操作执行不成功。如果省略该字段,则不报告操作执行不成功。

8.2.8 &alwaysReturns 字段规定是否总是返回执行该操作的效果,如果返回,则取值 TURE,否则取值 FALSE。

8.2.9 &Linked 字段(如果出现)规定一组操作,其中的任何一个都可在操作执行期间作为链接操作被调用。如果省略该字段,这一操作的调用不链接其他操作。

8.2.10 &synchronous 字段规定该操作是否同步,如果同步,则取值 TRUE,否则取值为 FALSE。如果 &returnResult 字段为 FALSE,该字段也应取值为 FALSE。当 &synchronous 字段置为 TRUE 时,这就意味着,在返回该操作之前,没有其他同步操作可由这一侧调用。

注: &alwaysReturns 字段和 &synchronous 字段的组合代替了 ISO/IEC 9072-1 中定义的“操作类”的早期概念。

8.2.11 &InvokePriority 字段规定能调用该操作的允许优先级(见 8.9)。

8.2.12 &ResultPriority 字段规定能返回该操作结果的允许优先级(见 8.9)。

8.2.13 &operationCode 字段(如果出现)规定标识该操作所使用的代码值(见 8.8),例如,当该操作要被调用时。

注:使用 Invoke{}PDU 不能调用不具有 &operationCode 的操作(见 9.3)。实际上,除打算在某一特定情况中使用外,所有的操作应具有分配的 &operationCodes,例如,作为一个联结操作。

8.2.14 &idempotent 字段规定操作是否是等幂的,如果是等幂的,取值 TRUE,否则取值 FALSE。

8.3 差错

8.3.1 差错是操作执行不成功的报告。规定信息客体类 ERROR(所有的差错都属于它)如下,各个字段在 8.3.2~8.3.5 中描述。

ERROR::=CLASS			
{			
	&ParameterType	OPTIONAL,	
	&ParameterTypeOptional	BOOLEAN OPTIONAL,	
	&ErrorPriority	Priority OPTIONAL,	
	&errorCode	Code UNIQUE OPTIONAL	
}			
WITH SYNTAX			
{			
	[PARAMETER	&ParameterType	[OPTIONAL ¶meterTypeOptional]]
	[PRIORITY	&ErrorPriority]	
	[CODE	&errorCode]	
}			

8.3.2 &ParameterType 字段规定差错的参数的数据类型。如果在某一差错中省略该字段,则该差错无参数值。

8.3.3 ¶meterTypeOptional 字段规定作为限定差错参数返回值的数据类型是否可任选地出现,只有当 &ParameterType 字段出现时,该字段才能出现。如果该字段不存在或取值 FALSE,在 ReturnError{}PDU 中不能省略 &ParameterType 的值(见 9.5)。

8.3.4 &ErrorPriority 字段规定能返回差错的允许优先级(见 8.9)。

8.3.5 &errorCode 字段(如果出现)规定用于标识该差错的代码值(见 8.8),例如,当返回差错时。

注:使用后面定义的 ReturnError{}PDU 不能返回没有 &errorCode 的差错。实际上,除打算在某一特定情况(例如,联结差错)中使用外,所有的差错都应具有分配的 &errorCode 字段。

8.4 操作包

8.4.1 操作包是一对通信客体所起作用的规范,这是从他们能相互调用的观点来说的。若该包是不对称的,则术语“消费者”和“供应者”可用于所涉及的两个客体。规定信息客体类 OPERATIONPACKAGE(所有这样的包均属于它)如下,各个字段在 8.4.2~8.4.7 中描述。

```
OPERATION-PACKAGE ::= CLASS
```

```
{
    &Both                OPERATION OPTIONAL,
    &Consumer             OPERATION OPTIONAL,
    &Supplier             OPERATION OPTIONAL,
    &id                   OBJECT IDENTIFIER UNIQUE OPTIONAL
}
```

```
WITH SYNTAX
```

```
{
    [OPERATIONS          &Both]
    [CONSUMER INVOKES    &Supplier]
    [SUPPLIER INVOKES    &Consumer]
    [ID                   &id]
}
```

8.4.2 &Both 字段规定了两个客体均能执行的操作的集。可以省略该字段。

8.4.3 &Consumer 字段规定了两客体中的某一客体能够执行的操作的集,该客体称为消费者。可以省略该字段,在这种情况下,消费者只能执行由 &Both 字段规定的操作。

8.4.4 &Supplier 字段规定了两客体中的某一客体能够执行的操作集,该客体称为供应者。可以省略该字段,在这种情况下,供应者只能执行由 &Both 字段规定的操作。

