

国际电信联盟

**ITU-T**

国际电信联盟  
电信标准化部门

**M.3020**

(07/2011)

M系列：电信管理，包括TMN和网络维护  
电信管理网

---

**管理接口规范方法**

ITU-T M.3020 建议书

ITU-T



ITU-T M系列建议书  
电信管理，包括 TMN 和网络维护

引言与维护和维护组织的一般原则	M.10-M.299
国际传输系统	M.300-M.559
国际电话电路	M.560-M.759
公共信道信令系统	M.760-M.799
国际电报系统和相片传真传输	M.800-M.899
国际租用一次群和超群链路	M.900-M.999
国际租用电路	M.1000-M.1099
移动通信系统和业务	M.1100-M.1199
国际公众电话网	M.1200-M.1299
国际数据传输系统	M.1300-M.1399
标志和信息交换	M.1400-M.1999
国际传送网	M.2000-M.2999
<b>电信管理网</b>	<b>M.3000-M.3599</b>
综合业务数字网	M.3600-M.3999
公共信道信令系统	M.4000-M.4999

欲了解更详细信息，请查阅ITU-T建议书目录。

### 摘要

ITU-T M.3020建议书描述了管理接口规范方法（MISM），描述了以用户要求、分析和设计（RAD）为基础得到接口规范的过程，给出了RAD使用统一建模语言（UML）标记的的准则；然而，其它的接口规范技术并未排除在外。本ITU-T建议书在较高层次描述了使用UML的准则。

### 历史沿革

版本	建议书	批准日期	研究组
1.0	ITU-T M.3020	1992-10-05	
2.0	ITU-T M.3020	1995-07-27	4
3.0	ITU-T M.3020	2000-02-04	4
4.0	ITU-T M.3020	2007-07-22	4
5.0	ITU-T M.3020	2008-07-29	4
6.0	ITU-T M.3020	2009-05-14	2
7.0	ITU-T M.3020	2010-09-06	2
8.0	ITU-T M.3020	2011-07-14	2

## 前言

国际电信联盟（ITU）是从事电信领域工作的联合国专门机构。ITU-T（国际电信联盟电信标准化部门）是国际电信联盟的常设机构，负责研究技术、操作和资费问题，并且为在世界范围内实现电信标准化，发表有关上述研究项目的建议书。

每四年一届的世界电信标准化全会（WTSA）确定 ITU-T 各研究组的研究课题，再由各研究组制定有关这些课题的建议书。

WTSA 第 1 号决议规定了批准建议书须遵循的程序。

属 ITU-T 研究范围的某些信息技术领域的必要标准，是与国际标准化组织（ISO）和国际电工技术委员会（IEC）合作制定的。

## 注

本建议书为简明扼要起见而使用的“主管部门”一词，既指电信主管部门，又指经认可的运营机构。

遵守本建议书的规定是以自愿为基础的，但建议书可能包含某些强制性条款（以确保例如互操作性或适用性等），只有满足所有强制性条款的规定，才能达到遵守建议书的目的。“应该”或“必须”等其它一些强制性用语及其否定形式被用于表达特定要求。使用此类用语不表示要求任何一方遵守本建议书。

## 知识产权

国际电联提请注意：本建议书的应用或实施可能涉及使用已申报的知识产权。国际电联对无论是其成员还是建议书制定程序之外的其它机构提出的有关已申报的知识产权的证据、有效性或适用性不表示意见。

至本建议书批准之日止，国际电联尚未收到实施本建议书可能需要的受专利保护的知识产权的通知。但需要提醒实施者注意的是，这可能并非最新信息，因此特大力提倡他们通过下列网址查询电信标准化局（TSB）的专利数据库：<http://www.itu.int/ITU-T/ipr/>。

© 国际电联 2014

版权所有。未经国际电联事先书面许可，不得以任何手段复制本出版物的任何部分。

## 目录

页码

1	范围 .....	1
2	参考文献 .....	1
3	定义 .....	2
3.1	在其他地方定义的术语 .....	2
3.2	本建议书定义的术语 .....	2
4	缩写 .....	3
5	惯例 .....	4
6	对于方法和标记支持的要求 .....	4
7	方法 .....	5
7.1	总则 .....	5
7.2	方法的应用和构成 .....	5
7.3	详细的方法 .....	5
8	管理接口规范 .....	8
9	过程中的可追溯性 .....	8
10	文档结构 .....	8
附件A	— 需求 .....	9
A.1	惯例 .....	9
A.2	需求模板 .....	12
A.3	简化的需求模板 .....	14
附件B	— 分析 .....	16
B.1	惯例 .....	17
B.2	分析模板 .....	19
B.3	IOC特性和继承 .....	28
附件C	— MISM UML 指令集 .....	30
C.1	引言 .....	30
C.2	基本模型元素 .....	30
C.3	构造型 .....	33
C.4	关联类 .....	40
C.5	抽象类 .....	40
C.6	<<InformationObjectClass>> 和<<SupportIOC>>的应用 .....	41
附件D	— 设计 .....	42
附件E	— 信息类型定义 — 类型目录 .....	43
E.1	基本类型 .....	43
E.2	枚举类型 .....	43
E.3	复数类型 .....	43
E.4	有用的类型 .....	43
E.5	关键字 .....	44

	页码
附件F — 关于IOC特性、继承和实体导入的准则.....	45
F.1    IOC特性 .....	45
F.2    继承.....	46
F.3    实体（接口、IOC 和属性）导入 .....	46
附录 I — 需求举例 .....	47
附录II — 分析举例.....	50
附录III — 与ITU-T Z.601建议书对比.....	58
附录IV — 有待进一步研究的问题 .....	59
IV.1    SOA .....	59
IV.2    UML .....	59
IV.3    可见性 .....	59
IV.4    类型定义 .....	59
附录V — 另外的UML使用例子 .....	60
V.1    代理类.....	60
附录VI — 关于需求编号的准则 .....	62
参考资料.....	63

## 管理接口规范方法

### 1 范围

本建议书描述了管理接口规范方法（MISM），描述了以用户要求、分析和设计（RAD）为基础得到机器-机器接口规范的过程，给出了RAD使用统一建模语言（UML）标记的准则；然而，其它的接口规范技术并未排除在外。本建议书描述了使用UML的准则，接口规范专注于[ITU-T M.3200]中定义的管理业务，和/或者支持[ITU-T M.3050.x]系列中定义的管理过程。这样一个规范可以支持部分的或者一项或多项管理业务，管理业务由管理功能组成，这些功能可能会引用专门适合于特定的被管理领域的、在[ITU-T M.3400]中定义的那些功能或[ITU-T M.3050.x]系列中定义的过程，或者认为合适的新功能。

本方法既适用于传统的管理站/代理样式的管理接口[ITU-T M.3010]，也适用于下一代网络管理体系结构所采用的面向服务的架构（SOA）原则[ITU-T M.3060]。

### 2 参考文献

下列ITU-T建议书和其他参考文献的条款，通过在本建议书中的引用而构成本建议书的条款。在出版时，所指出的版本是有效的。所有的建议书和其他参考文献都面临修订，使用本建议书的各方应探讨使用下列建议书和其他参考文献最新版本的可能性。当前有效的ITU-T建议书清单定期出版。本建议书中引用某个独立文件，并非确定该文件具备建议书的地位。

[ITU-T M.3010] ITU-T M.3010 建议书（2000年），电信管理网的原则。

[ITU-T M.3050.x] ITU-T M.3050.x建议书（2007年），增强的电信运营图（eTOM）。

[ITU-T M.3060] ITU-T M.3060建议书/Y.2401（2006年），下一代网络管理的原则。

[ITU-T M.3200] ITU-T M.3200建议书（1997年），TMN管理业务和电信管理域：概述。

[ITU-T M.3400] ITU-T M.3400建议书（2000年），TMN管理功能。

[ITU-T Q.812] ITU-T Q.812建议书（2004年），Q和X接口的上层协议概要。

[ITU-T X.680] ITU-T X.680建议书（2008年）| ISO/IEC 8824-1:2008，信息技术 — 抽象语法标记1（ASN.1）：基本标记规范。

[ITU-T X.681] ITU-T X.681建议书（2008年）| ISO/IEC 8824-2:2008，信息技术 — 抽象语法标记1（ASN.1）：信息对象规范。

- [ITU-T X.722] ITU-T X.722建议书（1992年）| ISO/IEC 10165-4:1992，信息技术 — 开放系统互连 — 管理信息的结构：被管理对象定义的准则。
- [ITU-T Z.100] ITU-T Z.100建议书（2007年），规格描述语言（SDL）。
- [OMG UML] OMG: *Unified Modelling Language Specification, Version 1.5*.  
非规范性参考文献清单见参考资料。

### 3 定义

#### 3.1 在其他地方定义的术语

本建议书使用了[ITU-T M.3010]的下列术语：

- 用户；
- 管理业务；
- 管理功能集。

本建议书使用了[OMG UML]的下列术语：

- 活动图；
- 参与者；
- 关联；
- 类；
- 类图；
- 类元；
- 协作图；
- 组合；
- modelElement；
- 顺序图；
- 状态图；
- stereotype；
- 用例。

本建议书使用了[ITU-T M.3060]的下列术语：

- 参考点。

#### 3.2 本建议书定义的术语

本建议书定义了下列术语：

**3.2.1 agent 代理：**封装管理功能的一个明确子集，通过管理接口与管理站进行交互，对于管理站来说，代理行为只有通过管理接口才可见。

注 — 认为相当于IRPAgent [b-3GPP TS 32.150]。

**3.2.2 information object class 信息对象类：**描述管理接口中传递/使用的信息，采用UML元模型中的构造型“类”建模，信息对象类的正式定义及其规范的结构详见附件B。



**3.2.3 information service 信息服务:** 描述与被管理的实体（网络资源或者支持对象）有关的信息，以及针对特定的功能域可能会采取的管理信息的方式。所有的 IRP 均定义了信息服务。

注 — 认为与[b-3GPP TS 32.150]中信息服务的定义相同。

**3.2.4 information type 信息类型:** 操作的输入参数类型的规范。

**3.2.5 integration reference point 集成参考点:** 为了定义管理接口的某一个方面，用一组规范描述的体系结构概念，包含需求规格说明、信息服务规范以及一个或多个解决方案集规范。

注 — 认为与[b-3GPP TS 32.150]中IRP的定义相同。

**3.2.6 management goals 管理目标:** 用户开展管理活动时的高级目标。

**3.2.7 management interface 管理接口:** 实现管理站和代理之间的管理能力，允许一个管理站使用多个代理，以及一个代理支持多个管理站。

注 — Q、C2B/B2B和Itf-N (3GPP)都是管理接口的例子。

**3.2.8 management role 管理角色:** 定义了预期是实施电信管理的操作人员或系统的活动，管理角色的定义与其它组件无关，即电信资源和管理功能。

**3.2.9 management scenario 管理情况:** 管理情况是来自管理业务的管理交互的一个例子。

**3.2.10 manager 管理站:** 模拟代理的一个用户，通过管理接口直接与代理进行交互。

由于管理站代表一个代理用户，因此它清晰地描述了代理应该做什么，对于代理来说，管理站的行为只有通过管理接口才可见。

注 — 认为相当于IRPManager[b-3GPP TS 32.150]。

**3.2.11 matching information 匹配信息:** 参数类型的技术要求（可能会引用IOC或者IOC的属性）。

**3.2.12 protocol-neutral specification 协议中立的规范:** 定义了支持管理能力的管理接口，与管理技术例如CORBA和XML等隐含的或者要求的协议和信息表示方法无关。

**3.2.13 protocol-specific specification 协议特有的规范:** 定义了适合于某一项具体选择的管理技术（例如，CORBA）、支持管理能力的管理接口。

注 — 认为相当于解决方案集[b-3GPP TS 32.150]。

**3.2.14 telecommunications resources 电信资源:** 电信资源是需要管理、使用管理业务的物理或逻辑实体。

## 4 缩写

本建议书使用了下列缩写：

3GPP	第三代合作计划
ADM	管理的（用法：需求类）
ASN.1	抽象语法标记1
CM	有条件的强制性

CO	有条件的可选
CON	概念上的（用法：需求类）
CORBA	公共对象请求代理体系结构
FUN	功能性的（用法：需求类）
GDMO	定义被管理对象的准则
IDL	接口定义语言
IOC	信息对象类
IRP	集成参考点
IS	信息服务
MISM	管理接口规范方法
NA	不适用的
NE	网元
NON	非功能性的（用法：需求类）
OMG	对象管理组
OO	面向对象的
OSI	开放系统互连
SDL	规格描述语言
SOA	面向服务的架构
SS	解决方案集
TS	技术规范
UML	统一建模语言
XML	可扩展标记语言

## 5 惯例

第A.1节包含适用于需求阶段的惯例。

第B.1节包含适用于分析阶段的惯例。

## 6 对于方法和标记支持的要求

在提出方法和选择标记的时候，适用下列要求：

- 1) 方法包括标记选择，应支持收集问题空间即电信管理的所有相关需求。
- 2) 方法为需求及其对应的分析|信息服务，以及它们对应的设计规范|解决方案集的产生提供方便。
- 3) 标记应有便于无歧义地产生目标管理协议子集中的规范，方法不会提出可能选择的协议服务（例如，CORBA安全服务）。  
注一 [ITU-T Q.812]规定了适合国际电联使用的管理协议。
- 4) 方法应容许在全部三个阶段中都具有强制项和可选项的规范，还规定了三个阶段之间的强制|可选项的关系。

- 5) 应该有可能由协议中立规范（分析IS）产生互操作语言专用定义，即设计IS（例如，从UML到IDL，从UML到GDMO/ASN.1）。

## 7 方法

### 7.1 总则

这个方法旨在提供关于得到机器—机器管理接口定义过程的描述。

### 7.2 方法的应用和构成

管理接口规范方法（MISM）规定了一个分为三个阶段的过程，其特点是允许跨越这三个阶段可追溯。这三个阶段运用了行业认可的、采用面向对象的分析和设计准则的技术，这三个阶段是需求、分析和设计，采用的技术应允许使用或者开发在市场上可以买到的支持工具。各个阶段可以根据问题的性质而采用不同的技术。

### 7.3 详细的方法

#### 7.3.1 概述

需求和分析阶段产生UML规范，设计阶段使用网络管理范例专用的标记，这三个阶段的输出是：

- 需求阶段 — 需求。
- 分析阶段 — 与实现无关的规范。
- 设计阶段 — 技术特有的规范。

起初，设计阶段会采用手工或者定制的方法进行开发，当能够采用工具产生互操作协议专用的定义时，就能够在设计阶段使用UML标记。

以下各节描述了这三个阶段。

#### 7.3.2 需求

关于要解决的问题的需求可分成两大类，第一类需求在这里被称作商务需求，课题方面的主要专家应能够判定需求充分地代表了要解决的管理问题的需要，第二类被称作规范需求，这些需求应提供足够的细节，使得能够制定分析阶段和设计阶段的接口定义。由于最终的接口定义必须要追溯到需求，因此这三个阶段之间进行交互可能是必需的，需求中的任何不确定必须要通过这种交互予以解决，从而确保能够形成一个可执行的规范。

人机接口数据可以在第二类需求中规定，这些需求可能会对后续阶段中设计的概念和数据产生很大的影响，更多细节参见附录三，以及关于人机接口数据设计的ITU-T M.1400系列建议书。

可以采用不同的技术来确定这两类需求，无论采用什么技术，需求的可读性是至关重要的。只要可能具有可读性和可追溯性，就不要要求需求自身采用机器可读的标记。为了描述可供追溯的不同需求，建议采取枚举需求的方法。

需求阶段包括确定例如安全策略、依据应用的问题域的范围、资源以及由资源假定的角色等方面，需求规定了角色、责任以及构成问题空间的实体之间的关系，可以采用不同的技术包括文本表示方法来规定商务级需求，为了方便在设计和实现阶段追溯这些需求，建议对需求进行枚举。

问题必须限定在一个特定的范围之内，确定范围的方式之一是使用[ITU-T M.3200]中确定的管理业务和[ITU-T M.3400]中确定的功能集，采用被管理的资源和管理功能来确定需求，[ITU-T M.3050.x]“增强的电信运营图（eTOM）”描述了管理业务方法的一个备选方法，该建议书提供了一个基于商务过程的方法。

[ITU-T M.3200]和[ITU-T M.3050]方法之间关系的描述见[ITU-T M.3050.x]。

在提出特定商务需求的应用内部，必须对管理功能进行分组并予以支持，因而，eTOM过程之间的链接、[ITU-T M.3200]管理业务、[ITU-T M.3400]管理功能集和管理功能对于促使这种分组清晰、有效十分重要，为了满足问题的商务需求，可能需要补充[ITU-T M.3400]。

