

中华人民共和国通信行业标准

YD/T 1279-2003

900/1800MHz TDMA 数字蜂窝移动 通信网网络管理接口技术要求 ——基于 CORBA 的接口定义

900/1800MHz TDMA Digital Celler Mobile Telecommunication
network management interface technical specification
——Interface definition based on CORBA

2003-07-07 发布

2003-07-07 实施

中华人民共和国信息产业部 发布

目 次

前 言	III
1 范围	1
2 规范性引用文件	1
3 术语、定义和缩略语	2
3.1 术语和定义	2
3.2 缩略语	2
4 接口的位置	3
5 接口需求	4
5.1 高层需求	4
5.1.1 角色	4
5.1.2 高层用例图	4
5.2 详述需求	5
5.2.1 故障管理功能集	5
5.2.2 配置管理功能集	7
5.2.3 性能管理功能集	9
5.2.4 安全管理功能集	11
5.2.5 公共管理功能集	11
6 接口分析	13
6.1 接口分析模型	13
6.2 被管理对象的建模	13
6.2.1 被管理对象的命名	13
6.2.2 被管理对象的属性表示	14
6.2.3 被管理对象粒度的选择	15
6.2.4 通用对象访问接口	15
6.3 NMC 和 OMC 接口的交互控制	17
6.3.1 会话接口	17
6.3.2 OMC 接口	19
6.4 公共管理功能部分	21
6.4.1 通知服务	21
6.4.2 日志服务	23
6.4.3 版本管理	26
6.4.4 批量数据传送服务	27
6.4.5 心跳服务	28
6.5 安全管理	30
6.5.1 鉴权	30
6.5.2 超时	30
6.5.3 访问控制	32
6.6 故障管理	32
6.6.1 FMHandler_v2 接口概述	32

6.6.2	FMHandler_v2 接口的操作	33
6.6.3	故障管理的场景描述	38
6.7	配置管理	40
6.7.1	CMHandler_v2 接口概述	40
6.7.2	CMHandler_v2 接口的操作	41
6.7.3	配置管理的场景描述	47
6.8	性能管理	50
6.8.1	PMHandler_v2 接口概述	50
6.8.2	PMHandler_v2 接口的操作	50
6.8.3	JobFileInfolterator 接口	54
6.8.4	性能管理场景描述	56
7	接口设计	57
7.1	接口定义	57
7.1.1	通用模块	58
7.1.2	OMC 接口模块	60
7.1.3	Session 接口模块	61
7.1.4	FMHandler 接口模块	61
7.1.5	PMHandler 接口模块	65
7.1.6	CMHandler 接口模块	71
7.1.7	MOHandler 接口模块	75
7.1.8	cTelHeartbeat 接口模块	76
附录 A (资料性附录)	接口协议栈描述	79
附录 B (规范性附录)	通知格式定义	80
附录 C (规范性附录)	文件格式定义	94
附录 D (规范性附录)	接口信息模型的 IDL 定义	113
附录 E (规范性附录)	故障原因定义	154
附录 F (规范性附录)	测量任务定义	163

前 言

本标准是《900/1800MHz TDMA 数字蜂窝移动通信网网络管理接口技术要求》系列标准中的第 2 项标准。该系列标准由两个标准组成，其名称如下：

- 《900/1800MHz TDMA 数字蜂窝移动通信网网络管理接口技术要求——基于 CMIP 的接口定义》；
- 《900/1800MHz TDMA 数字蜂窝移动通信网网络管理接口技术要求——基于 CORBA 的接口定义》。

本标准是参考国际电信联盟——电信标准部 (ITU-T) X.72x 系列、X.73x 系列、X.90x 系列、Q.82x 系列、M.3100、M.3020 等相关建议以及欧洲电信标准协会 (ETSI) 的 GSM12.xx 系列标准、OMG 的有关规范，并结合我国具体情况编制而成的。

附录 A 是资料性附录，附录 B、C、D、E 和 F 是规范性附录。

本标准由中国通信标准化协会提出并归口。

本标准起草单位：中国移动通信集团公司

北京邮电大学

本标准主要起草人：王智立 施万中 林 巍 魏丽红 陈颖慧 徐海东 邱雪松 王 焯

900/1800MHz TDMA 数字蜂窝移动通信网网络管理接口 技术要求——基于 CORBA 的接口定义

1 范围

本标准规定了 900/1800MHz TDMA 数字蜂窝移动通信网 NMC-OMC 网络管理接口的管理功能需求、管理信息模型和通信协议。本标准定义的接口类型为基于 CORBA 的网络管理接口。

本标准适用于 900/1800MHz TDMA 数字蜂窝移动通信网的网络管理。

2 规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本标准，然而，鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本标准。

YD/T 871-1996 (1996)	电信管理网通用网络信息模型
ITU-T E.502 (1992)	数字通信交换话务测量需求
ITU-T M.3020 (1999)	TMN 接口描述方法学
ITU-T Q.544 (1988)	数字交换测量
ITU-T Q.722 (1992)	电话消息和信号的通用功能
ITU-T Q.811 (1992)	Q3 和 X 接口的低层协议轮廓
ITU-T Q.812 (1992)	Q3 和 X 接口的高层协议轮廓
ITU-T Q.821 (1993)	Q3 接口的告警监视功能
ITU-T Q.822 (1994)	Q3 接口性能管理的第 1, 2, 3 阶段描述
ITU-T Q.823 (1996)	话务量管理的第 2 和第 3 阶段功能规范
ITU-T X.721 (1992)	信息技术—开放系统互连—管理信息结构：管理信息定义
ITU-T X.730 (1992)	信息技术—开放系统互连—系统管理：对象管理功能
ITU-T X.731 (1992)	信息技术—开放系统互连—管理信息结构：状态管理功能
ITU-T X.733 (1992)	信息技术—开放系统互连—系统管理：告警上报功能
ITU-T X.734 (1992)	信息技术—开放系统互连—系统管理：事件报告管理功能
ITU-T X.735 (1992)	信息技术—开放系统互连—系统管理：日志控制功能
ITU-T X.90x	开放分布式处理
ETSI GSM 02.16 (V4.3.4, 1994)	数字蜂窝通信系统（第二阶段）：国际移动台设备标识
ETSI GSM 04.08 (V4.10.1, 1995)	数字蜂窝通信系统（第二阶段）：移动无线接口第三层规范
ETSI GSM 04.11 (V5.3.0, 2000)	数字蜂窝通信系统（第二阶段）：在无线接口上的点到点短消息业务支持
ETSI 3GPP TS 04.60 (V8.9.0, 2001)	数字蜂窝通信系统（第二阶段）：通用分组无线业务—移动台与基站系统接口—无线链路控制/媒质接入控制协议
ETSI GSM 08.08 (V4.12.1, 1998)	数字蜂窝通信系统（第二阶段）：移动交换中心与基站系统（MSC-BSS）间接口的第三层规范
ETSI GSM 09.02 (V5.3.0, 1996)	数字蜂窝通信系统（第二阶段）：MAP 规范
ETSI GSM 12.00 (V4.5.1, 1996)	数字蜂窝通信系统（第二阶段）：网络管理的目标和结构

ETSI GSM 12.04 (V4.3.1, 2000)	数字蜂窝通信系统 (第二阶段): GSM 性能管理和测量
ETSI GSM 12.11 (V6.3.0, 1999)	数字蜂窝通信系统 (第二阶段): 基站子系统的故障管理
ETSI GSM 12.20 (V4.2.1 1996)	数字蜂窝通信系统 (第二阶段): 基站子系统的管理信息
OMG 规范 (1998)	公共对象请求代理体系结构和规范修订版 2.2
OMG 规范 (1999)	一致性建模语言规范版本 1.3
OMG 规范 (1998)	公共对象服务规范
OMG 规范 (1998)	电信事件日志服务规范
OMG 规范 (1998)	通知服务规范
IETF RFC1779 (1995)	DN 的字符串表示法

3 术语、定义和缩略语

3.1 术语和定义

下列术语和定义适用于本标准。

3.1.1

管理对象 Managed Object

从管理角度对网络中被管资源的一种抽象。

3.1.2

CORBA 对象 CORBA Object

在 CORBA 环境下, 可识别的、被封装的实体, 它可以向用户提供一个或多个服务。CORBA 对象由它所支持的接口来标识。

3.1.3

PLMN 子网 PLMN Subnetwork

由一个 OMC 及其所管辖的所有设备、资源组成的网络, 是整个 PLMN 的一个子集。

3.1.4

约束 Obligation

在客户机/服务器关系中, 双方在接口上都应遵循的要求。

3.1.5

许可 Permission

在客户机/服务器关系中, 对客户端所应具有能力的描述。

3.2 缩略语

下列缩略语适用于本标准。

AUC	鉴权中心 (Authentication Center)
ASN.1	抽象语法标记 No.1 (Abstract Syntax Notation number one)
BER	基本编码规则 (Basic Encoding Rules)
BHCA	忙时试呼次数 (Busy Hour Call Attempts)
BSC	基站控制器 (Base Station Controller)
BSS	基站子系统 (Base Station Subsystem)
BTS	基站收发信机 (Base Transceiver Station)
CM	配置管理 (Configuration Management)
COM	公共管理 (Common Management)
CORBA	公共对象请求代理体系结构 (Common Object Request Broker Architecture)
DN	可识别名 (Distinguished Name)
EIR	设备身份寄存器 (Equipment Identity Register)
FM	故障管理 (Fault Management)

FTP	文件传输协议 (File Transfer Protocol)
GPRS	通用分组无线业务 (General Packet Radio Service)
GSM	全球移动通信系统 (Global System for Mobile communication)
HLR	归属位置寄存器 (Home Location Register)
HO, HDO	切换 (Handover)
IDL	接口定义语言 (Interface Definition Language)
IIOIP	互联网 ORB 间通信协议 (Internet Inter-ORB Protocol)
IMSI	国际移动用户标识 (International Mobile Subscriber Identifier)
IOR	可互操作的对象引用 (Interoperable Object Reference)
MO	管理对象 (Managed Object)
MOC	管理对象类 (Managed Object Class)
MSC	移动交换中心 (Mobile Switching Center)
MSRN	移动用户漫游号码 (Mobile Subscriber Roaming Number)
NE	网元 (Network Element)
NMC	网络管理中心 (Network Management Center)
NR	网络资源 (Network Resource)
ODP	开放分布式处理 (Open Distributed Processing)
OMC	操作维护中心 (Operation & Maintenance Center)
ORB	对象请求代理 (Object Request Broker)
PCU	分组控制单元 (Packet Control Unit)
PLMN	公用陆地移动网 (Public Land Mobile Network)
PM	性能管理 (Performance Management)
PSTN	公用交换电话网 (Public Switched Telephone Network)
QoS	服务质量 (Quality of Service)
RDN	相对可识别名 (Relative Distinguished Name)
SMSC	短消息交换中心 (Short Message Switching Center)
UML	一致性建模语言 (Unified Modeling Language)
VLR	拜访位置寄存器 (Visiting Location Register)

4 接口的位置

900/1800MHz TDMA 数字蜂窝移动通信网 NMC-OMC 网络管理接口在 900/1800MHz TDMA 数字蜂窝移动通信网的管理网中位置如图 1 所示。

在图 1 中, OMC 由各设备供应商提供, 用于管理其自身的设备。NMC 在 OMC 的协同工作下, 负责监视和控制整个 900/1800MHz TDMA 数字蜂窝移动通信网的运行状态, 对 900/1800MHz TDMA 数字蜂窝移动通信网进行统一管理。

NMC 和 OMC 之间通过 Q 接口相连, 本标准定义了基于 CORBA 的 NMC 和 OMC 间 Q 接口的管理功能需求和管理信息模型。本标准第 5 章对接口的管理功能需求进行了规定, 分别从公共管理、配置管理、故障管理和性能管理等方面进行详述。第 6 章按照管理需求给出详细的接口分析过程, 第 7 章是基于 CORBA/IDL 的具体接口设计。本标准的附录 A 对基于 CORBA 的通信协议栈进行了描述, 附录 B 定义了通知的内容和格式, 附录 C 定义批量数据传送文件的内容和格式, 附录 D 规定了总体对象模型和具体的配置管理信息, 附录 E 列举了本接口中使用的故障可能原因, 附录 F 给出具体的性能测量类型及参数。

NMC 和 OMC 之间的物理连接可以采用 DDN、X.25 和其他的物理媒质。

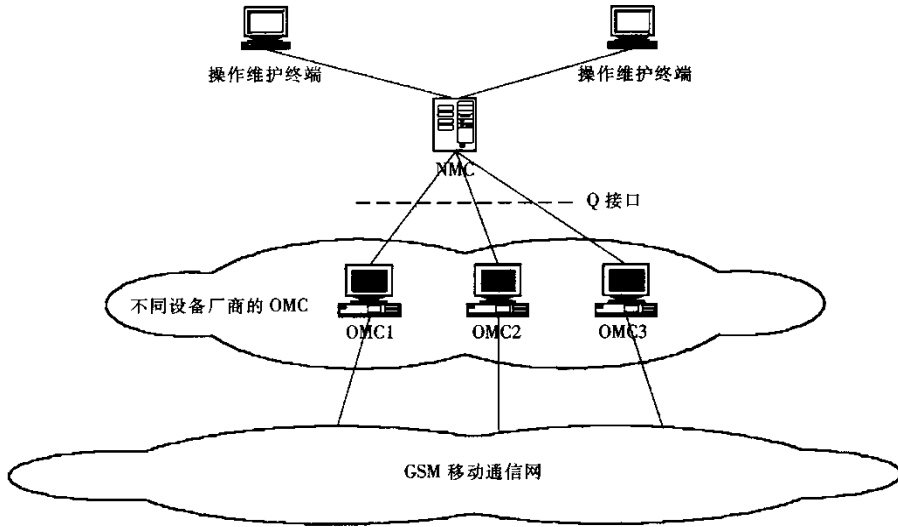


图 1 900/1800MHz TDMA 数字蜂窝移动通信网 NMC-OMC 网络管理接口在 900/1800MHz TDMA 数字蜂窝移动通信网管理网中的位置示意

5 接口需求

5.1 高层需求

本标准中给出的是与具体厂商无关、与版本无关、基于 CORBA 技术的 NMC 和 OMC 间的网管接口。通过该接口，NMC 可以使用如配置管理、性能管理和故障管理等不同管理功能域中的管理功能，动态获取网络的配置、性能和故障信息，从而实现网络层的管理功能。

5.1.1 角色

NMC：负责管理 OMC 的实体。NMC 可以管理由不同厂商所提供的 OMC。

OMC：负责 GSM 网络设备的操作、维护和管理实体。OMC 向上提供本标准所定义的接口，接受 NMC 的管理。

5.1.2 高层用例图

接口需求的高层管理功能描述如图 2 所示。

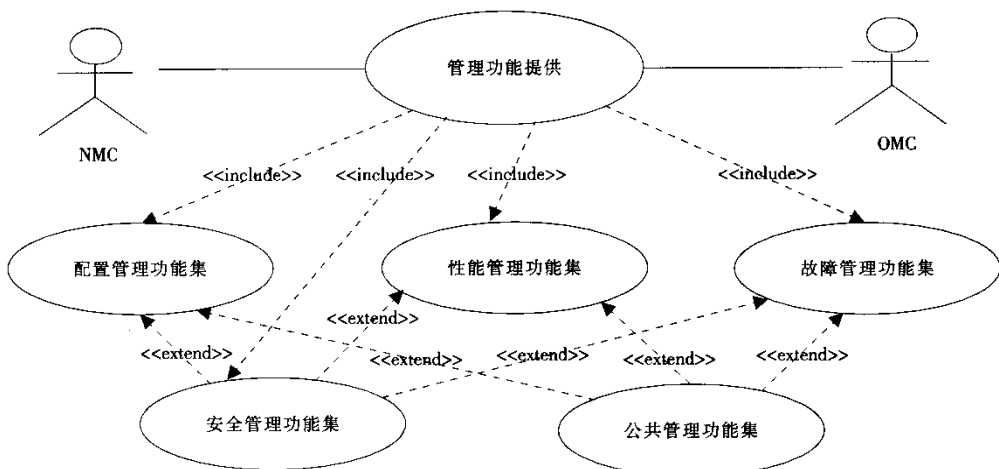


图 2 管理功能的高层用例图

管理功能包括配置管理功能集、性能管理功能集、故障管理功能集、安全管理功能集和公共管理功能集。同时，安全管理功能集和公共管理功能集对配置、性能和故障管理功能集提供支持。

5.2 详述需求

本节在两个层次上对接口需求进行了描述，首先根据不同的管理功能集描述需求，然后在每个功能集中，根据具体的管理功能项进行细化描述。

5.2.1 故障管理功能集

故障管理功能集支持 NMC 对 OMC 所管理网络中告警事件的监视，以及在系统初启或长时间通信中断后对 OMC 中告警信息的同步。它主要包括以下功能：

- OMC 向 NMC 及时准确地提供被管网元或其自身状态发生变化、异常时的告警信息；
- 保持 NMC 和 OMC 之间网元故障信息的同步；
- OMC 通过日志保存产生的告警信息以支持 NMC 可以动态获取历史告警信息。

5.2.1.1 角色

该功能与高层用例中的角色一致。

5.2.1.2 用例图

NMC 通过故障管理功能集与 OMC 进行交互，以获取故障的相关信息。故障管理功能集用例可进一步分解为订购告警通知、撤销告警通知订购、获取告警订购状态、同步告警、订购告警日志、撤销告警日志订购和查询告警日志记录 7 个功能项。如图 3 所示。

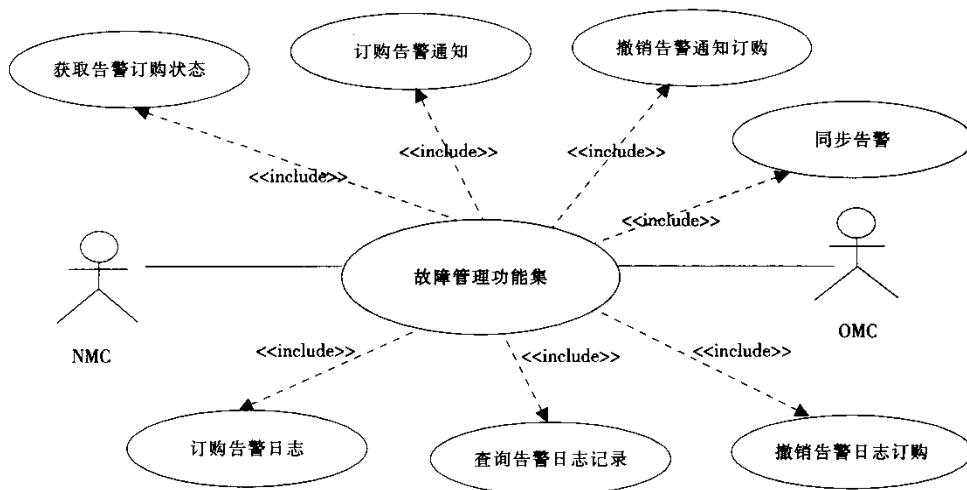


图 3 故障管理功能集用例图

5.2.1.3 故障管理功能项

5.2.1.3.1 订购告警通知

NMC 使用该功能向 OMC（告警通知的发送者）订购告警事件通知。

约束

- NMC 应给出接受告警通知的目的地；
- 如果订购失败，OMC 应将含有相应错误指示的信息返回给 NMC；
- 如果订购成功，OMC 应将惟一标识本次订购的标识符返回给 NMC。

许可

NMC 可以指定过滤条件来过滤告警通知。

5.2.1.3.2 撤销告警通知订购

NMC 使用该功能请求 OMC 撤销一个已存在的告警通知订购。

约束

NMC 必须给出订购的标识符，以确定将被撤销的告警订购；

如果 NMC 给出的订购标识符在 OMC 中不存在，OMC 应将含有相应错误指示的信息返回给 NMC。

5.2.1.3.3 同步告警

NMC 使用该功能请求 OMC 对告警信息进行同步，即返回 OMC 中的活跃告警信息。NMC 可以在任何时候使用告警同步功能以获取 OMC 中的所有活跃告警信息。

约束

NMC 应给出接受告警通知的目的地；

如果该操作失败，OMC 应将含有相应错误指示的信息返回给 NMC。

许可

NMC 可以指定过滤条件来过滤告警通知。

5.2.1.3.4 获取告警订购状态

NMC 使用该功能来获取存在于 OMC 中的某个告警订购的状态。

约束

NMC 应指定订购的标识符；

如果在 OMC 中不存在 NMC 所指定的告警订购，OMC 应将含有相应错误指示的信息返回给 NMC。

5.2.1.3.5 订购告警日志

NMC 使用该功能来订购告警的日志记录。

约束

如果订购不成功，OMC 应将含有相应错误指示的信息返回给 NMC；

如果订购成功，OMC 应将惟一的订购标识符返回给 NMC。

许可

NMC 可以指定过滤条件，要求 OMC 将符合过滤条件的告警通知存放到日志中；

NMC 可以指定日志的最大容量，以限定可容纳的最大日志记录数；

NMC 可以指定当日志达到最大容量时应采取的动作。动作可以是覆盖，即新的日志记录将覆盖最早生成的记录；也可以是停止，即 OMC 将停止告警日志记录的生成。

5.2.1.3.6 撤销告警日志订购

NMC 使用该功能请求 OMC 撤销一个已存在的告警日志订购。

约束

NMC 应指定订购的标识符；

如果 OMC 中不存在 NMC 所指定的日志订购，OMC 应将含有相应错误指示的信息返回给 NMC。

5.2.1.3.7 查询告警日志记录

NMC 使用该功能获取指定告警日志订购中的告警日志记录。OMC 接收到此命令后，将收集告警日志记录数据，并将结果数据以文件形式传送给 NMC。

约束

NMC 应指定告警日志订购标识符；

如果 NMC 给出的标识符所指示的告警日志订购不存在，OMC 应将含有相应错误指示的信息返回给 NMC；

当 OMC 完成数据收集，应向 NMC 发送含有结果文件信息和日志订购标识符的文件准备就绪通知；

如果在数据收集过程中发生错误，OMC 应向 NMC 发送含有错误级别的文件准备错误通知。如果错误级别为提示 (warning) 或次要 (minor)，OMC 应继续文件准备工作，否则 OMC 将停止数据收集。

许可

NMC 可以指定过滤条件来对告警日志记录进行过滤。

5.2.2 配置管理功能集

配置管理功能集包括以下功能：对网络中因配置信息改变而产生的事件进行实时监控、查询管理对象的状态和属性、对网络配置信息进行及时快速的更新。

5.2.2.1 角色

该功能与高层用例中的角色一致。

5.2.2.2 用例图

NMC 通过配置管理功能集与 OMC 进行交互以获取配置相关信息。配置管理功能集用例可进一步分解为订购配置通知、撤销配置通知订购、获取配置订购状态、同步配置数据、订购配置日志、撤销配置日志订购、查询配置日志记录、获取网络拓扑信息、查询管理对象属性值和获取所包含的管理对象 10 个功能项，如图 4 所示。

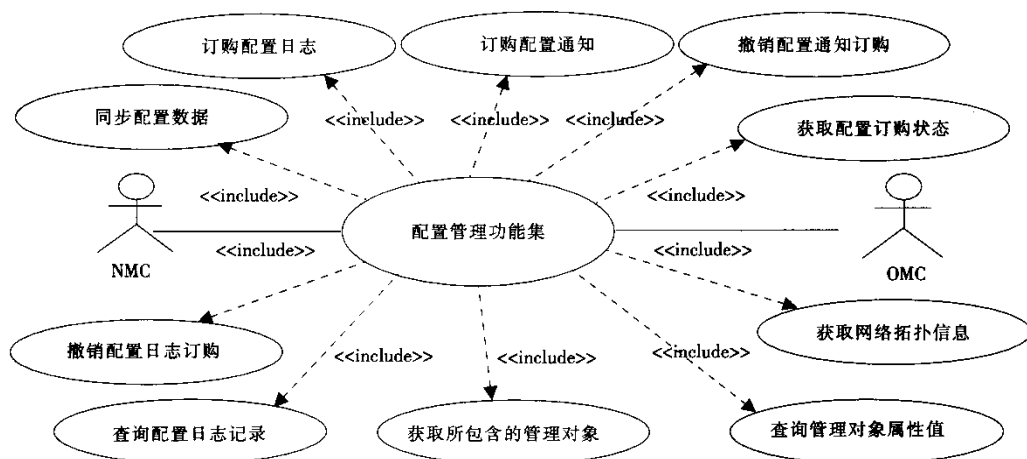


图 4 配置管理功能集的用例图

5.2.2.3 配置管理功能项

5.2.2.3.1 订购配置通知

NMC 使用该功能向 OMC（告警通知的发送者）订购配置事件通知。

约束

NMC 应给出接收配置通知的目的地；

如果订购失败，OMC 应将含有相应错误指示的信息返回给 NMC；

如果订购成功，OMC 应将本次订购的惟一标识符返回给 NMC。

许可

NMC 可以指定过滤条件来过滤配置事件。

5.2.2.3.2 撤销配置通知订购

NMC 使用该功能请求 OMC 撤销已存在的某个配置通知订购。

约束

NMC 必须给出订购的标识符，以确定所要撤销的配置通知订购；

如果 NMC 给出的订购标识符在 OMC 中不存在，OMC 应将含有相应错误指示的信息返回给 NMC。

5.2.2.3.3 获取配置订购状态

NMC 使用该功能来获取一个存在于 OMC 中的配置订购的状态。

约束

NMC 应指定订购的标识符；

如果 OMC 中不存在 NMC 所指定的配置订购，OMC 应将含有相应错误指示的信息返回给 NMC。

5.2.2.3.4 同步配置数据

NMC 使用该功能请求 OMC 进行配置数据的同步。NMC 可以在任何时候使用同步配置数据功能以获取 OMC 所管理网络的配置信息，从而保持 NMC 与 OMC 中存储的配置数据的一致性。OMC 接收到此命令后，将收集相应配置数据，并将结果数据以文件形式传送给 NMC。

约束

NMC 应指定所要进行配置数据同步的包含子树的基对象；

如果 OMC 中不存在 NMC 所指定的基对象，OMC 应将含有相应错误指示的信息返回给 NMC；

当 OMC 完成数据收集，应向 NMC 发送含有结果文件信息和同步配置数据操作标识符的文件准备就绪通知；

如果在数据收集过程中发生了错误，OMC 应向 NMC 发送含有错误级别的文件准备错误通知。如果错误级别为提示 (warning) 或次要 (minor)，OMC 应继续收集数据，否则 OMC 将停止数据收集；

如果操作成功，OMC 应将本次操作的惟一标识符返回给 NMC。

5.2.2.3.5 获取网络拓扑信息

NMC 使用该功能来获取 OMC 所管理的整个网络的拓扑信息 (反映包含关系)。OMC 接收到此命令后，将收集相应的配置数据，并将结果数据以文件形式传送给 NMC。

约束

如果操作成功，OMC 应返回该操作的惟一标识符；

当 OMC 完成数据收集，应向 NMC 发送含有结果文件信息和相应的获取网络拓扑操作标识符的文件准备就绪通知；

如果在数据收集过程中发生了错误，OMC 应向 NMC 发送含有错误级别的文件准备错误通知。如果错误级别为提示 (warning) 或次要 (minor)，OMC 应继续收集数据，否则 OMC 将停止数据收集。

5.2.2.3.6 查询管理对象属性值

NMC 使用该功能查询管理对象的属性值。

约束

NMC 应指定管理对象的标识符，对象标识符包括对象的类名和对象的可区分名 (DN)；

如果 NMC 所指定的对象不存在，OMC 应将含有相应错误指示的信息返回给 NMC；

如果 NMC 指定了属性值列表，但给出的一些属性在指定的对象中不存在，OMC 应将含有相应错误指示的信息返回给 NMC；

如果操作成功，NMC 应返回对象所有属性或所有指定的部分属性的属性值。

许可

NMC 可以明确指定所要获取的属性值的属性名。如果 NMC 没有指定特定的属性列表，则 OMC 将返回对象所有属性的属性值。

5.2.2.3.7 获取所包含的管理对象信息

NMC 使用该功能来获取指定对象下所包含对象的信息。

约束

NMC 应指定基对象的标识符，对象标识符包括对象的类名和对象的可区分名 (DN)。

NMC 应指定包含树的范围类型。类型可以是单层 (IndividualLevel)，即包含树中指定的第 N 层的所有对象；从基层到第 N 层 (BaseToNLevel)，即包含树中从基对象到指定的第 N 层的所有对象；基层 (BaseLevel)，即基对象本身；以及整棵子树 (WholeSubTree)，即整棵包含子树中的所有对象。