注:提供的参数化的操作包 switch{}(见 10.18),通过转换另一客体的作用来获得一个操作包。

8.4.5 &id 字段(如果出现)规定用于标识该操作包的客体标识符(OBJECT IDENTIFIER)值。例如,如果该包是要被通告或协商的。

注:没有 &id 的包不能被通告或协商。

8.4.6 由 &Both、&Consumer 和 &Supplier 字段包括的(直接或间接)所有操作应具有不同的 &operationCode 值。

8.4.7 在被包含的所有操作(见 8.4.6)的 &Error 字段中包括所有差错应具有不同的 &errorCode 值。

8.5 连接包

8.5.1 连接包是在动态地建立和释放联系时所涉及的操作和 QOS 的规范。仅当联结和断联操作用来分别动态地建立和释放联系时,才应规定连接包。规定信息客体类 CONNECTIONPACKAGE(所有的连接包均属于它)如下,各种字段在 8.5.2~8.5.6 中描述。

CONNECTION-PACKAGE::=CLASS

```
{
    &bind                OPERATION DEFAULT emptyBind,
    &unbind              OPERATION DEFAULT emptyUnbind,
    &responderCanUnbind  BOOLEAN DEFAULT FALSE,
    &unbindCanFail       BOOLEAN DEFAULT FALSE,
    &id                  OBJECT IDENTIFIER UNIQUE OPTIONAL
}
```

WITH SYNTAX

```
{
    [BIND                &bind]
    [UNBIND              &unbind]
    [RESPONDER UNBIND    &responderCanUnbind]
    [FAILURE TO UNBIND   &unbindCanFail]
    [ID                  id]
}
```

8.5.2 &bind 字段规定了作为建立联系的一部分而被执行的操作(OPERATION)。这样的操作必须具有其置为 TRUE 的 &returnResult 和 &alwaysReturns 字段,以及出现在 &Errors 字段中的单一差错。如果成功地建立该联系,联结操作返回一个结果。如果建立联系不成功,联结操作报告其差错。如果没有明确包括 &bind 字段,连接包应包括 emptyBind 操作(见 10.2)。

8.5.3 &unbind 字段规定了作为释放联系的一部分而被执行的操作。这样的操作必须具有置为 TRUE 的 &returnResult 和 &alwaysReturns 字段,以及由 &Errors 字段定义的单一差错。如果成功释放联系,断联操作返回一个结果。如果释放联系不成功,断联操作不返回结果,而报告其差错。如果没有明确包括 &unbind 字段,连接包应包括 emptyUnbind 操作(见 0.3)。

8.5.4 &responderCanUnbind 字段表示联系响应者(也可以是发起者)是否能调用 &unbind 操作。

8.5.5 &unbindCanFail 字段表示在 &unbind 操作已指明差错后,联系是否有可能仍然存在。

8.5.6 &id 字段(如果出现)规定用于标识该连接包的 OBJECT IDENTIFIER 值,例如,如果该连接包是要被通告或协商的。

注:没有 &id 的连接包不能被通告或协商。

8.6 联系约定

8.6.1 联系约定是相互建立联系的一对通信客体的作用的规范。规定信息客体类 CONTRACT(所有这样的约定均属于它)如下,各个字段在 8.6.2~8.6.6 中描述:

CONTRACT::=CLASS

```
{
    &connection          CONNECTION-PACKAGE OPTIONAL,
    &OperationsOf         OPERATION-PACKAGE OPTIONAL,
    &InitiatorConsumerOf  OPERATION-PACKAGE OPTIONAL,
    &InitiatorSupplierOf  OPERATION-PACKAGE OPTIONAL,
    &id                  OBJECT IDENTIFIER UNIQUE OPTIONAL
}
```

WITH SYNTAX

```
{
    [CONNECTION          &connection]
    [OPERATIONS OF       &OperationsOf]
    [INITIATOR CONSUMER OF &InitiatorConsumerOf]
    [RESPONDER CONSUMER OF &InitiatorSupplierOf]
    [ID                  &id]
}
```