UML用例和情况应该用于收集商务需求时与主要专家的交互，需求还应标识商务过程可以见到的失败条件。

注一 不要求将每条需求都表示为一个用例。

产生的需求必须是完整和详细的，方法的递归性可用于取得这种完整性，需求的完整性（清晰且证明文件齐全）会推动分析和设计阶段。

关于需求构成和标识的准则以及模板在第A.1.2节描述。

用例是通过一系列步骤而达到的目标，每个步骤可以看作该用例的一个子目标，因此，每个步骤可以代表另一个用例（下级的用例）或者一个位于案例分解最底层的自主活动。

关于用例的准则和模板在第A.1.2节描述。

从附录I 可以获得需求定义的一个例子。

### 7.3.3 分析

在分析阶段，需求用于确定交互的实体、它们的特性以及相互之间的关系，这使得可以定义由实体提供的接口，在UML标记中，这些实体变成为类，类的描述连同接触到的接口应能追溯到需求，分析规范中定义的类和设计规范中类之间的关系未必是一一对应。

这个阶段应考虑人机接口数据的需要（即信息模型必须包含足够的信息，使得能够基于分析结果开展设计）。

本建议书提供了关于使用UML标记的高级指导，以便支持管理接口规范；然而，SDL [ITU-T Z.100]可能会用于扩充UML定义。

分析阶段应与设计约束无关，例如，可以采用OO原则记录分析，即使设计可能会采用非面向对象的技术，分析阶段中确定的信息包括类描述、数据定义、类关系、交互图（顺序图和/或者协作图）、状态转换图和活动图，类定义包括作为注释或者文本描写记录的操作规范、通知、属性和行为。

为了支持管理接口调整，在分析阶段应重用协议中立的公共管理业务（如果存在的话）-或其它现有的业务。

关于用例的准则和模板的描述见附件A。

分析模板将信息类型当作一个特性用于描述IOC属性和操作/通知参数。能够使用的有效信息类型及其语义的定义见附件E。

## 7.3.4 设计

### 7.3.4.1 概述

在设计阶段，会产生可实现的互操作接口规范，这将包括选择目标规范语言，设计阶段的规范取决于特定的管理范式（例如，用于CORBA接口的IDL）。

这个阶段要区分三种数据规范：与管理范式（例如，XML）相关的、经多个接口传送数据的设计（例如，故障和性能），经各个单独的接口传送的消息（例如，告警报告）以及与某一个特殊范式一致的数据编码方法（例如，压缩的XML）。

特定管理范式的选择可参见其它的ITU-T建议书，下面各节提供了一个概述。

在设计阶段，建议参考需求和分析阶段的UML描述，来扩充行为规范，例如，GDMO的行为定义能够参考分析阶段的状态图、顺序图和类定义，如果需要的话，可以包含另外的描述实体之间交互的UML图，相当于特定的协议范式。

由于管理采用了另外的范式，因此这些范式定义的标记/语言将被使用。

### 7.3.4.2 CORBA

在基于CORBA管理的语境下，采用IDL定义信息模型。

### 7.3.4.3 GDMO

在基于OSI系统管理的范式的语境下[ITU-T X.722]，设计规范为采用GDMO模板的信息模型规范，该模板适用于被管理对象类、属性、行为、通知、活动、类的命名实例和错误/异常规范。信息的语法采用ASN.1标记[ITU-T X.680]规定。

在GDMO中，对象类层次规定了管理所需要的对象类的特性，为了最多地从规范重用中受益，需要广泛地使用继承（超类和子类）。采用来自[ITU-T X.722]的模板规定对象类，定义信息模型的模板应采用ASN.1对象标识符的值进行登记（根据[ITU-T X.722]的规则），对于那些在其它ITU-T建议书和ISO标准中已经规定的对象类，只需要引用特定的建议书和对象类。命名既不是对象类层次的一部分，也不是对象类层次的目标。

#### 7.3.4.4 XML

有待进一步研究。

### 8 管理接口规范

管理接口规范包含第7章中讨论的需求、分析和设计规范，附件A、B和C提供了用于确定这些规范的结构。

当设计一个系统时，这些技术和支持标记也适用于管理接口规范，即使系统设计并未作为ITU-T管理建议书的一部分。它们有助于描述在一个系统例如NE的内部，如何使用接口规范管理资源。

### 9 MISM过程中的可追溯性

为了获得需求、分析和设计之间的可追溯性，有必要指定合适的标识，可以通过每个阶段内以及各个阶段间确定的实体之间的引用来实现对可追溯性的支持，可追溯性是指从设计|解决方案集到分析|信息服务，以及从分析|信息服务到需求。可追溯性更适合于需求规范的产物之间，以及分析|信息服务的产物之间，例如用例和文本需求之间，应按照第7.3.2节中的描述来确定需求，分析阶段的输出为各种不同的用例确定了更为详细的信息需求，设计阶段应指向分析阶段输出中的各种图表和文本，指针可能是引用适当的小节。

从设计阶段到主题级需求的可追溯通常是间接的，这是必须的，因为这个阶段的输出会被描述成不同的详细程度。

附件B描述了需求阶段和分析阶段之间可追溯性的准则。

推荐下列在其它文档规定的、适用于追溯需求以及其它的机制（可能并不遵循所提倡的标识模式）：

forum/body "::<" document ID "::<" id

其中，"id"可以是下列其中之一：

- 1) 需求ID；
- 2) 用例ID；
- 3) 需求题目/正文；
- 4) 用例题目；
- 5) 唯一标识某一个需求或用例的文档的小节。

例子：

3GPP::32.111-1::getAlarmList

ITU-T::M.3016::1.5.1.2

### 10 文档结构

虽然有三个阶段，但是接口文档可以将它们的输出合并成一个或多个文档，建议将需求和分析合并，为每个特定的网络管理协议范式开发单独的设计文档。

## 附件A

### 要求

(本附件是本建议书的组成部分)

#### A.1 惯例

##### A.1.1 UML标记用于需求

##### A.1.2 用例模板

##### A.1.3 需求类型

#### A.2 需求模板

##### 1 概念和背景

##### 2 商务级需求

###### 2.1 需求

###### 2.2 参与者角色

###### 2.3 电信资源

###### 2.4 高层用例

##### 3 规范级需求

###### 3.1 需求

###### 3.2 参与者角色

###### 3.3 电信资源

###### 3.4 用例

#### A.3 简化的需求模板

##### 1 概念和背景

##### 2 需求

以下是针对需求规格说明的准则，使用这个模板的例子见附录I。

标准的（或者完整格式）需求模板见第A.2节，此外，第A.3节中还定义了一个简化的需求模板。

#### **A.1 惯例**

##### **A.1.1 UML标记用于需求**

表A.1标识了管理概念和UML标记之间的对应关系，本建议书规定了不同阶段使用的高级概念和标记，构造型被用于扩充UML标记，本建议书包含了经批准可以在管理环境内部使用的构造型（见附件C）。

表A.1 — 需求概念

管理概念	UML 标记	备注
用户	参与者	用户模拟成一个参与者。
管理角色	参与者	参与者扮演一个角色，通常每个参与者最好只模拟一个角色。
管理功能	用例	管理功能用一个或多个用例来模拟。
管理功能集	用例	管理功能集是一个组合的用例，每项管理功能（可能的）用一个单独的用例模拟。
管理业务	用例	管理业务用一个高层用例来模拟。
管理情况	顺序图	相比协作图，更推荐采用顺序图。
电信资源类型	类	类图以适合于方法阶段的详细程度描述了电信资源类型的详细特性。
管理目标	-	由于没有适合的UML标记，所以用文本描写记录管理目标。

### A.1.2 用例模板

当提供用例时，应遵循下列惯例和模板。

表A.2 — 用例模板

用例段	进展/规范	<<Uses>> 相关的使用
目标 <sup>(*)</sup>	这是用例力争达到的目标/最终结果，应该是对用例在成功情况下达到状况的简要描述。 可能会有相对于其它用例的优先级以及所要求的用例性能的描述，例如： <ul style="list-style-type: none"> <li>• 实时。</li> <li>• 接近实时。</li> <li>• 非实时。</li> </ul>	
参与者和角色 <sup>(*)</sup>	用例涉及的参与者/角色名称，包括每个参与者的角色特征。	
电信资源	用例涉及的电信资源名称。	
假定	提供用例语境的环境描述。 假定与前置条件互斥。 假定与静态特性有关。	
前置条件	在能够触发用例之前必须是真实的所有系统和环境条件的列表。 前置条件与假定互斥。 前置条件与动态特性有关，并且能够产生异常，而假定则不会。	



表A.2 — 用例模板

用例段	进展/规范	<<Uses>> 相关的使用
什么时候开始	触发用例开始的单一事件的名称。 可选地并且通常不会用于规定触发器，例如“当管理站必须检索信息时”。	
第1步 <sup>(*)</sup> (M O)	一个用例描述了实现用例的目标而必需的一系列步骤（手工的和自动的）。 各个步骤可以调用其它的用例。 要对各个步骤进行编号以便可追溯。 每个步骤要标识为强制的（M）或可选的（O）。 子步骤是相对包含多步的步骤而标识的，例如： 第n步 第n.1步 第n.2步 这里，n.1和n.2是第n步的子步聚。	可参考一个使用过的用例。
第n步 (M O)	由于需要而增加的、具有逻辑次序的步骤。	
什么时候结束 <sup>(*)</sup>	表示用例完成的事件列表。 注-在这个语境下，“事件”应被看作是最广义的，不限于通过管理接口交换的通知，例如，过程完成能够被看作是表示用例完成的一个事件。	
异常	用例在其运行期间检测到的异常条件和故障的汇总列表。	
后置条件	当用例已经结束时必须是真实的所有系统和环境条件的列表，后置条件的描述决定着预期用例将会完全地成功、部分地成功或者甚至已经失败以便完成该用例。	
可追溯性 <sup>(*)</sup>	该用例揭示的需求或用例。	
注 — 标记了“*”的字段对于所有用例规范都是强制性的，其它的字段只有与特定的用例有关时才是强制性的。		

### A.1.3 需求类型

将需求分成不同的类型是有用的，下列类型被认为与MISM有关：

- 概念上的（CON）— 确定一个概念、数据类型、关系、格式或者结构。
- 功能性的（FUN）— 确定功能能力、动态情况、序列、定时参数或者交互。
- 非功能性的（NON）— 非功能性需求，包括异常条件、错误条件和性能界限。
- 管理的（ADM）— 与用例正常运行不相关的系统管理和运行需求。

需求书写应根据以下模板：

REQ-Label-Category-Number {Category, number} Details {Source Citation}

其中，“Label”是建议书（或其中一部分）的缩写，对标签集不作限定，并且不要求标准化。

关于需求编号的准则见附录 VI。

## A.2 需求模板

### 1 概念和背景

定义这个规范的主要目标、结果以及管理接口（和参考点），将[ITU-T M.3200]分类用作确定这个接口所支持管理业务的出处。

本小节应提供一个清晰的有关用户从中受益的描述，即开展这项管理业务的原因，必要时应增加背景和语境，但解释性的和描述性的部分应是单独的，需要时，应将支持的背景信息放入附录。

#### 1.a SubClauseTitle

SubClauseTitle是小节的名称。

“a”代表一个数字，从1开始，每个新的小节加1。

小节的使用是可选的。

### 2 商务级需求

#### 2.1 需求

##### 2.1.a SubSetTitle

SubSetTitle是商务级需求子集的名称。

“a”代表一个数字，从1开始，每个新子集加1。

子集的使用是可选的，所有商务级需求都在第2.1小节（需求）中描述。

文本中列出了主要的需求，并且确定了用例以及参与者/角色和资源。高层用例（下面的第2.4小节）应产生商务级需求，并且由于没有细化到更低的层次而有别于规范需求，第2.4节包含了高层用例组成的许多例子，与策略相关的信息（例如，安全性，持续性）是这个层级很可能纳入的内容，对需求进行编号是可追溯性的需要。

应按第A.1.3节中的描述确定需求，在需求规格说明内部，建议应将需求写在第A.1.3节的结果中（对于整个规格说明或者对于每个子集）。

使用需求类型是可选的，当使用需求类型时，就能使用类型的子集。

例如，标记为‘SM’的建议书中第23号概念上的需求确定如下：

标识符	定义
REQ-SM-CON-23	业务情况包括名称、地址、电话号码、业务描述和可选的用于联系的传真号码{T1M1.5 246号文档11/96}

必要时，可以使用一个或多个表格以及表格之间的支持性文本。

#### 2.2 参与者角色

这里包含了参与者的文本描述（见第3节）。

## 2.3 电信资源

这里呈现的是支持用例所需的相关资源（见第3节）的文本描述。

## 2.4 高层用例

将会提供高层用例图，为了让主要专家能够理解用例，应为他们补充每个用例的文字描述，文字描述有两个用途：收集领域专家的知识，相对于需求来验证分析和设计阶段的模型，附录一给出了高层用例图的一个例子。

### 2.4.a UseCaseName

UseCaseName是用例的名称。

“a”代表一个数字，从1开始，每个新定义的用例加1。

对于每个为接口规范需求而定义的高层用例，重复本小节。

高层用例可能会确定[ITU-T M.3400]中定义的各种功能集，或者[ITU-T M.3050.x]中定义的管理过程，按照下面规范级需求小节中的描述，通过使用构造型例如“包含”和“扩充”，可以进一步细化这些用例。

适当时，可以使用顺序图，然而，不期望在高层需求中使用这些图，当这个层级的用例在下一级需求被进一步分解时，这些图可能会更加合适。

从这个层级需求到下一级需求的可追溯性，可以通过各个功能集是如何使用新用例来进一步细化的来确认。

使用表A.2中定义的模板的一组用例表格，可能会用于表示在适合所分析问题的抽象层研究的重要能力。

用例提供的详细程度和覆盖范围取决于编写团队对专题的熟悉度，因此是主观的，较低の詳細程度最有可能是分析的分析，而不是需求收集。

允许继续对更高抽象层用例的每个步骤进行更加详细地分析，这可以通过引用为此目的而保留的表格中的更加详细的用例来实现，需要强调的是这不是必须要做的，并且是主观的，取决于作者/编写组的需要。

提供下面列表以帮助最始鉴别合适的用例：

- 该系统的主要目的是什么？
- 什么类型的人/系统需要与该系统交互？
- 这些人/系统如何分组或者抽象成角色的？
- 该系统的启动、正常运行、故障和恢复状况是怎样的？
- 可能需要来自该系统的什么类型的报告或数据？
- 哪些特定的活动是必需的（例如，基于时间和网络负载）？

以常用的方式记录用例是有用的，建议采用以下结构：

- <用例表>（见表A.2）
- <可选的顺序图>
- <可选的状态图>

## 3 规范级需求

### 3.1 需求

这里，使用来自[ITU-T M.3400]的管理功能，对商务级需求作了进一步细化，由于[ITU-T M.3400]不够详尽，没有提出适用于所有被管理领域的所有管理业务，因此可能会需要新的功能，新的功能应包含在需求中，如下所述。

#### 3.1.a SubSetTitle

SubSetTitle表示规范级需求的一个子集的名称。

“a”代表一个数字，从1开始，每个新的子集加1。

子集的使用是可选的，所有规范级需求均在第3.1小节（需求）中描述。

文本中列出了主要的详细的和具体的需求，并且确定了用例以及参与者/角色和资源，第3.4小节中的用例应产生较低详细程度的规范级需求，并且相对于商业级用例需求，该用例应更加地面向实现。对需求进行编号是可追溯性的需要。

应按照第A.1.3中的描述确定需求，在需求规格说明内部，建议应将需求写在第A.1.3节结果中（对于整个规格说明或者对于每个子集）。

使用需求类型是可选的，当使用需求类型时，就能够使用类型的子集。

例如，标记为‘OM’的建议书中第33号功能性需求确定如下：

标识符	定义
REQ-OM-FUN-33	挂起操作能够被发起者取消。

必要时，可以使用一个或多个表格以及表格之间的支持性文本。

规范级需求应遵循第A.1节中定义的惯例和模板。

### 3.2 参与者角色

这里包含了所有参与者的一个列表和商务级需求中尚未定义的参与者的文本描述。

### 3.3 电信资源

这里呈现的是所有无源资源的一个列表和商务级需求中尚未定义的资源文本描述。

### 3.4 用例

这里，使用多个规范级用例对高层用例作了进一步的细化，对于其中的每一个用例，下面的小节将作进一步地说明，如下所述。

#### 3.4.a UseCaseName

UseCaseName是用例的名称。

“a”代表一个数字，从1开始，每个新定义的用例加1。

适当时，可以使用顺序图和状态图。

注—适用于顺序图、状态图的准则和标准有待进一步研究。

用例规范应遵循第A.1节中定义的惯例和模板。

## A.3 简化的需求模板

简化的需求模板是在只需要文本需求的时候案例使用的备选模板，定义一个单独的模板是为了避免给第A.2节中描述的完整模板增加可选项将会产生的不确定性。

### 1 概念和背景

定义这个规范的主要目标、结果以及管理接口（和参考点），将[ITU-T M.3200]分类用作确定这个接口所支持管理业务的出处。

本节应提供一个清晰的有关用户从中受益的描述，即开展这项管理业务的原因，必要时应增加背景和语境，但解释性的和描述性的部分应是单独的，需要时，应将支持的背景信息放入附录。