如果需要，NMC 应指定包含层次的深度，即第几层。

如果 NMC 指定的基对象不存在，OMC 应将含有相应错误指示的信息返回给 NMC。

如果 NMC 指定的包含层次超过了该基对象所确定的子树的最大深度且范围的类型为单层，OMC 应将含有相应错误指示的信息返回给 NMC。

如果 NMC 指定的包含层次超过了该基对象所确定的子树的最大深度且范围的类型为从基层到第 N 层或整棵子树，则由该基对象所确定的整棵子树上的对象信息都将返回。

如果操作成功，OMC 将返回所有符合条件的对象的信息。

5.2.2.3.8 订购配置日志

NMC 使用该功能来订购配置的日志记录。同时，OMC 开始将满足给定条件的配置事件记入日志中。

约束

如果订购不成功，OMC 应将含有相应错误指示的信息返回给 NMC；

如果订购成功，OMC 应将惟一的订购标识符返回给 NMC。

许可

NMC 可以指定过滤条件，要求 OMC 将符合过滤条件的配置通知存放至日志中；

NMC 可以指定日志的最大容量，以限定可容纳的最大日志记录数；

NMC 可以指定当日志达到最大容量时应采取的动作。动作可以是覆盖，即新的日志记录将覆盖最早生成的记录；也可以是停止，即 OMC 将停止告警日志记录的生成。

5.2.2.3.9 撤销配置日志订购

NMC 使用该功能请求 OMC 撤销一个已存在的配置日志订购。

约束

NMC 应指定订购的标识符；

如果 OMC 中并不存在 NMC 所指定的日志订购，OMC 应将含有相应错误指示的信息返回给 NMC。

5.2.2.3.10 查询配置日志记录

NMC 使用该功能获得指定配置订购中的配置日志记录。OMC 接收到此命令后，将收集配置日志记录数据，并将数据放入文件中。

约束

NMC 应指定日志订购标识符。

如果 NMC 给出的标识符所指示的日志订购并不存在，OMC 应将含有相应错误指示的信息返回给 NMC。

当 OMC 完成数据收集，应向 NMC 发送含有结果文件信息和日志订购标识符的通知。

如果在数据收集过程中发生错误，OMC 应向 NMC 发送含有错误级别的文件准备错误通知。如果错误级别为提示 (warning) 或次要 (minor)，OMC 应继续文件准备工作，否则 OMC 将停止数据收集。

许可

NMC 可以指定过滤条件来对配置记录进行过滤。

5.2.3 性能管理功能集

性能管理功能集支持 NMC 在不同的时间段，不同的时间间隔，根据不同的目的和不同的数据要求 (包括业务质量、网络规划、网络维护及性能下降等问题的监视和预警等)，获得相关的性能数据。

5.2.3.1 角色

该功能与高层用例中的角色一致。

5.2.3.2 用例

NMC 通过性能管理功能集同 OMC 进行交互以获取性能的相关信息。性能管理功能集用例可进一步分解为开启测量任务、加入测量任务、停止测量任务、获取当前测量任务信息和获取测量任务的结果文件信息 5 个用例，如图 5 所示。

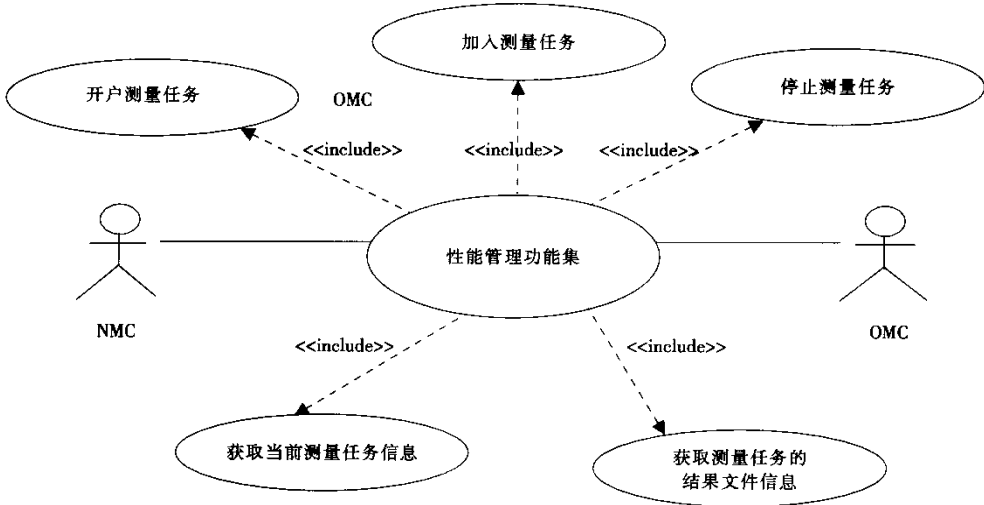


图 5 性能管理功能集的用例图

5.2.3.3 性能管理功能项

5.2.3.3.1 开启测量任务

NMC 使用该功能请求 OMC 对指定的性能数据进行采集，并将收集到的数据放入文件中。如果有多个 NMC 对 OMC 中同一类型的性能数据进行采集，OMC 可以只启动一个测量任务，并将结果同时送给几个 NMC。OMC 应支持的性能数据采集的最小采集周期粒度为 15min。

约束

- NMC 应指定测量结果发送的目的地；
- 如果操作成功，OMC 应将该测量任务的唯一标识符返回给 NMC；
- 如果操作失败，OMC 应将含有相应错误指示的信息返回给 NMC；
- 当 OMC 完成数据收集，OMC 应向 NMC 发送含有结果文件信息和测量任务标识符的通知；
- 如果在数据收集过程中发生错误，OMC 应向 NMC 发送含有错误级别的文件准备错误通知。如果错误级别为提示 (warning) 或次要 (minor)，OMC 应继续收集数据，否则 OMC 将停止数据收集。

许可

- NMC 可以选择对哪些对象及其中的性能属性进行测量；
- NMC 可以指定相邻两次数据收集的时间间隔；
- NMC 可以指定相邻两次 OMC 向 NMC 进行测量数据上报的时间间隔；
- NMC 可以指定测量任务的开启时间；
- NMC 可以指定测量任务的结束时间。当指定的结束时间到达后，该测量任务将自动被中止，这与在 NMC 调用停止测量任务操作所带来的结果是一样的。

5.2.3.3.2 停止测量任务

NMC 使用该功能请求 OMC 中止对某个测量任务的数据收集。

约束

- NMC 应指定测量任务的标识符；
- 如果在 OMC 中不存在 NMC 所指定的测量任务，OMC 应将含有相应错误指示的信息返回给 NMC；
- 如果 NMC 未订购该测量任务 (开启或加入)，OMC 应将含有相应错误指示的信息返回给 NMC；
- 如果操作成功，OMC 应将该测量任务的当前状态返回，以指示任务是停止了还是由于其他 NMC 依然使用此测量任务，此测量任务仍在运行。

5.2.3.3.3 加入测量任务

NMC 使用该功能来加入一个已被其他 NMC 开启的测量任务。

约束

NMC 应指定测量任务的标识符；

如果 NMC 成功加入了该测量任务，OMC 应返回本次订购操作的标识符；

如果在 OMC 中不存在 NMC 指定的测量任务，OMC 应将含有相应错误指示的信息返回给 NMC。

许可

NMC 可以指定加入测量任务的时间；

NMC 可以指定停止订购该测量任务的时间。

5.2.3.3.4 获取当前测量任务信息

NMC 使用该功能请求 OMC 返回当前 OMC 中运行的所有测量任务的信息（包括该 NMC 自己开启的或其他 NMC 开启的测量任务，可以是这在运行的或简要运行的测量任务）。

约束

如果动作成功，OMC 向 NMC 返回测量任务的信息，包括测量任务标识符、测量信息和测量任务当前状态。

5.2.3.3.5 获取测量任务的结果文件信息

NMC 使用该功能获取指定的测量任务的结果文件信息，包括文件的基本信息和文件的访问信息。

约束

只有订购了该测量任务的 NMC 有权限调用该操作；

NMC 应指定测量任务的标识符和对给测量任务订购的标识符；

如果指定的测量任务目前还没有开始运行，OMC 应将含有相应错误指示的信息返回给 NMC；

如果操作成功，OMC 将向 NMC 返回测量任务的结果文件信息，包括文件名、文件大小、文件创建时间、文件预计的删除时间、文件压缩类型、文件路径、用户登录访问名和访问密码。

5.2.4 安全管理功能集

5.2.4.1 角色

该功能与高层用例中的角色一致。

5.2.4.2 安全管理功能项

由于 NMC 和 OMC 之间的信息交互是基于公用网的，因此必须有一定的安全控制功能以确保系统不会受到蓄意的非法侵入。

约束

当 NMC 开始同 OMC 进行交互时，OMC 必须检查该 NMC 是否合法；

每一次 NMC 向 OMC 发起请求时，OMC 必须对该 NMC 进行身份认证；

如果发起请求的 NMC 身份不合法或鉴权验证失败，OMC 应将含有相应错误指示的信息返回给 NMC。

5.2.5 公共管理功能集

本节主要描述了支持故障、配置和性能管理功能集的一些公共管理功能。

5.2.5.1 角色

该功能集与高层用例中的角色一致。

5.2.5.2 通知服务功能

OMC 依靠通知服务功能向 NMC 报告实时的管理信息，例如 OMC 或网元中产生的异步或突发事件。需要提供一种组织良好的机制来支持该功能需求。

约束

OMC 应确保通知的正确发送；

NMC 应可以自由地向 OMC 订购通知或撤销订购。

许可

- 允许 OMC 向多个目的地发送通知；
- NMC 可以同时向多个 OMC 订购通知；
- NMC 可以指定过滤条件来对上报的通知进行过滤；
- 在通知的发送中可以指定优先级。

5.2.5.3 日志服务功能

对 OMC 而言，将 OMC 或网元中发生的异常事件信息进行保存是很有必要的。NMC 通过日志服务功能，可以随时从 OMC 的日志中对这些信息进行查询。

约束

- 支持 NMC 订购或撤销订购日志；
- 支持 NMC 对日志记录进行查询或删除。

许可

- NMC 可以指定过滤条件来对要计入日志的事件进行过滤；
- NMC 可以修改过滤条件；
- NMC 应能控制日志的执行状态，如挂起日志和恢复日志。

5.2.5.4 批量数据传输功能

该接口应能够支持在 NMC 和 OMC 之间的大量数据传输，包括配置信息、告警事件和性能测量数据。批量数据传输主要发生在以下情况：

NMC 开启一个测量任务，当测量任务的结果准备好后，NMC 使用批量数据传输机制从 OMC 中获取测量结果；

NMC 请求 OMC 返回该 OMC 所管理的网络的配置信息和拓扑信息；

NMC 请求获得 OMC 中贮存的配置和告警日志记录；

当 OMC 系统初启或 OMC 与 NMC 之间连接发生了较长时间的中断，NMC 使用批量数据传输来获取 OMC 中的告警信息；

NMC 要从 OMC 中获取大量的日志信息。

约束

- 需要使用一种标准的文件传输协议；
- 传输的文件的格式应标准化，且应当是通用的工业标准。

许可

- 如果 NMC 和 OMC 双方同意，则在传输中可以使用约定的备份传输协议；
- 如果 NMC 和 OMC 双方同意，则在传输中可以使用约定的文件压缩方法。

5.2.5.4.1 版本管理

由于在网络中新网元和新管理功能的引入，管理接口也需要进行升级，因此在定义的接口中要求有一种明确的、定义完整的接口版本升级机制来支持管理接口的演进。¹

约束

- 支持接口平滑升级；
- 如果 NMC 和 OMC 均能支持多个接口版本，则需采用一种有效的方法来获得双方均能支持的版本。

5.2.5.5 心跳服务功能

NMC 和 OMC 实体之间在应用层的连接有时可能中断，此时如果 OMC 向 NMC 发送通知，则可能会造成数据的丢失。为了能尽早地发现连接中断的情况，NMC 应定期从 OMC 中收到心跳通知。当 NMC 连续几次丢失这样的心跳通知时，则可以发现连接中断。这样，NMC 和 OMC 之间的连接就得到了监视。

注：1. 在本标准中，版本指的是标准中定义的接口版本，而不是在 OMC 中的接口实现的版本。

约束

OMC 应周期性地发送心跳通知给 NMC。

许可

NMC 可以获取和修改 OMC 中心跳通知的上报周期。

6 接口分析

6.1 接口分析模型

在本节中，首先给出分析阶段的总体分析模型，其中的每个具体接口以及其间的关系，将在后面的章节中给出详细的描述。

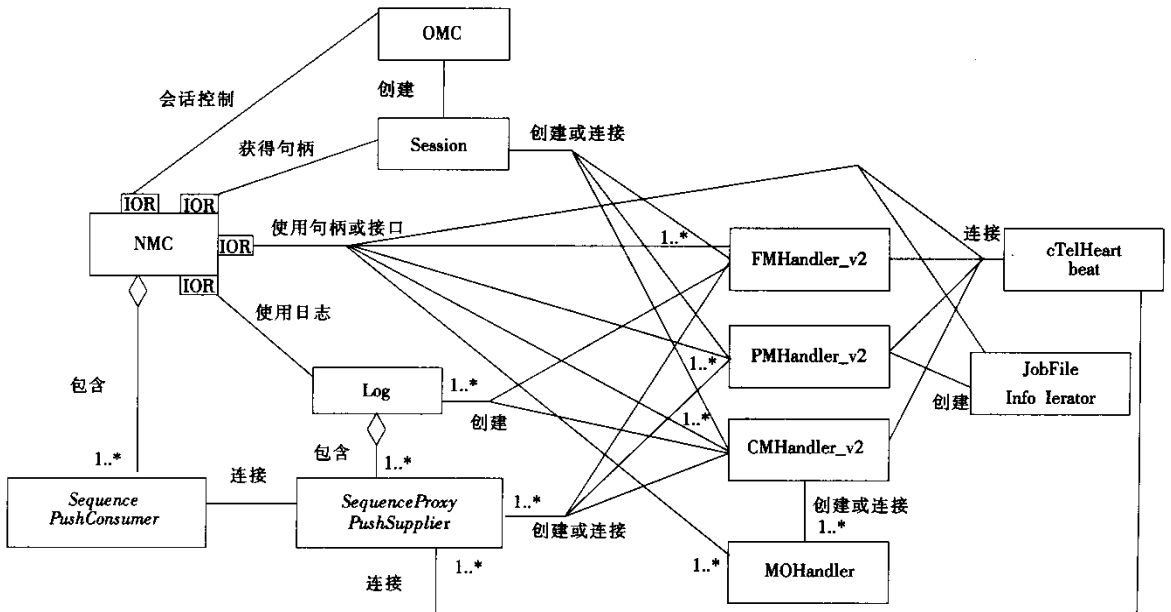


图 6 总体分析模型的 UML 类图

图 6 给出了总体的分析模型图，其中浅灰色背景接口图不在本标准的范围之内，仅仅是用来明确阐释接口之间的关系。由斜体文字所表示的接口在本标准的范围之内，但这是从其他接口规范中导入的。其中，*SequenceProxyPushSupplier* 和 *SequencePushConsumer* 接口是从 OMC 的 CORBA 通知服务规范中导入的，而 *log* 接口是 OMC 的电信日志服务规范中导入的。

接口之间可能的关联关系通过关联名称明确指定，例如 *SequencePushConsumer* 和 *SequencePushSupplier* 之间的“连接”关联。关联中的数量关系也由表示关联的直线的末端表示出来。

图 6 中的 NMC 代表网络管理系统，OMC 接口是本接口规范中的入口，NMC 在执行所有的管理操作之前，首先要访问该接口，进行鉴权并开启一个会话 (Session)。会话接口用于交互控制，提供了安全控制机制和访问具体管理域句柄的方法。

FMHandler、PMHandler、CMHandler 分别是故障管理、性能管理和配置管理的句柄接口，按照不同的管理域提供相应的管理功能。而 MOHandler 接口是访问具体的管理对象实例的通用接口，用于获取具体的管理对象实例的属性等信息。

6.2 被管理对象的建模

6.2.1 被管理对象的命名

在 OMC 中的每一个管理对象均需要有一个唯一的标识符来标识。在本标准中，使用可区分名 (DN)

作为对象的标识。

6.2.1.1 命名约定

本标准仿照以 GDMO 作为描述语言的信息模型中的命名约定，采用可区分名 (DN) 作为对象的标识符。ITU-T 建议 X.500 中定义了 DN 和 RDN 的概念，其中 DN 是由 RDN 组成的序列，而 RDN 则是由属性名和属性值构成的二元组。

在本标准中，采用字符串作为 DN 的类型，完整的定义将在下一节中给出。根据一个对象的 DN，可以在包容树中得到包含该对象的对象的 DN (如果存在)。DN 中所承载的惟一关系就是包容关系。在每个 RDN 中，所选择的属性名应是对象的命名属性。

6.2.1.2 字符串形式 DN 的 BNF 范式表示

以下给出了字符串形式 DN 的 BNF 范式表示：

```
DistinguishedName ::= RDNSequence
<spaced-separator> ::= <optional-space> <separator> <optional-space>
<separator> ::= “,”
<optional-space> ::= ( <CR> ) * ( “ ” ) *
RDNSequence ::= RDNSequence <spaced-separator> RDNSequence | RelativeDistinguishedName
RelativeDistinguishedName ::= NamingAttributeNameAndValue
NamingAttributeNameAndValue ::= NamingAttributeName ‘=’ NamingAttributeValue
NamingAttributeName ::= <one or more StringChar, which form the naming attribute>
NamingAttributeValue ::= <one or more StringChar>
<special> ::= “,” | “=” | <CR> | <LF> | “+” | “<” | “>” | “#” | “;” | “/”
StringChar ::= any character except <special> or “\” or “”
```

示例 1:

omcFunction 对象的 DN 可以表示为:

“networkId=1, managedElementId=1, omcFunctionId=1”。

示例 2:

hlrFunction 对象的 DN 可以为:

“networkId=1, managedElementId=2, hlrFunctionId=1”。

6.2.2 被管理对象的属性表示

在管理对象中，属性以一个名值对 (Name-Value Pair) 来表示。由于属性名的不同，属性值可以取不同的类型，因此本标准使用 IDL 中的 any 类型来表示不同类型的属性值。对象所有属性值的列表在本标准中定义为名值对的序列。

基于 IDL 的定义如下所示:

```
typedef string AttrNameType;
typedef any AttrValueType;
struct AttrInfoType
{
    AttrNameType name;
    AttrValueType value;
};
typedef sequence< AttrInfoType> AttrInfoListType;
typedef sequence< AttrNameType> AttrNameListType;
```

6.2.3 被管理对象粒度的选择

OMC 中允许 NMC 管理和访问的实体都应该模型化为一个被管理对象。在 CORBA 环境中，管理对象应该可以通过 CORBA 接口直接或间接地访问到。

管理对象建模的方法，从访问粒度的角度来讲主要可以分为以下几种：

1) 细粒度方式。即每个管理对象都是一个 CORBA 对象实例，该对象在 CORBA 环境中可以直接被管理系统访问。

2) 按管理对象类划分的粗粒度方式。每个管理对象都不是 CORBA 对象，而每个管理对象类至少有一个 CORBA 对象与之相对应，管理系统可以通过该 CORBA 对象来访问此管理对象类的实例。

3) 按照管理域划分的粗粒度方式。所有的管理对象都可以通过一个 CORBA 接口来访问，并且按照管理域的不同来定义相应的 CORBA 接口，从而提供不同的管理功能。

以上方法各有利弊。本标准从简单易行和产品成熟度的角度考虑，采用的是第 3 种建模方式。

6.2.4 通用对象访问接口

在本标准中，定义了一个通用的接口 MOHandler（管理对象句柄）作为管理对象实例的通用访问接口。

6.2.4.1 MOHandler 的类图

MOHandler 接口包含了 OMC 所支持的可以对管理对象进行操作的通用方法。OMC 中属于不同对象类的管理对象均可通过 MOHandler 通用接口或其派生接口来访问。该接口的类图如图 7 所示。

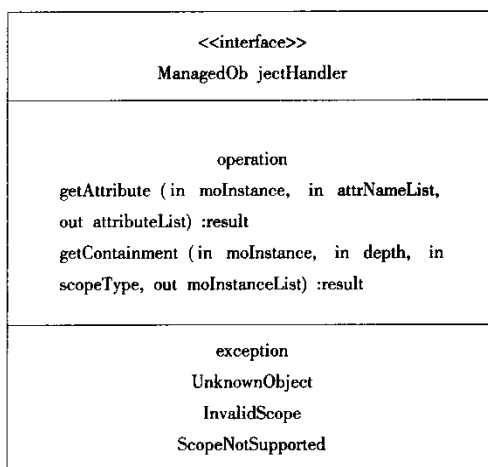


图 7 MOHandler 接口的 UML 类图表示

6.2.4.2 MOHandler 的操作

6.2.4.2.1 getAttribute 操作

行为

“该操作支持 NMC 查询指定对象的属性值。如果在 NMC 给出的属性名列表中含有该对象中不存在的属性或不允许读取的属性，则在输出参数属性列表中不返回该属性，且动作的返回值应为失败 (failure)，以指示并没有返回所有要求的属性值。”

输入参数

moInstance: ManagedObjectType

输入参数 moInstance 标识了一个特定的管理对象。其类型为一个由类名和管理对象可区分名组成的结构。

attrNameList: AttrNameListType

输入参数 attrNameList 指定要取值的属性列表。如果该序列的长度为 0，则该对象所有属性值都应返回。

输出参数:

attributeList: AttrButeListType

输出参数 attributeList 返回所要求的属性信息列表, 包括属性名和属性值。

前提条件

命题 1: “指定的对象在 OMC 中存在”。

命题 2: “对象实例包含所指定的属性且均是可以查询的”。

后继条件

如果操作成功, 则 result= “success”;

如果前提条件中的命题 2 不成立, 则 result= “failure”。

返回值

result: eResultType

返回值 result 表明操作的成功与否。

异常

如果前提条件命题 1 不成立, 则应 OMC 应给出异常 “UnknowObject”。

6.2.4.2.2 getContainment 操作

行为

“NMC 使用该操作来获取某个指定对象所包含对象的信息。”

输入参数

moInstance: ManagedObjectType

输入参数 moInstance 标识了一个特定的管理对象。其类型为一个由类名和管理对象可区分名组成的结构。

depth: unsigned long

输入参数 depth 给出所要求的包含树的深度。

scope: eScopeType

输入参数 scope 给出包含树的范围, 值可以为 BaseLevel (基层)、IndividualLevel (单层)、BaseToNLevel (基层到第 N 层对象的) 和 WholeSubTree (整棵子树)。如果值为 WholeSubTree 或 BaseLevel, 则参数 depth 的值将被忽略。(在现阶段, 厂商可以只支持 BaseLevel, N=1 的 IndividualLevel 和 N=1 的 BaseToNLevel)。

输出参数

moInstanceList: ManagedObjectListType

输出参数 moInstanceList 返回指定对象所包含对象的信息列表, 每一序列元素中都包括对象的类名和对象的区分名。

前提条件

命题 1: “指定的管理对象在 OMC 中存在”。

命题 2: “参数 depth 的值超过了包含树的最大深度且参数 scope 的值为 IndividualLevel”。

命题 3: “参数 depth 的值超过了包含树的最大深度且参数 scope 的值为 BaseToNLevel”。

命题 4: “厂商目前支持指定的定界类型”。

后继条件

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”;

如果前提条件中的命题 3 成立, 则 result= “failure”。

返回值

result: eResultType

返回值 result 表明操作的成功与否。

异常

如果前提条件命题 1 不成立，则应给出异常 “UnknownObject”；

如果前提条件命题 2 成立，则应给出异常 “InvalidScope”；

如果前提条件命题 4 不成立，则应给出异常 “ScopeNotSupported”。

6.3 NMC 和 OMC 接口的交互控制

6.3.1 会话接口

6.3.1.1 会话的概念

在 CORBA 应用中，公布和存贮对象引用的通用方法是把对象引用放在命名服务中。由于在多厂商环境中，直接访问别的厂商的命名服务是比较危险的，因此，在 NMC 和 OMC 接口之间不宜建立公共的命名服务。引入会话概念则可以解决这个问题，同时会话概念还可以处理服务的初始化和权限验证。

当 NMC 开始执行管理功能时，首先必须通过权限验证才能开启一个会话。在会话期间，NMC 可以执行配置、故障和性能等管理域中的管理功能。当 NMC 结束管理活动时，NMC 应关闭这次会话。

OMC 可以规定所能支持的最大会话数，以避免 OMC 由于性能原因造成系统瘫痪，最大会话数的具体数目由 OMC 根据自己的系统决定。

6.3.1.2 Session 接口概述

NMC 通过 Session（会话）接口获取某个特定管理域的句柄。会话接口是 NMC 调用不同管理域句柄所提供操作的入口点。图 8 是 Session 接口的类图表示。

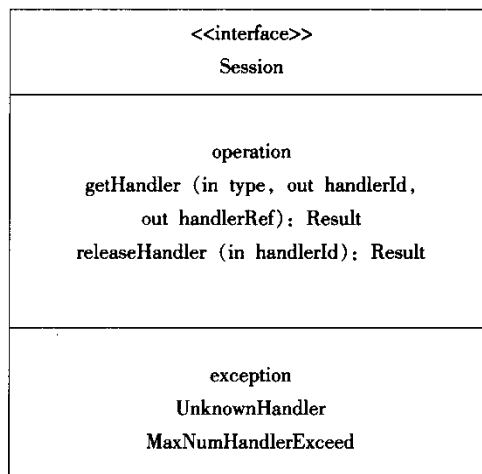


图 8 会话接口的 UML 类图表示

6.3.1.3 Session 接口的操作

6.3.1.3.1 getHandler 操作

行为

“NMC 使用该操作来获取特定域（配置、性能或故障）的句柄。”

输入参数

handlerType: eHandlerType

输入参数 handlerType 表示 NMC 要获取的管理域对象的句柄的类型，可以是配置、故障或性能。

输出参数

handlerId: HandlerIdType

输出参数 handlerId 在 OMC 中惟一标识返回的句柄。

handlerRef: CORBA.: Object

输出参数 handlerRef 返回句柄的 IOR。NMC 可以通过返回的 IOR 执行调用。

前提条件

命题 1: “在 OMC 中的句柄数小于 OMC 所能支持的最大句柄数”。

后继条件

如果操作成功, 则 result= “success”;
 如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 表明操作的成功与否。

异常

如果前提条件命题 1 不成立, 则应给出异常 “MaxNumHandlerExceed”。

6.3.1.3.2 releaseHandler 操作

行为

“NMC 使用 releaseHandler 操作释放一个句柄。当该句柄为配置管理的句柄 (CMHandler) 时, 所有的该对象所用到的 MOHandler 都应该被隐式释放, 以保证数据的一致性。”

输入参数

handlerId: HandlerIdType

输入参数 handlerId 标识所要释放的句柄。

前提条件

命题 1: “NMC 指定的句柄在 OMC 中存在”。

后继条件

如果操作成功, 则 result= “success”;
 如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 表明操作的成功与否。