8.6.2 &connection 字段的出现表示由该联系约定控制的联系通过相应的联结和断联操作动态地建立

和释放,这些联结和断联操作为连接包的一部分来规定,该字段的内容规定了所涉及的操作包。

8.6.3 当联系存在时,并且联系是对称的或联系的发起者起着消费者和供应者的作用时,&OperationsOf 字段(如果出现)规定一个或多个适用的操作包。

8.6.4 当联系存在时,并且认为联系的发起者起消费者的作用时,&InitiatorConsumerOf 字段(如果出现)规定一个或多个适用的操作包。

注:提供了参数化的操作包 switch{}(见 10.2),它允许通过转换另一客体的作用产生一个操作包。

8.6.5 当联系存在,并且认为联系的发起者起供应者的作用时,&InitiatorSupplierOf 字段(如果出现)规定一个或多个适用的操作。

8.6.6 &id 字段(如果出现)规定用于标识该联系约定的 OBJECT IDENTIFIER 值,例如,如果该联系约定是要被通告或协商时所用的值。

注:无 &id 的联系约定不能被通告或协商。

8.7 ROS 客体类

8.7.1 ROS 客体类定义了一组 ROS 客体的能力,这些客体具有共同使用一组特殊的约定与其他 ROS 客体交互(作用)的能力。规定信息客体类 ROS-OBJECT-CLASS 如下,各种字段在 8.7.2~8.7.6 中描述:

ROS-OBJECT-CLASS ::= CLASS	
{	
&Is	ROS-OBJECT-CLASS OPTIONAL,
&Initiates	CONTRACT OPTIONAL,
&Responds	CONTRACT OPTIONAL,
&InitiatesAndResponds	CONTRACT OPTIONAL,
&id	OBJECT IDENTIFIER UNIQUE
}	
WITH SYNTAX	
[IS	&Is]
[BOTH	&InitiatesAndResponds]
[INITIATES	&Initiates]
[RESPONDS	&Responds]
ID	&id
}	

8.7.2 &Is 字段(如果出现)规定了一组 ROS 客体类,它是已定义类的超类。已定义类的 ROS 客体能够支持每一个被规定超类所有客体含有的全部约定,以及下面 8.7.3~8.7.5 定义的字段中直接出现的那些约定。

8.7.3 &InitiatesAndResponds 字段规定了一组 CONTRACT,对于这组 CONTRACT,该类的 ROS 客体能够作为发起者和响应者。可以省略该字段。

8.7.4 &Initiates 字段规定了一组 CONTRACT,对于这组 CONTRACT,该类的 ROS 客体能够作为发起者。可以省略该字段,在这种情况下,这些 ROS 客体只能够启动由 &InitiatesAndResponds 字段规定的联系约定。

8.7.5 &Responds 字段规定了一组 CONTRACT,对于这组 CONTRACT,该类的客体能够作为响应者。可以省略该字段,在这种情况下,这些客体只能够对由 &InitiatesAndResponds 字段规定的联系约定作出响应。

8.7.6 &id 字段规定了用于标识该客体类的 OBJECT IDENTIFIER 值,例如,如果该客体类是要被通告或协商的。

8.8 代码

Code(代码)类型对操作的 &operationCode 字段和差错的 &errorCode 字段提供了值。规定代码类型如下:

```
Code ::= CHOICE
```

```
{
    local      INTEGER,
    global     OBJECT IDENTIFIER
}
```

8.9 优先级

规定 Priority(优先级)类型如下:

```
Priority ::= INTEGER(0..MAX)
```

为了在两个客体间进行交换,该参数定义了相对于其他调用(及其返回),指派给相应的调用(操作的)或其返回的传送的优先级。

该值越低,优先级越高。

9 类属 ROS 协议

9.1 引言

本章提供了一批用来规定实现 ROS 概念的协议的定义。主要定义有:

- a) 关于操作的调用和返回的参数化的 PDU 集(ROS{});
- b) 关于联结操作的调用和返回的参数化的 PDU(Bind{});
- c) 关于断联操作的调用和返回的参数化的 PDU(Unbind{}).

另外,还有一些上述定义所依赖的辅助定义。

9.2 ROS

9.2.1 参数化的类型 ROS{} 为包含 PDU 的抽象语法的定义提供了一个基础,这些 PDU 包括操作调用的 PDU、结果和差错返回的 PDU 以及拒绝无效的 PDU。规定 ROS{} 如下:

```
ROS{InvokeId;InvokeIdSet;OPERATION;Invokable;OPERATION;Returnable} ::= CHOICE
```

```
{
    invoke      [1]      Invoke{{InvokeIdSet},{Invokable}},
    returnResult [2]      ReturnResult{{Returnable}},
    returnError  [3]      Return{{Errors{{Returnable}}}},
    reject       [4]      Reject
}
```

(CONSTRAINED BY{--必须与上面的定义一致--})

! RejectProblem;general-unrecognizedPDU)

注:在实际实现中,使用不能自标识和自限定的编码规则时,不允许内部级别和外部级别限制违反有差别。

9.2.2 为了产生充分定义的类型,必须提供的 ASN.1 参数如下:

a) InvokeIdSet——类型 InvokeId 的值集定义了可用以标识调用以及与以后报告相关的值。如果在某一实现中可用另外的方法实现这些相关功能,就可以设置值集 NoInvokeId(见 9.9)。

注

1 应用设计者可以选择该参数为非限制的 INTEGER,或提供确切的范围,或将该参数扩展为某一抽象语法参数(见 10.14、10.15 和 10.16)。

2 对于 Invoke、ReturnResult 和 ReturnErrorPDU,OSI 实现并不支持值 noInvokeId(见 9.9.1)。