#### 1.a SubClauseTitle

SubClauseTitle是小节的名称。

“a”代表一个数字，从1开始，每个新的小节加1。

使用小节是可选的。

## 2 需求

### 2.a SubSetTitle

SubSetTitle商务级需求的一个子集的名称。

“a”代表一个数字，从1开始，每个新子集加1。

使用子集是可选的，所有的商务级需求均在第2节（需求）中描述。

文本中列出了主要的需求，并且确定用例以及参与者/角色和资源。用例应产生高层需求，并且由于没有细化到更低层次而有别于规范需求，与策略相关的信息（例如，安全性，持续性）是这个层级候选纳入的内容，对需求进行编号是可追溯性的需要。

应按第A.1.3节中的描述确定需要，在需求规格说明内部，建议应将需求写在第A.1.3节的结果中（对于整个规格说明或者对于每个子集）。

使用需求类型是可选的，当使用需求类型时，就能够使用类型的子集。

例如，标记为'SM'的建议书中第23号概念上的需求确定如下：

标识符	定义
REQ-SM-CON-23	业务情况包括名称、地址、电话号码、业务描述和可选的用于联系的传真号码{T1M1.5 246号文档11/96}

必要时，可以使用一个或多个表格以及表格之间的支持性文本。

## 附件 B

### 分析

(本附件是本建议书的组成部分)

#### B.1 惯例

##### B.1.1 强制的、可选的和有条件的限定符

#### B.2 分析模板

##### 1 概念和背景

##### 2 信息对象类

##### 2.1 导入的信息实体和本地标签

##### 2.2 类图

##### 2.2.1 属性和关系

##### 2.2.2 继承

##### 2.3 信息对象类定义

##### 2.3.a *InformationObjectClassName*

##### 2.4 信息关系定义

##### 2.4.a *InformationRelationshipName (supportQualifier)*

##### 2.5 信息属性定义

##### 2.5.1 定义和合法值

##### 2.5.2 约束

##### 2.6 公共通知

##### 2.7 系统状态模型

##### 3 接口定义

##### 3.1 表示接口的类图

##### 3.2 一般规则

##### 3.b 接口 *InterfaceName (supportQualifier)*

##### 3.b.a 操作 *OperationName (supportQualifier)*

##### 3.b.b 通知 *NotificationName (supportQualifier)*

##### 3.c 情况

#### B.3 IOC特性、继承和导入

##### B.3.1 特性

##### B.3.2 继承

##### B.3.3 导入

以下是适用于分析阶段结果规范的准则。

分析模板是以3GPP信息服务[b-3GPP TS 32.151]为基础的，并作了扩充以满足对该方法的附加要求（例如，可追溯性）。

对于管理接口规范，应使用第B.2节所指出“分析”模板的第2.2和2.3小节。对于信息模型（例如，网络资源模型），应只使用第2.2小节。

分析模板将信息类型作为一个特性用来描述IOC属性和操作/通知参数，附件E定义了能够使用的有效信息类型及其语义。

使用这个模板的一个例子见附录二。

构造“分析|信息服务”和“设计|解决方案”集用于表示由ITU-T和3GPP提出的、内容相当但命名不同的规范。

## B.1 惯例

### B.1.1 强制的、可选的和有条件的限定符

本小节定义了多个用于限定分析|信息服务、设计|解决方案集之间的关系及其对接口实现影响的术语，本小节中定义的限定符只用于限定代理的行为，对于管理接口规范来说这就足够了。

M、O、CM、CO、C分析规范|IS规范定义了IOC属性、接口、操作、通知、操作参数和通知参数，它们可以具有以下支持/读/写限定符：M、O、CM、CO、C。

限定符M的定义（强制的）：

- 用于应支持的项。

限定符O的定义（可选的）：

- 用于可以支持也可以不支持的项。

限定符CM的定义（有条件的强制）：

- 用于明确地在一定条件下是强制性的项：
  - 具有支持限定符CM的所有项应具有建议书|IS规范中定义的相应约束，如果指定的约束得到满足，则支持该项。

限定符CO的定义（有条件的可选）：

- 用于明确地在一定条件下是可选的项：
  - 具有支持限定符CO的所有项应具有建议书|IS规格中定义的相应约束，如果满足了指定的约束得到满足，则支持该项。

限定符C的定义（SS-有条件的）：

- 用于只适合于某些但不是全部的设计|解决方案集（SS）的项。

设计|SS规范定义了IOC属性、操作、通知、操作参数和通知参数的SS-等同物。这些SS-等同物可以具有以下支持/读/写限定符：M、O、CM和CO。

分析|IS定义的构造的限定符到相应的SS-构造的限定符之间的映射定义如下：

- 对于限定符M、O、CM和CO，各个IS定义的项（操作和通知、操作的输入和输出参数、通知的输入参数、信息关系和信息属性）应被映射到所有SS中的等同物，映射的等同物应具有和IS定义的限定符相同的限定符。
- 对于限定符C，各个IS定义项应被映射到至少一个SS中的等同物，映射的等同物能够具有支持限定符M或O。

表B.1从来自代理的支持的角度，定义了等同物限定符的语义。

表B.1 — 用于设计|解决方案集的限定符的语义

映射的SS等同物	强制的	可选的	有条件的强制 (CM)	有条件的可选 (CO)
映射的通知等同物	代理应产生此通知。	代理可以产生也可以不产生它。	如果该项的约束得到满足，代理应产生这个通知。	代理可以选择是否产生它，如果代理选择产生它，则必须满足这个通知的约束。
映射的操作等同物	代理应支持它。	代理可能支持也可能不支持这个操作，如果代理不支持这个操作，代理应以表明代理不支持这个操作的某一个理由拒绝操作调用。拒绝连理由均应返回至管理站。	如果该项的约束得到满足，代理应支持这个操作。	如果该项的约束得到满足，代理可能会支持这个操作。
映射的操作等同物的输入参数	代理应接受它并按照它的值工作。	代理可能支持也可能不支持这个输入参数。如果代理不支持这个输入参数，并且如果该参数传送含意（即它不会传送没有信息的语义），则代理应以某一个理由（不支持该参数的理由）拒绝调用。拒绝连同理由均应返回至管理站。	如果该项的约束得到满足，代理应接受它并按照它的值工作。	如果该项的约束得到满足，代理可能会接受并按照它的值工作。
映射的通知等同物的输入参数和映射的操作等同物的输出参数	代理应提供这个参数。	代理可能会提供这个参数。	如用该项的约束得到满足，代理应提供这个参数。	如果该项的约束得到满足，代理可能会提供这个参数。
映射的IOC属性等同物	代理应支持它。	代理可能会支持它。	如果该项的约束得到满足，代理应支持这个属性。	如果该项的约束得到满足，代理可能会支持这个属性。



## B.2 分析模板

### 1 概念和背景

本节应提供管理接口规范分析的介绍。

#### 1.a SubClauseTitle

SubClauseTitle是小节的名称。

“a”代表一个数字，从1开始，每个新小节加1。

使用小节是可选的。

### 2 信息对象类

本小节应该用于所有的规范（管理接口规范和专门的信息模型规范）。

#### 2.1 导入的信息实体和本地标签

本小节确定了一个已经在其它规范中定义的并且已经导入当前文档的信息实体（例如，信息对象类、接口、信息关系、信息属性）的列表，所有导入的实体应该与当前规范本地定义的实体一样对待，导入的一个目的是用于继承，该列表中的每个元素都是一个（标签引用、本地标签）对，标签引用包括定义这个标签的规范的名称、信息实体的类型及其名称，于是导入的信息实体的本地标签就能够在整个规格说明中使用，代替标签引用。

此信息以表格的形式提供。

标签引用	本地标签

导入的元素应来自以这个方法作为基础的、协议中立的定义，但是必要时，为了随着时间迁移协议特有的规范，可能会导入许多来自其它规范的元素，。

关于实体导入以及IOC特性和继承的准则见附件F。

### 2.2 类图

#### 2.2.1 属性和关系

第一组图代表这个IS中定义的所有信息对象类以及它们所有的关系和属性，包括与导入的IOC的关系（如果有的话）。这些图应包含信息对象类基数（用于关联以及约束关系），也可能包含关联名和角色名，这些图应均为符合UML的类图（参见附件C）。

导入的信息对象类的特性（关系）不需要在图中重复，信息对象类应采用构造型<<InformationObjectClass>>来定义。

#### 2.2.2 继承

第二组图代表这个IS中定义的所有信息对象类的继承层次，不需要在这些图中包含完整的继承层次，但至少应包含当前文档中定义的所有信息对象类的父信息对象类，缺省地，一个信息对象类从信息对象类“顶端”继承下来，这些图应均为符合UML的类图。

导入的信息对象类的特性（属性、关系）不需要在这个图中重复，信息对象类应采用构造型<<InformationObjectClass>>来定义。

注1 — 能够在第2.2.1节中重复第2.2.2小节显示的一些继承关系，以提高可读性。

注2 — 接口继承在第3.1小节显示，不是在本小节。

## 2.3 信息对象类定义

各个信息对象类采用下列结构定义。

不应显示继承的项（属性等），因为它们由父IOC定义，因而对于所有的子类均有效。

### 2.3.a InformationObjectClassName

*InformationObjectClassName*是信息对象类的名称。

“a”代表一个数字，从1开始，每个新定义的IOC加1。

#### 2.3.a.1 定义

<定义>小节采用自然语言书写，<定义>小节涉及信息对象类自身，在这个定义中不会发现与对象特性类和其它对象类的关系有关的特性。读者必须查阅关系定义才能发现这类信息，在这里应确定了与继承有关的信息。

这里，也应定义关于回溯到一个或多个由这个IOC支持的需求的可追溯性的信息，采取下面的形式：

引用	需求标签	备注

#### 2.3.a.2 属性

<属性>小节提供了属性列表，这些属性是对象类的可管理的特性。每个元素均是一个元组（attributeName、supportQualifier、readQualifier、writeQualifier）：

- supportQualifier表示属性是否是强制的（M）、可选的（O）、有条件的强制（CM）、有条件的可选（CO）、SS-有条件的（C）或不支持的（-），允许的值是：强制的、可选的、有条件的或者不支持（分别是“M”、“O”、“C”或“-”）。
- readQualifier表示属性是否对于管理站可读。可能的值是：强制的（M）、可选的（O）、有条件的强制（CM）、有条件的可选（CO）、SS-有条件的（C）或不支持的（-），允许的值是：强制的（M）、可选的（O）和不支持（-）。
- writeQualifier表示属性是否对于管理站可写，对于“M”、“O”和“-”，writeQualifier的语义等同于supportQualifier，允许的值是：强制的（M）、可选的（O）和不支持（-）。

在supportQualifier、readQualifier和writeQualifier之间存在着依赖关系，supportQualifier表示对属性支持的需求，对于任何给定的属性，无论supportQualifier的值是多少，至少readQualifier或writeQualifier中的一个必须是“M”，supportQualifier为“O”的含意是该属性是可选的；然而，读和写限定符表示应该怎样支持可选的属性，是否应该支持可选的属性。

每个定义的私有或者代理内部属性对于IRPManager总是不可写，因此，它们的writeQualifier总是“-”。

所支持的公有属性的readQualifier和writeQualifier不可能都为“-”。

supportQualifier中“-”专用来记录对“原型”IOC（见第C.3.5小节）所定义属性的支持，supportQualifier为“-”的属性不是由实现“原型”所定义属性的一个子集的IOC实现的，在这种情况下，readQualifier和writeQualifier没有关联性。然而，一个不支持的属性既不可读也不可写，出于这一原因，对于不支持的属性，readQualifier和writeQualifier应为“-”。

对于任何使用一个或多个来自“原型”的属性的IOC，应使用一个单独的表格表示所支持的属性，如果不支持“原型”属性，则不需要这个表格。例如，如果一个特殊的IOC具有规定的属性（即，“原型”没有定义的属性）并且封装了来自两个“原型”的属性，则上述IOC的全部属性将被包含在三个单独的表格中。

此信息以表格的形式提供。

属性名	支持限定符	读限定符	写限定符	需求ID

### 2.3.a.3 属性约束

<属性约束>小节表示始终保持为真的属性之间的约束，那些特性在属性的生命周期内始终保持为真，特别是不需要在操作或通知的前置或后置条件中重复它们。

注 — 当不需要定义属性约束时，就不需要本小节。

### 2.3.a.4 关系

<关系>小节表示的是这个类所涉及到的关系的列表，各个元素是一个relationshipName。

关系将以表格的形式列出，如下所示：

关系	需求ID

每个关系名应引用（也可能是一个超级链接）第2节的适当小节（信息对象类）。

注 — 本小节是可选的，由于第2.2.1小节的类图描绘了所有的关系，因此可以取消本小节。

### 2.3.a.5 状态图

<状态图>小节包含状态图，信息对象类的状态图定义了这个信息对象类的允许状态以及那些状态之间的转换，将状态表示为各个属性值或者属性值的组合或者包含定义的信息对象类的关系，这应是一个符合UML的状态图。

注 — 当不需要定义状态图时，就不需要本小节。

### 2.3.a.6 通知

对于这个IOC，<通知>小节提供：

- a) 可选地引用第2.6小节定义的、对这个IOC有效的公共通知，并且
- b) 可选地，一个应该排除在适合这个IOC的公共通知列表（在第2.6小节中定义）之外的通知列表（注意到不能排除由父IOC继承下来的通知），  
以及
- c) 可选地，适用于这个IOC的通知列表，以及在第2.6小节公共通知中可能定义了也可能没有定义的通知。

本小节规定的通知是能够经管理接口发出的通知，其中，这些通知报头的“对象类”和“对象实例”参数（见注释2）标识着一个由封装本小节（即第2.3.a小节）定义的IOC的实例。

本小节规定的通知可能源自实现对象，该对象的标识符在实现时会被映射为经管理接口使用的对象实例标识符，因此，在本小节（即第2.3.a.6小节）中出现的通知并不意味或者标识着那些通知是源自一个由封装本小节（即第2.3.a小节）定义的IOC实例。

与上述选项c)有关的信息以表格的形式提供，这样表格的一个例子如下所示：

名称	限定符	需求ID	注释

注 1 — 可以缺少本小节和表格。

注 2 — 通知标题的定义见通知IRP信息服务[b-3GPP TS 32.302]。

注 3 — 通知表格中确定的通知的限定符，表示这样的通知是否能够传送该通知中的实例 DN，在管理规范中确定的通知限定符表示与主体通知发行有关的支持层级。

当且仅当满足以下条件时，管理站才能接收传送类-ABC实例的DN的通知-XYZ：

- 1) 类-ABC通知表定义了通知-XYZ，且
- 2) 类-ABC实例实现支持这个通知-XYZ，且
- 3) 管理接口定义了通知-XYZ，且
- 4) 管理接口实现支持这个通知-XYZ。

## 2.4 信息关系定义

本小节首先在下面表格中列出了这个建议书|规范所支持的所有关系，将支持限定符定义为第B.1节中的属性。

关系	支持限定符	需求ID

采用下列结构定义各个信息关系。

不应显示继承关系，因为它们由父IOC定义，因而对所有的子类均有效。

### 2.4.a InformationRelationshipName (supportQualifier)

InformationRelationshipName是信息关系的名称，其后跟有一个限定符（见第B.1节）。

“a”代表一个数字，从1开始，每个新定义的信息关系加1。

#### 2.4.a.1 定义

<定义>小节采用自然语言书写。

#### 2.4.a.2 角色

<角色>小节确定了对象类在关系中扮演的角色，每个元素是一个 (roleName、roleDefinition) 对。

此信息以表格形式提供。

名称	定义

#### 2.4.a.3 约束

<约束>小节包含了用于规定关系中必须保留的语义不变量的特性列表，每个元素是一个 (propertyName, propertyDefinition) 对，那些特性在关系的生命周期内始终保持为真，并且在操作或通知的前置或后置条件中不需要被重复。

此信息以表格的形式提供。

关系	支持限定符	需求ID

## 2.5 信息属性定义

采用下列结构定义各个信息属性。

不会显示继承的属性，因为它们是在父IOC中定义的，因而对于所有的子类均有效。

### 2.5.1 定义和合法值

对于每个定义的属性，本小节包含它的属性名、用自然语言书写的定义、信息类型（见附件E）和一个可选的这个属性所支持的合法值列表。

在合法值可以被枚举的情况下，每个元素都是一个（合法值名、合法值语义）对，除非一个合法值语义适用于多个数值，在这种情况下，只需要提供一次语义。当合法值不能被枚举时，合法值列表由单个定义规定。

此信息以表格的形式提供。

属性名	定义	信息类型/合法值

### 2.5.2 约束

<约束>小节指出是否存在影响属性的任何约束，每个约束由一个元组（propertyName、受影响的属性、propertyDefinition）定义，PropertyDefinitions用自然语言表示。