异常

如果前提条件命题 1 不成立, 则应给出异常 “UnknownHandler”。

6.3.1.4 会话管理的场景描述

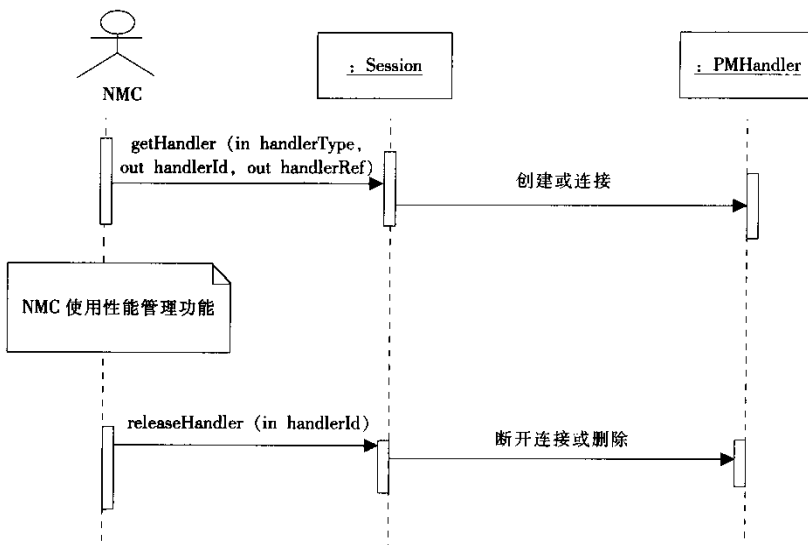


图 9 会话管理的 UML 序列图表示

图9描述了NMC和OMC通过会话接口进行一次交互的过程。当会话打开后，NMC可以获得某个特定域的句柄从而实现不同的管理功能。在交互完成后，NMC应调用releaseHandler操作来显式地释放会话接口对象。

6.3.2 OMC接口

6.3.2.1 OMC接口概述

为实现NMC和OMC之间的初始交互，OMC应向NMC提供一个入口访问对象，NMC使用该对象建立与OMC的连接，并从而调用相关操作来执行特定的管理功能。

OMC负责将自己向NMC注册，即向NMC提供一个入口对象，该入口对象是OMC中的一个永久对象(persistent)。NMC可以通过NMC的命名服务或文件存储等方式来获得OMC入口对象的IOR(不在本标准的范围之内)。

该接口作为OMC系统的入口对象使用，负责NMC-OMC间的接口交互控制。该接口提供的操作如图10所示。从性能方面考虑，OMC可以限制它所能同时支持的会话数目。具体的数目由OMC实现厂商自己决定，不在本标准的范围之内。

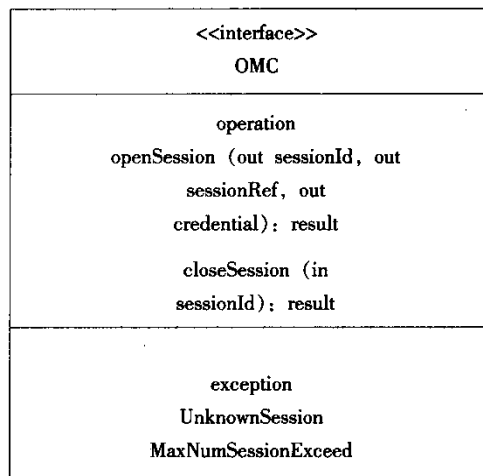


图10 OMC接口的UML序列图表示

6.3.2.2 OMC接口的操作

6.3.2.2.1 openSession操作

行为

“NMC使用该操作开启一次会话。”

输出参数

sessionId: SessionIdType

输出参数sessionId标识所开启的会话。

sessionRef: Session

输出参数sessionRef是所开启的会话接口对象的IOR。

前提条件

命题1：“当前OMC中激活的会话数小于OMC所能支持的最大会话数”。

后继条件

如果操作成功，则result=“success”；

如果操作失败，则result=“failure”。

返回值

result: eResultType

返回值 result 标识动作的成功与否。

异常

如果前提条件命题 1 不成立，则应给出异常 “MaxNumSessionExceed”。

注：OMC 能同时支持的最大的会话数目由厂商自己决定，但应该在系统实现完成时候给出明确的声明。

6.3.2.2 closeSession 操作

行为

“NMC 使用该操作关闭 OMC 中一个已经开启的会话。在操作成功后，该 NMC 通过该会话获取的所有句柄对象都将无效。”

输入参数

sessionId: SessionIdType

输入参数 sessionId 标识要关闭的会话。

前提条件

命题 1: “指定的会话在 OMC 中存在”。

后继条件

如果操作成功，则 result= “success”;

如果操作失败，则 result= “failure”。

返回值

result: eResultType

返回值 result 标识了动作的成功与否。

异常

如果前提条件命题 1 不成立，则应给出异常 “UnknownSession”。

6.3.2.3 OMC 管理的场景描述

图 11 描述了 NMC 通过 OMC 接口创建和释放会话的交互实例。

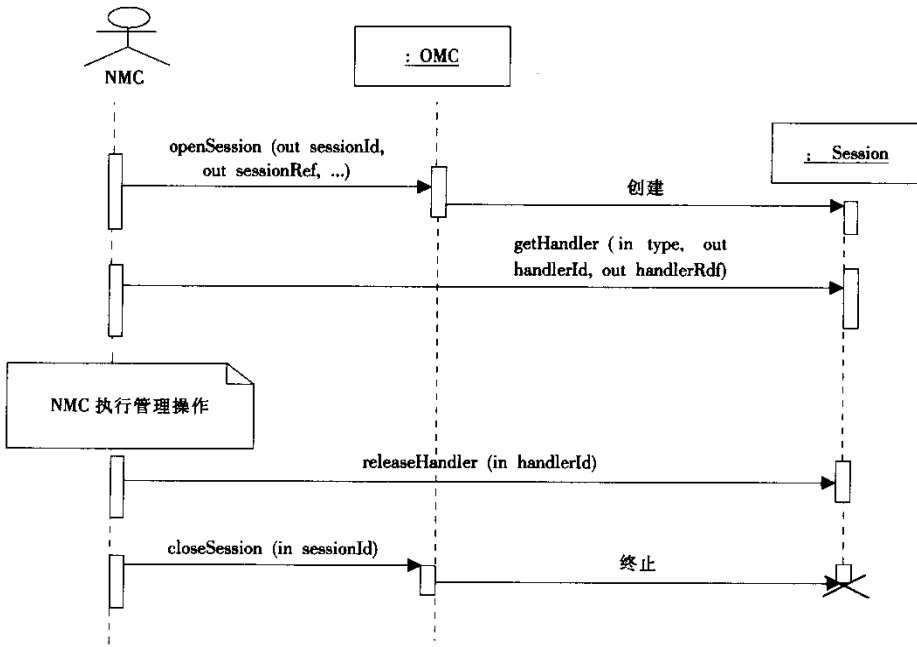


图 11 OMC 接口对会话接口的管理 UML 序列图

6.4 公共管理功能部分

6.4.1 通知服务

6.4.1.1 事件管理模型

本标准中采用 CORBA 的事件服务和通知服务来实现接口的事件功能集的需求。在推 (Push) 和拉 (Pull) 两种模型中, 本标准选择了推模型的实现方法。

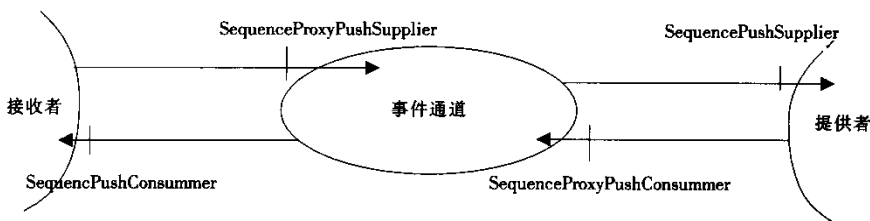


图 12 事件通道的推模型

在推模型中, 事件提供者将事件数据主动发送给事件通道; 事件通道再将事件数据转发给事件接收者。图 12 描述了事件提供者和事件通道, 事件通道和事件接收者之间推模型的通信机制。

为建立事件通信, 事件提供者 (OMC) 和事件接收者 (NMC) 都应向事件通道注册。事件提供者和接收者的注册都分为两步。首先, 产生事件的应用程序从事件通道获得一个序列代理推接收者 (Sequence Proxy Push Consumer) 对象; 然后将自己提供的序列推提供者 (Sequence Push Supplier) 与该序列代理推接收者对象相连。

与此类似, 接收事件的应用进程首先从事件通道获得一个序列代理推提供者 (Sequence Proxy Push Supplier) 对象, 然后应用进程将自己实现的一个序列推接收者 (Sequence Push Consumer) 与该序列代理推提供者 (Sequence Proxy Push Supplier) 相连。

本标准中采用的是结构化的事件消息, 它是一个事先定义好的, 允许多种事件类型映射的数据结构。图 13 给出了结构化事件的基本结构。

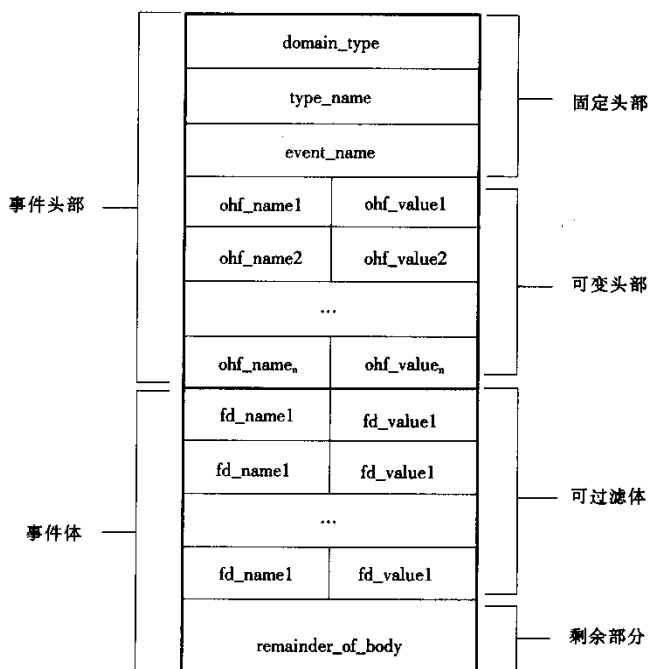


图 13 结构化事件的基本结构

结构化事件头部的固定域包括标识不同领域的域名 (domain_type)，在给定领域中惟一标识事件类型的类型名 (type_name)，以及全局惟一标识该通知的事件名 (event_name) 这 3 个参数。

在本标准中，通知服务的业务质量 (QoS) 应该支持 “EventReliability=Persistent 且 ConnectionReliability=Persistent”，即事件被发送到事件通道后，在终止期限内事件通道必须确保已注册的事件接收者收到该事件。如果通道和事件接收者之间的通信因为某种原因而中断，通道应持久保存该事件直至时间超过了所设置的终止期限；或者等两者之间的通信恢复后，将事件发送给所有注册的事件接收者。另外，若由于故障而重启，事件通道应负责自动重建与所有在故障发生时已建立连接的事件接收者和事件发送者之间的连接。

一般来讲，服务顺序的策略应按照事件到达事件通道的次序来进行，如先进先出 (FIFO)，也可以作为可选来支持优先级的服务策略，即将事件按照优先级的次序存入缓冲区，从而使高优先级的事件可以比低优先级的事件早发送。

事件接收者可以指定过滤条件来选择所要接收的通知。每个过滤条件由以下两部分组成：

—数据结构的序列。每一个结构体指定一种事件类型，如 NOTIFY_NEW_ALARM、NOTIFY_OBJECT-CREATION 等。

—字符串形式的布尔表达式。表达式中的语法遵照一定的约束文法。目前所使用的文法为 CORBA 通知服务中的默认过滤约束语言。布尔表达式的子项应根据事件信息来指定。

6.4.1.2 所支持的事件

NMC-OMC 接口中的事件服务支持 OMC 向 NMC 发送事件和告警，其类型包括：

- 管理对象创建通知；
- 管理对象删除通知；
- 对象状态改变通知；
- 对象属性值改变通知；
- 故障告警，包括新产生的告警，状态变化的告警和被清除的告警；
- 批量数据传输准备就绪通知，包括性能数据文件、配置同步和拓扑信息文件以及告警信息文件；
- 批量数据传输准备错误通知；
- 当前活跃告警上报通知；
- 心跳通知；
- 请求配置信息同步通知。

6.4.1.2.1 管理对象创建通知

触发条件

当 OMC 中新创建一个管理对象。

汇报的事件

OMC 中创建了一管理对象。

6.4.1.2.2 管理对象删除通知

触发条件

当 OMC 中一个管理对象被删除。

汇报的事件

OMC 中删除了一管理对象。

6.4.1.2.3 状态改变通知

触发条件

当 OMC 中一个管理对象的状态值改变。

汇报的事件

OMC 中一个管理对象的状态值发生了改变。

6.4.1.2.4 属性值改变通知

触发条件

当 OMC 中一个管理对象的属性值改变。

汇报的事件

OMC 中一个管理对象的属性值发生了改变。

6.4.1.2.5 故障管理中的告警通知

触发条件

当 OMC 中的告警状态发生改变，包括新的告警、告警变化和告警清除。

汇报的事件

OMC 中的告警状态发生了改变，包括新的告警生成、告警变化和告警清除。

6.4.1.2.6 批量数据传输准备就绪通知

触发条件

OMC 完成相应的数据收集。

汇报的事件

通知 NMC 可以开始获取要传输的数据文件。

6.4.1.2.7 批量数据传输准备错误通知

触发条件

当 OMC 在准备数据文件时发生了错误。

汇报的事件

OMC 在数据收集集中发生了错误。

6.4.1.2.8 告警总结汇报通知

触发条件

当 NMC 调用告警同步功能，并且 OMC 完成了相应的告警同步信息的数据采集。

汇报的事件

OMC 中所有的活跃告警信息。

6.4.1.2.9 心跳通知

触发点

当 cTelHeartbeat 对象在每个心跳上报周期结束时。

汇报的事件

告诉 NMC 相应通道是否工作正常。

6.4.1.2.10 请求配置信息同步通知

触发点

当 OMC 引入铃行动资源和修改铃很多网络资源时，可能会产生非常多的对象创建、对象删除、对象属性值/状态改变通知。此时，OMC 可以指发送该通知告诉 NMC 该 OMC 所管理的部分网络资源需要进行同步。如果 OMC 发送了该通知，则与该根对象以及其下面所有的管理对象相关的对象创建、对象删除或属性值、状态改变通知都应该被抑止，不必再发送给 NMC。

汇报的事件

告诉 NMC 需要进行配置信息同步的管理对象实例树上的根对象。

6.4.2 日志服务

OMC 负责收集相关事件信息，并进行永久性的保存。NMC 可以在任何时候请求 OMC 返回所需的事件信息。

本标准采用了基于 OMG 的电信事件日志服务 (OMG Telecom Log Service) 的日志管理模型来实现日志服务功能，该规范可能被 CORBA3.0 规范所采纳。²

注：2. 目前，日志服务对 OMC 厂商而言是可选的。

6.4.2.1 日志管理模型

由于日志负责收集事件信息，因此日志是一个事件的接收者。同时日志也可以转发事件，因此它又是一个事件的提供者。因此，日志服务是从事件/通知服务派生而来的。日志可以看作是事件/通知通道，Log 接口继承了事件/通知通道。提供事件的应用进程通过和日志中的事件/通知通道相连就可以将事件发送给日志对象。日志管理模型如图 14 所示。

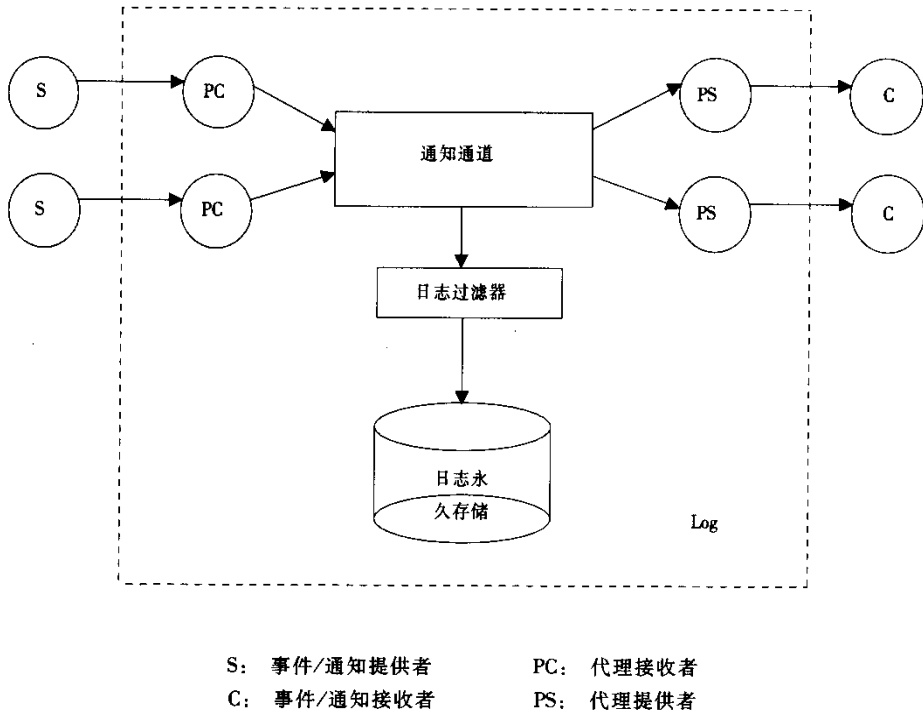
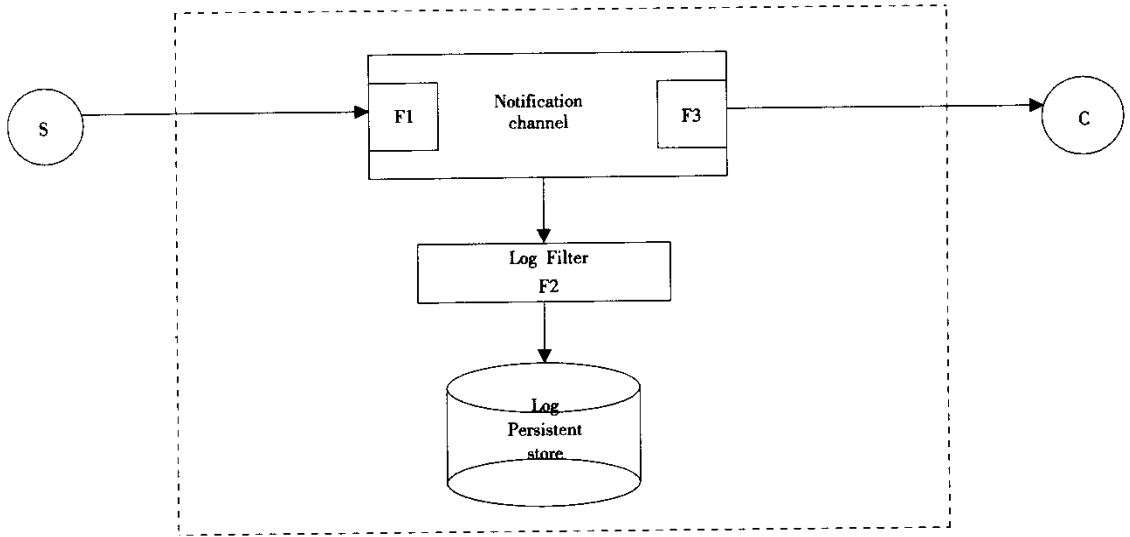


图 14 日志管理模型

在该模型中，事件提供者、日志和事件接收者（包括其他日志对象）构成了事件通信的主体。和 CORBA 中的事件/通知服务类似，要形成事件通信也需要两步注册。事件提供者向日志对象所支持的事件通道发送事件，日志对象根据所设置的过滤条件决定是否生成一条日志记录。同时，事件也可以转发给其他日志对象。事件日志的处理全过程可以分为以下主要步骤（参照图 15）：

- 事件提供者通过日志所支持的事件/通知通道接口和日志对象相连；
- 日志的客户（需要日志记录信息的实体，如本标准中的 NMC）可以指定通知通道提供者侧的过滤条件（F1）；
- 日志的客户可以指定日志对象的存储过滤条件（F2）；
- 日志的客户可以指定通知通道的接收者侧的过滤条件（F3）；
- 事件提供者使用推（push）操作将事件发给日志对象；
- 对输入事件使用通知过滤条件（F1），以过滤那些不满足条件的事件；
- 日志过滤条件 F2 将对每一个通过 F1 的事件进行过滤；
- 对于每个通过 F1 和 F2 的事件，将产生一条相应的日志记录。



S: 事件/通知提供者 F1, F3: 通知过滤器
 F2: 日志过滤器 C: 事件/通知接收者

图 15 事件的日志存储和转发

在本标准中，日志本身所产生的事件不在日志的存储范围之内。存储在日志中的事件可以是任意类型的事件，只要日志的存储库支持其存储即可。每一个日志记录中包括存储时间、日志记录标识（由日志对象所分配的唯一的数字）和所要存储的事件信息。客户可以根据日志记录的时间、日志记录标识以及时间内容来对日志记录进行相应的分类查询。

表 1 日志记录的基本结构

名称	描述
Log record Id	必选。惟一标识该日志记录
Logging time	必选。指日志记录的产生时间
Event information	必选。日志记录中所包含的事件信息

日志对象支持客户对日志记录的操作，包括查询和删除。在查询操作中，客户可以根据日志记录信息（包括日志产生时间、日志记录标识和所存储的事件信息）来指定过滤条件。

6.4.2.2 Log 接口

6.4.2.2.1 Log 接口概述

在本标准中，所有存储的事件是结构化的，因此规范采用了 OMG 日志服务中所定义的 NotifyLog 接口来实现日志服务功能。NotifyLog 接口的详细定义在 OMG 电信事件日志服务中。

6.4.2.2.2 日志管理的场景描述

图 16 描述了一个日志管理的场景。NMC 向 OMC 调用日志订购操作。OMC 随即创建日志对象来向 NMC 提供日志服务。NMC 可以调用日志对象的操作，例如设置过滤条件、设置日志对象的属性等。NMC 也可以调用日志对象中与日志记录相关的操作，例如检索日志记录、删除日志记录等。如果 NMC 不再需要日志功能，可以调用 OMC 系统的 unsubscribeLog 操作来撤销日志订购，OMC 将根据情况决定是否停止该日志服务。

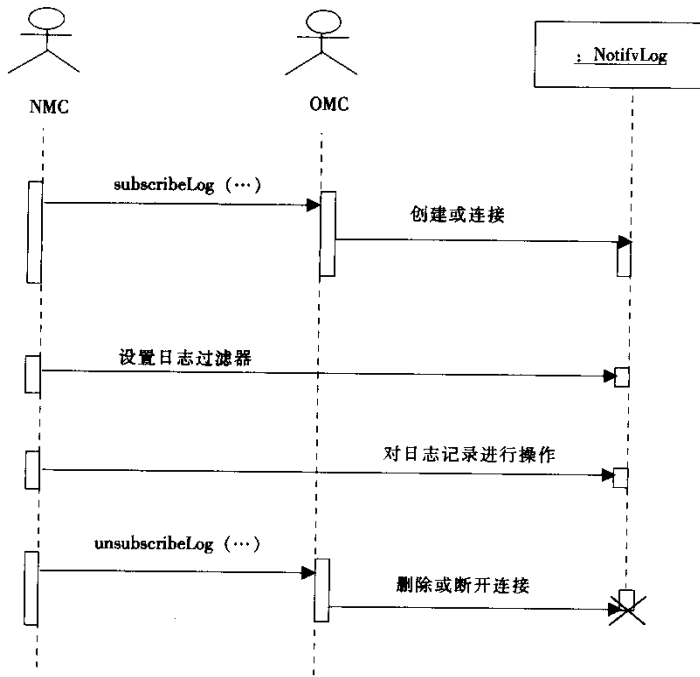


图 16 日志管理的 UML 序列图

6.4.3 版本管理

为解决接口版本的升级问题，本标准采用以下两种方案：首先，NMC-OMC 接口支持版本号的指示；其次，在 NMC-OMC 接口上支持版本协商。

6.4.3.1 版本号的指示

每个 IDL 接口都应有版本名与其相关联以指示接口中所使用的版本号，这可以通过在 IDL 接口名后加版本号的后缀来实现。例如，CMHandler 接口名为 CMHandler_V1，这意味着使用的是版本号为 V1 的 CMHandler 接口。

由于 NMC 和 OMC 的接口软件的升级是相互独立的，因而在一般情况下要求双方都能支持多个接口版本。接口的版本号仅仅表明接口所使用的是哪个版本，而并无其他含义，例如，版本号为 V2 的 MOHandler 接口并不意味着它一定是从版本号为 V1 的 MOHandler 接口继承而来。如果厂商可以支持这种语义，则不属于本标准的范围之内。

6.4.3.2 版本协商

为支持版本协商，Session 接口的 getHandler 和 CMHandler 接口的 getMOHandler 操作需要增加一个附加的输入参数 versionNum (版本号) 和输出参数 supportVersionList (所支持的版本列表)。NMC 使用参数 versionNum 来指示所要接口的版本号。如果 OMC 支持 NMC 需要的版本，操作的返回结果为 success，输出参数 supportVersionList 将返回 NMC 在 versionNum 中所输入的版本号。如果 OMC 不能支持 NMC 需要的版本，该操作的返回结果为 failure，输出参数 supportVersionList 将返回 OMC 所能支持的该接口的所有版本的版本号，NMC 可以从返回的版本号列表中选择满足需要的版本的版本号来作为输入参数 versionNum 的值，然后重新调用会话接口的 getHandler 或 CMHandler 接口的 getMOHandler 操作。如果 NMC 不能使用返回版本中的任何一项，则 NMC 和 OMC 将无法实现交互。版本协商的完整场景描述如图 17 所示。

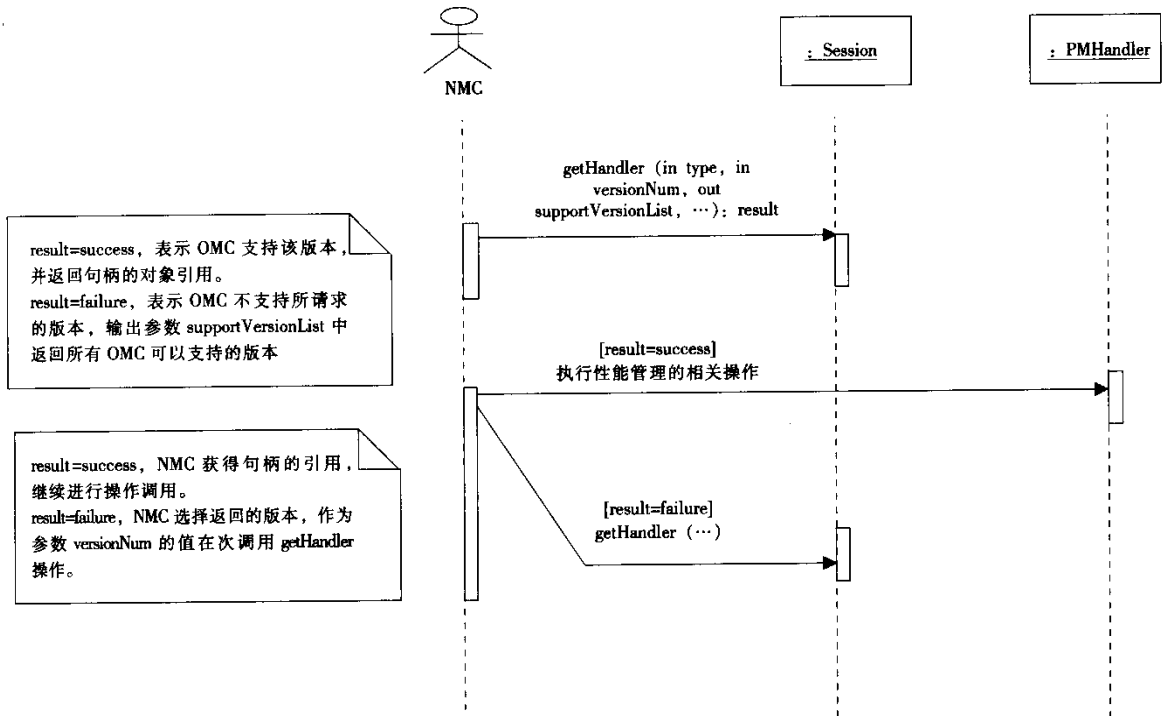


图 17 版本协商的 UML 序列图

当 NMC 调用 `getHandler` 操作时, NMC 指定所需要的接口版本的版本号。如果 OMC 可以支持, 则操作返回结果为 `success`, 并返回相应句柄的 IOR; 否则结果为 `failure`, 输出参数 `supportVersionList` 将返回 OMC 所能支持的该接口所有版本的版本号。NMC 可以从中间选择一个来获取相应管理域句柄的对象引用。

6.4.4 批量数据传送服务

6.4.4.1 服务实现

在 NMC 和 OMC 之间传送数据的方式有下面的两种:

- CORBA 总线;
- 文件传输协议 (FTP)。

当要传送的数据量相当大时, 应用 CORBA 总线方式效率很低, 本标准中采用了 FTP 来实现接口上批量数据的传送, 其机制描述如下: 当 OMC 准备好结果文件后, 向 NMC 发出一个文件准备就绪通知, NMC 在收到该通知后采用 FTP 服务来获取结果文件; 如果 OMC 在文件准备过程中发生了错误, OMC 应向 NMC 发出一个含有错误级别的批量数据文件准备错误通知, 当错误级别为提示 (`warning`) 或次要 (`minor`) 时, OMC 应继续收集数据, 但若错误级别为主要 (`major`)、紧急 (`critical`) 或不确定 (`indetermined`), OMC 将停止数据收集。

这些场景将在相应的接口中进行描述。

将具体的错误原因映射为相关的级别可以参见本标准附录 E 中的定义。

6.4.4.2 传送文件的格式

根据文件的内容, 批量数据传输的文件可分为性能数据文件、配置信息文件、拓扑信息文件、告警同步数据文件和日志记录文件五类。

所传输的文件采用 ASN.1/BER 格式。目前不采用压缩工具对文件进行压缩。

各种数据文件的具体格式在附录 C 中给出。

6.4.4.3 传送文件的文件名

为了能描述所要传送文件的必要信息，本标准中对所传送文件的文件名定义了一套标准的命名规则。具体的命名格式如下：

```
File_name: = <OMC_name> <separator1> <file_type> <separator2> <generation_time>
[<separator2> <suffix>]
```

其中：

- 参数“OMC_name”标识了产生该文件的 OMC 的名称，其中不允许含有字符“_”和“.”。
- 参数“separator1”为字符“_”。
- 参数“file_type”标识了所传输的文件的类型，可以是“PM”、“CM”、“TOP”、“FM”、“CMLOG”或者“FMLOG”。
- 参数“separator2”为“.”。
- 参数“generation_time”的格式为“MMDDYYYYHHmmss”，其中 2 位的 MM 代表月份，2 位的 DD 代表天，4 位的 YYYY 代表文件产生的年份，2 位的 HH 代表小时，2 位的 mm 代表分钟，2 位的 ss 代表秒。例如，一个在 1998 年 11 月 10 日上午 8 时生成的文件的名字中参数“generation time”应当是“11101998080000”。
- 参数“suffix”用于一个结果存放在几个文件的情况。其顺序应当反映文件生成的先后顺序。参数的格式为数字型。例如，一个测量任务返回两个结果文件，首先生成的文件的参数“suffix”的值为 1，另一个则为 2。

6.4.5 心跳服务

6.4.5.1 心跳服务描述

心跳服务用于检查被管系统中通知通道的运行状况，以及在管理系统和被管系统之间的通信网的状况。它周期性地向关心该状况的管理系统发送一个小的通知，其中包含发送的系统名以及所经过的通道的标识。在配置完该服务后，管理系统可以在其所关心的通知通道上设置过滤条件以接收心跳通知，以使这种网络监视功能能够正常运转。因为这些心跳通知与其他资源所产生的通知所经过的是同样的通道、软件和网络，因此可以周期性地检查这些资源的运行状况。

心跳服务中含有名为 systemLabel 的属性，以及读取和设置心跳周期的操作。systemLabel 是一个用户提供的标识符，用于管理系统区分不同的厂商发来的心跳通知。

心跳通知周期性地向每一个与之相关联的通知通道发送一个心跳通知。OMC 中的每个事件通道都必须与一个，并且是惟一的一个 cTelHeartbeat 对象实例相关联，而每个 cTelHeartbeat 对象实例则可以与一个或多个通知通道相关联，甚至可以是被管的 OMC 系统中的所有通道。心跳通知不是由管理对象发送的，而且要发给所有的通知通道。在每个心跳周期结束的时候，心跳服务向每个与之相关联的通道发送一个通知。发送给每个通道的通知中含有该通道的通道标识符。

心跳的周期可以由 periodSet 操作来设置，该操作的输入参数值就是要设置的周期，即心跳服务在两个相邻的通知之间所等待的时间（单位：s）。对心跳周期的更新则会使心跳服务立即发送一个含有新周期值的心跳通知，并开始以新的周期来工作。如果将周期值设置为 0，则该服务只发送一个周期值为 0 的心跳通知，以后就不再发送（直到该周期值被重新设置）。

每个通知包括属性系统标签的值、所通过的通道的标识符、当前的周期值以及时间戳。

OMC 系统应满足下面的需求：

a) 被管理系统至少要实例化一个 cTelHeartbeat 的对象实例，该对象可以由管理系统通过 CMHandler_v2、FMHandler_v2 和 PMHandler_v2 接口对象通过 getHeartbeat 操作来获得。

b) 更新心跳通知将使该服务向所有与其相关联的通道发送一个带有新的周期值的心跳通知，并开始以新的周期来计时工作。将周期值设置为 0，则该服务只发送一个周期值为 0 的通知，然后就停止发送（直到该值被重新设置）。

c) 在每个周期中，心跳服务必须向所有其管理的通知通道发送一次心跳通知，除非其周期被修改。在两个相邻的心跳通知之间的时间间隔不应超过周期值的两倍。

d) 当 OMC 中的每个通知通道初始化时，它应与唯一的一个 cTelHeartbeat 实例相关联。至于如何将通知通道映射到合适的 cTelHeartbeat 实例则是 OMC 系统的内部问题。

6.4.5.2 cTelHeartbeat 接口

6.4.5.2.1 cTelHeartbeat 接口概述

cTelHeartbeat 接口实现了上节中介绍的心跳服务的功能。图 18 为 cTelHeartbeat 接口的类图。

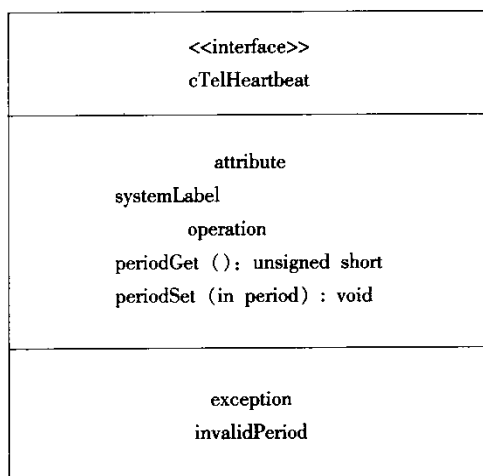


图 18 cTelHeartbeat 接口的 UML 类图

6.4.5.2.2 cTelHeartbeat 接口的属性和操作

6.4.5.2.2.1 systemLabel 属性

行为

“该属性指明了 OMC 系统的用户友好名。NMC 使用该属性的读操作来获取 OMC 系统的用户友好名，NMC 也可以通过该属性的写操作来修改其属性值。该值将出现在心跳通知中，用于 NMC 区分不同的 OMC 系统所发出的心跳通知。”

6.4.5.2.2.2 periodGet 操作

行为

“NMC 用该属性读操作来获取 cTelHeartbeat 对象的心跳周期。”

输入参数

无。

输出参数

无。

返回值

result: eResultType

返回指 result 表示操作是成功还是失败。

6.4.5.2.2.3 periodSet 操作

行为

“NMC 使用该操作来改变 cTelHeartbeat 对象的心跳周期。”

输入参数

period: unsigned short

输入参数 period 指明要设置的新的心跳周期。

输出参数

无。

前提条件

命题 1：“周期值的设置是合理的，如为非零值则不应低于 OMC 所能支持的心跳周期的下限”。

后继条件

当该操作执行成功后，若周期值为非零值，cTelHeartbeat 对象实例应立即向其所管理的通知通道发送一个含有新周期值的心跳通知，并且此后将按照新的周期值来发送通知；如果将周期值设置为 0，则 cTelHeartbeat 对象实例只立即发送一个周期值为 0 的心跳通知，以后就不再发送（直到周期值被重新设置）。

返回值

无。

异常

如果前提条件命题 1 不成立，则给出异常 “InvalidPeriod”。

6.5 安全管理

为实现 NMC 和 OMC 之间交互的安全，本标准中定义了鉴权、超时和访问控制 3 种机制。

6.5.1 鉴权

鉴权是第一级的安全检查，在 NMC 开启会话时执行。为实现鉴权机制，需对 OMC 接口做一定的修改，在 openSession 操作中增加用户名 (user) 和口令 (password) 两个输入参数。（其中 password 是否加密不在本标准的范围中。如果 NMC 和 OMC 均同意采用一种或多种加密算法，则可依据所协商的算法对所传递的 password 参数进行加密。）用户信息和密码信息应在 OMC 进行注册，由此 OMC 可以确定某个用户是否合法。如果输入的密码不正确，OMC 将向 NMC 抛出相应的异常。

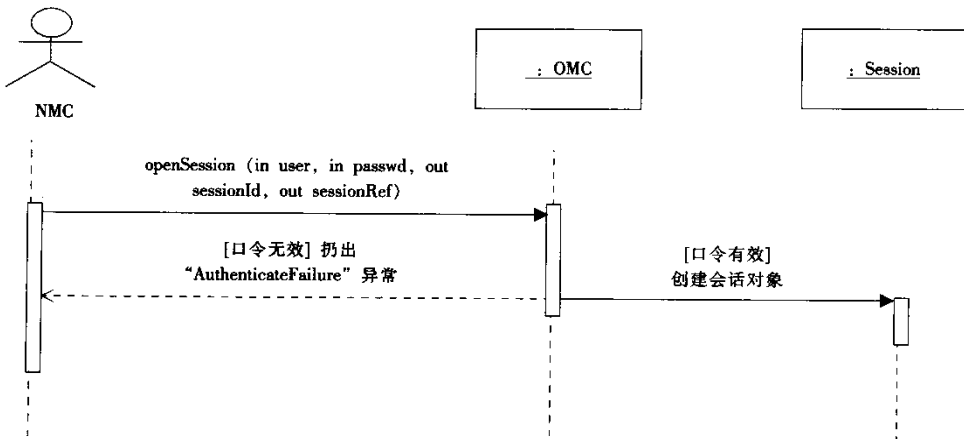


图 19 鉴权检查的 UML 序列图

图 19 描述了一个完整的鉴权过程。NMC 在调用 openSession 操作时必须提供用户名和密码信息作为输入参数（用户名和密码信息 NMC 和 OMC 双方都知道）。OMC 判断用户名和密码是否正确。如果鉴权失败，OMC 将向 NMC 给出相应的“鉴权失败”（“AuthenticateFailure”）异常。

6.5.2 超时

从性能角度考虑，OMC 不可能支持无限多的会话和句柄对象的数目。由此在本标准中引入了会话和句柄接口的超时机制，以提高 OMC 系统的性能。

当一个由 NMC 开启的会话处于闲置状态的时间超过了预先规定的最大时间，则超时事件就会发生，

相应的会话接口对象将被自动关闭；同时对于该 NMC 而言，通过被关闭的会话获得的所有句柄也都将无效。（但是在 OMC 中，有可能这些句柄由于同时被其他 NMC 的会话使用，而不一定是真正释放。）当 NMC 使用从已超时的会话所提供的身份验证信息调用句柄中的操作时，OMC 将给出 InvalidCredential 异常，其中的 credentialInvalidReason 参数值为 SESSION_TIMEOUT。如果 NMC 使用已超时的会话接口调用操作时，ORB 将检测到对象不存在并抛出相应的对象不存在系统异常。

如果句柄对象处于闲置状态的时间超过了预先规定的最大时间，该句柄对象将被自动释放，但不影响其他句柄对象的生命周期。

为实现超时机制，会话接口和所有的句柄接口中需增加一个属性 timeout 来存储超时时间（最大的闲置时间）。NMC 在调用 openSession 或 getHandler 操作时，应当指定相应接口对象的超时时间。超时时间的值作为 CORBA::Context 对象的一部分值进行传送。值的类型是字符串形式的无符号长整型值（单位：min）。如果操作参数中没有指明超时时间，则缺省的超时时间为 60min。如果 NMC 所给的超时时间过长，则 OMC 可以给出 InvalidTimeout 异常，指明出错原因，同时拒绝发来的 openSession 或 getHandler 操作。最大缺省超时时间的取值不在本标准的范围内。

每次当 NMC 向会话接口对象调用操作时，该会话接口的闲置时间将重置为 0；每次 NMC 向句柄对象调用操作时，句柄和会话的闲置时间也都将重置为 0。如果一个句柄被多个会话所共享，对于 NMC 而言应当是透明的，即 NMC 可以认为该句柄对象被该会话所独占。因此，对于不同的会话，句柄的闲置时间应当根据不同会话而分别进行计时和设置。

图 20 描述了一个超时场景的例子。当会话的闲置时间超过了 openSession 操作中上下文对象中所给出的最大超时时间时，会话接口对象将被自动释放，但是对于 PMHandler 接口，由于被其他会话共享而仍然存在。如果 NMC 继续用已超时会话中提供的身份验证信息调用 PMHandler 对象的操作，则 OMC 将向 NMC 给出“身份信息无效”（InvalidCredential）异常。

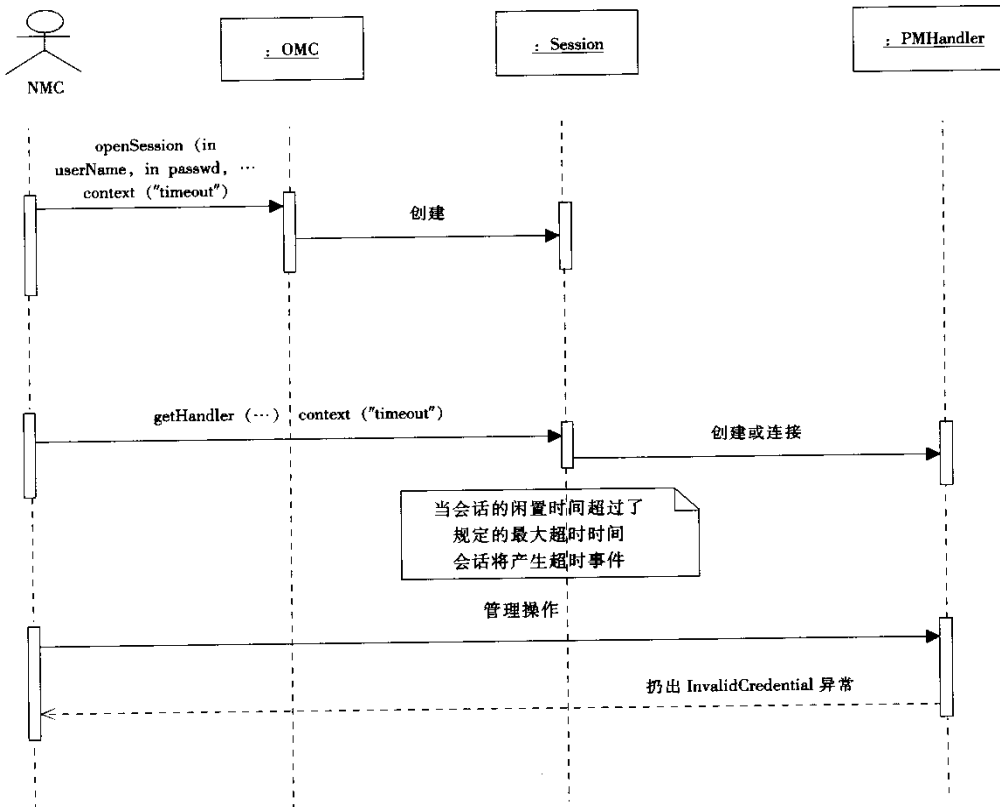


图 20 超时场景的 UML 序列图

6.5.3 访问控制

OMC 应使用访问控制机制来检查调用操作的 NMC 是否有权限执行该管理操作。首先，在 openSession 操作的结果中，OMC 将向 NMC 返回用户身份验证信息，此信息将在该会话中由 NMC 使用。其次，采用上下文对象 (CORBA::Context) 来实现访问控制，即对于每一个向 OMC 调用的管理操作，在相应的 CORBA::Context 对象中都应该包含由 openSession 操作所返回的用户身份验证信息。如果用户身份验证信息非法，OMC 应给出指示违反访问控制策略的异常。

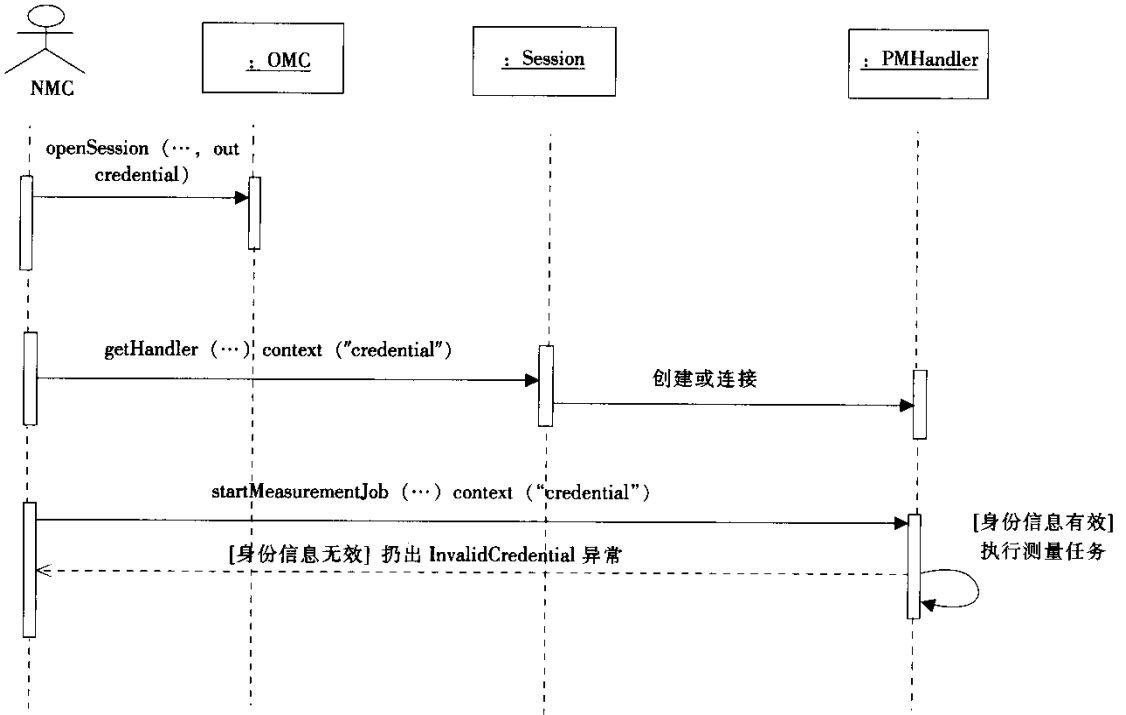


图 21 访问控制的 UML 序列图

图 21 描述了实施访问控制机制的一个完整交互过程。NMC 在向 OMC 接口调用 openSession 操作时获得该会话的身份验证信息。在该会话的整个阶段，如果 NMC 要对 OMC 中的任何其他接口进行操作调用，NMC 必须在上下文对象中给出身份验证信息。OMC 判断调用者是否有权执行操作。如果身份验证信息非法，OMC 应给出“身份信息无效” (InvalidCredential) 异常。图中所示的例子假设在 getHandler 操作中 NMC 使用的身份验证信息合法，而在调用 startMeasurementJob 操作中使用了非法的身份验证信息。

6.6 故障管理

为了实现故障管理功能集，本标准中还定义了 FMHandler 接口。该接口提供了在功能需求部分描述的故障管理功能。

6.6.1 FMHandler_v2 接口概述

FMHandler_v2 接口向 NMC 提供了执行各种故障管理功能的操作。其接口如图 22 所示。

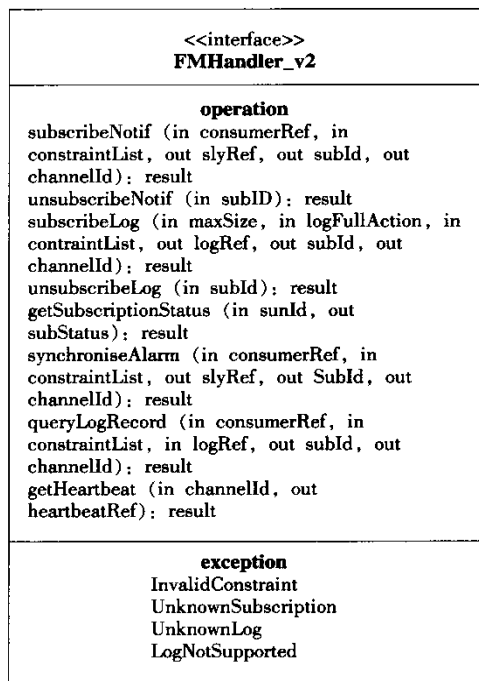


图 22 FMHandler_v2 接口的 UML 类图

6.6.2 FMHandler_v2 接口的操作

6.6.2.1 subscribeNotif_v2 操作

行为

“NMC 调用该操作向 OMC 请求告警通知订购。”

输入参数

consumerRef: SequencePushConsumer

输入参数 consumerRef 是 NMC 侧事件接收者对象的 IOR，事件通道通过对 consumerRef 调用 push 操作将事件信息传送给 NMC。该参数的类型是从 CORBA 事件服务中 CosNotifyComm 模块导入的 SequencePushConsumer 类型。

constraintList: ConstrainExpSeq

输入参数 constraintList 定义过滤条件，由 OMC 设置事件通道，从而实现对告警的过滤。其类型为从 CORBA 事件服务中 CosNotifyComm 模块导入的 ConstrainExpSeq 类型。若值为空则表示无需对事件进行过滤。其中，ConstraintExp 里的 constraint_expr 域应遵从 OMG 通知服务中的缺省过滤约束语言的语法。

输出参数

slyRef: SequenceProxyPushSupplier

输出参数 slyRef 是由通知通道所创建的代理对象。NMC 可以通过该对象控制通知和过滤条件。

subId: SubscriptionIdType

输出参数 subId 是通知订购的标识符。NMC 通过 subId 可以控制该通知订购。

channelId: ChannelIdType

输出参数 channelId 是 consumerRef 所连接的通知通道的标识符。该参数将在以后的 getHandler 操作中使用，来获得相应的 cTelHeartbeat 对象的 IOR。

前提条件

命题 1: “给定的过滤条件是合法的”。

后继条件

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立, 则应给出异常 “InvalidConstraint”。

6.6.2.2 unsubscribeNotif 操作

行为

“NMC 使用该操作来撤销一个已存在的告警通知订购。”

输入参数

subId: SubscriptionIdType

输入参数 subId 标识要撤销的告警通知订购。

前提条件

命题 1: “OMC 中存在指定的告警通知订购”。

后继条件

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立, 则应给出异常 “UnknownSubscription”。

6.6.2.3 getSubscriptionStatus 操作

行为

“NMC 使用该操作来查询某个指定的告警订购的当前状态。”

输入参数

subId: SubscriptionIdType

输入参数 subId 标识所要查询的告警订购。

输出参数

subStatus: eSubscriptionStatusType

输出参数 subStatus 表明告警订购的当前状态。

前提条件

命题 1: “OMC 中存在指定的告警订购”。

后继条件

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立，则应给出异常 “UnknownSubscription”。

6.6.2.4 subscribeLog_v2 操作

行为

“NMC 调用该操作向 OMC 请求告警日志订购。”

输入参数

constrainList: ConstrainExpSeq

输入参数 constrainList 定义过滤条件，由 OMC 设置事件通道，从而实现对要存入日志的告警的过滤。其类型为从 CORBA 事件服务中 CosNotifyComm 模块导入的 ConstrainExpSeq 类型。若值为空则表示无需对事件进行过滤。其中 ConstraintExp 里的 constraint_expr 域应遵从 OMG 通知服务中的缺省过滤约束语言的语法。

logFullAction: LogFullAction

输入参数 logFullAction 指当日志对象中存储的总记录超过给定的最大容量时，日志对象应采取的行动。它可以为覆盖 (wrap) 或停止 (halt)。

maxsize: unsigned long

输入参数 maxsize 指日志对象中能存储的日志记录的最大容量（以字节为单位）。日志对象的容量可以是无限的。若值为 0，则假设日志的容量无限。

输出参数

logRef: NotifyLog

输出参数 logRef 是日志对象的 IOR。NMC 可以通过 logRef 来控制日志的过滤条件，并可调用日志记录的相关操作。

subId: SubscriptionIdType

输出参数 subId 是日志订购的标识符。NMC 通过 subId 可以控制该日志订购。

channelId: ChannelIdType

输出参数 channelId 是 consumerRef 所连接的通知通道的标识符。该参数将在以后的 getHandler 操作中使用，来获得相应的 cTelHeartbeat 对象的 IOR。

前提条件

命题 1：“给定的过滤条件是合法的”。

命题 2：“OMC 支持日志服务功能”。

后继条件

如果操作成功，则 result= “success”；

如果操作失败，则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立，则应给出异常 “InvalidConstraint”；

如果前提条件命题 2 不成立，则应给出异常 “LogNotSupported”；

6.6.2.5 unSubscribeLog 操作

行为

“NMC 使用该操作来撤销一个已存在的告警日志订购。”

输入参数

subId: SubscriptionIdType

输入参数 subId 标识要撤销的告警日志订购。

前提条件

命题 1: “OMC 中存在指定的告警日志订购”。

命题 2: “OMC 支持日志服务功能”。

后继条件

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立, 则应给出异常 “UnknownSubscription”;

如果前提条件命题 2 不成立, 则应给出异常 “LogNotSupported”。

6.6.2.6 queryLogRecord_v2 操作

行为

“NMC 调用该操作来获取告警日志记录。当 OMC 完成数据收集后, 它应向 NMC 发送指示消息, 通知 NMC 结果已就绪。”

输入参数

consumerRef: SequencePushConsumer

输入参数 consumerRef 是 NMC 侧事件接收者对象的 IOR, 事件通道通过对 consumerRef 调用 push 操作通知 NMC 日志结果文件已准备就绪。该参数的类型是从 CORBA 事件服务中 CosNotifyComm 模块导入的 SequencePushConsumer 类型。

constraintList: ConstrainExpSeq

输入参数 constraintList 定义过滤条件, 由 OMC 设置事件通道, 从而实现对告警日志的过滤。其类型为从 CORBA 事件服务中 CosNotifyComm 模块导入的 ConstrainExpSeq 类型。若值为空则表示无需对事件进行过滤。其中 ConstraintExp 里的 constraint_expr 域应遵从 OMG 通知服务中的缺省过滤约束语言的语法。

logRef: NotifyLog

输入参数 logRef 是 NMC 所订购的日志接口对象的 IOR。

输出参数

subId: SubscriptionIdType

输出参数 subId 是为该日志记录查询操作所分配通知订购的标识符。NMC 可以通过 subId 控制该通知订购。

channelId: ChannelIdType

输出参数 channelId 是 consumerRef 所连接的通知通道的标识符。该参数将在以后的 getHandler 操作中使用, 来获得相应的 cTelHeartbeat 对象的 IOR。

前提条件

命题 1: “给定的过滤条件是合法的”。

命题 2: “指定的 log 对象在 OMC 中存在”。

命题 3: “OMC 支持日志服务”。

后继条件

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立，则应给出异常 “InvalidConstraint”；

如果前提条件命题 2 不成立，则应给出异常 “UnknownLog”；

如果前提条件命题 3 不成立，则应给出异常 “LogNotSupported”。

6.6.2.7 synchronizeAlarm_v2 操作

行为

“NMC 调用该操作来获取 OMC 中满足过滤条件的所有活跃告警信息。当 OMC 完成了告警信息的采集，则告警数据将在一个告警总结报告中。这些数据可以有两种方式进行传送：一种是通过事件通道，另一种是通过 FTP 服务在第一种情况中。当 OMC 完成数据采集后，它将向 NMC 发送一个或多个 NOTIFY_CURRENT_ALARM_SUMMARY_REPORT 通知，报告中的数据可以分割为好几个通知，以使每个通知不至于过大。在第二种情况下，当 OMC 完成数据采集后，可以将报数据写入一个到多个文件中，并向 NMC 发送 NOTIFY_FM_SYNC_TRANSFER_UP_READY 通知来告诉 NMC 来获取文件。这通常用于告警同步的数据量很大，而不适用用通知通道来传送的情况。OMC 可以只选择其中的一种机制来实现，在第一阶段，文件方式是可选的。”