此信息以表格的形式提供。

名称	受影响的属性	定义

## 2.6 公共通知

这个<公共通知>小节提供了一个能够被这个管理接口规范定义的任何IOC引用的通知列表，在第2.3.a.6小节中这些通知只适用于引用该小节的IOC。

此信息以表格的形式提供。

名称	限定符	注释

注 — 当没有公共通知时，就不需要本小节。

## 2.7 系统状态模型

一些信息配置足够的特殊或者复杂，以至于应该使用状态图来阐明它们，本小节中的状态图定义了系统的允许状态以及那些状态之间的转换，将状态表示为属性值约束的组合或者包含一个或多个信息对象类的关系。

## 3 接口定义

本小节应该用于所有的管理接口规范，对于单独的信息模型规范则是可选的。

### 3.1 表示接口的类图

图中定义了各个接口，这应是一个符合UML的类图（参见附件C）。

接口采用构造型<<接口>>定义，每个接口包含强制性的一组操作或通知，或者可选的单一操作或单一通知，构造型（见附件C）用于规定可选的或者强制的接口，在类图中，应在每次操作和通知之前将接口中的每个操作和通知用附加符号“+”限定为“公有的”。

注—本小节能够显示接口继承。

### 3.2 一般规则

下列规则与所有的规范均有关，它们应作为规范的一部分被直接复制。

规则1：各个具有至少一个输入参数的操作支持前置条件valid\_input\_parameter，该前置条件表示所有输入参数相对于它们的信息类型应是有效的，此外，每个这样的操作支持异常operation\_failed\_invalid\_input\_parameter，该异常是在前置条件valid\_input\_paramete为假的时候出现的。异常具有相同的进入和退出状态。

规则2：各个具有至少一个可选输入参数的操作支持一组前置条件supported\_optional\_input\_parameter\_xxx，这里“xxx”是操作输入参数的名称，前置条件表示该操作支持指定的可选输入参数。此外，每个这样的操作支持异常operation\_failed\_unsupported\_optional\_input\_parameter\_xxx，该异常是在前置条件supported\_optional\_input\_parameter\_xxx为假并且指定的可选输入参数传送信息的时候出现，异常具有相同的进入和退出状态。

规则3：每个操作应支持一般的异常operation\_failed\_internal\_problem，该异常是在发生内部问题并且操作不能完成的时候才出现，异常具有相同的进入和退出状态。

注—安全性考虑事项以及由此得出的一般规则有待进一步研究。

#### 3.b 接口InterfaceName (supportQualifier)

InterfaceName是接口的名称，其后跟有一个限定符（见第B.1节）。

“b”代表一个数字，从3开始，每个新定义的接口加1。

各个接口由它的名称以及下面规定的一系列操作或通知定义。

采用下列结构定义各个操作。

注—接口内容和接口命名的分组操作/划分有待进一步研究。

##### 3.b.a 操作OperationName (supportQualifier)

OperationName是操作的名称，其后跟有一个限定符（见第B.1节）。

“a”代表一个数字，从1开始，每个新定义的操作加1。

##### 3.b.a.1 定义

<定义>小节用自然语言书写。

这里，还定义了关于返回到由这个操作所支持的一个或多个需求的可追溯性的信息，采取下面的形式：

引用	需求标签	备注

### 3.b.a.2 输入参数

操作的输入参数的列表。每个元素是一个元组（参数名称、支持限定符、信息类型（见附件E和第E.2节中的注释）和可选的这个参数所支持的合法值列表、备注），第B.1节中规定了适用于支持限定符的合法值。

此信息以表格的形式提供。

参数名	支持限定符	匹配的信息类型/合法值	备注

注 — 信息类型限定参数名称的参数，在合法值能够被枚举的情况下，每个元素是一个（合法值名、合法值语义）对，除非一个合法值语义适用于多个数值，在这种情况下，只提供一次定义。当合法值不能被枚举时，合法值列表由单一定义规定。

### 3.b.a.3 输出参数

操作的输出参数的列表。每个元素是一个（参数名称、支持限定符、匹配的信息/信息类型（见附件E和第E.2中的注释）和可选的这个参数所支持的合法值列表、备注），第B.1节中规定了适用于支持限定符的合法值。

此信息以表格的形式提供。

参数名	支持限定符	匹配的信息/信息类型/合法值	备注

注 — 信息类型限定了参数名称的参数，在合法值能够被枚举的情况下，每个元素是一个（合法值名称、合法值语义）对，除非一个合法值语义适用于多个数值，在这种情况下，只提供一次定义。当合法值不能被枚举时，合法值列表由单一定义规定。

这个表格还应包含一个用来指示操作完成状态的（成功、部分成功、失败原因等）的特殊参数‘状态’。

### 3.b.a.4 前置条件

前置条件是一个由AND、OR和NOT逻辑运算符结合成的断言集，在调用操作之前必须保持前置条件为真。

各个断言的定义是一个（propertyName、propertyDefinition）对，构成前置条件的所有断言均以表格的形式提供。

断言名	定义

### 3.b.a.5 后置条件

后置条件是一个由AND、OR和NOT逻辑运算符结合而成的断言集，在操作完成之后必须保持后置条件为真。当有关信息实体的后置条件什么都没说时，则假设是这个信息实体与前置条件中的描述对比没有发生变化。

各个断言的定义是一个 (propertyName、propertyDefinition) 对，构成后置条件的所有断言均以表格的形式提供。

断言名	定义

### 3.b.a.6 异常

操作可能产生的异常列表，每个元素是一个元组 (exceptionName、条件、ReturnedInformation、exitState)。

#### 3.b.a.6.c exceptionName

ExceptionName是一个异常的名称。

“c”代表一个数字，从1开始，每个新定义的异常加1。

此信息以表格的形式提供。

异常名	定义	
	条件	
	返回信息	
	退出状态	
	条件	
	返回信息	
	退出状态	

### 3.b.a.7 约束

<约束>小节提出针对操作或其参数的约束。

注 — 当不需要定义约束时，就不需要本小节。

### 3.b.b 通知NotificationName (supportQualifier)

NotificationName是通知的名称，其后跟有一个限定符（见第B.1节）。

“b”代表一个数字，从1开始，每个新定义的通知加1。

#### 3.b.b.1 定义

<定义>小节采用自然语言书写。

这里，还应定义关于这个通知所支持的回溯至一个或多个需求的可追溯性的信息，采用下面的形式：

引用	需求标签	备注

#### 3.b.b.2 输入参数

通知的输入参数列表，每个元素是一个元组（参数名称、限定符、匹配的信息/信息类型（见附件E和第E.2条中的注释）以及可选的这个参数所支持的合法值列表、备注）。



“限定符”栏包含了两个限定符，支持限定符（见第B.1节）和过滤限定符，用逗号隔开。过滤限定符表示通知的参数能否被过滤，值为是（Y）或否（N）。

此信息以表格的形式提供。

参数名	限定符	匹配的信息/ 信息类型/ 合法值	备注

注 — 信息类型限定了参数名称的参数，在合法值能够被枚举的情况下，每个元素是一个（合法值名称、合法值语义）对，除非合法值语义适用于多个数值，在这种情况下定义只提供一次。当合法值不能被枚举时，合法值列表由单个定义规定。

### 3.b.b.3 触发事件

发送通知的触发事件是由“开始状态”小节规定的信息类型到“到达状态”小节定义的信息类型的转换。

#### 3.b.b.3.1 开始状态

本小节是一个由AND、OR和NOT逻辑运算符结合而成的断言集。

每个断言的定义是一个(propertyName, propertyDefinition)对。构成“开始状态”的所有断言均以表格的形式提供。

断言名	定义

#### 3.b.b.3.2 到达状态

本小节是一个由AND、OR和NOT逻辑运算符结合而成的断言集。当关于信息实体的到达状态什么都没说明时，则假设是这个信息实体与开始状态中的表述对比尚未发生变化。

每个断言的定义是一个(propertyName, propertyDefinition)对。构成“到达状态”的所有断言均以表格的形式提供。

断言名	定义

### 3.b.b.4 约束

<约束>小节提供关于通知或通知的参数的约束。

注 — 当不需要定义约束时，就不需要本小节。

### 3.c 情况

本小节包含一个或多个顺序图，每个图描绘一种可能的情况，这些图应均为符合UML的顺序图，这是一个可选的小节。

## B.3 IOC特性和继承

### B.3.1 特性

从以下几个方面定义 IOC的特性（不包括支持IOC）：

- a) 包括语义和语法、合法值范围和支持限定的IOC属性。IOC属性不限于配置管理，而且包括那些相关的管理，例如1)性能管理（即测量类型），2)踪迹管理和3)会计管理。
- b) 与IOC有关联的非属性特有的行为（见注释1）  
注 1 — 例如，A和B之间的链接是可选的，如果A实例属于一个ManagedElement实例，而B实例属于另一个ManagedElement实例，那么它们之间的链接是强制性的。这种链接行为是非属性特有的行为，可以预期这种行为和其它行为一样，将会被继承。
- c) IOC与另一个IOC的关系。
- d) IOC通知类型及其限定。
- e) IOC和它的父类的关系（见注释2），有三个互斥的案例：
  - 1) IOC是抽象类，尚未指定父类。
  - 2) IOC是抽象类，已经指定了所有可能的父类，不管子类IOC可能会被指定为一个根IOC。
  - 3) IOC不是抽象类，已经指定了所有可能的父类，不管子类IOC可能会被指定为一个根IOC。

IOC实例是一个根IOC或者有且只有一个父类。

注 2 — 本小节中父类与子类的关系是父类名包含子类名的关系。

- f) IOC和其子类的关系。有三个互斥的案例：
  - 1) IOC不应有任何子（名称包含关系）IOC。
  - 2) IOC可以有子IOC，可以规定每个子IOC的实例的最大数量，IOC可以指明不允许厂家特有对象作为子IOC。
  - 3) IOC只有一个特定的IOC（或它们的子类），可以规定每个IOC的实例的最大数量，IOC可以指明不允许厂家特有的对象作为子IOC。
- g) IOC是否能够用示例说明（即，IOC是否为一个抽象IOC）。
- h) 用于命名的属性。

### B.3.2 继承

IOC（子类）继承另一个IOC（超类），在这种情况下子类应具有超类的所有特性。

子类能够将继承的支持限定由可选改为强制，但反过来不行，子类能够将继承的支持限定由有条件可选改为有条件的强制，但反过来不行。

IOC可能是多个IOC的超类，子类不能有一个以上的超类。

子类能够：

- a) 增加（相对于其超类的那些属性）独特的属性包括它们的行为、合法值范围和支持限定。每个附加的属性应有自己唯一的属性名（在所有增加的和继承的属性之间）。

- b) 在IOC基础上增加非属性行为，这种行为不会与继承的超类行为矛盾。
- c) 增加和IOC的关系，每个附加的关系应有自己唯一的名称（在所有增加的和继承的关系之间）。
- d) 增加附加的通知类型及其限定。
- e) 在超类具有Property-e-1的情况下指定所有可能的父类（和它们的子类），IOC将会具有Property-e-2或Property-e-3。如果超类具有Property-e-2或Property-e-3，则限制可能的父类（和它们的子类），并且/或者去除子类成为一个根IOC的能力。
- f) 在超类具有Property-f-2的情况下增加子IOC，IOC将会具有Property-f-3。如果超类具有Property-f-3，则限制允许的子IOC（或它们的子类）。
- g) 规定IOC能否用示例说明（即，IOC是一个抽象IOC）
- h) 限制具有合法值范围的超类属性的合法值范围。

### **B.3.3 导入**

为了促进IOC定义在IRP规范之间的重用，IRP规范（被称为主体IRP）可以使用导入机制，以便重用在另一个IRP规范中规定的IOC定义，当主体IRP规范导入一个IOC时，它不能改变导入的IOC的特性，如果需要改变导入的IOC，必须使用继承来定义它自己的新类。

## 附件C

### MISM UML指令集

(本附件是本建议书的组成部分)

下面是适合于分析阶段结果规范的准则，以3GPP统一建模语言（UML）指令集[b-3GPP TS 32.152]为基础。

#### C.1 引言

UML为模拟分布式系统提供了一组丰富的概念、标记和模型元素，对于分析规范来说，不需要使用所有的UML标记和模型元素。本附件记录了必要且充分的UML标记和模型元素组，包括通过UML扩充机制<< stereotype>>建立的标记和模型元素，用于开发协议中立的规范。总的来说，这组标记和模型元素被称作UML建模指令集。

遵循该方法的建议书应采用本指令集的UML标记和模型元素，也可以采用认为必需的其它UML标记和模型元素。

#### C.2 基本模型元素

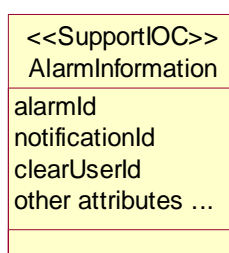
##### C.2.1 概述

UML定义了大量基本模型元素，本小节列出了指令集所选用的子集，所选择子集的语义定义见[OMG UML]。

##### C.2.2 属性

见[OMG UML]的第3.25小节。

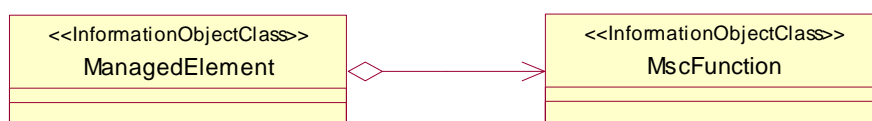
这个例子显示了被列为类AlarmInformation属性部分的字符串的一些属性。



##### C.2.3 聚合

见[OMG UML]的第3.43.2.5小节。

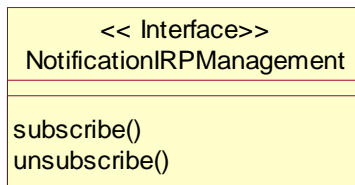
这个例子显示了附加在指示聚合的路径末端的一个空心菱形，该菱形附属于聚合的类。



##### C.2.4 操作

见[OMG UML]的第3.26小节。

这个例子显示了可能会要求NotificationIRPManagement的实例执行的两个操作，这两个操作显示为类NotificationIRPManagement操作部分的字符串，操作有一个名字例如订阅，以及一个自变量列表（没有显示出来）。

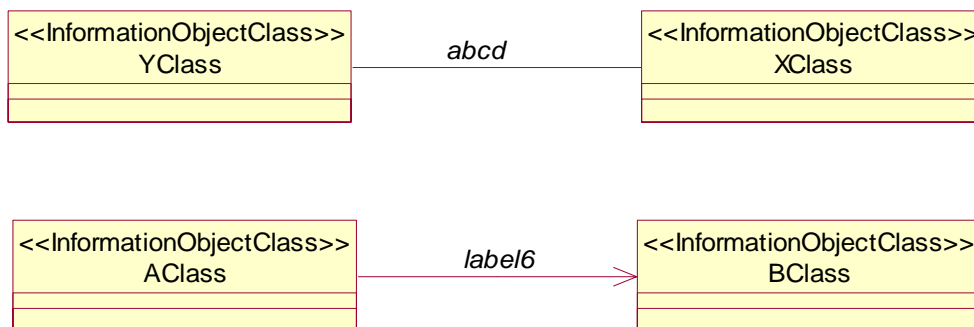


### C.2.5 关联和关联名

见[OMG UML]的第3.41小节。

这两个例子显示了正好两个模型元素之间的二元关联，关联可以包括模型元素与自身相关的可能性。第一个例子显示的是双向关联，在这种关联中，每个模型元素都知道对方，第二个例子显示的是单向关联（用目标模型元素端的公开箭头表示），在这种关联中，只是源模型元素知道目标模型元素，反过来不行。

可以对关联进行命名，例如下面例子中的*abcd* 和 *label6*。



### C.2.6 实现关系

见[OMG UML]的第2.5.2.1小节。

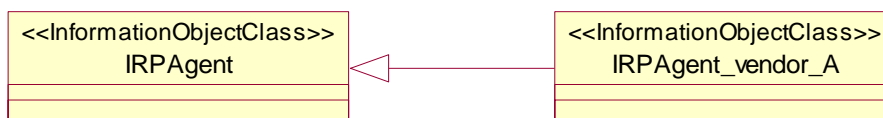
这个例子显示了模型元素AlarmIRPOperations\_1和另一个模型元素AlarmIRP之间的实现关系，后者（目标模型元素）实现前者，目标模型元素必须是<<Interface>>。



### C.2.7 泛化关系

见[OMG UML]的第3.50小节。

这个例子显示的是一个比较通用的元素（代理）和比较专用的元素（Agent\_vendor\_A）之间的泛化关系，后者与第一个元素完全一致，并且增加了额外的信息。



### C.2.8 依赖关系

见[OMG UML]的第3.51小节。

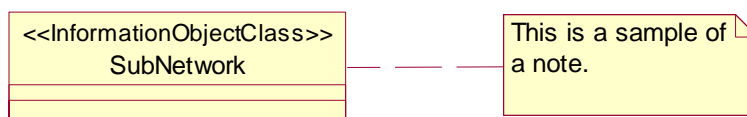
这个例子显示BClass实例和AClass实例有语义上的关系，表示一种情况，在这种情况下改变目标元素会需要改变具有依赖关系的源元素。