输入参数

consumerRef: SequencePushConsumer

输入参数 consumerRef 是 NMC 侧事件接收者对象的 IOR，事件通道通过对 consumerRef 调用 push 操作将 NOTIFY_FM_SYNC_TRANSFER_UP_READY 通知或 NOTIFY_FM_SYNC_TRANSFER_PREPARATION_ERROR 通知传送给 NMC。该参数的类型是从 CORBA 事件服务中 CosNotifyComm 模块导入的 SequencePushConsumer 类型。

constraintList: ConstrainExpSeq

输入参数 constraintList 定义过滤条件，由 OMC 设置事件通道，从而实现对告警通知的过滤。其类型为从 CORBA 事件服务中 CosNotifyComm 模块导入的 ConstrainExpSeq 类型。若值为空则表示无需对事件进行过滤。其中 ConstraintExp 里的 constraint_expr 域应遵从 OMC 通知服务中的缺省过滤约束语言的语法。

输出参数

slyRef: SequenceProxyPushSupplier

输出参数 slyRef 是由通知通道所创建的代理对象。NMC 可以通过该对象控制通知和过滤条件。

subId: SubscriptionIdType

输出参数 subId 是为该告警同步操作所分配通知订购的标识符。NMC 可以通过 subId 控制该通知订购。

channelId: ChannelIdType

输出参数 channelId 是 consumerRef 所连接的通知通道的标识符。该参数将在以后的 getHandler 操作中使用，来获得相应的 cTelHeartbeat 对象的 IOR。

前提条件

命题 1：“给定的过滤条件是合法的”。

后继条件

如果操作成功，则 result= “success”；

如果操作失败，则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立，则应给出异常 “InvalidConstraint”。

6.6.2.8 getHeartbeat 操作

行为

“NMC 使用该操作来获取与某个指定的通知通道相关联的 cTelHeartbeat 对象的对象引用。”

输入参数

channelId: ChannelIdType

输入参数 channelId 指 NMC 由目标 cTelHeartbeat 对象所管理的通知通道的标识。

输出参数

heartbeatRef: cTelHeartbeat

输出参数 heartbeatRef 指与给定的通知通道相关联的 cTelHeartbeat 对象的对象引用。

前提条件

命题 1：“OMC 中存在指定的通知通道”。

后继条件

如果操作成功，则 result= “success”；

如果操作失败，则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立，则应给出异常 “UnknowChannel”。

6.6.3 故障管理的场景描述

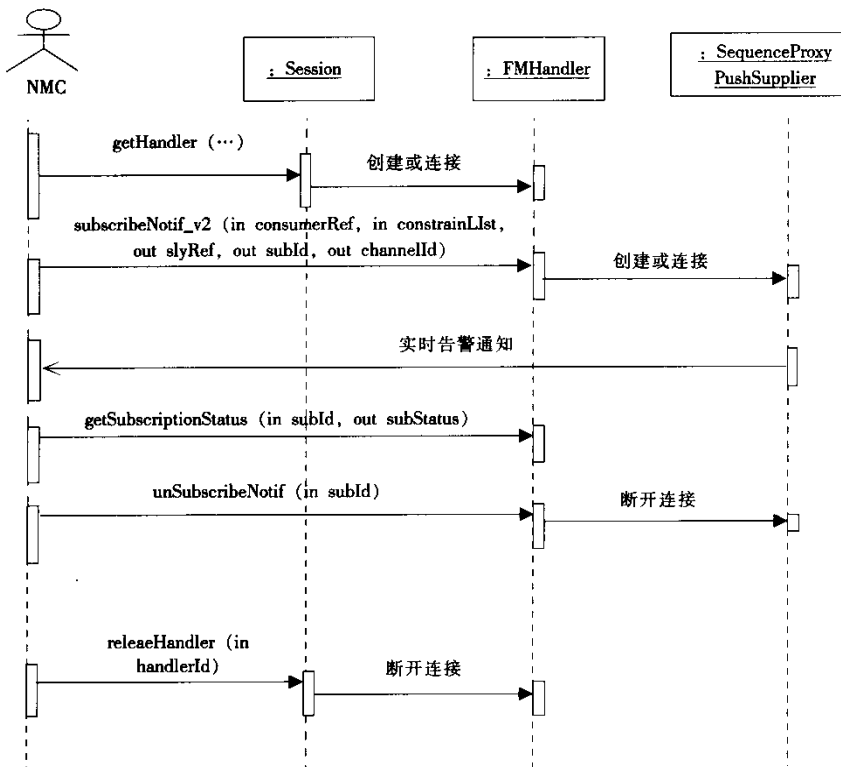


图 23 告警通知订购的 UML 序列图

图 23 描述了一个完整的通知订购的场景。NMC 可输入适当的参数调用 FMHandler_v2 接口中的订购通知操作。在该操作成功后，NMC 就可以收到满足所设过滤条件的事件通知。NMC 可以通过调用 Fmandler_v2 的 synchronizeAlarm_v2 操作来获取 OMC 中当前所有的活跃告警。如果 NMC 不想再接收告警通知，可以调用该接口中的 unSubscribeNotif 操作来撤销本次告警通知的订购。

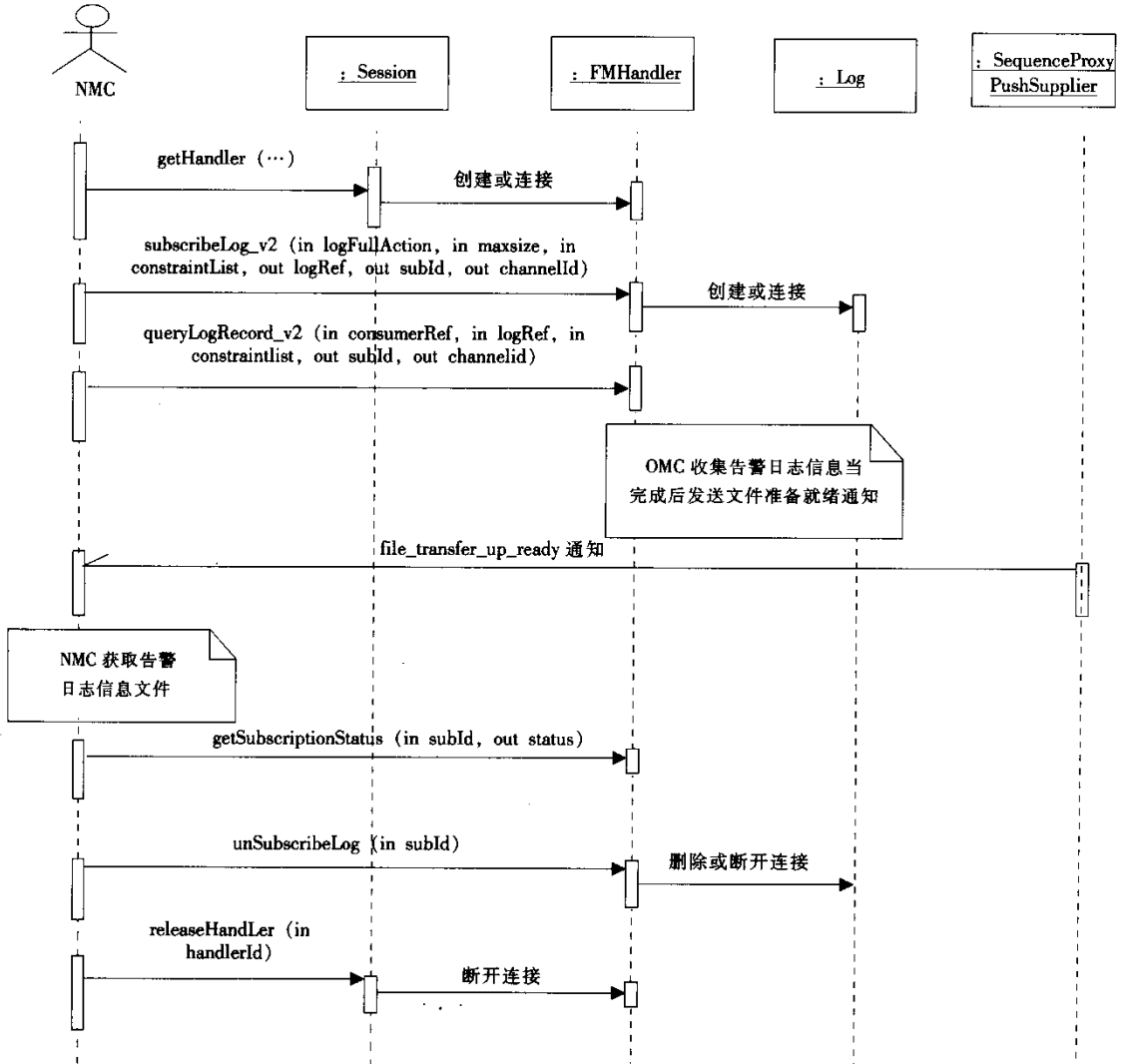


图 24 告警日志订购的序列图

图 24 描述了一个完整的告警日志订购的交互场景。NMC 输入适当的参数来调用 FMHandler_v2 接口中的订购告警日志操作。若操作成功，则完成了一次告警日志订购。OMC 将负责对满足过滤条件事件通知持久保存为日志记录。NMC 可以显式地调用 FMHandler_v2 接口的 queryLogRecord 操作来获取满足过滤条件的日志记录（或通过 Log 接口直接查询日志信息）。若 NMC 要撤销告警日志订购，可以调用接口中的 unSubscribeLog 操作。

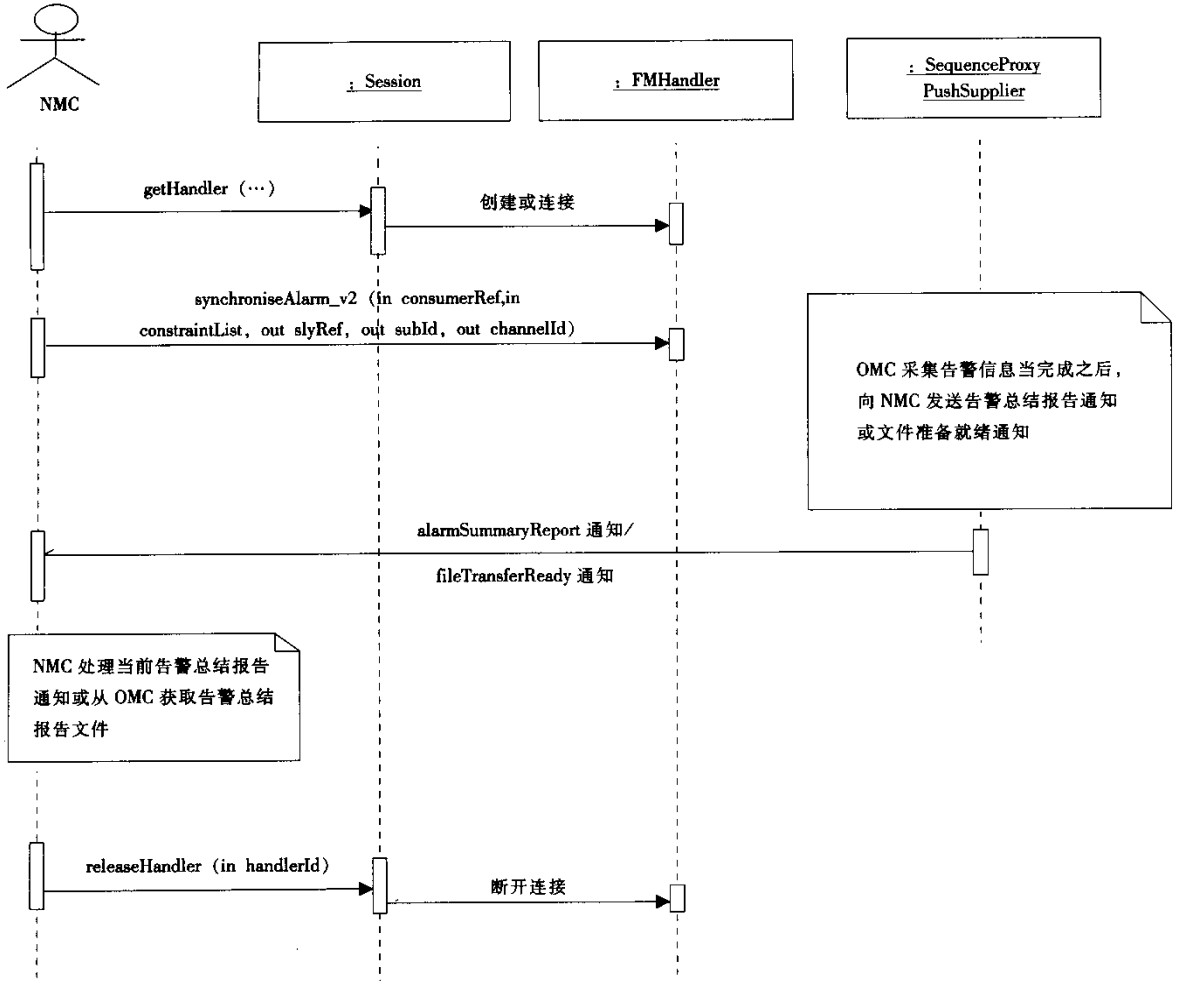


图 25 同步告警的场景序列图

图 25 描述了一个完整的告警同步的场景。NMC 调用 FMHandler_v2 的 synchroniseAlarm_v2 操作。OMC 将收集系统中所有的活跃告警。当 OMC 完成收集采集后，它将向 NMC 发送告警同步通知来上报传送结果。或者 OMC 发送文件传输准备就绪通知，NMC 收到之后将使用 FTP 服务来获取结果文件。

6.7 配置管理

为实现配置管理功能集，本标准中定义了 CMHandler 接口。该接口提供了在功能需求部分描述的配置管理功能。

6.7.1 CMHandler_v2 接口概述

CMHandler_v2 接口向 NMC 提供了以下类型的操作：执行各种配置管理功能，控制 MOHandler 接口，其接口的类图如图 26 所示。

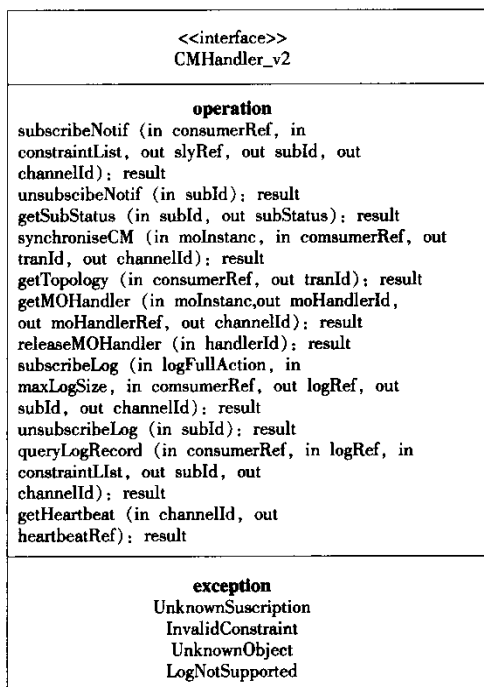


图 26 CMHandler_v2 接口的 UML 类图

6.7.2 CMHandler_v2 接口的操作

6.7.2.1 subscribeNotif_v2 操作

行为

“NMC 调用该操作向 OMC 请求配置通知订购。”

输入参数

consumerRef: SequencePushConsumer

输入参数 consumerRef 是 NMC 侧事件接收者对象的 IOR，事件通道通过对 consumerRef 调用 push 操作将事件信息传送给 NMC。该参数的类型是从 CORBA 事件服务中 CosNotifyComm 模块导入的 SequencePushConsumer 类型。

constraintList: ConstrainExpSeq

输入参数 constraintList 定义过滤条件，由 OMC 设置事件通道，从而实现对配置通知的过滤。其类型为从 CORBA 事件服务中 CosNotifyComm 模块导入的 ConstrainExpSeq 类型。若值为空则表示无需对事件进行过滤。其中 ConstraintExp 里的 constraint_expr 域应遵从从 OMG 通知服务中的缺省过滤约束语言的语法。

输出参数

slyRef: SequenceProxyPushSupplier

输出参数 slyRef 是由通知通道所创建的代理对象。NMC 可以通过该对象控制通知和过滤条件。

subId: SubscriptionIdType

输出参数 subId 是通知订购的标识符。NMC 通过 subId 可以控制该通知订购。

channelId: ChannelIdType

输出参数 channelId 是 consumerRef 所连接的通知通道的标识符。该参数将在以后的 getHandler 操作中使用，来获得相应的 cTelHeartbeat 对象的 IOR。

前提条件

命题 1: “给定的过滤条件是合法的”。

后继条件

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立, 则应给出异常 “InvalidConstraint”。

6.7.2.2 unsubscribeNotif 操作

行为

“NMC 使用该操作来撤销一个已存在的配置通知订购。”

输入参数

subId: SubscriptionIdType

参数 subId 标识要撤销的配置通知订购。

前提条件

命题 1: “OMC 中存在指定的配置通知订购”。

后继条件

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立, 则应给出异常 “UnknownSubscription”。

6.7.2.3 getSubscriptionStatus 操作

行为

“NMC 使用该操作来查询某个指定的配置通知订购的当前状态。”

输入参数

subId: SubscriptionIdType

输入参数 subId 标识所要查询的配置通知订购。

输出参数

subStatus: eSubscriptionStatusType

输出参数 subStatus 表明配置通知订购的当前状态。

前提条件

命题 1: “OMC 中存在指定的配置通知订购”。

后继条件

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立，则应给出异常 “UnknownSubscription”。

6.7.2.4 subscribeLog_v2 操作

行为

“NMC 调用该操作向 OMC 请求配置通知日志订购。”

输入参数

constrainList: ConstrainExpSeq

输入参数 constrainList 定义过滤条件，由 OMC 设置事件通道，从而实现对要存入日志的配置通知的过滤。其类型为从 CORBA 事件服务中 CosNotifyComm 模块导入的 ConstrainExpSeq 类型。若值为空则表示无需对事件进行过滤。其中 ConstraintExp 里的 constraint_expr 域应遵从 OMG 通知服务中的缺省过滤约束语言的语法。

logFullAction: LogFullAction

输入参数 logFullAction 指当日志对象中存储的总记录超过给定的最大容量时，日志对象应采取的行动。它可以为覆盖 (wrap) 或停止 (halt)。

maxsize: unsigned long

输入参数 maxsize 指日志对象中能存储的日志记录的最大容量 (以字节为单位)。日志对象的容量可以是无限的。若值为 0，则日志的容量无限。

输出参数

logRef: NotifyLog

输出参数 logRef 是日志对象的 IOR。NMC 可以通过 logRef 来控制日志的过滤条件，并可调用日志记录的相关操作。

subId: SubscriptionIdType

输出参数 subId 是日志订购的标识符。NMC 通过 subId 可以控制该日志订购。

channelId: ChannelIdType

输出参数 channelId 是 consumerRef 所连接的通知通道的标识符。该参数将在以后的 getHandler 操作中使用，来获得相应的 cTelHeartbeat 对象的 IOR。

前提条件

命题 1: “给定的过滤条件是合法的”。

命题 2: “OMC 支持日志服务功能”。

后继条件

如果操作成功，则 result= “success”；

如果操作失败，则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立，则应给出异常 “InvalidConstraint”；

如果前提条件命题 2 不成立，则应给出异常 “LogNotSupported”。

6.7.2.5 unsubscribeLog 操作

行为

“NMC 使用该操作来撤销一个已存在的配置日志订购。”

输入参数

subId: SubscriptionIdType

输入参数 subId 标识要撤销的配置日志订购。

前提条件

命题 1: “OMC 中存在指定的配置日志订购”。

命题 2: “OMC 支持日志服务功能”。

后继条件

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立, 则应给出异常 “UnknownSubscription”;

如果前提条件命题 2 不成立, 则应给出异常 “LogNotSupported”。

6.7.2.6 getTopology_v2 操作

行为

“NMC 调用该操作以获取 OMC 子网的拓扑信息。当 OMC 完成数据收集后, 应向 NMC 发送结果文件就绪消息, 然后 NMC 可以使用批量数据传输功能获取结果文件。如果在数据收集过程中发生了错误, OMC 应向 NMC 发送指示性的通知。”

输入参数

consumerRef : SequencePushConsumer

输入参数 consumerRef 是 NMC 侧事件接收者对象的 IOR, 事件通道通过对 consumerRef 调用 push 操作将 NOTIFY_TOP_SYNC_TRANSFER_UP_READY 通知或 NOTIFY_TOP_SYNC_TRANSFER_PREPARATION_ERROR 通知传送给 NMC。该参数的类型是从 CORBA 事件服务中 CosNotifyComm 模块导入的 SequencePushConsumer 类型。

输出参数

tranId: TransactionIdType

输出参数 tranId 标识这次获取拓扑操作。在 OMC 发送的结果文件就绪通知中含有 tranId 参数, 用于 NMC 将结果文件与其对应的操作相关联。

channelId: ChannelIdType

输出参数 channelId 是 consumerRef 所连接的通知通道的标识符。该参数将在以后的 getHandler 操作中使用, 来获得相应的 cTelHeartbeat 对象的 IOR。

后继条件

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”。

返回值

result : eResultType

返回值 result 标识操作的成功与否。

6.7.2.7 synchronizeCM_v2 操作

行为

“NMC 调用该操作来获取 OMC 中包容树的配置信息数据。当 OMC 完成数据收集后, 应向 NMC 发送结果文件就绪通知, 然后 NMC 可以使用批量数据传输功能获取结果文件。如果在数据收集过程中发生了错误, OMC 应向 NMC 发送指示性的通知。”

输入参数

moInstance: ManagedObjectType

输入参数 moInstance 指配置拓扑信息所需的包容树的基对象。该参数的类型是由类名和管理

对象的可区分名所组成的结构。

consumerRef: SequencePushConsumer

输入参数 consumerRef 是 NMC 侧事件接收者对象的 IOR，事件通道通过对 consumerRef 调用 push 操作将 NOTIFY_CM_SYNC_TRANSFER_UP_READY 通知 或 NOTIFY_CM_SYNC_TRANSFER_PREPARATION_ERROR 通知传送给 NMC。该参数的类型是从 CORBA 事件服务中 CosNotifyComm 模块导入的 SequencePushConsumer 类型。

输出参数

tranId: TransactionIdType

输出参数 tranId 标识这次配置同步操作。在 OMC 发送的结果文件就绪通知中含有 tranId 参数，用于 NMC 将结果文件与其对应的操作相关联。

channelId: ChannelIdType

输出参数 channelId 是 consumerRef 所连接的通知通道的标识符。该参数将在以后的 getHandler 操作中使用，来获得相应的 cTelHeartbeat 对象的 IOR。

前提条件

命题 1：“OMC 中存在 NMC 指定的基对象实例”。

后继条件

如果操作成功，则 result= “success”；

如果操作失败，则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立，则应给出异常 “UnknownObject”。

6.7.2.8 queryLogRecord_v2 操作

行为

“NMC 调用该操作来获取配置通知的日志记录。当 OMC 完成数据收集后，它应向 NMC 发送指示消息，通知 NMC 结果文件已就绪。”

输入参数

consumerRef: SequencePushConsumer

输入参数 consumerRef 是 NMC 侧事件接收者对象的 IOR，事件通道通过对 consumerRef 调用 push 操作通知 NMC 日志结果文件已准备就绪。该参数的类型是从 CORBA 事件服务中 CosNotifyComm 模块导入的 SequencePushConsumer 类型。

constrainList: ConstrainExpSeq

输入参数 constraintList 定义过滤条件，由 OMC 设置事件通道，从而实现了对配置通知日志的过滤。其类型为从 CORBA 事件服务中 CosNotifyComm 模块导入的 ConstrainExpSeq 类型。若值为空则表示无需对事件进行过滤。其中 ConstraintExp 里的 constraint_expr 域应遵从 OMC 通知服务中的缺省过滤约束语言的语法。

logRef: NotifyLog

输入参数 logRef 是 NMC 所订购的日志接口对象的 IOR。

输出参数

subId: SubscriptionIdType

输出参数 subId 是为该日志记录查询操作所分配通知订购的标识符。NMC 可以通过 subId 控制该通知订购。

channelId: ChannelIdType

输出参数 channelId 是 consumerRef 所连接的通知通道的标识符。该参数将在以后的 getHandler 操作中使用, 来获得相应的 cTelHeartbeat 对象的 IOR。

前提条件

- 命题 1: “给定的过滤条件是合法的”。
- 命题 2: “指定的 log 对象在 OMC 中存在”。
- 命题 3: “OMC 支持日志服务”。

后继条件

- 如果操作成功, 则 result= “success”;
- 如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

- 如果前提条件命题 1 不成立, 则应给出异常 “InvalidConstraint”;
- 如果前提条件命题 2 不成立, 则应给出异常 “UnknownLog”;
- 如果前提条件命题 3 不成立, 则应给出异常 “LogNotSupported”。

6.7.2.9 getMOHandler 操作

行为

“NMC 调用该操作以获取管理某个特定管理对象的句柄。”

输入参数

moInstance: ManagedObjectType

输入参数 moInstance 指需要获得句柄的管理对象。该参数的类型是由类名和管理对象的可区分名所组成的结构。

输出参数

moHandlerRef: CORBA::Object

输出参数 moHandlerRef 返回接口 MOHandler 对象实例的 IOR。NMC 可以通过它对管理对象实施相应的操作。

handlerId: HandlerIdType

输出参数 handlerId 惟一标识了返回的 MOHandler 接口对象实例。

前提条件

- 命题 1: “OMC 中存在 NMC 指定的基对象实例”。

后继条件

- 如果操作成功, 则 result= “success”;
- 如果操作失败, 则 result= “failure”, 且 moHandlerRef=NULL。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

- 如果前提条件命题 1 不成立, 则应给出异常 “UnknownObject”。

6.7.2.10 releaseMOHandler 操作

行为

“NMC 调用该操作来释放一个管理对象的句柄。”

输入参数

handlerId: HandlerId

输入参数 handlerId 标识了要释放的 MOHandler 接口对象实例。

前提条件

命题 1: “OMC 中存在 NMC 指定的 MOHandler 对象”。

后继条件

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立, 则应给出异常 “UnknownHandler”。

6.7.2.11 getHeartbeat 操作

行为

“NMC 使用该操作来获取与某个指定的通知通道相关联的 cTelHeartbeat 对象的对象引用。”

输入参数

channelId: ChannelIdType

输入参数 channelId 指 NMC 由目标 cTelHeartbeat 对象所管理的通知通道的标识。

输出参数

heartbeatRef: cTelHeartbeat

输出参数 heartbeatRef 指与给定的通知通道相关联的 cTelHeartbeat 对象的对象引用。

前提条件

命题 1: “OMC 中存在指定的通知通道”。

后继条件

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立, 则应给出异常 “UnknowChannel”。

6.7.3 配置管理的场景描述

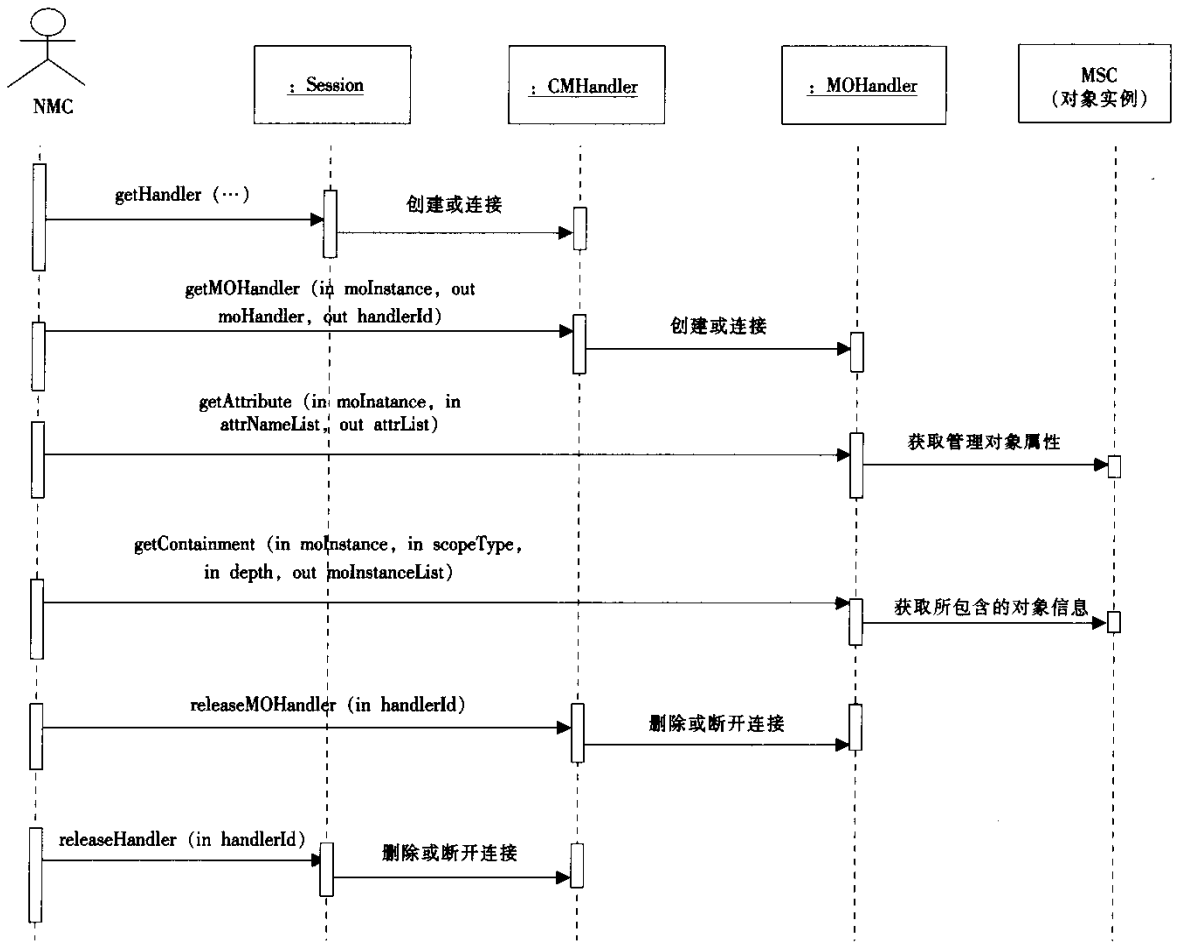


图 27 管理对象句柄的场景

在如图 27 所示的场景中，NMC 首先调用 CMHandler_v2 接口中的 getMOHandler 操作获得 MOHandler 接口的 IOR。随后，NMC 可以调用 MOHandler 接口中的操作来访问管理对象的信息。NMC 在完成对管理对象信息的访问后，可以调用 releaseMOHandler 操作来释放相应的 MOHandler 接口对象实例。

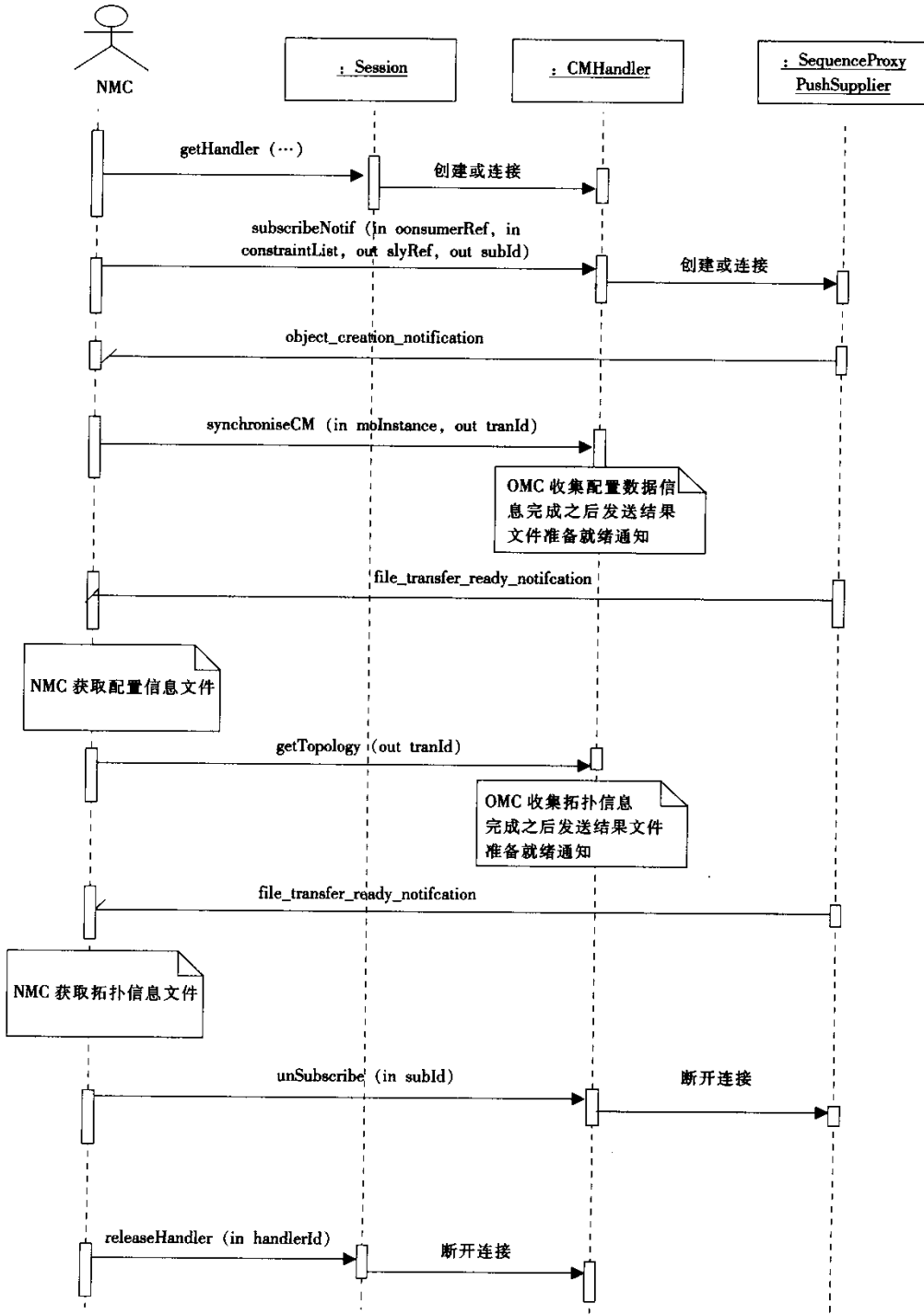


图 28 配置管理场景的序列图

在如图 28 所示场景中，NMC 首先调用 `getHandler` 获得 `CMHandler` 接口对象实例。随后 NMC 可以调用 `CMHandler_v2` 接口中的相关操作来指示 OMC 进行配置信息的收集。在 OMC 完成了信息收集并以文件形式组织好数据后，向 NMC 发送结果文件准备就绪通知。NMC 使用 FTP 服务来获取结果文件。

6.8 性能管理

为实现性能管理功能集，本标准中定义了 `PMHandler` 接口。该接口提供了在功能需求部分描述的性能管理功能。

6.8.1 `PMHandler_v2` 接口概述

`PMHandler_v2` 接口向 NMC 提供了执行各种性能管理功能的操作。其接口如图 29 所示。

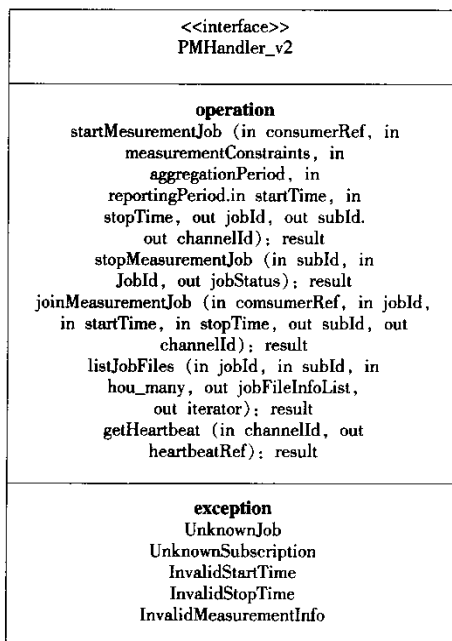


图 29 `PMHandler_v2` 接口的 UML 类图

6.8.2 `PMHandler_v2` 接口的操作

6.8.2.1 `startMeasurementJob_v2` 操作

行为

“NMC 调用 `startMeasurementJob` 操作请求 OMC 开启一项测量工作。”

输入参数

consumerRef: SequencePushConsumer

输入参数 `consumerRef` 是 NMC 侧接收事件的对象的 IOR，通过对 `consumerRef` 调用 `push` 操作可以通知 NMC 结果文件已准备就绪。参数的类型是从 CORBA 事件服务中 `CosNotifyComm` 模块导入的 `SequencePushConsumer` 类型。

measurementInfoList: MeasurementInfoListType

输入参数 `measurementInfoList` 指定测量工作中需进行性能数据采集的管理对象、测量属性以及测量中其他一些必须的信息。如果参数中只给出了管理对象类，而并未指明具体的管理对象 DN，则该测量工作将测量所有该对象类的实例。如果属性值列表的长度为 0，则表示在该测量任务中的所有测量属性都应被返回。

aggregationPeriod: AggregationPeriodType

输入参数 aggregationPeriod 指连续两次性能数据采集的时间间隔，其单位是 15min。

reportingPeriod: ReportingPeriodType

输入参数 reportingPeriod 指连续两次由 OMC 向 NMC 发送结果文件准备就绪通知的时间间隔。该值应为 aggregationPeriod 参数值的整数倍，以 15min 为单位。

startTime: GeneralizedTimeType

输入参数 startTime 指测量工作的实际开始时间。如果值为空，则 OMC 应立即开启测量工作。该参数的类型为字符串类型，格式遵照 ASN.1 中定义的 GeneralizedTime。

stopTime: GeneralizedTimeType

输入参数 stopTime 指测量工作的结束时间。如果该值为空，则表示该测量工作将一直运行，直到 NMC 调用 stopMeasurementJob 操作来中止它。参数的类型为字符串类型，格式遵照 ASN.1 中定义的 GeneralizedTime。若实际时间到了参数 stopTime 所指定的值，则 OMC 应和收到 stopMeasurementJob 操作请求的结果行为一样，将自动停止对该测量任务的订购。

输出参数**jobId: JobIdType**

输出参数 jobId 在 OMC 中惟一标识该项测量任务。

subId: SubscriptionIdType

输出参数 subId 是 NMC 订购该测量任务的标识符。对于 OMC 而言，subId 表示不同的 NMC 对该测量任务的订购。

channelId: ChannelIdType

输出参数 channelId 是 consumerRef 所连接的通知通道的标识符。该参数将在以后的 getHandler 操作中使用，来获得相应的 cTelHeartbeat 对象的 IOR。

后继条件

命题 1：“指定的测量属性是有效的”。

命题 2：“测量任务指定的起始时间不在于当前时间”。

命题 3：“指定的结束时间不早于该测量任务的开始时间”。

后继条件

如果操作成功，则 result= “success”；

如果操作失败，则 result= “failure”。

返回值**result: eResultType**

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立，则应给出异常 “InvalidMeasurementInfo”；

如果前提条件命题 2 不成立，则应给出异常 “InvalidStartTime”；

如果前提条件命题 3 不成立，则应给出异常 “InvalidStopTime”。

6.8.2.2 joinMeasurementJob_v2 操作**行为**

“NMC 调用该操作来加入一项已经被其他 NMC 开启的测量任务，相当于订购了该测量任务。”

输入参数**jobId: JobIdType**

输入参数 jobId 指 OMC 中某项测量任务的惟一标识。

consumerRef: SequencePushConsumer

输入参数 consumerRef 是 NMC 侧接收事件的对象的 IOR，通知通道可以通过对 consumerRef

调用 `push` 操作告诉 NMC 结果文件已准备就绪。该参数的类型是从 CORBA 事件服务中 `CosNotifyComm` 模块导入的 `SequencePushConsumer` 类型。

startTime: GeneralizedTimeType

输入参数 `startTime` 指定 NMC 实际参与测量任务的时间。如果该值为空，则表示 NMC 立即参与该测量任务。该参数的类型为字符串类型，格式遵照 ASN.1 中定义的 `GeneralizedTime`。该参数值不能早于开启该测量任务的 NMC 设定的任务起始运行时间，否则 OMC 可以给出 `InvalidStartTime` 异常。

stopTime: GeneralizedTimeType

输入参数 `stopTime` 指定此 NMC 结束对该测量任务订购的时间。如果值为空，则表示 NMC 一直订购该测量工作，直到 NMC 调用 `stopMeasurementJob` 动作来停止它。该参数的类型为字符串类型，格式遵照 ASN.1 中定义的 `GeneralizedTime`。若实际时间到了参数 `stopTime` 所指定的值，则 OMC 应和收到 `stopMeasurementJob` 操作请求的结果行为一样，将自动停止对该测量任务的订购。

输出参数

subId: SubscriptionIdType

输出参数 `subId` 是 NMC 订购该测量任务的标识符。对于 OMC 而言，`subId` 表示不同的 NMC 该测量任务的订购。

前提条件

命题 1：“OMC 中存在指定的测量任务”。

命题 2：“NMC 指定的测量任务订购时间 `startTime` 晚于当前时间”。

命题 3：“NMC 指定的结束任务订购时间晚于其相应的起始时间”。

后继条件

如果操作成功，则 `result=“success”`；

如果操作不成功，则 `result=“failure”`。

返回值

result: eResultType

返回值 `result` 标识操作的成功与否。

异常

如果前提条件命题 1 不成立，则应给出异常 `“UnknownJob”`；

如果前提条件命题 2 不成立，则应给出异常 `“InvalidStartTime”`；

如果前提条件命题 3 不成立，则应给出异常 `“InvalidStopTime”`

6.8.2.3 stopMeasurementJob 操作

行为

“NMC 调用该操作来停止对一项测量任务的订购。如果此时该任务只为这一个 NMC 运行，则操作完成之后该测量任务就应停止运行了。”

输入参数：

jobId: JobIdType

输入参数 `jobId` 指 OMC 中该项测量任务的惟一标识。

subId: SubscriptionIdType

输入参数 `subId` 是 NMC 订购该测量任务的标识符。

输出参数

jobStauts: eJobStatusType

输出参数 `jobStauts` 返回了测量任务的当前状态。在此操作中，该参数的取值可以为 `JOB_DISCONNECTED_BUT_STILL_RUNNING`，表示该测量任务仍在为其他 NMC 服务；或 `JOB_STOPPED`，表示测量任务已经停止。

前提条件

命题 1: “OMC 中存在指定的测量任务”。

命题 2: “NMC 订购了该测量任务”。

后继条件

命题 3: “该测量任务停止运行”。

命题 4: “已被该 NMC 撤销订购, 但测量任务仍在为其他的 NMC 所运行”。

如果命题 3 成立, 则 jobStatus=JOB_STOPPED。

如果命题 4 成立, 则 jobStatus=JOB_JOB_DISCONNECTED_BUT_STILL_RUNNING。

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立, 则应给出异常 “UnknownJob”;

如果前提条件命题 2 不成立, 则应给出异常 “UnknownSubscription”。

6.8.2.4 listMeasurementJob 操作

行为

“NMC 调用该操作来获取当前 OMC 中所有已存在的测量任务的信息。”

输出参数

jobInfoList: JobInfoListType

输出参数 jobInfoList 返回当前 OMC 中所有已经运行或将要运行的测量任务的信息。

后继条件

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

6.8.2.5 listJobFiles_v2 操作

行为

“NMC 调用该操作来获取 OMC 中指定的测量任务的结果文件信息以及文件的访问信息。只有订购了该测量任务的 NMC 有权限调用该操作。”

输入参数

jobId: JobIdType

输入参数 jobId 指 OMC 中该项测量任务的惟一标识。

subId: SubscriptionIdType

输入参数 subId 是 NMC 订购该测量任务的标识符。

how_many: unsigned long

输入参数 how_many 指 OMC 将通过输出参数 jobFileInfoList 返回的文件信息的个数。如果还有剩余的文件信息, 则将通过输出参数 iterator 来获得。

输出参数

jobFileInfoList: FileInfoListType

输出参数 jobFileInfoList 返回该测量任务的结果文件信息以及文件的访问信息。

iterator: JobFileInfoIterator

输出参数 iterator 是接口 JobFileInfoIterator 的对象实例。当文件信息没有全部通过输出参数 jobFileInfoList 返回时, 该参数将用于获取剩余的文件信息。如果所有的结果都由 jobFileInfoList 返回了, 则该参数的值是空。

前提条件

命题 1: “OMC 中存在指定的测量任务”。

命题 2: “NMC 订购了该测量任务”。

后继条件

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立, 则应给出异常 “UnknownJob”;

如果前提条件命题 2 不成立, 则应给出异常 “UnknownSubscription”。

6.8.2.6 getHeartbeat 操作

行为

“NMC 使用该操作来获取与某个指定的通知通道相关联的 cTelHeartbeat 对象的对象引用。”

输入参数

channelId: ChannelIdType

输入参数 channelId 指 NMC 由目标 cTelHeartbeat 对象所管理的通知通道的标识。

输出参数

heartbeatRef: cTelHeartbeat

输出参数 heartbeatRef 指与给定的通知通道相关联的 cTelHeartbeat 对象的对象引用。

前提条件

命题 1: “OMC 中存在指定的通知通道”。

后继条件

如果操作成功, 则 result= “success”;

如果操作失败, 则 result= “failure”。

返回值

result: eResultType

返回值 result 标识操作的成功与否。

异常

如果前提条件命题 1 不成立, 则应给出异常 “UnknowChannel”。

6.8.3 JobFileInfolerator 接口

6.8.3.1 JobFileInfolerator 接口概述

该接口用于 NMC 获取 listJobFiles_v2 操作中所剩余的测量任务的结果文件信息。该接口提供了两个操作: getNext 和 destroy。getNext 操作用于获取剩余的文件信息, destroy 操作用于获取所有的文件信息后释放该对象。图 30 是 JobFileInfolerator 接口的 UML 类图。

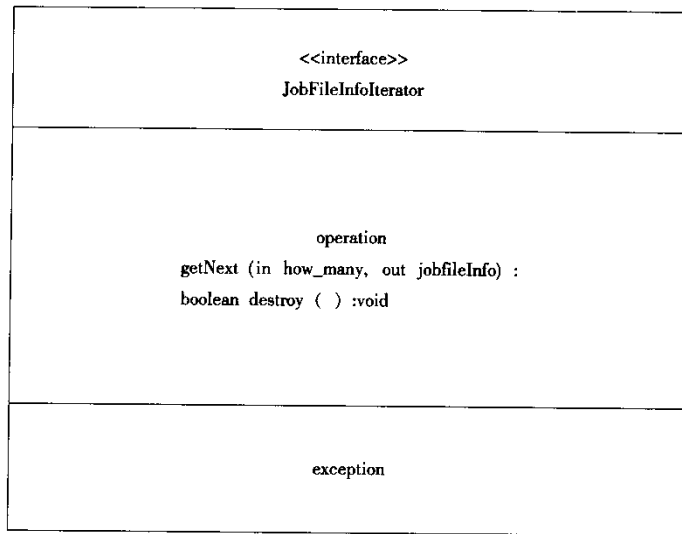


图 30 JobFileInfoterator 接口的 UML 类图

6.8.3.2 JobFileInfoterator 接口的操作

6.8.3.2.1 getNext 操作

行为

“NMC 使用该操作来获取全部或部分剩余文件信息。如果一次 getNext 操作仍没有将所有的信息全部获取完，则 NMC 可以再次调用该操作，直到所有的文件信息全部获取完为止。”

输入参数

how_many: unsigned long

输入参数 how_many 指 OMC 将通过输出参数 jobFileInfoList 返回的文件信息的个数。

输出参数

jobFileInfo: jobFileType

输出参数 jobFileInfo 包含由 listJobFiles_v2 操作所剩余的文件信息。输出的文件信息数目由输入参数 how_many 确定。如果本次调用后已没有剩余的文件信息，则输出的文件信息数目可以小于或等于有 how_many 指定的个数，根据实际的文件个数来确定。否则，输出的文件信息数应等于 how_many。在这种情况下，至少还需要一次调用。

后继条件

如果 JobFileInfoterator 对象中还有剩余的文件信息，则 result= “TRUE”；

如果 JobFileInfoterator 对象中已没有剩余的文件信息，则 result= “FALSE”。

返回值

result: boolean

返回值 result 的值表示 JobFileInfoterator 对象中是否还有剩余的文件信息。

6.8.3.2.2 destroy

行为

“当 NMC 已经获得了所有与某测量任务相关的结果文件信息后，可使用该操作释放该 JobFileInfoterato 对象。”

输入参数

无。

输出参数

无。

返回值

无。

6.8.4 性能管理场景描述

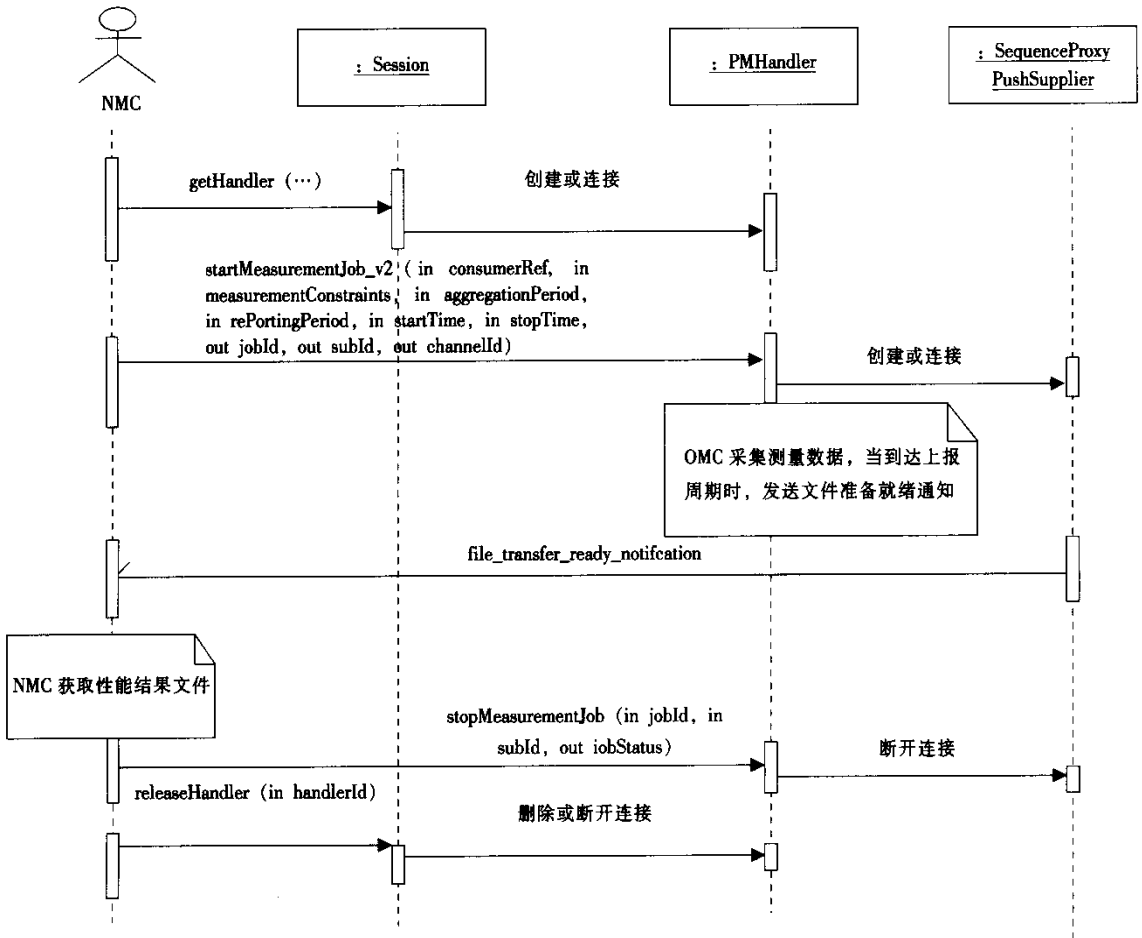


图 31 性能管理的 UML 序列图 1

图 31 描述了 NMC 如何开启一个测量任务并获取测量结果。NMC 首先调用 PMHandler_v2 中的 startMeasurementJob_v2 操作请求 OMC 开始收集性能数据。OMC 根据 startMeasurementJob_v2 操作的输入参数 reportingPeriod 的值，每隔一个上报周期时间，向 NMC 发送测量结果文件准备就绪通知。NMC 在收到通知后，使用 FTP 获取结果文件。NMC 可以调用 stopMeasurementJob 操作来结束其对此项测量任务结果的订购。该操作的输出参数 jobStatus 将返回测量任务的当前状态：停止或仍为其他 NMC 服务。

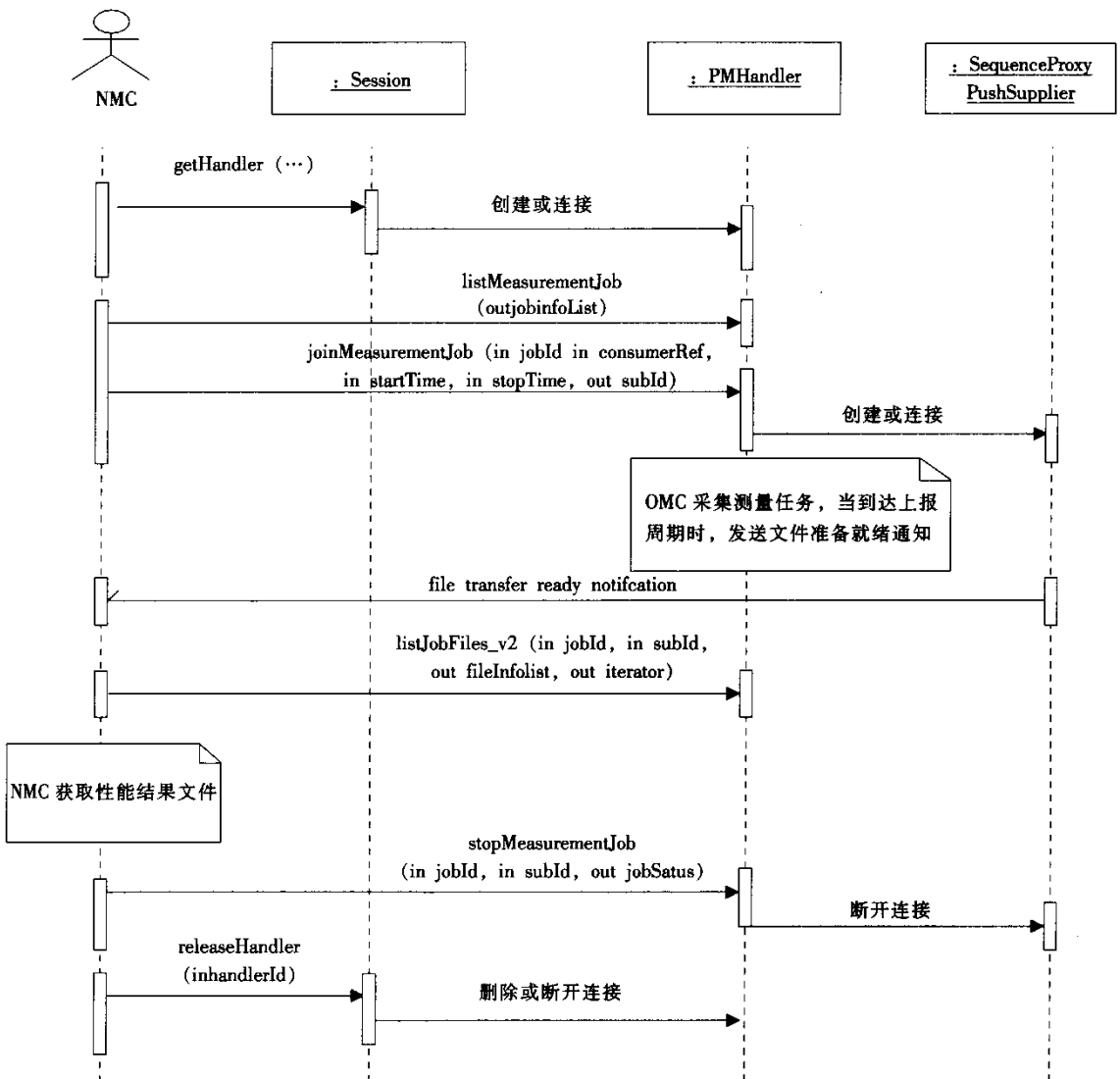


图 32 性能管理的 UML 序列图 2

图 32 描述了 NMC 如何加入一项已由其他 NMC 开启的测量任务，并获取结果文件，其前提是测量任务已由其他 NMC 所开启。首先 NMC 调用 joinMeasurementJob_v2 操作来加入测量任务，在收到 OMC 发送的结果文件准备就绪通知后，NMC 就可以使用 FTP 获取通知中指定的性能结果文件。