### C.2.9 注释

见[OMG UML]的第3.11小节。

这个例子将注释显示为一个右上角有“折角”的矩形，注释可包含任意的文本，它在特定的图上出现，可以通过虚线连接零个或多个建模元素。

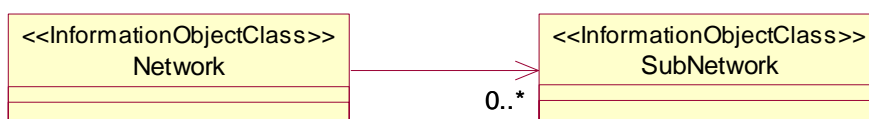


### C.2.10 多重性也称作基数

见[OMG UML]的第3.44小节。

这个例子显示了附加在关联路径一端的多重性，这个多重性的含义是一对多，网络实例能与零个、一个或多个SubNetwork实例相关联。

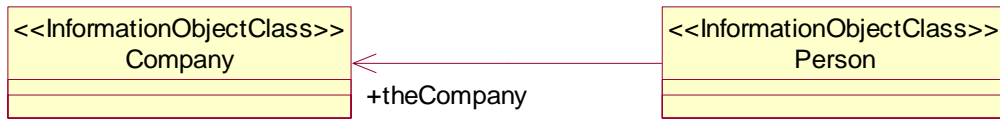
在 [b-3GPP TS 32.152] 以前的版本中，基数零能够表示IOC具有所谓的“瞬态”特性，例如，它表示实例尚未建立，但处于建立的过程之中。在该方法的这个版本中，基数零将不能用于表示这种特性，因为这样的特性被认为是所有IOC固有的，认为所有规定的IOC都有这种固有的“瞬态”特性。



### C.2.11 角色名

见[OMG UML]的第3.43.2.6小节。

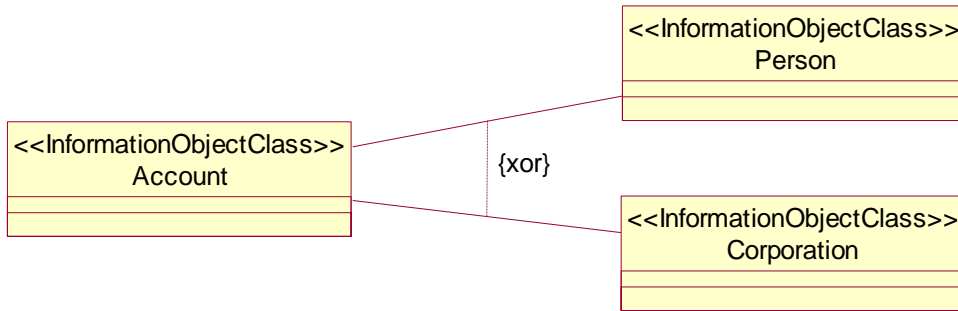
这个例子显示了一个人（比如说实例John）与一个公司的关联（比如说，其DN为“Company=XYZ”），我们可以通过使用相对的关联末端定位这个关联，这样John的Person.the company将保留该DN，即“Company=XYZ”，采用名词作为角色名。



### C.2.12 Xor约束

见[OMG UML]的第2.5.2.3和3.42.5.1小节。

这个例子显示了一个与某个人（例如，John Smith）或公司（例如，ABC有限公司）有关联的帐户（例如，帐户0906）。



## C.3 构造型

### C.3.1 概述

本小节列出了允许在管理接口规范中使用的所有构造型，构造型<<Interface>>的定义见[OMG UML]，本建议书将它列出是为了便于参考和完整性，本建议书还定义了其它的构造型。

表C.3-1 — 实体构造型

构造型	基类	受影响的元模型元素
Interface	类	
ProxyClass	类	
Notification	类	
Archetype	限定符（[OMG UML]的第2.5.2.10小节）	
InformationObjectClass	限定符	
SupportIOC	限定符	
use	关联	
may use	关联	
may realize	关联	
names	组合	

### C.3.2 <<Interface>>

[OMG UML]的第2.5.2.25小节：

“接口是一组指定的、描述元素行为特性的操作，在元模型中，接口包含了一组操作，这些操作共同定义了由实现接口的类元提供的一种服务，一个类元可以提供多种服务，这意味着它可以实现多个接口，多个类元可以实现相同的接口。

接口[可以有或者]可以没有属性、关联或方法，如果接口看不见关联，则可以参与关联；即，类元（不同于接口）可能具有一个到接口的关联，从类元可以导航到这个关联，从接口则不行。”

摘自[OMG UML]的第2.5.4.6小节：

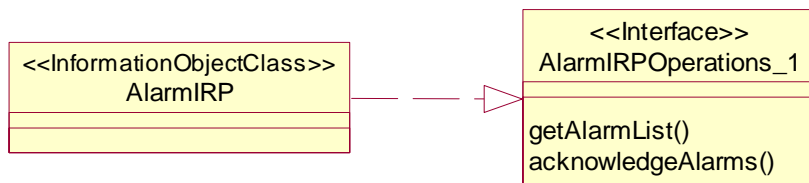
“接口的用途是收集构成类元提供的相关服务的一组操作，接口提供了一种划分操作组并且描述操作组特性的方法，接口只是一个有名字的操作集合，不能直接用示例说明。”

摘自[OMG UML]的第2.5.4.6小节：

“多个类元可以实现相同的接口，所有这些类元必须至少包含与接口所包含的操作相匹配的操作，操作规范包含了操作的签名（即，它的名称，参数的类型和返回类型）。接口不会隐含实现类元的任何内部结构，例如，它不会包含实现操作所采用的算法，然而，操作可能会包含调用结果的说明[例如，有前置条件和后置条件]。”

### C.3.2.1 例子

这个例子显示了具有两个操作的AlarmIRPOperations\_1 <<Interface>>，操作的输入参数和输出参数被隐藏（即，不显示），AlarmIRP与<<Interface>>具有单向的强制性实现关系。



<<Interface>>标记

### C.3.3 <<ProxyClass>>

#### C.3.3.1 概述

这代表多个 <<InformationObjectClass>>，它封装了存在于所代表的 <<InformationObjectClass>>中的属性、链接、方法（或操作）和交互。

<<ProxyClass>>的语义是 <<ProxyClass>>的所有行为都存在于所代表的 <<InformationObjectClass>>中，由于这个类只是其它类的代表，因此除了由所代表的 <<InformationObjectClass>>已经定义的那些行为以外，这个类不能定义自己的行为。

特定的 <<InformationObjectClass>>可以用零个、一个或多个 <<ProxyClass>>或者 <<Archetype>>表示，例如，ManagedElement <<InformationObjectClass>>能够拥有 MonitoredEntity <<ProxyClass>>和ManagedEntity <<ProxyClass>>。

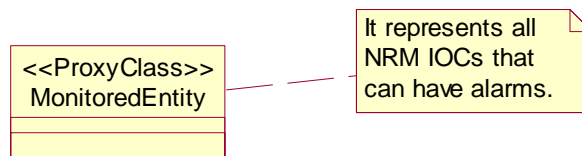
<<ProxyClass>>的属性能被源实体访问，该实体与 <<ProxyClass>>存在关联。

#### C.3.3.2 例子

这个例子显示了一个命名为MonitoredEntity的 <<ProxyClass>>，它代表所有的其实例正在用于告警条件监测的 NRM <<InformationObjectClass>>（例如，GgsnFunction <<InformationObjectClass>>）。



注意到<<MonitoredEntity>>没有定义任何的属性，属性已经由<<MonitoredEntity>>代表的所有<<InformationObjectClass>>定义了。



### <<ProxyClass>>标记

使用<<ProxyClass>>的更多例子见附录五。

## C.3.4 <<Archetype>>

### C.3.4.1 概述

这代表了多个公共类特性（例如，所代表的<<InformationObjectClass>>的典型属性、链接、操作和交互）。

<<Archetype>>的语义是<<Archetype>>封装的所有属性、链接操作和交互可能会也可能不会出现在所代表的<<InformationObjectClass>>中。<<Archetype>>代表一个在技术中立分析模型中最有用的占位符类，该分析模型在一个更为复杂的结构模型内部需要更进一步的规范和/或映射。

### C.3.4.2 例子

这个例子显示了一个命名为StateManagement的<<Archetype>>，还显示了依赖于这个StateManagement的<<InformationObjectClass>>代理，注意到StateManagement已经定义了多个属性（没有显示在UML图中），依赖于这个StateManagement的类可能会也可能不会使用所有的StateManagement属性，换言之，至少有一个StateManagement属性会出现在这个代理中，该代理所使用StateManagement属性的精确设置在代理规范中规定。



### <<Archetype>>标记

## C.3.5 <<InformationObjectClass>>

### C.3.5.1 概述

这代表一个IOC，每个<<InformationObjectClass>>代表一组具有相似结构、行为和关系的实例。

这个<<InformationObjectClass>>和其它的信息类例如<<Interface>>，会被映射到技术特有的模型元素例如适合CMIP技术的GDMO被管理对象类。从协议中立建模结构到技术特有建模结构的映射收录在相应的协议特有的规范中。

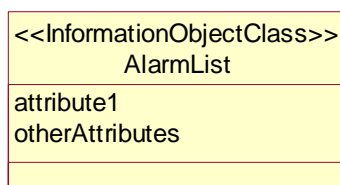
<<InformationObjectClass>>名称的作用域在规范这个类的建议书之内，该名称在该建议书内所有的<<InformationObjectClass>>名称之中必须是唯一的，建议书名称被认为与UML数据包名称类似。

除了未包含/定义方法或操作以外， <<InformationObjectClass>>与UML类相同。

[OMG UML]的第3.22.1小节：“类代表被模拟系统内部的一个概念，类具有数据结构、行为以及与其它元素的关系。”

### C.3.5.2 例子

这个例子显示了一个AlarmList <<InformationObjectClass>>。



### <<InformationObjectClass>> 标记

### C.3.6 <<use>>和<<may use>>

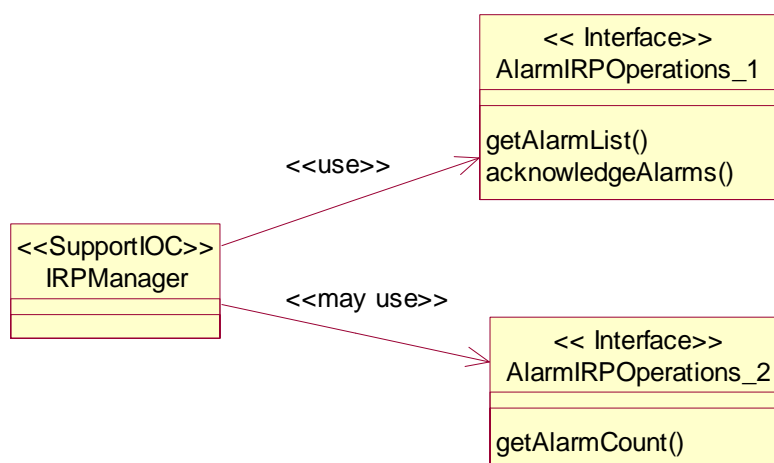
<<use>> 和 <<may use>> 均为单向关联，目标必须是<<Interface>>或<<Notification>>。

在目标是<<Interface>>的情况下， <<use>>规定源类必须具有使用目标<<Interface>>的能力，这样它就能够调用<<Interface>>定义的操作。源实体支持该能力是强制性的。 <<may use>>规定源类可以具有使用目标<<Interface>>的能力，这样它就可以调用<<Interface>>定义的操作，源实体支持该能力是可选的。

在目标是<<Notification>>的情况下， <<use>>规定源类必须是目标<<Notification>>定义通知的发起者，源实体支持该能力是强制性的， <<may use>>规定源类可以是<<Notification>>定义通知的发起者，源实体支持该能力是可选的。

#### C.3.6.1 目标<<Interface>>的例子

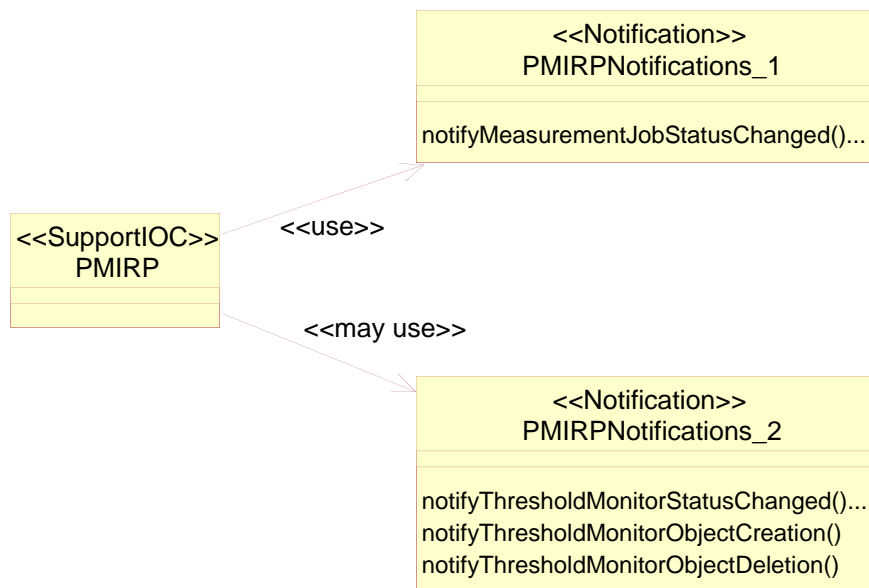
这个例子显示IRPManager应使用AlarmIRPOperations\_1定义的操作，以及可能会使用AlarmIRPOperations\_2定义的操作。



### 用于目标<<Interface>>的<<use>> 和<<may use>>标记

### C.3.6.2 目标<<Notification>>的例子

这个例子显示PMIRP应具有发出或者发起PMIRPNotifications\_1所定义通知的能力，以及可能会具有发出或发起PMIRPNotifications\_2所定义通知的能力。



用于目标<<Notification>>的<<use>>和<<may use>>

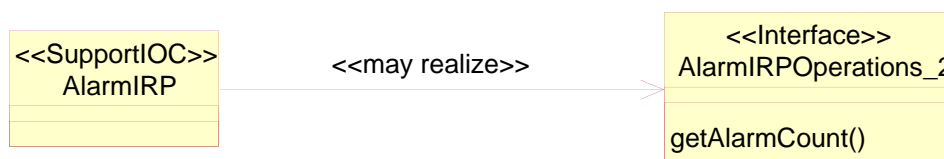
### C.3.7 <<may realize>>

<<may realize>>是一个单向关联，目标必须是<<Interface>>，<<may realize>>显示源实体可以实现目标<<Interface>>定义的操作。

注意到UML基本元素已经定义了实现关联（因此，不需要定义这种关联的构造型），实现关联显示源实体必须实现（或执行）目标<<Interface>>定义的操作。

#### C.3.7.1 例子

这个例子显示AlarmIRP可能会实现AlarmIRPOperations\_2的操作。



<<may realize>>标记

### C.3.8 <<names>>

它规定了一个单向的组合，在源实体的名称空间内，相同目标类元的所有其它目标实例之中，以及具有和源实体的相同<<names>>组合的其它类元的其它目标实例之中，目标实例是唯一的。