7 接口设计

7.1 接口定义

本节根据第 6 章的接口分析模型，给出设计阶段基于 IDL 的接口定义。接口定义是按照不同的模块来进行划分的，每个模块都是一个独立的 IDL 文件。

注：由于目前日志服务对设备厂商而言是可选的，所以与日志相关的操作也是可选的。但如果厂商能够支持日志服务，则必须按照本标准中定义的操作来实现。

7.1.1 通用模块

```

// File: NmclIfComm.idl
// Common part of the NMC-OMC interface definition. In this file, base types and
// exception that will be referenced in other modules are defined.

#ifndef _NMC_IF_COMMON_IDL_
#define _NMC_IF_COMMON_IDL_

#include <NmclIfIMBaseType.idl>

module NmclIfComm
{
    typedef string HandlerIdType;
    typedef string VersionIdType;
    typedef sequence<VersionIdType> VersionIdListType;
    typedef unsigned long SubscriptionIdType;
    typedef unsigned long SessionIdType;
    typedef NmclIfIMBaseType:: eClassIdType MOClassNameType;
    typedef string ChannelIdType;
    //In this OMC-NMC interface, class adopted is defined in Appendix D.4
    typedef string DistinguishedNameType; //string format should conform to what defined in section
                                        //2.1.1.1 of this specification.
    typedef sequence<DistinguishedNameType> DNListType;
    struct ManagedObjectType
    {
        MOClassNameType    className;
        DistinguishedNameType moDN;
    };
    typedef sequence<ManagedObjectType> ManagedObjectListType;
    typedef NmclIfIMBaseType:: eAttrIdType AttrNameType;
    //in this OMC-NMC interface, attribute that can be adopted is defined in Appendix D.4
    typedef any AttrValueType;
    typedef sequence<AttrNameType> AttrNameListType;

    struct AttrInfoType
    {
        AttrNameType name;
        // here any is used as attrValueType
        // actual value type is determined by attribute name based on
        // information defined in file NmclIfIMComm.idl
        // and all referenced types are defined in file NmclIfIMCommType.idl
        AttrValueType value;
        // value is based on the attribute name.
    };
};

```

```

typedef sequence<AttrInfoType> AttrInfoListType;
typedef string CredentialType;
enum eCredentialInvalidReasonType
{
    UNKNOWN_CREDENTIAL,
    SESSION_TIMEOUT
};

enum eSubscriptionStatusType
{
    SUBSCRIPTION_OK,
    SUBSCRIPTION_SUSPEND,
    SUBSCRIPTION_CLOSED
};

enum eResultType
{
    success,
    failure
};

enum eObjectInvalidReasonType
{
    UNKNOWN_OBJECT_CLASS,
    UNKNOWN_DISTINGUISHED_NAME
};

typedef string GeneralizedTimeType;

exception InvalidCredential {eCredentialInvalidReasonType reason;};
exception ExceedMaxNumberOfHandler {};
exception UnknownHandler {HandlerIdType handlerId;};
exception UnknownSubscription {};
// imported from CosNotifyFilter module
//exception CosNotifyFilter:: InvalidConstraint {ConstraintExp constr;};

exception UnknownAttribute {AttrNameType unknowAttr;};
exception AuthenticateFailure {};
exception UnknownObject
{
    ManagedObjectType unknownMOInstance;
    eObjectInvalidReasonType invalidReason;
};
exception UnknownLog {};

```

```

exception LogNotSupported {}; // for optional LogService
exception InvalidTimeout { string reason;}; // for unreasonable timeout values;
exception UnknownChannel {};
};

```

```

#endif

```

7.1.2 OMC 接口模块

```

// File: NmcIfOMC.idl

```

```

// This file contains interface OMC definition

```

```

#ifndef _NMC_IF_OMC_IDL_

```

```

#define _NMC_IF_OMC_IDL_

```

```

#include <NmcIfComm.idl>

```

```

#include <NmcIfSession.idl>

```

```

module NmcIfOMC

```

```

{
    exception ExceedMaxNumSession {};
    exception UnknownSession { NmcIfComm:: SessionIdType unknownSessionId;};

```

```

    interface OMC

```

```

    {
        NmcIfComm:: eResultType openSession (
            in string userName,
            in string password,
            out NmcIfComm:: SessionIdType sessionId,
            out NmcIfSession:: Session sessionRef,
            out NmcIfComm:: CredentialType credential) //content given by OMC, NMC
                //inserts to CORBA:: Context
            raises (NmcIfComm:: AuthenticateFailure,
                ExceedMaxNumSession,
                NmcIfComm:: InvalidTimeout)
            context ("timeout");
        NmcIfComm:: eResultType closeSession (
            in NmcIfComm:: SessionIdType sessionId)
            raises (NmcIfComm:: InvalidCredential,
                UnknownSession)
            context ("credential");
    };
};

```

```

#endif

```


7.1.3 Session 接口模块

```
// File: NmclSession.idl
```

```
// In this file, session interface is defined.
```

```
#ifndef _NMC_IF_SESSION_IDL_
```

```
#define _NMC_IF_SESSION_IDL_
```

```
#include <NmclComm.idl>
```

```
module NmclSession
```

```
{
```

```
    enum eHandlerType
```

```
    {
```

```
        FAULT_MANAGEMENT,
```

```
        CONFIGURATION_MANAGEMENT,
```

```
        PERFORMANCE_MANAGEMENT
```

```
    };
```

```
    interface Session
```

```
    {
```

```
        readonly attribute unsigned long timeout;
```

```
        NmclComm:: eResultType getHandler (
```

```
            in eHandlerType          handlerType,
```

```
            in NmclComm:: VersionIdType versionId,
```

```
            out NmclComm:: VersionIdListType supportVersionList,
```

```
            out NmclComm:: HandlerIdType  handlerId,
```

```
            out Object                handlerRef)
```

```
        raises (NmclComm:: ExceedMaxNumberOfHandler,
```

```
                NmclComm:: InvalidCredential,
```

```
                NmclComm:: InvalidTimeout)
```

```
        context ("credential", "timeout");
```

```
        NmclComm:: eResultType releaseHandler (
```

```
            in NmclComm:: HandlerIdType handlerId)
```

```
        raises (NmclComm:: UnknownHandler,
```

```
                NmclComm:: InvalidCredential)
```

```
        context ("credential");
```

```
    };
```

```
};
```

```
#endif
```

7.1.4 FMHandler 接口模块

```
// File: NmclFMHandler.idl
```

// This file contains interface FMHandler definition.

#ifndef _NMC_IF_FMHANDLER_IDL_

#define _NMC_IF_FMHANDLER_IDL_

#include <NmcIfComm.idl>

#include <CosNotifyChannelAdmin.idl>

#include <DsNotifyLogAdmin.idl>

#include <NmcIfHeartbeat.idl>

module NmcIfFMHandler

```
{
    interface FMHandler_v1
    {
        readonly attribute unsigned long timeout;

        NmcIfComm:: eResultType subscribeNotif (
            in CosNotifyComm:: SequencePushConsumer consumerRef,
            in CosNotifyFilter:: ConstraintExpSeq constrainList,
            out NmcIfComm:: SubscriptionIdType subId,
            out CosNotifyChannelAdmin:: SequenceProxyPushSupplier slyRef)
            raises (CosNotifyFilter:: InvalidConstraint,
                NmcIfComm:: InvalidCredential)
            context ("credential");

        NmcIfComm:: eResultType subscribeLog (
            in DsLogAdmin:: LogFullAction logFullAction,
            in unsigned long maxsize, //based on octet
            in CosNotifyFilter:: ConstraintExpSeq constrainList,
            out NmcIfComm:: SubscriptionIdType subId,
            out DsNotifyLogAdmin:: NotifyLog logRef)
            raises (NmcIfComm:: InvalidCredential,
                CosNotifyFilter:: InvalidConstraint,
                NmcIfComm:: LogNotSupported)
            context ("credential");

        NmcIfComm:: eResultType unsubscribeNotif (
            in NmcIfComm:: SubscriptionIdType subId)
            raises (NmcIfComm:: UnknownSubscription,
                NmcIfComm:: InvalidCredential)
            context ("credential");

        NmcIfComm:: eResultType unsubscribeLog (
            in NmcIfComm:: SubscriptionIdType subId)
```

```

raises (NmclComm:: UnknownSubscription,
        NmclComm:: InvalidCredential,
        NmclComm:: LogNotSupported)
context ("credential");

```

```

NmclComm:: eResultType synchronizeAlarm (
    in CosNotifyComm:: SequencePushConsumer consumerRef,
    in CosNotifyFilter:: ConstraintExpSeq constrainList,
    out CosNotifyChannelAdmin:: SequenceProxyPushSupplier slyRef,
    out NmclComm:: SubscriptionIdType subId)
raises (CosNotifyFilter:: InvalidConstraint,
        NmclComm:: InvalidCredential)
context ("credential");

```

```

NmclComm:: eResultType queryLogRecord (
    in CosNotifyComm:: SequencePushConsumer consumerRef,
    in string constrainList,
    in DsNotifyLogAdmin:: NotifyLog logRef,
    out NmclComm:: SubscriptionIdType subId)
raises (NmclComm:: UnknownLog,
        CosNotifyFilter:: InvalidConstraint,
        NmclComm:: InvalidCredential,
        NmclComm:: LogNotSupported)
context ("credential");

```

```

NmclComm:: eResultType getSubscriptionStatus (
    in NmclComm:: SubscriptionIdType subId,
    out NmclComm:: eSubscriptionStatusType subStatus)
raises (NmclComm:: UnknownSubscription,
        NmclComm:: InvalidCredential)
context ("credential");

```

```
};
```

```
interface FMHandler_v2 : FMHandler_v1
```

```
{
```

```

/*****
// The following operations inherited from FMHandler_v1 won't be used in
// FMHandler_v2, other operations and attribute are used as before:
// 1) subscribeNotif (),
// 2) subscribeLog (),
// 3) synchronizeAlarm (),
// 4) queryLogRecord ();
// which are replaced by subscribeNotif_v2 (), subscribeLog_v2 (),
// synchronizeAlarm_v2 (), and queryLogRecord_v2 () respectively;

```

// One more operation getHeartbeat () is added.

*****/

```
NmcIfComm:: eResultType subscribeNotif_v2 (
    in CosNotifyComm:: SequencePushConsumer consumerRef,
    in CosNotifyFilter:: ConstraintExpSeq constrainList,
    out NmcIfComm:: SubscriptionIdType subId,
    out CosNotifyChannelAdmin:: SequenceProxyPushSupplier slyRef,
    out NmcIfComm:: ChannelIdType channelId)
raises (CosNotifyFilter:: InvalidConstraint,
        NmcIfComm:: InvalidCredential)
context ("credential");
```

```
NmcIfComm:: eResultType subscribeLog_v2 (
    in DsLogAdmin:: LogFullAction logFullAction,
    in unsigned long maxsize, //based on octet
    in CosNotifyFilter:: ConstraintExpSeq constrainList,
    out NmcIfComm:: SubscriptionIdType subId,
    out DsNotifyLogAdmin:: NotifyLog logRef,
    out NmcIfComm:: ChannelIdType channelId)
raises (NmcIfComm:: InvalidCredential,
        CosNotifyFilter:: InvalidConstraint,
        NmcIfComm:: LogNotSupported)
context ("credential");
```

```
NmcIfComm:: eResultType synchronizeAlarm_v2 (
    in CosNotifyComm:: SequencePushConsumer consumerRef,
    in CosNotifyFilter:: ConstraintExpSeq constrainList,
    out CosNotifyChannelAdmin:: SequenceProxyPushSupplier slyRef,
    out NmcIfComm:: SubscriptionIdType subId,
    out NmcIfComm:: ChannelIdType channelId)
raises (CosNotifyFilter:: InvalidConstraint,
        NmcIfComm:: InvalidCredential)
context ("credential");
```

```
NmcIfComm:: eResultType queryLogRecord_v2 (
    in CosNotifyComm:: SequencePushConsumer consumerRef,
    in string constrainList,
    in DsNotifyLogAdmin:: NotifyLog logRef,
    out NmcIfComm:: SubscriptionIdType subId,
    out NmcIfComm:: ChannelIdType channelId)
raises (NmcIfComm:: UnknownLog,
        CosNotifyFilter:: InvalidConstraint,
        NmcIfComm:: InvalidCredential,
```

```

        NmcIfComm:: LogNotSupported)
    context ("credential");

    NmcIfComm:: eResultType getHeartbeat (
        in NmcIfComm:: ChannelIdType channelId,
        out NmcIfHeartbeat:: cTelHeartbeat heartbeatRef)
    raises (NmcIfComm:: UnknownChannel,
        NmcIfComm:: InvalidCredential)
    context ("credential");
};

};

#endif

```

7.1.5 PMHandler 接口模块

```

// File: NmcIfPMHandler.idl
// This file contains interface PMHandler definition.

#ifndef _NMC_IF_PM_HANDLER_IDL_
#define _NMC_IF_PM_HANDLER_IDL_
#include <CosNotifyComm.idl>
#include <NmcIfComm.idl>
#include <NmcIfIMMeasurement.idl>
#include <NmcIfIMCommType.idl>
#include <NmcIfHeartbeat.idl>

module NmcIfPMHandler
{
    exception AlreadySubscribed {};
    exception InvalidStartTime {};
    exception InvalidStopTime {};
    typedef unsigned long JobIdType;
    typedef string GeneralizedTimeType; //format conforms to GeneralizedTime defined in ASN.1
    typedef unsigned long AggregationPeriodType; // time period is based on 15minutase.
    typedef unsigned long ReportingPeriodType; //time period is based on 15 minutes.
    typedef sequence<NmcIfIMMeasurement:: mAttrIdType>MeasurementAttrListType;

    typedef CosNotification:: PropertySeq PropertyListType;

    const string ObservedCell = "ObservedCell";
    const string AdjacentCell = "AdjacentCell";
    // for use of value in CosNotification:: Property.

```

```

enum eMeasurementCategoryType
{
    circuitEndpointSubGroup, // 0
    observedDestination, // 1
    trafficFlow, // 2
    bscHDOStatistics, // 3
    bscTrafficStatistics, // 4
    bscProcessorLoad, // 5
    bscGRPSSubLayerTraffic, // 6
    btsTrafficData, // 7
    powerLevelOfBS, // 8
    btsHDOStatistics, // 9
    btsLocationUpdate, // 10
    btsGPRSRadioResource, // 11
    btsGPRSPacketService, // 12
    eirPerformanceStatistics, // 13
    eirProcessorLoad, // 14
    handoverObservation, // 15
    hlrPerformanceStatistics, // 16
    hlrProcessorLoad, // 17
    mscPerformance, // 18
    mscQoS, // 19
    mscProcessorLoad, // 20
    ssfTraffic, // 21
    smsPerformanceStatistics, // 22
    smsProcessorLoad, // 23
    vlrPerformanceStatistics, // 24
    vlrProcessorLoad, // 25
    signallingLinkSetTPTraffic // 26
};

```

```

struct MeasurementInfoType
{
    NmcIfComm::ManagedObjectType moInstance;
    //if only the class name is given, the measurement job will be applied to all instances of the classe
    MeasurementCategoryType measurementCategory;
    MeasurementAttrListType measurementAttrNameList;
    // if the length of measurementAttrNameList is 0, it means
    // all the attributeName for the specified measurementJob.
    PropertyListType additionalInfo;
    // optional. Can be null, but in special cases can contain special property
    // for example, in handoverObservation measurement it should contain
    // the observed cell and adjacent cell.
    // set the "ObservedCell" or "AdjacentCell" constant in Property.name

```

```

    // and set the value of "NmclfIMCommType:: CellGlobalIdType" in Property.value,
    // which specifies the cell Id.
};

typedef sequence<MeasurementInfoType> MeasurementInfoListType;
enum eJobStatusType
{
    JOB_OK,
    JOB_FAIL,
    JOB_TO_BE_RUN,
    JOB_DISCONNECTED_BUT_STILL_RUNNING,
    JOB_STOPPED
};

struct JobInfoType
{
    JobIdType      jobId;
    AggregationPeriodType aggregationPeriod;
    ReportingPeriodType reportingPeriod;
    MeasurementInfoListType measurementInfoList;
    GeneralizedTimeType firstStartTime; //specifies time when first NMC started the job
                                         // or the earliest time when the job will be started
    GeneralizedTimeType lastStopTime; //specifies the latest time when the job will
                                         // be stopped to several NMC
    eJobStatusType  jobStatus;
};
typedef sequence<JobInfoType> JobInfoListType;
struct FileInfoType
{
    string          fileDirectory;
    string          fileName;
    unsigned long fileSize;
    string          fileCompression; // empty string means no compression.
    GeneralizedTimeType creationTime;
    GeneralizedTimeType deletionTime; // the file should be held for at least two weeks.
};
typedef sequence<FileInfoType> FileInfoListType;

struct JobFileInfoType
{
    JobIdType      jobId;
    string         serverAddr;
    string         userName;
    string         passwd;
};

```

```

        FileInfoListType relatedFileList;
    };

    exception UnknownJob { JobIdType unknownJobId;};
    exception InvalidMeasurementInfo { MeasurementInfoType invalidInfo;};

    interface PMHandler_v1
    {
        readonly attribute unsigned long timeout;

        NmcIfComm:: eResultType startMeasurementJob (
            in CosNotifyComm:: SequencePushConsumer consumerRef,
            in MeasurementInfoListType measurementInfoList,
            in AggregationPeriodType aggregationimePeriod,
            in ReportingPeriodType reportingPeriod,
            in GeneralizedTimeType startTime,
            in GeneralizedTimeType stopTime,
            out JobIdType jobId,
            out NmcIfComm:: SubscriptionIdType subId)
            raises ( InvalidMeasurementInfo,
                InvalidStartTime,
                InvalidStopTime,
                NmcIfComm:: InvalidCredential)
            context ("credential");

        NmcIfComm:: eResultType stopMeasurementJob (
            in NmcIfComm:: SubscriptionIdType subId,
            in JobIdType jobId,
            out eJobStatusType jobStatus)
            raises (UnknownJob,
                NmcIfComm:: UnknownSubscription,
                NmcIfComm:: InvalidCredential)
            context ("credential");

        NmcIfComm:: eResultType joinMeasurementJob (
            in JobIdType jobId,
            in CosNotifyComm:: SequencePushConsumer consumerRef,
            in GeneralizedTimeType startTime,
            in GeneralizedTimeType stopTime,
            out NmcIfComm:: SubscriptionIdType subId)
            raises (UnknownJob,
                InvalidStartTime,
                InvalidStopTime,
                NmcIfComm:: InvalidCredential)
    }

```



```

context ("credential");

NmclIfComm:: eResultType listMeasurementJob (
    out JobInfoListType jobInfoList)
    raises (NmclIfComm:: InvalidCredential)
    context ("credential");

NmclIfComm:: eResultType listJobFiles (
    in JobIdType          jobId,
    in NmclIfComm:: SubscriptionIdType subId,
    out JobFileInfoType  jobFileInfo)
    raises (UnknownJob,
           NmclIfComm:: UnknownSubscription,
           NmclIfComm:: InvalidCredential)
    context ("credential");
};

interface JobFileInfoIterator;

interface PMHandler_v2 : PMHandler_v1
{
    /*****
    // The following operations inherited from PMHandler_v1 won't be used in
    // PMHandler_v2, other operations and attribute are used as before:
    // 1) startMeasurementJob (),
    // 2) joinMeasurementJob (),
    // 3) listJobFiles (),
    // whcih are replaced by startMeasurementJob_v2 (), joinMeasurementJob_v2 (),
    // listJobFiles_v2 (), respectively;
    // One more operation getHeartbeat () is added.
    *****/

    NmclIfComm:: eResultType startMeasurementJob_v2 (
        in CosNotifyComm:: SequencePushConsumer consumerRef,
        in MeasurementInfoListType measurementInfoList,
        in AggregationPeriodType aggregationimePeriod,
        in ReportingPeriodType reportingPeriod,
        in GeneralizedTimeType startTime,
        in GeneralizedTimeType stopTime,
        out JobIdType          jobId,
        out NmclIfComm:: SubscriptionIdType subId,
        out NmclIfComm:: ChannelIdType channelId)
        raises (InvalidMeasurementInfo,
              InvalidStartTime,

```

```

        InvalidStopTime,
        NmcIfComm:: InvalidCredential)
    context ("credential");

```

```

NmcIfComm:: eResultType joinMeasurementJob_v2 (
    in JobIdType jobId,
    in CosNotifyComm:: SequencePushConsumer consumerRef,
    in GeneralizedTimeType startTime,
    in GeneralizedTimeType stopTime,
    out NmcIfComm:: SubscriptionIdType subId,
    out NmcIfComm:: ChannelIdType channelId)
    raises (UnknownJob,
        InvalidStartTime,
        InvalidStopTime,
        NmcIfComm:: InvalidCredential)
    context ("credential");

```

```

NmcIfComm:: eResultType listJobFiles_v2 (
    in JobIdType jobId,
    in NmcIfComm:: SubscriptionIdType subId,
    in unsigned long how_many,
    out JobFileInfoType jobFileInfo,
    out JobFileInfolterator iterator)
    raises (UnknownJob,
        NmcIfComm:: UnknownSubscription,
        NmcIfComm:: InvalidCredential)
    context ("credential");

```

```

NmcIfComm:: eResultType getHeartbeat (
    in NmcIfComm:: ChannelIdType channelId,
    out NmcIfHeartbeat:: cTelHeartbeat heartbeatRef)
    raises (NmcIfComm:: UnknownChannel,
        NmcIfComm:: InvalidCredential)
    context ("credential");

```

```
};
```

```
interface JobFileInfolterator
```

```
{
```

```

    boolean getNext (
        in unsigned long how_many,
        out JobFileInfoType jobFileInfo);

```

```

    void destroy ();

```

```
};
```

```
};
```

```
#endif
```

7.1.6 CMHandler 接口模块

```
// File: NmcIfCMHandler.idl
```

```
// This file contains idl definition for CMHandler interface
```

```
#ifndef _NMC_IF_CMHANDLER_IDL_
```

```
#define _NMC_IF_CMHANDLER_IDL_
```

```
#include <NmcIfComm.idl>
```

```
#include <CosNotifyChannelAdmin.idl>
```

```
#include <DsNotifyLogAdmin.idl>
```

```
#include <NmcIfHeartbeat.idl>
```

```
module NmcIfCMHandler
```

```
{
```

```
    typedef unsigned long TransactionIdType;
```

```
    interface CMHandler_v1
```

```
    {
```

```
        readonly attribute unsigned long timeout; //count in minutes
```

```
        NmcIfComm:: eResultType getTopology (
            in CosNotifyComm:: SequencePushConsumer consumerRef,
            out TransactionIdType tranId)
            raises (NmcIfComm:: InvalidCredential)
            context ("credential");
```

```
        NmcIfComm:: eResultType synchronizeCM (
            in CosNotifyComm:: SequencePushConsumer consumerRef,
            in NmcIfComm:: ManagedObjectType moInstance,
            out TransactionIdType tranId)
            raises (NmcIfComm:: InvalidCredential,
                NmcIfComm:: UnknownObject)
            context ("credential");
```

```
        NmcIfComm:: eResultType getMOHandler (
            in NmcIfComm:: ManagedObjectType moInstance,
            in NmcIfComm:: VersionIdType versionId,
            out NmcIfComm:: VersionIdListType supportVersionList,
            out NmcIfComm:: HandlerIdType handlerId,
            out Object moHandlerRef)
```

```

raises (NmclIfComm:: InvalidCredential,
       NmclIfComm:: UnknownObject,
       NmclIfComm:: ExceedMaxNumberOfHandler,
       NmclIfComm:: InvalidTimeout)
context ("credential", "timeout");

```

```

NmclIfComm:: eResultType releaseMOHandler (
    in NmclIfComm:: HandlerIdType handlerId)
raises (NmclIfComm:: InvalidCredential,
       NmclIfComm:: UnknownHandler)
context ("credential");

```

```

NmclIfComm:: eResultType subscribeNotif (
    in CosNotifyComm:: SequencePushConsumer consumerRef,
    in CosNotifyFilter:: ConstraintExpSeq constraintList,
    out NmclIfComm:: SubscriptionIdType subId,
    out CosNotifyChannelAdmin:: SequenceProxyPushSupplier slyRef)
raises (CosNotifyFilter:: InvalidConstraint,
       NmclIfComm:: InvalidCredential)
context ("credential");

```

```

NmclIfComm:: eResultType unsubscribeNotif (
    in NmclIfComm:: SubscriptionIdType subId)
raises (NmclIfComm:: UnknownSubscription,
       NmclIfComm:: InvalidCredential)
context ("credential");

```

```

NmclIfComm:: eResultType subscribeLog (
    in DsLogAdmin:: LogFullAction logFullAction,
    in unsigned long maxsize,
    //maxsize is based on octet
    in CosNotifyFilter:: ConstraintExpSeq constraintList,
    out NmclIfComm:: SubscriptionIdType subId,
    out DsNotifyLogAdmin:: NotifyLog logRef)
raises (NmclIfComm:: InvalidCredential,
       CosNotifyFilter:: InvalidConstraint,
       NmclIfComm:: LogNotSupported)
context ("credential");

```

```

NmclIfComm:: eResultType unsubscribeLog (
    in NmclIfComm:: SubscriptionIdType subId)
raises (NmclIfComm:: UnknownSubscription,
       NmclIfComm:: InvalidCredential,
       NmclIfComm:: LogNotSupported)

```

```

context ("credential");

NmclIfComm:: eResultType queryLogRecord (
    in CosNotifyComm:: SequencePushConsumer consumerRef,
    in string constrainList,
    in DsNotifyLogAdmin:: NotifyLog logRef,
    out NmclIfComm:: SubscriptionIdType subId)
raises (NmclIfComm:: UnknownLog,
        CosNotifyFilter:: InvalidConstraint,
        NmclIfComm:: InvalidCredential,
        NmclIfComm:: LogNotSupported)
context ("credential");

NmclIfComm:: eResultType getSubscriptionStatus (
    in NmclIfComm:: SubscriptionIdType subId,
    out NmclIfComm:: eSubscriptionStatusType subStatus)
raises (NmclIfComm:: UnknownSubscription,
        NmclIfComm:: InvalidCredential)
context ("credential");
};

interface CMHandler_v2 : CMHandler_v1
{
    /******
    // The following operations inherited from CMHandler_v1 won't be used in
    // CMHandler_v2, other operations and attribute are used as before:
    // 1) getTopology (),
    // 1) synchronizeCM (),
    // 2) subscribeNotif (),
    // 3) subscribeLog (),
    // 4) queryLogRecord ();
    // whcih are replaced by getTopology_v2 (), synchronizeCM_v2 (),
    // subscribeNotif_v2 (), subscribeLog_v2 (),
    // and queryLogRecord_v2 () respectively;
    // One more operation getHeartbeat () is added.
    *****/

    NmclIfComm:: eResultType getTopology_v2 (
        in CosNotifyComm:: SequencePushConsumer consumerRef,
        out TransactionIdType tranId,
        out NmclIfComm:: ChannelIdType channelId)
raises (NmclIfComm:: InvalidCredential)
context ("credential");