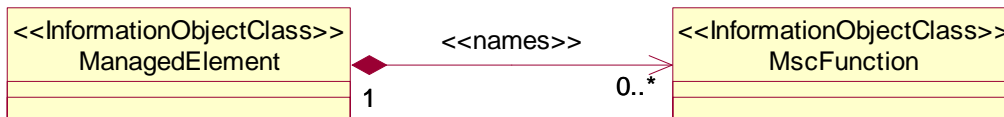
源类元和目标类元都应具有命名属性。

用作名称包含行为的组合提供了域和包含的指定元素之间整体-部分关系的语义，即使只通过名称。从管理的角度来说，进入部分要经过整体，应在这种关系的两端指出多重性。

目标实例不能拥有和多个源的多个<<names>>，即目标实例不能参与或属于多个名称空间。

### C.3.8.1 例子

这个例子显示MscFunction的所有实例在ManagedElement实例的名称空间内是唯一可识别的。



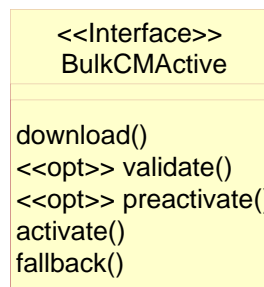
<<names>> 标记

### C.3.9 <<opt>>

<<opt>>（或<<optional>>）使得能在UML图中显示可选的属性、参数和操作（分别地）。

在没有构造型的情况下，所述的属性、参数或操作是强制性的。

#### C.3.9.1 例子



用于操作的<<opt>>标记

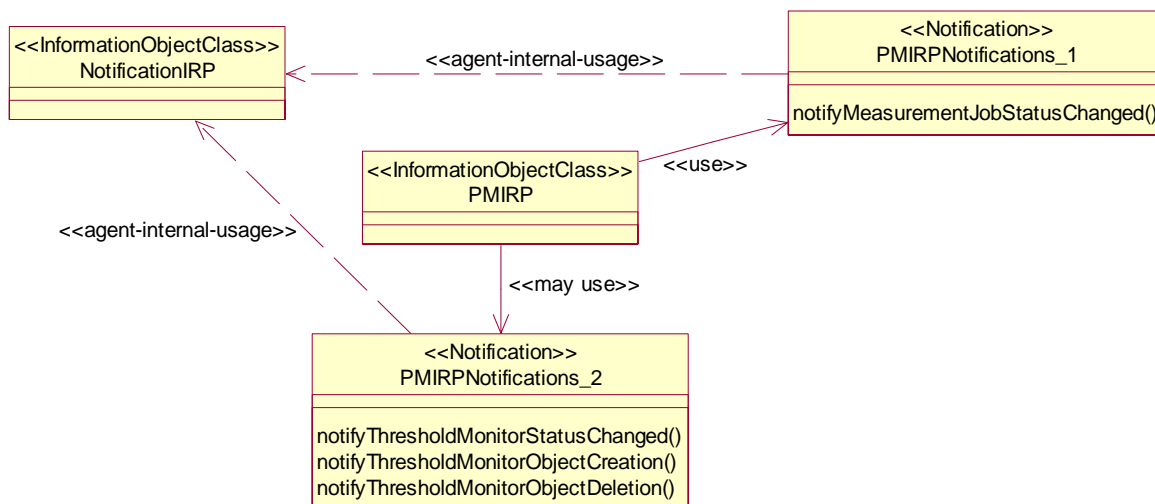
### C.3.10 <<Notification>>

#### C.3.10.1 概述

<<Notification>>是一个指定的通知集。

#### C.3.10.2 例子

这个例子显示了一个命名为“PMIRPNotifications\_1”的<<Notification>>和另外一个命名为“PMIRPNotifications\_2”的<<Notification>>，两者都有通知，通知的一个例子是notifyMeasurementJobStatusChanged()。



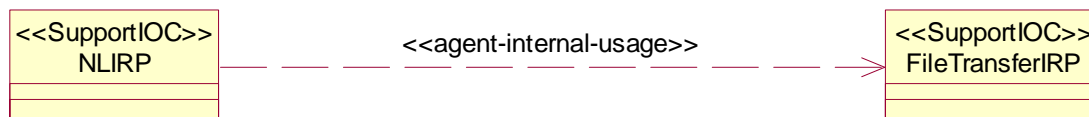
### <<Notification>>标记

#### C.3.11 <<agent-internal-usage>>

这是一个单向关联，源传递网络管理信息给目标，源和目标是在不同IRP实体中运行的实体或过程，例如AlarmIRP、PMIRP。实例可能是相同的IRP Agent或不同IRP Agent实例包含的名称，被传递的准确的网络管理信息以及信息传递机制不是标准化的，是厂家特有的。

##### C.3.11.1 例子

这个例子表明NLIRP（NotificationLog IRP）能够传递一些网络管理信息给FTIRP（FileTransferIRP）。



### <<agent-internal-usage>> 标记

#### C.3.12 <<SupportIOC>>

它是一组管理能力的描述符。

除了没有包含/定义方法或操作以外，<<SupportIOC>>和UML类相同。

[OMG UML] 的第3.22.1小节：“类代表被模拟系统内部的一个概念，类具有数据结构和行为以及与其它元素的关系。”

##### C.3.12.1 例子

这个例子显示了一个AlarmList <<SupportIOC>>。



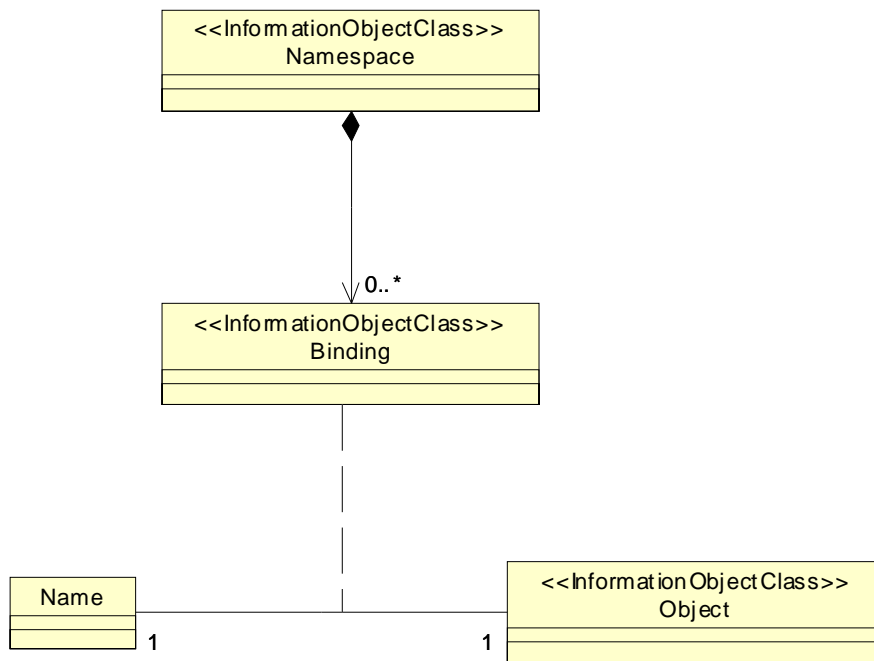
### <<SupportIOC>> 标记

## C.4 关联类

[OMG UML]的第3.46小节定义了一个关联类：

“关联类是一个也具有类特性的关联（或者一个具有关联特性的类），即使被描绘成一个关联和一个类，但它实际上只是单一模型元素。”

关联类适用于“InformationObjectClass”需要保持与其它多个“InformationObjectClass”的关联并且“包含的”“InformationObjectClass”范围内的关联成员之间存在关系的时候，例如，名称空间维持着一组绑定，一个绑定将一个名称与一个对象联系在一起，绑定“IOC”能够用一个关联类来建模，该关联类给名称与其它一些“InformationObjectClass”之间的关系提供了绑定语义。如下图所示。（只是示范性的，并非摘自另外的建议书。）



关联类的例子

## C.5 抽象类

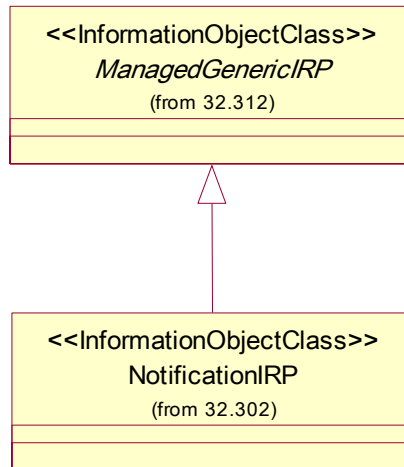
### C.5.1 概述

它将<<InformationObjectClass>>规定为一个被子类继承的基类，抽象类不能用示例说明。

抽象类标记是在图中相应的<<InformationObjectClass>>类名称上使用斜体字。

### C.5.2 例子

这个例子显示ManagedGenericIRP是一个抽象的<<InformationObjectClass>>。



### 抽象类标记

#### C.6 <<InformationObjectClass>> 和 <SupportIOC>>的应用

<<InformationObjectClass>>和<<SupportIOC>>均为构造型，这两个构造型有着相同的用途，因为每个都是一个指定的管理特性集，然而，它们的应用在支持通过管理接口的管理的语境下能够各不相同，本小节重点介绍这些应用的相似点和区别。

	<<InformationObjectClass>>	<<SupportIOC>>
它能是一个抽象类吗？	是	是
它能是一个具体类吗？	是	是
它能从<<InformationObjectClass>>继承吗？	是	否
它能从<<SupportIOC>>继承吗？	否	是
它是<<InformationObjectClass>>包含的名称吗？	是	是
它是<<SupportIOC>>包含的名称吗？	否	是
实例能有一个DN吗？	<<InformationObjectClass>>必须是一个命名树类，意味着其所有的实例必须有一个DN。	<<SupportIOC>>可能会被规范作者用于命名树内部的一个类，如果这样的话，这意味着它的所有实例将有一个DN。
管理站能够通过其objectClass和objectInstance参数传送实例DN的通知接收到信息吗？	是。 发出通知的类型用与类定义有关联的通知表格显示。	是，如果<<SupportIOC>>是一个命名树类的话。 发出通知的类型用与类定义有关联的通知表格显示。 否，如果<<SupportIOC>>不是一个命名树类的话。

## 附件D

### 设计

(本附件是本建议书的组成部分)

本附件提供协议特有设计规范的准则，有待进一步研究。



## 附件E

### 信息类型定义 — 类型目录

(本附件是本建议书的组成部分)

本附件定义了一个应该在概念模型（分析模型/信息服务）中用于规定类型信息的类型目录。

该目录规定了ASN.1[ITU-T X.680]所定义类型的一个子集，结合由ASN.1所定义的类型导出的类型（第E.4节）。

表E.1归纳了用于各个类型的关键字。

#### E.1 基本类型

基本类型是能够直接用于定义属性和参数的类型，基本类型也能用于构造复杂类型，基本类型包括以下ASN.1类型：

**E.1.1 整数型** [ITU-T X.680]的第19节。

**E.1.2 实数型** [ITU-T X.680]的第21节。

**E.1.3 布尔型** [ITU-T X.680]的第18节。

**E.1.4 比特串型**[ITU-T X.680]的第22节。

**E.1.5 空型** [ITU-T X.680]的第24节。

**E.1.6 广义时间类型** [ITU-T X.680]的第38节。

#### E.2 枚举类型

[ITU-T X.680]的第20节枚举类型代表一个枚举的数值，可能被特定属性或参数使用的所有数值应列出在合法值栏中，列出的名称样式只适用于概念模型，即具体的值（数字或字符串）的识别留给具体的设计模型。

注 — 如果这些值的数量超过了50，则建议用一个附录或者一个独立的文档来规定它们。

#### E.3 复杂类型

可以采用以下概念定义复杂类型：

**E.3.1 序列类型** [ITU-T X.680]的第25节

**E.3.2 选择类型**[ITU-T X.680]的第29节

**E.3.3 集合类型** [ITU-T X.680]的第27节

此外，通过使用以下类型支持复杂类型的序列和集合：

**E.3.4 单一序列类型** [ITU-T X.680]的第26节

**E.3.5 单一集合类型** [ITU-T X.680]的第28节

#### E.4 有用的类型

##### E.4.1 字符串类型

字符串代表一串字符，字符集不限，即：

String ::= UnrestrictedCharacterStringType [ITU-T X.680]的第44节。

## E.4.2 名称类型

名称代表对象实例在名称空间内的唯一名字，它可能包括对象包含树层次信息，但是它与实现相关，并且不在本建议书的范畴内。形式上，名称类型定义如下：

Name ::= TYPE-IDENTIFIER [ITU-T X.681]的附件A

## E.5 关键字

表E.1规定了在分析模板中用于信息类型定义的关键字列表，例如：

参数名	支持限字符	信息类型/合法值	备注
...			
eventIdList	M	SET OF INTEGER/-	要被确认告警的列表。

表 E.1 — 关键字

类型	关键字
整数型	INTEGER
实数型	REAL
布尔型	BOOLEAN
比特串型	BIT STRING
空型	NULL
广义时间类型	GeneralizedTime
枚举型	ENUMERATED
序列类型	SEQUENCE
选择类型	CHOICE
集合类型	SET
单一序列类型	SEQUENCE OF
单一集合类型	SET OF
字符串型	String
名称类型	Name

## 附件F

### 关于 IOC特性、继承和实体导入的准则

(本附件是本建议书的组成部分)

下列准则以[b-3GPP TS 32.150]为基础。

#### F.1 IOC特性

从以下几个方面规定IOC的特性（包括支持IOC）：

- a) 包括语义和语法、合法值范围和支持限定的IOC属性。IOC属性不限于配置管理，而且包括那些相关的管理，例如1)性能管理（即测量类型），2)踪迹管理和3)会计管理。
- b) 与IOC有关联的非属性特有的行为。

注 1 — 例如，MscServerFunction 和 CsMgwFunction 之间的链接是可选的，如果MscServerFunction实例属于一个ManagedElement实例，而CsMgwFunction实例属于另一个ManagedElement实例，那么它们之间的链接是强制性的。这种链接行为是非属性特有的行为，可以预期这种行为和其它行为一样，将会被继承。

- c) IOC与另一个IOC的关系。
- d) IOC通知类型及其限定。
- e) IOC和它的父类的关系（见注释2），有三个互斥的案例：
  - 1) IOC能够有任何的父类，在UML图中，该类有一个父类Any。
  - 2) IOC是抽象类，所有可能的父类均已指定，不管子类IOC可能会被指定为一个根IOC，在UML图中，子类有特定类的零个或者多个可能的父类（除了Any以外）。
  - 3) IOC是具体类，所有可能的父类均已指定，不管IOC可能会被指定为一个根IOC，在UML图中，该类有特定类的零个或者多个可能的父类（除了Any以外）。

IOC实例是一个根IOC或者有且只有一个父类，只有3GPP SA5可能会指定一个IOC类为一个可能的根IOC，当前，只有SubNetwork、ManagedElement或者MeContext IOC可以是根IOC。

注 2 — 本小节中父类与子类关系是父类名包含子类名的关系。

- f) IOC与其子类的关系，有三种互斥的案例：
  - 1) IOC不应有任何子（名称包含关系）IOC，在UML图中，该类没有子类。
  - 2) IOC能够有子IOC，可以规定每个子IOC的实例的最大数量，IOC可以指明不允许厂家特有对象作为子IOC，在UML图中，该类有一个子类Any。
  - 3) IOC只能有特定的子IOC（或它们的子类），可以规定每个子IOC的实例的最大数量，IOC可以指明不允许厂家特有的对象作为子IOC，在UML图中，该类有特定类的一个或多个子类（除了Any以外）。
- g) IOC能否用示例说明（即，IOC是否为一个抽象IOC）。
- h) 用于命名目的的属性。

## F.2 继承

IOC（子类）继承另一个IOC（超类），在这种情况下子类应具有超类的所有特性。

子类能够将继承的支持限定从可选改为强制，但反过来不行，子类能够将继承的支持限定从有条件的可选改为有条件的强制，但反过来不行。

IOC能够是许多IOC的一个超类，子类不能有一个以上的超类。

子类能够：

- a) 增加（与其超类的那些属性相比）独特的属性，包括它们的行为、合法值范围和支持限定，每个附加的属性应有自己唯一的属性名称（在所有增加的和继承的属性之间）。
- b) 在IOC基础上增加非属性行为，这种行为不会与继承的超类行为矛盾。
- c) 增加和IOC的关系，每个附加的关系应有自己唯一的名称（在所有增加的和继承的关系之间）。
- d) 增加另外的通知类型及其限定。
- e) 在超类具有Property-e-1的情况下指定所有可能的父类（和它们的超类），这样IOC将会具有Property-e-2或Property-e-3。如果超类具有Property-e-2或Property-e-3，则限制可能的父类（和它们的超类），并且/或者去除该子类成为一个根IOC的能力。
- f) 在超类具有Property-f-2的情况下增加子IOC，这样IOC将会具有Property-f-3。如果超类具有Property-f-3，则限制允许的子IOC（或它们的子类）。
- g) 规定IOC能否用示例说明（即，IOC是一个抽象IOC）。
- h) 限制具有合法值范围的超类属性的合法值范围。

## F.3 实体（接口、IOC和属性）导入

管理接口规范定义了实体（例如，IOC、接口和属性），为了促进实体定义在接口规格说明之间的重用，要使用导入机制。当管理接口规范（主体规格说明）导入一个由另一个管理接口规范定义的实体时，可以认为该主体规范已经在其规范中定义了所导入的实体，此外，主体规范不能改变导入实体的特性，如果要求一个实体与导入的实体不同但相似时，应定义一个继承导入实体的新实体，并在新的实体定义中引入变化。

# 附录 I

## 需求举例

(本附录不是本建议书的组成部分)

注 — 下列例子以告警管理为基础，但只是用于说明性目的，不能作为一组完整的或者正确的告警管理需求。

### 1 概念和背景

对于NE和整个网络健康状况的任何评估都需要检测网络中的故障，因而也需要将告警通知OS（EM和/或NM）。

### 2 商务级需求

#### 2.1 需求

可能网络中出现的故障可以归于下列类型中的一种：

- 硬件故障，即NE内部某些物理资源不能正常工作。
- 软件问题，例如，软件缺陷、数据库不一致。

##### 2.1.1 故障检测

大部分的故障应具有明确规定的宣布其存在或消失的条件，即故障发生和故障清除条件，在本附录中，所有这样的事件均应被称为ADAC故障，当一个以前被检测到的ADAC故障不再存在即故障清除时，网络实体应该能够使用类似于它们用于检测故障发生的技术予以识别。

##### 2.1.2 告警清除

需要清除由于故障引起的告警，为了清除一个告警，通常需要修理相应的故障。

...

REQ-FM-FUN-02 每当一个告警被清除时，代理应产生一个适当的清除告警事件，清除告警被规定为一种告警。

##### 2.1.3 告警前转和过滤

REQ-FM-FUN-03 对于每个被检测到的故障，适当的告警（故障通知）应由有故障的网络实体产生。

...

### 2.2 参与者角色

被管理的系统 履行代理角色的实体。

管理系统 履行管理站角色的实体。

### 2.3 电源资源

在本建议书中，被管理的网络设备被看作是相关的电信资源。

## 2.4 高层用例图

### 2.4.1 报告告警

图I.1中的第一个概括性用例图显示了告警接口的总的交互过程。

第一个概括性用例显示了与报告一个被检测到的故障有关的交互过程。



M.3020(11)\_F.1.1

图 I.1 — 报告告警

## 3 规范级需求

### 3.1 需求

没有规范级需求。

### 3.2 参与者角色

见本模板的第2.2小节。

### 3.3 电信资源

见本模板的第2.3小节。

### 3.4 用例

#### 3.4.1 故障通知

用例段	进展/规范	<<Uses>> 相关的使用
目标 (*)	根据故障条件检测，被管理系统发送一个告警报告通知，经过相关类型的接口Q到达管理系统。	
参与者和角色(*)	管理系统是来自被管理系统的通知的一个用户。	
电信资源	所有被管理的实体。	
假定	故障条件被检测到。	
前置条件	在管理系统和被管理系统之间有一个开放的通信信道。	
什么时候开始	故障条件被检测到。	
第1步 (*)	根据故障条件检测，产生适当的告警报告或安全性告警报告。	
什么时候结束	代理发出告警报告或者安全性告警报告。	

用例段	进展/规范	<<Uses>> 相关的使用
异常	通信或处理失败可能会导致无法将告警递送到管理系统，告警同步用例涵盖了这种情况。	
后置条件	将被管理系统中的故障条件通知管理系统。	
可追溯性(*)	REQ-FM-FUN-01、REQ-FM-FUN-02等。	
<p><b>3.4.2 告警清除</b></p> <p>...</p> <p><b>3.4.3 确认告警</b></p> <p>...</p>		

## 附录 II

### 分析举例

(本附录不是本建议书的组成部分)

注 — 下面例子是以告警管理为基础，但只是用于说明性目的，并不打算作为一组完整的或正确的告警管理需求。

#### 1 概念和背景

对于NE和整个网络健康状况的任何评估都需要检测网络中的故障，因而也需要将告警通知OS（EM和/或NM）。

...

#### 2 信息对象类

##### 2.1 导入的信息实体和本地标签

标签引用	本地标签
3GPP TS 32.302, 信息对象类, NotificationIRP	NotificationIRP
3GPP TS 32.302, 接口, notificationIRPNotification	NotificationIRPNotification
3GPP TS 32.622, 信息对象类, IRPAgent	IRPAgent
3GPP TS 32.312, 信息对象类, ManagedGenericIRP	ManagedGenericIRP

##### 2.2 类图

本小节介绍了封装代理内部信息的信息对象类（IOC）的集合，旨在确定操作和通知发出的AlarmAgent实现所需要的信息，本小节提供了UML中所有支持的对象类的概述，后续的小节提供了这些支持对象类各个方面的更加详细的规范。



## 2.2.1 属性和关系

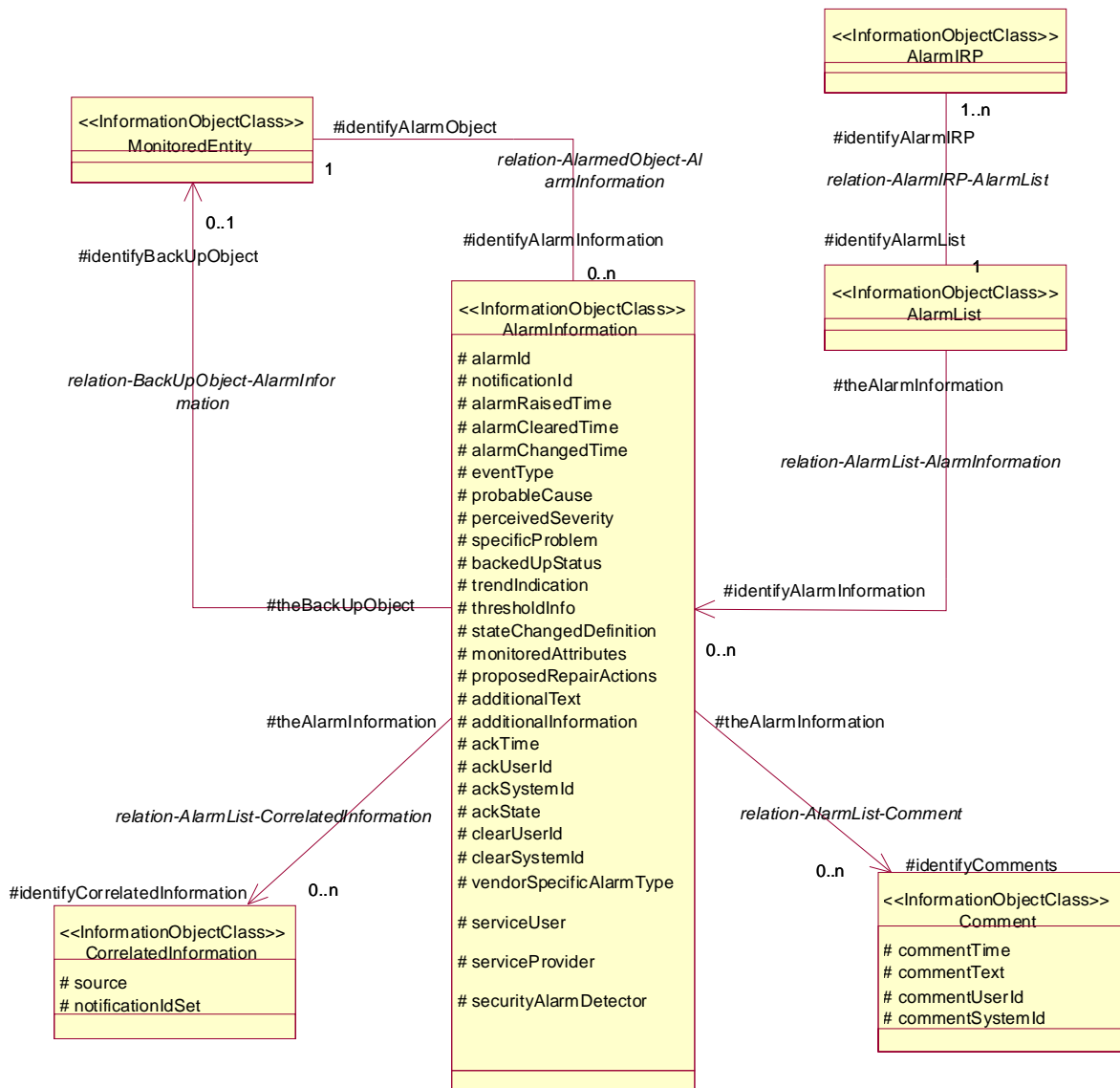
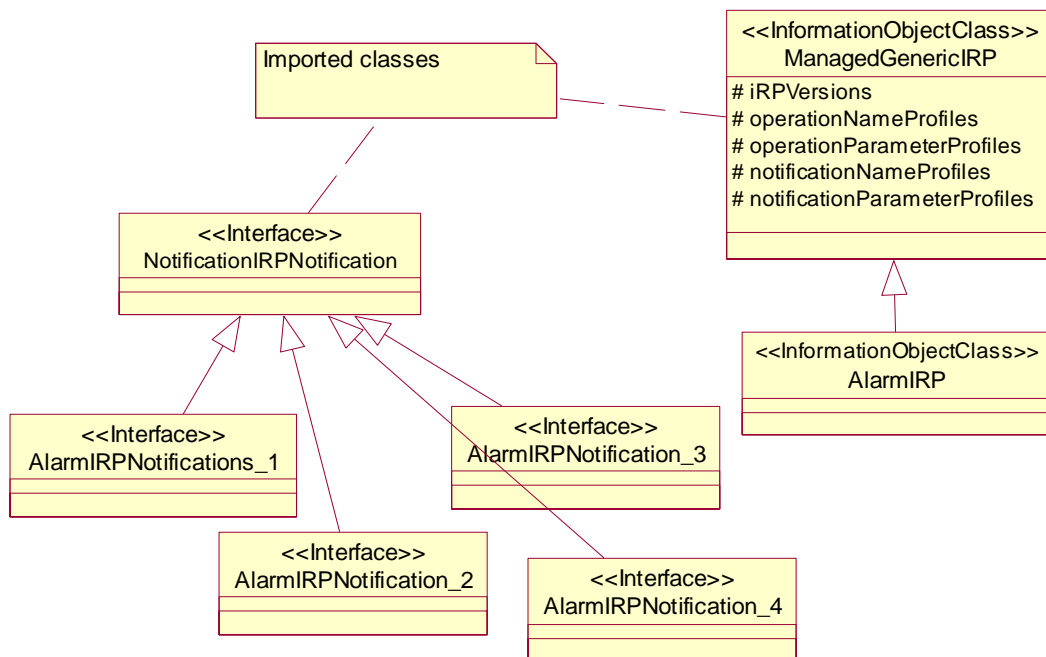


图 II.1 — 告警管理信息对象类

## 2.2.2 继承



图II.2 — 告警管理IOC继承

## 2.3 信息对象类定义

类名	限定符	需求 ID
AlarmInformation	M	REQ-FM-FUN-01, REQ-FM-FUN-02, etc.
AlarmList	M	REQ-FM-FUN-n
...		

### 2.3.1 AlarmInformation

#### 2.3.1.1 定义

AlarmInformation 包含关于产生告警的MonitoredEntity的告警条件的信息。

....

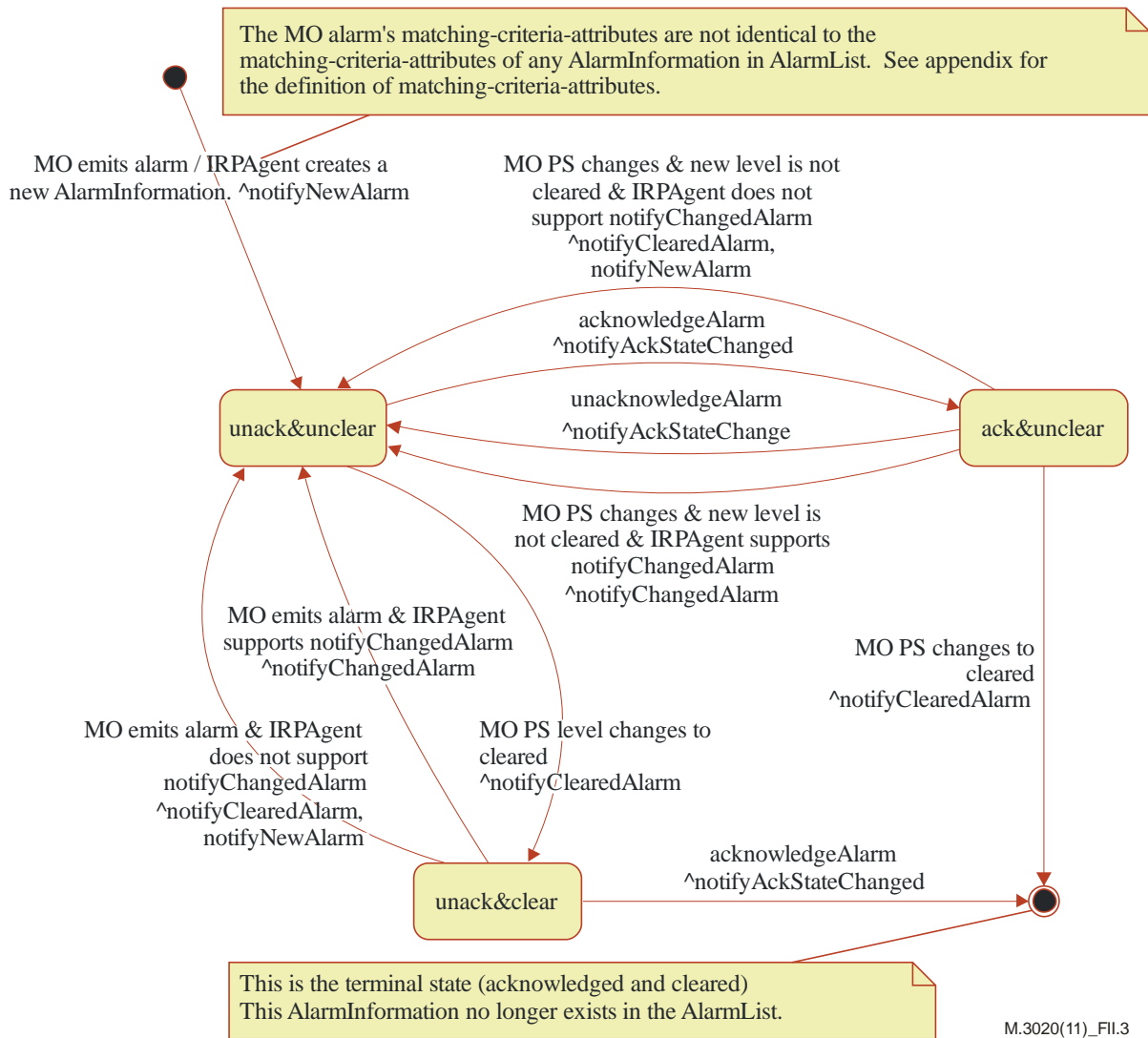
#### 2.3.1.2 属性

属性名	支持限定符	读限定符	写限定符	需求ID
alarmed	M	M	M	
probableCause	C	M	C	
structuredProbableCause	C	M	C	
perceivedSeverity	M	M	M	
specificProblem	O	O	O	
...				
...				

### 2.3.1.3 状态图

告警有多个状态。

...



图II.3 — 告警信息状态图

### 2.3.2 AlarmList

#### 2.4 信息关系定义

关系	支持限定符支持限定符	需求ID
relation-AlarmIRP-AlarmList	M	REQ-FM-FUN-x
...		

#### 2.4.1 relation-AlarmIRP-AlarmList (M)

##### 2.4.1.1 定义

这代表AlarmIRP和AlarmList之间的关系。

### 2.4.1.2 角色

名称	定义
identifyAlarmIRP	代表获得一个或多个AlarmIRP身份的能力。
identifyAlarmList	代表获得一个AlarmList身份的能力。

### 2.4.1.3 约束

这个关系没有约束。

### 2.4.2 relation-AlarmList-AlarmInformation (M)

...

## 2.5 信息属性定义

### 2.5.1 定义和合法值

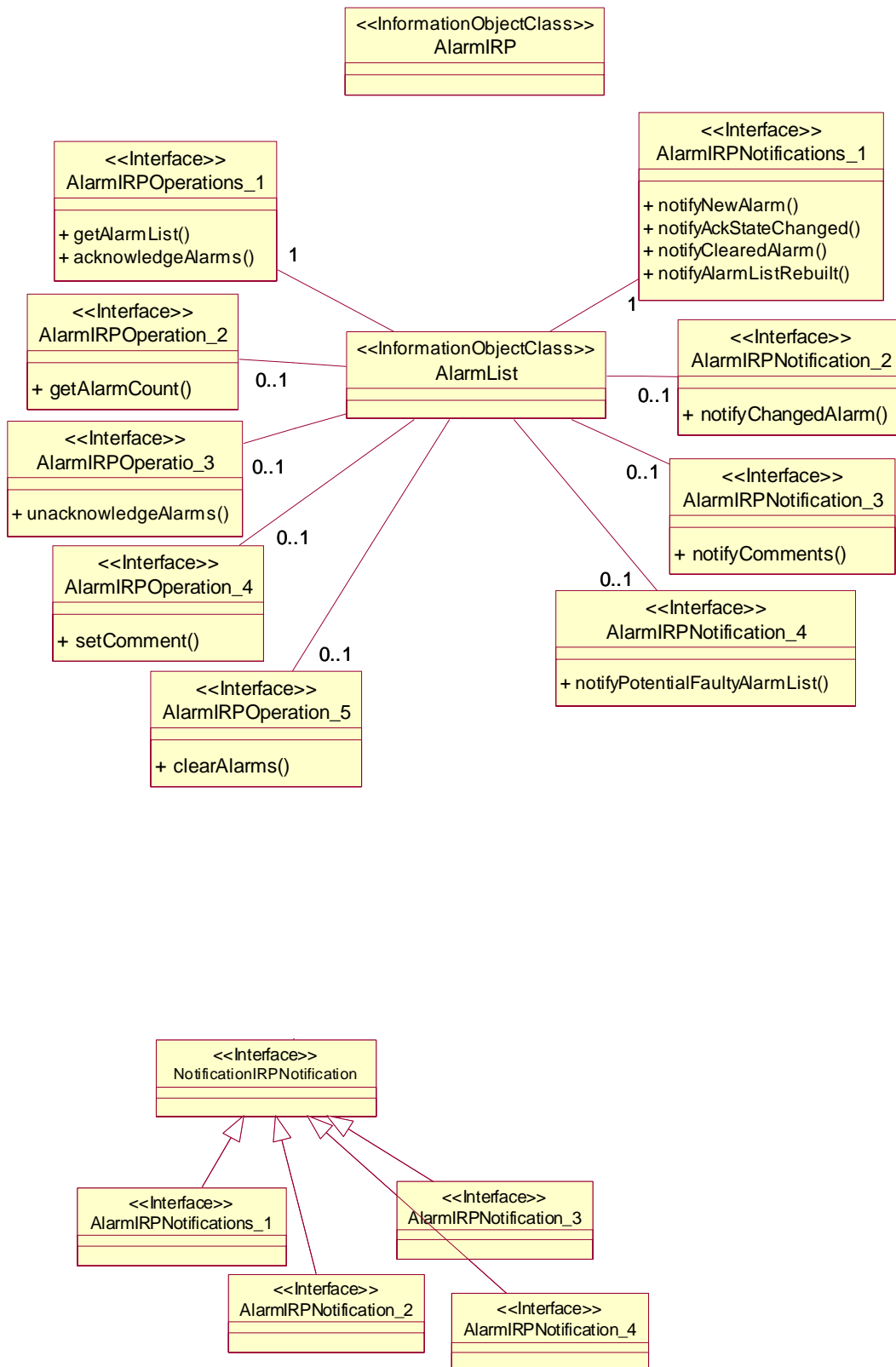
名称	定义	信息类型/ 合法值
alarmed	标识AlarmList中的一个AlarmInformation,	INTEGER
notificationId	标识传送AlarmInformation的通知。	INTEGER
ntfSubscriptionState	表示预约的活动状态。	枚举/“suspended”：预约被挂起。 “notSuspended”：预约是有效的。ENUMERATED/ “suspended”：the subscription is suspended. 预约被挂起。 “notSuspended”：the subscription is active. 预约是有效的。

### 2.5.2 约束

名称	受影响的属性	定义
inv_notificationId	notificationId	NotificationIds对于特别的被管理对象（代表NE）在告警相关很重要的整个时间段内的所有通知应是唯一的，完成告警相关所采用的算法超出了本IRP的范畴。

### 3 接口定义

#### 3.1 表示接口的类图



图II.4 — 告警管理IRP类图

### 3.2 一般规则

**规则1:** 各个具有至少一个输入参数的操作支持前置条件`alid_input_parameter`，该条件表示所有的输入参数应相对于其信息类型是有效的，此外，每个这样的操作支持异常`operation_failed_invalid_input_parameter`，该异常在前置条件`valid_input_parameter`为假的时候出现。异常具有相同的进入和退出状态。

**规则2:** 各个具有至少一个可选输入参数的操作支持一组前置条件`supported_optional_input_parameter_xxx`，这里，“xxx”是可选输入参数的名称，预置条件表示该操作支持指定的可选输入参数。此外，每个这样的操作支持异常`operation_failed_unsupported_optional_input_parameter_xxx`，该异常在以下情况出现：

- a) 预置条件`supported_optional_input_parameter_xxx`为假；且
- b) 指定的可选输入参数在传送信息。

异常具有相同的进入和退出状态。

**规则3:** 各个操作应支持当发生内部问题且操作不能完成时出现的一般异常`operation_failed_internal_problem`，异常具有相同的进入和退出状态。

### 3.3 接口AlarmIRPOperations\_1 (O)

操作名	限定符	需求ID
<code>acknowledgeAlarms</code>	M	REQ-FM-FUN-x, REQ-FM-FUN-y
<code>getAlarmList</code>	M	...

#### 3.3.1 操作`acknowledgeAlarms` (M)

##### 3.3.1.1 定义

管理站调用这个操作以确认一个或者多个告警。

##### 3.3.1.2 输入参数

参数名	支持限定符支持限定符	信息类型/合法值	备注
...			
<code>eventIdList</code>	M	SET OF INTEGER/-	要被确认告警的列表。

##### 3.3.1.3 输出参数

参数名	支持限定符匹配	匹配的信息/信息类型/合法值	备注
...			
Status	M	-- / ENUM / “OperationSucceeded”：如果 <code>allAlarmsAcknowledged</code> 为真， “OperationPartiallySucceeded”：如果 <code>someAlarmAcknowledged</code> 为真， “OperationFailed”：如果 <code>operationFailed</code> 为真。	

### 3.3.1.4 前置条件

atLeastOneValidId。

断言名	定义
atLeastOneValidId	AlarmInformationReferenceList包含至少一个标识AlarmList中AlarmInformation的标识符，这个标识的AlarmInformation应具有指示“未确认”的ackState，以及相同的perceivedSeverity，如果提供该信息的话。

### 3.3.1.5 后置条件

someAlarmAcknowledged 或者 allAlarmsAcknowledged。

断言名	定义
someAlarmAcknowledged	...
allAlarmsAcknowledged	...

### 3.3.1.6 异常

名称	定义
operation_failed	<b>条件：</b> 前置条件为假或者后置条件为假。 <b>返回的信息：</b> 输出参数状态。 <b>退出状态：</b> 进入状态。

### 3.3.2 操作getAlarmList (M)

...

## 附录 III

### 与ITU-T Z.601建议书对比

(本附录不是本建议书的组成部分)

本附录提供了关于本建议书和[b-ITU-T Z.601]之间关系的信息，[b-ITU-T Z.601]用于制定ITU-T M.1400系列建议书中的建议书。

尽管本建议书提供了一种用于确定两个物理系统之间管理接口的方法，[b-ITU-T Z.601]提供了一个适用于系统开发的框架，这个数据体系结构确定了系统内部的备选接口以及系统边界上的接口，边界上的这些接口将位于系统之间。

本建议书规定的方法主要旨在制定一组管理接口建议书，而不是开发各个系统，数据体系结构没有规定与需求阶段相似的需求收集，因为它只规定了各个系统的规范，没有规定它们与组织有关的用途。

[b-ITU-T Z.601]重点关注于终端用户感知的外部术语和语法的规范，本建议书重点关注终端用户可能不会感知的管理接口规范。

在本建议书中，关于要解决的问题的需求可分成两大类，第一类需求被称为商务需求；第二类被称作规范需求，规范需求可能会包含支持终端用户在他们的人机接口进行交互的需求，这些需求的其中一些可能会规定任何管理接口均支持的语法需求，语法需求相当于[b-ITU-T Z.601]中描述的数据体系结构的外部术语摘要。

分析阶段的输出将是一个信息模型，这相当于[b-ITU-T Z.601]中描述的数据体系结构的概念模式，如果来自分析阶段的信息模型没有传递所有必需的来自语法需求的信息，则实现设计可能会需要包含一个来自语法需求的映射。

来自实现设计阶段的文档将由两部分组成：

- 1) 多个接口公共的、技术相关的数据规范，例如，采用GDMO或者CORBA IDL，相当于符合[b-ITU-T Z.601]中数据体系结构的内部术语摘要。
- 2) 各个接口的技术相关规范，例如，采用CMIP或者CORBA IDL，相当于符合[b-ITU-T Z.601]中数据体系结构的分配模式。



## 附录 IV

### 有待进一步研究的问题

(本附录不是本建议书的组成部分)

本附录标识了已知的、需要进一步研究的问题。

#### IV.1 SOA

[ITU-T M.3060] (用于下一代网络管理的准则) 的批准表明管理方法由面向对象向面向服务的改变, 需要对这一变化的影响进行研究, 以便在本建议书未来的版本中标识出需要的所有改变。

#### IV.2 UML

ITU-T M.3020的这个版本引用了UML版本1.5, 以便与相应的3GPP规范保持一致, 修订的ITU-T M.3020应引用最近版本的UML:

- OMG MOF元一元模型集成了UML 2.x作为元模型, UML 2.x得到了主流行业工具厂家的支持, 在UML 2.0之前, 没有综合的元-元模型, UML自身也没有标准化, MOF支持增加和建立其它新的通过OCL以精确方式定义的元模型, OCL是一种谓词演算语言。
- 行业(电信、政府和军队)和工具厂家都趋同于OMG MOF模型。
- MOF元一元模型的好处是它支持一系列能够用于定义对象模型、HCI关系、各种技术特有的实现的元模型, 允许以标准的方式进行模型之间的转换, 这在UML 1.5中是不可能实现的, 因为UML 1.5存在于分离的高层元模型之中。

#### IV.3 可见性

已经建议缺省的可见性应该是属性私有的、操作公有的, 以便促进数据封装并减少定义实现模型花费的时间和精力。

#### IV.4 类型定义

当记录一个新的、基于这个方法的规范时, 有必要规定参数和属性的类型。本建议书的当前版本没有正式的类型定义, 因此, 对于不同规范中的相同含意, 类型定义可能会不同并且不一致, 例如, 一个整数数组可能会定义成一个整数列表, 或者一个整数序列, 或者一个整数集合。

附件E定义了能够在概念模型中使用的类型。

## 附录 V

### 另外的UML使用例子

(本附录不是本建议书的组成部分)

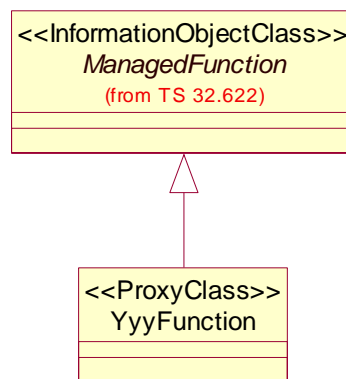
这个附录包含了关于附件C中描述的UML使用的另外例子。

#### V.1 代理类

##### V.1.1 第一个例子

这个例子显示了一个命名为YyyFunction的<<ProxyClass>>，它代表在UML图下方的注释中列出的所有IOC，在这个例子的语境中，所有列出的IOC均继承ManagedFunction IOC。

<<ProxyClass>>的使用消除了画多个UML<<InformationObjectClass>>方框的需要，即将那些类的名称罗列在UML图的注释中。



注 — YyyFunction <<ProxyClass>>代表 represents AsFunction、AucFunction、BgFunction等。

#### <<ProxyClass>>标记例子V.1

##### V.1.2 第二个例子

这个例子显示了一个命名为YyyFunction的<<ProxyClass>>，它代表直接位于UML图下方的注释中列出的所有IOC，在这个例子的语境中，所有列出的IOC均具有链接（内部的和和外部的）关系。

在与YyyFunction相关的X.Y小节下面，列出了InternalYyyFunction <<ProxyClass>>和ExternalYyyFunction <<ProxyClass>>代表的IOC的实际名称，例如，在适用于AsFunction的X.Y.1，增加了两个段落来列举与AsFunction链接的所有对等内部实体和外部实体，参见下面引用中例子，该例子使用AsFunction作为YyyFunction的一个例子。

在与YyyFunction相关的X.Y小节下面，列出了Link\_a\_z <<ProxyClass>>和ExternalLink\_a\_z <<ProxyClass>>代表的IOC的实现名称，例如，在适用于AsFunction的X.Y.1，增加了两个段落来列举Link\_a\_z和ExternalLink\_a\_z所代表的IOC，参见下面引用的文本，该文本使用AsFunction作为YyyFunction的一个例子。

"

## X.Y.1 AsFunction

### X.Y.1.1 定义

本IOC代表As功能，关于As的更多信息见[b-3GPP TS 23.002]。

链接的InternalYyyFunction <<ProxyClass>>代表SlsFunction、CscfFunction、HlrFunction

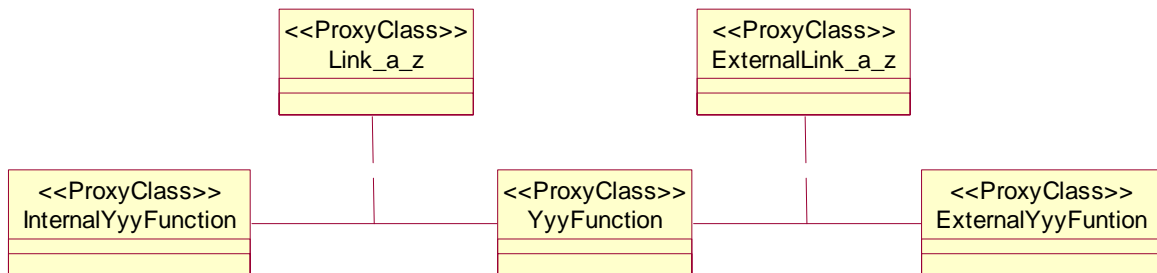
...

链接的ExternalYyyFunction <<ProxyClass>> 代表...

Link\_a\_z <<ProxyClass>> 代表Link\_As\_Scscf、Link\_Bgcf\_Scscf ...

ExternalLink\_a\_z <<ProxyClass>>代表...

"



注 — 'Yyy' of YyyFunction <<ProxyClass>>代表 AsFunction、 AucFunction等。

### <<ProxyClass>>标记例子V.2

## 附录 VI

### 关于需求编号的准则

(本附录不是本建议书的组成部分)

需求编号的格式如下：

**REQ-Label-Category-Number**

这里“Label”是建议书的缩写（或者其中一部分），对标签集不作限定，并且不需要标准化。本建议书定义了类型集。

一些问题如下：

- 在一个大规模的需求规格说明中如何构造标签？
- 如何处理需求的删减和增加？

下列准则被认为是有益的：

- 需求切勿被重编号，这种情况的唯一例外是规范的首次公布，但即便在这种情况下避免重编号也许会更好，因为草案形式的该规范有可能已经被采用了。
- 假定需求没有被重编号，不能期望需求在整个规格说明中被按顺序地编号。
- 标签能够用于将编号划分为逻辑部分，例如，建议“A\_B”样式用于标识“B”作为“A”的一个逻辑部分，然而，只要保留采用“-”分隔需求号码段的结构，就能使用其它的样式。
- 不建议使用后缀或者前缀标记，即在“号码”的前面或者在“号码”的后面增加某些东西，因为“号码”部分不是为了传送语义信息。
- 作为“A\_B”样式的备选方式，规范的作者可以选择分配一个号码范围给一组需求，这个方法应该可以得到允许。

## 参考资料

- [b-ITU-T M.1401] ITU-T M.1401 建议书（2006年），运营商电信网络之间互连指定的形式化。
- [b-ITU-T M.1403] ITU-T M.1403建议书（2007年），一般指令的形式化。
- [b-ITU-T M.1404] ITU-T M.1404建议书（2007年），运营商网络之间互连指令的形式化。
- [b-ITU-T Z.601] ITU-T Z.601建议书（2007年），软件系统的数据体系结构。
- [b-3GPP TS 23.002] 3GPP TS 23.002 (in force), *Network architecture*.
- [b-3GPP TS 32.101] 3GPP TS 32.101 V10.0.0 (2010), *Telecommunication management; Principles and high level requirements*.
- [b-3GPP TS 32.150] 3GPP TS 32.150 V10.2.0 (2011), *Telecommunication management; Integration Reference Point (IRP) Concept and definitions*.
- [b-3GPP TS 32.151] 3GPP TS 32.151 V10.1.0 (2010), *Telecommunication management; Integration Reference Point (IRP) Information Service (IS) template*.
- [b-3GPP TS 32.152] 3GPP TS 32.152 V10.0.0 (2010), *Telecommunication management; Integration Reference Point (IRP) Information Service (IS) Unified Modelling Language (UML) repertoire*.
- [b-3GPP TS 32.302] 3GPP TS 32.302 V10.0.0 (2010), *Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP); Information Service (IS)*.





## ITU-T 系列建议书

A系列	ITU-T工作的组织
D系列	一般资费原则
E系列	综合网络运行、电话业务、业务运行和人为因素
F系列	非话电信业务
G系列	传输系统和媒质、数字系统和网络
H系列	视听及多媒体系统
I系列	综合业务数字网
J系列	有线网络和电视、声音节目及其它多媒体信号的传输
K系列	干扰的防护
L系列	电缆和外部设备其它组件的结构、安装和保护
<b>M系列</b>	<b>电信管理，包括TMN和网络维护</b>
N系列	维护：国际声音节目和电视传输电路
O系列	测量设备的技术规范
P系列	电话传输质量、电话设施及本地线路网络
Q系列	交换和信令
R系列	电报传输
S系列	电报业务终端设备
T系列	远程信息处理业务的终端设备
U系列	电报交换
V系列	电话网上的数据通信
X系列	数据网、开放系统通信和安全性
Y系列	全球信息基础设施、互联网协议问题和下一代网络
Z系列	用于电信系统的语言和一般软件问题