```

```

NmclIfComm:: eResultType synchronizeCM_v2 (
    in CosNotifyComm:: SequencePushConsumer consumerRef,
    in NmclIfComm:: ManagedObjectType moInstance,
    out TransactionIdType tranId,
    out NmclIfComm:: ChannelIdType channelId)
raises (NmclIfComm:: InvalidCredential,
        NmclIfComm:: UnknownObject)
context ("credential");

NmclIfComm:: eResultType subscribeNotif_v2 (
    in CosNotifyComm:: SequencePushConsumer consumerRef,
    in CosNotifyFilter:: ConstraintExpSeq constraintList,
    out NmclIfComm:: SubscriptionIdType subId,
    out CosNotifyChannelAdmin:: SequenceProxyPushSupplier slyRef,
    out NmclIfComm:: ChannelIdType channelId)
raises (CosNotifyFilter:: InvalidConstraint,
        NmclIfComm:: InvalidCredential)
context ("credential");

NmclIfComm:: eResultType subscribeLog_v2 (
    in DsLogAdmin:: LogFullAction logFullAction,
    in unsigned long maxsize,
    //maxsize is based on octet
    in CosNotifyFilter:: ConstraintExpSeq constraintList,
    out NmclIfComm:: SubscriptionIdType subId,
    out DsNotifyLogAdmin:: NotifyLog logRef,
    out NmclIfComm:: ChannelIdType channelId)
raises (NmclIfComm:: InvalidCredential,
        CosNotifyFilter:: InvalidConstraint,
        NmclIfComm:: LogNotSupported)
context ("credential");

NmclIfComm:: eResultType queryLogRecord_v2 (
    in CosNotifyComm:: SequencePushConsumer consumerRef,
    in string constrainList,
    in DsNotifyLogAdmin:: NotifyLog logRef,
    out NmclIfComm:: SubscriptionIdType subId,
    out NmclIfComm:: ChannelIdType channelId)
raises (NmclIfComm:: UnknownLog,
        CosNotifyFilter:: InvalidConstraint,
        NmclIfComm:: InvalidCredential,
        NmclIfComm:: LogNotSupported)
context ("credential");

```

```

NmclIfComm:: eResultType getHeartbeat (
    in NmclIfComm:: ChannelIdType channelId,
    out NmclIfHeartbeat:: cTelHeartbeat heartbeatRef)
raises (NmclIfComm:: UnknownChannel,
        NmclIfComm:: InvalidCredential)
context ("credential");
};

```

```
};
```

```
#endif
```

7.1.7 MOHandler 接口模块

```
// File: NmclMOHandler.idl
```

```
// This file contains idl definition for MOHandler interface
```

```
#ifndef _NMC_IF_MANAGED_OBJECT_HANDLER_IDL_
```

```
#define _NMC_IF_MANAGED_OBJECT_HANDLER_IDL_
```

```
#include <NmclIfComm.idl>
```

```
module NmclIfMOHandler
```

```
{
```

```
    exception InvalidScope {};
```

```
    enum eScopeType
```

```
    {
```

```
        BaseLevel,
```

```
        IndividualLevel,
```

```
        BaseToNLevel,
```

```
        WholeSubTree
```

```
    };
```

```
    exception ScopeNotSupported
```

```
    {
```

```
        eScopeType scope;
```

```
        string reason;
```

```
    };
```

```
    interface MOHandler_v1
```

```
    {
```

```
        readonly attribute unsigned long timeout ;
```

```
        NmclIfComm:: eResultType getContainment (
```

```
            in NmclIfComm:: ManagedObjectType moInstance,
```

```
            in unsigned long depth,
```

```
            in eScopeType scope,
```

```
            out NmclIfComm:: ManagedObjectListType moInstanceList)
```

```

        raises (NmclIfComm:: UnknownObject,
               InvalidScope,
               ScopeNotSupported,
               NmclIfComm:: InvalidCredential)
        context ("credential");
NmclIfComm:: eResultType getAttribute (
    in NmclIfComm:: ManagedObjectType moInstance,
    in NmclIfComm:: AttrNameListType attrNameList,
    out NmclIfComm:: AttrInfoListType attrInfoList)
    raises (NmclIfComm:: UnknownObject,
           NmclIfComm:: InvalidCredential)
    context ("credential");
};
};
/** *****
/** * It is up to vendor to adopt different approaches to implement managed object
/** * modeling , any of the three approaches are ok.
/** * specific object handler can be derived from the MOHandler interface with special
/** * operations or attributes.
/** * for example :
/** *     interface MSCClassHandler : MOHandler
/** *     {
/** *         // MSC class specificdc attributes operations can be added here.
/** *     };
/** *****
#endif

```

7.1.8 cTelHeartbeat 接口模块

```

// File: Heartbeat.idl
// This file contains interface Heartbeat definition

```

```

#ifndef _NMC_IF_HEARTBEAT_IDL_
#define _NMC_IF_HEARTBEAT_IDL_

```

```

module NmclIfHeartbeat

```

```

{
/**
This interface defines a service used to periodically test the
operation of the notification channels on a system. The service
supporting this interface periodically emits a short "heartbeat"
notification on each channel on the system.
*/

/**

```


The InvalidPeriod exception is used by OMC when the period value to be set by NMC is not a reasonable in OMC's implementation. A very short period may cause OMC to send many heartbeatReport in a short time, which may decrease the performance of OMC. To prevent this, OMC may set the lower limit period in its system implementation.

When the period to be set is shorter the lower limit period, OMC may throw this exception and reject to set the period to new value.

Note: set the period to zero must be allowed. The behaviour of setting period to zero will be described below.

*/

```
exception InvalidPeriod
```

```
{
```

```
    unsigned short periodLowerLimit;
```

```
    string reason;
```

```
};
```

```
interface cTelHeartbeat {
```

```
    /** The systemLabel attribute is sent in heartbeat
    notifications. It is used to identify the heartbeat service
    instance from which the notification came. Resetting this does
    not cause the service to immediately emit a notification, but
    the new value will be sent with the next notification. */
```

```
    attribute string systemLabel;
```

```
    /** The period is the interval, in seconds, at which the
    heartbeat service emits the heartbeat notification. If it is
    zero, the service does not emit notifications. */
```

```
    unsigned short periodGet ();
```

```
    /** Updating of the period shall cause the service to deliver a
    notification to all channels with the new period value and then
    begin a new period. Setting the period to zero shall cause the
    service to emit one final notification with a period value of
    zero, then no more (until the period is reset) . An attempt to
    set the period to a value outside the range supported will
    result in an ApplicationError with the error code set to
    invalidParameter. */
```

```
    void periodSet (in unsigned short period)
        raises (InvalidPeriod);
```

```
/** This operation signature defines the notification emitted  
    by the heartbeat service.  
    @param systemLabel the current value of the Heartbeat  
    service systemLabel attribute.  
    @param channelId the Identifier of the channel through which the  
        notification was sent.  
    @param period the current value of the Heartbeat  
        service period attribute.  
    @param timeStamp the current time when the notification  
        is emitted.
```

```
*/
```

```
}; // end of Heartbeat interface
```

```
};
```

```
#endif
```

附录 A
(资料性附录)
接口协议栈描述

A.1 简介

该附录给出了基于 CORBA 环境的 NMC-OMC 接口所需的协议栈。

A.2 协议栈描述

NMC-OMC 接口的协议栈体系结构如图 A.1 所示。

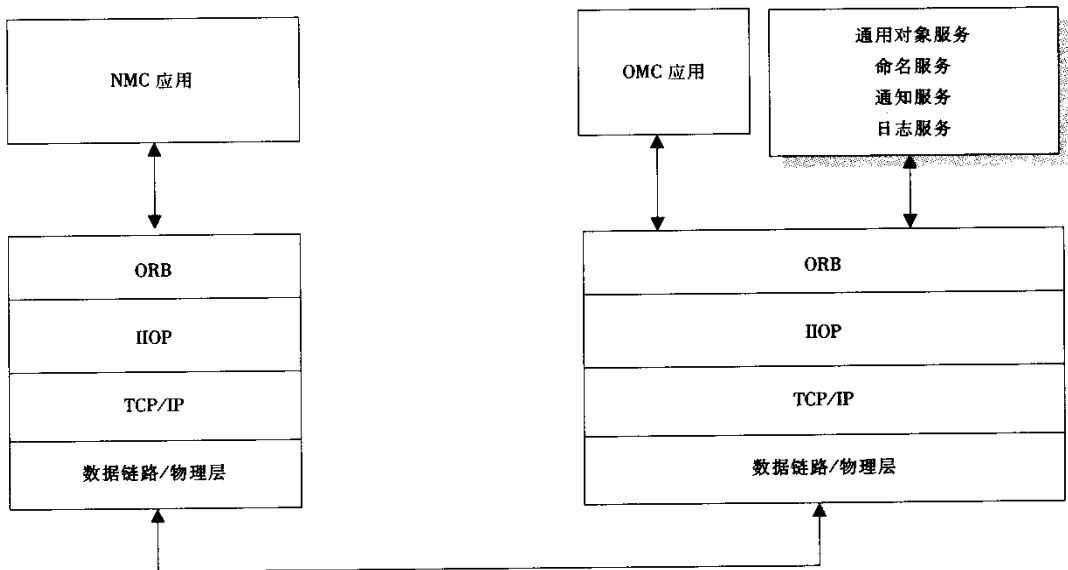


图 A.1 NMC-OMC 接口的协议栈体系结构

NMC 和 OMC 的应用进程运行于 ORB 之上，ORB 间的通信采用了运行于 TCP/IP 之上的 IIOP。本标准中没有定义采用何种数据链路层和物理层协议。只要可以支持 TCP/IP，任何数据链路层和物理层协议均可以适用。

NMC 和 OMC 的应用系统都基于 CORBA 的公共对象服务 (Common Object Service)，如命名服务、通知服务、日志服务等。

附录 B
(规范性附录)
通知格式定义

B.1 简介

本附录定义了该接口所支持通知的格式。由于本标准中的事件服务是基于 CORBA 事件服务和通知服务来实现的，因此通知的格式遵守 CORBA 通知服务中所定义的通知的格式。

B.2 所支持的通知类型

在本接口中支持以下通知：

- 对象创建通知；
- 对象删除通知；
- 对象属性值改变通知；
- 对象状态改变通知；
- 告警通知（包括新告警产生、告警变化和告警清除）；
- 批量数据传送准备就绪通知；
- 批量数据传送准备错误通知；
- 告警总结汇报通知；
- 心跳通知；
- 请求配置信息同步通知。

B.3 结构化事件的信息

本节定义了 NMC-OMC 接口在 CORBA 环境中通知的格式及属性，OMC-NMC 通知是基于 OMC 通知服务中定义的结构化事件而实现的，其各部分组成如下：

事件头部 (Header)

- 固定事件头 (fixed_header)：
 - 事件类型 (event_type)，包括域名 (domain_name) 和类型名 (type_name)；
 - 事件名 (event_name)。
- 可变事件头 (variable_header)。

主体 (Body)

- 可滤过体域 (filterable_data)；
- 保留体 (remainder_of_body)。

结构化事件的参数说明见表 B.1。

表 B.1 结构化事件的参数说明

字段名称	子参数名称	类型	取值
fixed_header	domain_name	string	标识了事件所属的工业领域。 在本标准中，可以取以下值： GSM_CM_NERWORK_MGMT, GSM_FM_NERWORK_MGMT, GSM_PM_NERWORK_MGMT

表 B.1 (续)

字段名称	子参数名称	类型	取值
fixed_header	type_name	string	标识了给定管理域中的事件类型。例如，可以取值 NOTIFY_STATE_CHANGE、NOTIFY_PM_TRANSFER_UP_READY 等
	event_name	string	在 OMC 中惟一地标识该事件。这里，事件名又可称为通知标识符，它与 systemName 一起可以在 GSM 网管领域中惟一标识一个通知
variable_header	—	名值对序列	可选
filterable_data	—	名值对序列	该域包含了名值对 (name-value pair) 序列。各通知中的名值对将在下面通知的各个表中定义
remainder_of_body	—	any	可选

B.4 通知格式

B.4.1 对象创建通知格式

管理对象创建通知的固定事件头见表 B.2。

表 B.2 管理对象创建通知的固定事件头

名称	取值
domain_name	GSM_CM_NETWORK_MGMT
type_name	NOTIFY_OBJECT_CREATION
event_name	<通知标识符>

管理对象创建通知的可滤过体域见表 B.3。

表 B.3 管理对象创建通知的可滤过体域

名称	类型	注解
SystemName	string	必选。标识了产生该通知的 OMC
MOInstance	ManagedObjectType	必选。标识了所创建的管理对象实例
AttributeList	AttrInfoListType	必选。给出了该管理对象实例的属性值列表

B.4.2 对象删除通知格式

管理对象删除通知的固定事件头见表 B.4。

表 B.4 管理对象删除通知的固定事件头

名称	取值
domain_name	GSM_CM_NETWORK_MGMT
type_name	NOTIFY_OBJECT_DELETION
event_name	通知标识符

管理对象删除通知的可滤过体域见表 B.5。

表 B.5 管理对象删除通知的可滤过体域

名称	类型	注解
SystemName	string	必选。标识了产生该通知的 OMC
MOInstance	ManagedObjectType	必选。标识了所删除的管理对象实例

B.4.3 对象属性值/状态值改变通知格式

对象属性值/状态值改变通知的固定事件头见表 B.6。

表 B.6 对象属性值/状态值改变通知的固定事件头

名称	取值
domain_name	GSM_CM_NETWORK_MGMT
type_name	NOTIFY_ATTRIBUTE_VALUE_CHANGE/NOTIFY_STATE_CHANGE
event_name	<通知标识符>

对象属性值/状态值改变通知的可滤过体域见表 B.7。

表 B.7 对象属性值/状态值改变通知的可滤过体域

名称	类型	注解
SystemName	string	必选。标识了产生该通知的 OMC
MOInstance	ManagedObjectType	必选。标识了属性值/状态值发生改变的管理对象实例
AttrChangeDef	AttrChangeInfoListType	必选。给出了发生改变的属性或的相关信息。其类型为结构 attrChangeInfo 的序列，每个 attrChangeInfo 由属性名，原属性值（可选）和新属性值组成

B.4.4 故障告警对象通知格式

故障管理告警通知的固定事件头见表 B.8。

表 B.8 故障管理告警通知的固定事件头

名称	取值
domain_name	GSM_FM_NETWORK_MGMT
type_name	NOTIFY_NEW_ALARM/ NOTIFY_ALARM_CHANGED/ NOTIFY_ALARM_CLEARED
event_name	<通知标识符>

故障管理告警通知的可滤过体域见表 B.9。

表 B.9 故障管理告警通知的可滤过体域

名称	类型	注释
SystemName	string	必选。标识了产生该通知的 OMC
AlarmId	string	必选。在所有的 OMC 中产生的告警中惟一标识了该告警
MOInstance	ManagedObjectType	必选。标识了产生告警的管理对象
EventTime	string	必选。指定了产生该告警的时间，其格式为定义在 X.208 中的 GeneralizedTime
AlarmType	eAlarmType	必选。指明告警的类型。 可以为定义在 X.733 中的下列告警： Communication_alarm Quality_of_service_alarm Processing_error_alarm Equipment_alarm Environmental_alarm
ProbableCause	string	必选。为数字型字符串，标识了产生告警的原因。告警原因定义在附录 E 中。当该值为“1000000”时，表示告警原因是由厂商自定义的。在该通知的 specificProbelem 域中可以得到真实的告警原因
PerceivedSeverity	eSeverityType	必选。表示告警级别，可以有下面的取值： ——INDETERMINATE； ——CRITICAL； ——MAJOR； ——MINOR； ——WARNING； ——CLEARED。
SpecificProblem	string	条件可选。指出引起告警的由厂商自定义的原因（非本标准定义）。其类型为字符串，格式为“域名，域内定义的原因”。其中域名为厂商名，告警原因可以只是一个整数表示的原因
CorrelatedNotifications	AlarmIdentifierListType	条件可选。是一个由结构 AlarmIdentifier 所确定的告警组成的序列，标识了一系列与和告警相关的告警事件。结构 AlarmIdentifier 由 systemName 和 alarmId 构成。如果告警级别为“CLEARED”则表示一系列被消除的告警。对于发生改变的告警，则表示一系列收影响的告警
AdditionalInfo	AddiPropertyListType	条件可选。是一个名值对的列表，描述一些附加的信息。目前，建议在该域中填入下面的子域，分别为： <ul style="list-style-type: none"> • locatingInfo: 网络实体产生该告警事件的具体时间； • originalType: 设备厂商指配的原始告警类型； • originalSeverity: 设备厂商指配的原始告警级别； • detailedContext: 设备厂商提供的详细告警上下文； • repairProposal: 设备厂商提供的维修建议。 上面的每一项都对应于 AdditionalInfo 中的一个名值对，其中在“name”域中应填入“locatingInfo”，“originalType”、“originalSeverity”、“detailedContext”或“repairProposal”，并将其相应的描述采用字符串类型填在“value”域中

B.4.5 批量数据传输准备就绪通知

批量数据传输准备就绪通知的固定事件头见表 B.10。

表 B.10 批量数据传输准备就绪通知的固定事件头

名称	取值
domain_name	GSM_CM_NETWORK_MGMT/ GSM_FM_NETWORK_MGMT/ GSM_PM_NETWORK_MGMT
type_name	NOTIFY_PM_TRANSFER_UP_READY / NOTIFY_FM_SYNC_TRANSFER_UP_READY/ NOTIFY_CM_SYNC_TRANSFER_UP_READY / NOTIFY_TOP_SYNC_TRANSFER_UP_READY/ NOTIFY_CM_LOG_RECORD_TRANSFER_UP_READY/ NOTIFY_FM_LOG_RECORD_TRANSFER_UP_READY
event_name	<通知标识符>

批量数据传输准备就绪通知的可滤过体域见表 B.11。

表 B.11 批量数据传输准备就绪通知的可滤过体域

名称	类型	注释
SystemName	string	必选。标识了产生该通知的 OMC
TransferId	string	必选。传输工作的标识符，可以为 tranId, subId 或 jobId
JobFileInfoList	FileInfoListType	必选。包含所有和结果文件相关的信息。其类型为结构 FileInfo 的序列。结构 FileInfo 中包括文件所在目录、文件名、文件大小、文件压缩信息、文件生成时间和文件的预计删除时间等信息
IPAddr	string	必选。存放结果文件的主机的 IP 地址
UserName	string	必选。传输文件所使用 FTP 服务登录时的用户名
Password	string	必选。传输文件所使用 FTP 服务登录时，与上述用户名相对应的口令

B.4.6 文件传输准备错误通知

批量数据传输准备错误通知的固定事件头见表 B.12。

表 B.12 批量数据传输准备错误通知的固定事件头

Name	VALUE
domain_name	GSM_CM_NETWORK_MGMT/ GSM_PM_NETWORK_MGMT/ GSM_FM_NETWORK_MGMT

表 B.12 (续)

Name	VALUE
type_name	NOTIFY_TOP_SYNC_TRANSFER_PREPARATION_ERROR/ NOTIFY_FM_SYNC_TRANSFER_PREPARATION_ERROR/ NOTIFY_CM_SYNC_TRANSFER_PREPARATION_ERROR/ NOTIFY_PM_TRANSFER_PREPARATION_ERROR/ NOTIFY_CM_LOG_RECORD_TRANSFER_PREPARATION_ERROR/ NOTIFY_FM_LOG_RECORD_TRANSFER_PREPARATION_ERROR
event_name	<通知标识符>

批量数据传输准备错误通知的可滤过体域见表 B.13。

表 B.13 批量数据传输准备错误通知的可滤过体域

名称	类型	注释
SystemName	string	必选。标识了产生该通知的 OMC
TransferId	string	必选。传输工作的标识符，可以为 tranId、subId 或 jobId
ProbableCause	string	必选。指示文件传输准备错误的失败原因。可能的原因定义在 X.721 中
PerceivedSeverity	eSeverityType	必选。给出了错误级别，可以为 MAJOR（主要）、MINOR（次要）WARNING（提示）
AdditionalText	string	可选。可以以字符串的形式提供了关于错误原因的更详细信息

B.4.7 当前告警总结汇报通知

当前告警总结汇报通知的固定事件头见表 B.14。

表 B.14 当前告警总结汇报通知的固定事件头

名称	取值
domain_name	GSM_FM_NETWORK_MGMT
type_name	NOTIFY_CURRENT_ALARM_SUMMARY_REPORT
event_name	<通知标识符>

当前告警总结汇报通知的可滤过体域见表 B.15。

表 B.15 当前告警总结汇报通知的可滤过体域

名称	类型	注释
SystemName	string	必选。标识了产生该通知的 OMC
AlarmSummary	AlarmSummaryDataType	必选。给出了 OMC 中当前所有活跃告警的信息。详细的类型定义请参见附录 B.5
SubscriptionId	SubscriptionIdType	必选。指该通知与那个告警同步操作相关联。其取值应与相应的“synchronizeAlarm”操作的输出参数 subId 的值相一致

表 B.15 (续)

名 称	类 型	注 释
IsLast	boolean	必选。如果本通知是当前告警总结汇报的最后一个通知（或只有一个），则应取值为 TRUE；否则，它应为 FALSE，表示还有后继的通知来传送剩余的信息
FormerNotificationId	string	条件可选。当本通知不是当前告警总结汇报的第一个通知时该域就应存在。它给出了上一个通知的标识符，以便 NMC 可以通过将一系列的拆分后通知重新组装成一个当前告警同步汇报
UnsuccessFlag	boolean	条件可选。当“IsLast”域为 TRUE，而 OMC 由于内部原因，又无法将其管理的所有设备的当前告警信息收集完整时该域就应存在（如 OMC 于 NE/NR 之间的连接故障）。当该域存在时其值总是 TRUE。当 NMC 收到这样的通知后，应该在适当的时候再调用一次“synchronizeAlarm”操作来进行当前活路告警信息的同步

B.4.8 心跳通知格式

心跳通知的固定事件头见表 B.16。

表 B.16 心跳通知的固定事件头

名 称	取 值
domain_name	GSM_COMMON_NETWORK_MGMT
type_name	NOTIFY_HEARTBEAT
event_name	<Notification Id>

心跳通知的可过滤体域见表 B.17。

表 B.17 心跳通知的可过滤体域

名 称	类 型	注 释
SystemName	string	必选。该域应为 cTelHeartbeat 对象的属性 systemLabel 的当前值，指明了发送该心跳通知的是哪个 OMC
ChannelId	string	必选。指明了该通知所经过的通知通道的标识符。其类型为字符串，惟一标识一个 OM 从中的通知通道
Period	unsigned short	必选。指心跳服务的心跳周期。其值应为 cTelHeartbeat 中的属性 period 的当前值
TimeStamp	UtcT	必选。时间戳，指发送该通知的时间

B.4.9 请求配置同步通知格式

请求配置同步通知的固定事件头见表 B.18。

表 B.18 请求配置同步通知的固定事件头

名称	取值
domain_name	GSM_CM_NETWORK_MGMT
type_name	NOTIFY_REQUEST_CM_SYNCHRONIZATION
event_name	<Notification Id>

请求配置同步通知的可过滤体域见表 B.19。

表 B.19 请求配置同步通知的可过滤体域

名称	类型	注释
SystemName	string	必选。标识了产生该通知的 OMC
BaseMOInstance	ManagedObjectType	必选。表示需要 NMC 进行同步的网络资源所对应的管理对象实例的根对象
EventTime	GeneralizedTimeType	必选。表示给通知发生的时间

B.5 通知格式的 IDL 定义

本节采用 IDL 定义了接口中所支持的通知的格式。基本的通知格式定义是基于 CORBA 通知服务中的 CosNotification 模块中的相关定义。

```
// File: NmclIfNotifFormat.idl
```

```
// This file specifies the format of notifications involved in this interface.
```

```
#ifndef _NMC_IF_NOTIF_FORMAT_IDL_
```

```
#define _NMC_IF_NOTIF_FORMAT_IDL_
```

```
#include <NmcIfComm.idl>
```

```
#include <NmcIfIMCommType.idl>
```

```
#include <CosNotification.idl>
```

```
module NmclIfNotifFormat
```

```
{
```

```
/**
*****
```

```
/** * the following IDL definition is contained in CosNotification module in
```

```
/** * CORBA Notification service.
```

```
*****
```

```
/** * typedef string Istring;
```

```
/** * typedef Istring PropertyName;
```

```
/** * typedef any PropertyValue;
```

```
/** * struct Property {
```

```
/** *   PropertyName name;
```

```
/** *   PropertyValue value;
```

```

/* * };
/* * typedef sequence<Property> PropertySeq;
/* * // The following are the same, but serve different purposes.
/* * typedef PropertySeq OptionalHeaderFields;
/* * typedef PropertySeq FilterableEventBody;
/* * struct EventType {
/* *     string domain_name;
/* *     string type_name;
/* * };
/* * typedef sequence<EventType> EventTypeSeq;
/* * struct FixedEventHeader {
/* *     EventType event_type;
/* *     string event_name;
/* * };
/* * struct EventHeader {
/* *     FixedEventHeader fixed_header;
/* *     OptionalHeaderFields variable_header;
/* * };
/* * struct StructuredEvent {
/* *     EventHeader header;
/* *     FilterableEventBody filterable_data;
/* *     any remainder_of_body;
/* * }; // StructuredEvent
/* * typedef sequence<StructuredEvent> EventBatch;
/*****

// The following constant declarations define the standard domain name in this
// interface.
typedef string DomainNameType;
const DomainNameType GSM_CM_NETWORK_MGMT = "GSM_CM_NETWORK_MGMT";
const DomainNameType GSM_FM_NETWORK_MGMT = "GSM_FM_NETWORK_MGMT";
const DomainNameType GSM_PM_NETWORK_MGMT = "GSM_PM_NETWORK_MGMT";
const DomainNameType GSM_COMMON_NETWORK_MGMT =
"GSM_COMMON_NETWORK_MGMT";

// The following constant declarations define the standard event type name.
typedef string TypeNameType;
const TypeNameType NOTIFY_ATTRIBUTE_VALUE_CHANGE =
"NOTIFY_ATTRIBUTE_VALUE_CHANGE";
const TypeNameType NOTIFY_STATE_CHANGE = "NOTIFY_STATE_CHANGE";
const TypeNameType NOTIFY_PM_TRANSFER_UP_READY =
"NOTIFY_PM_TRANSFER_UP_READY";
const TypeNameType NOTIFY_FM_SYNC_TRANSFER_UP_READY =
"NOTIFY_FM_SYNC_TRANSFER_UP_READY";

```

```

const      TypeNameType      NOTIFY_CM_SYNC_TRANSFER_UP_READY =
"NOTIFY_CM_SYNC_TRANSFER_UP_READY";
const      TypeNameType      NOTIFY_TOP_SYNC_TRANSFER_UP_READY =
"NOTIFY_TOP_SYNC_TRANSFER_UP_READY";
const      TypeNameType      NOTIFY_CM_LOG_RECORD_TRANSFER_UP_READY =
"NOTIFY_CM_LOG_RECORD_TRANSFER_UP_READY";
const      TypeNameType      NOTIFY_FM_LOG_RECORD_TRANSFER_UP_READY =
"NOTIFY_FM_LOG_RECORD_TRANSFER_UP_READY";
const      TypeNameType      NOTIFY_TOP_SYNC_TRANSFER_PREPARATION_ERROR =
"NOTIFY_TOP_SYNC_TRANSFER_PREPARATION_ERROR";
const      TypeNameType      NOTIFY_FM_SYNC_TRANSFER_PREPARATION_ERROR =
"NOTIFY_FM_SYNC_TRANSFER_PREPARATION_ERROR";
const      TypeNameType      NOTIFY_CM_SYNC_TRANSFER_PREPARATION_ERROR =
"NOTIFY_CM_SYNC_TRANSFER_PREPARATION_ERROR";
const      TypeNameType      NOTIFY_PM_TRANSFER_PREPARATION_ERROR =
"NOTIFY_PM_TRANSFER_PREPARATION_ERROR";
const      TypeNameType      NOTIFY_CM_LOG_RECORD_TRANSFER_PREPARATION_ERROR =
"NOTIFY_CM_LOG_RECORD_TRANSFER_PREPARATION_ERROR";
const      TypeNameType      NOTIFY_FM_LOG_RECORD_TRANSFER_PREPARATION_ERROR =
"NOTIFY_FM_LOG_RECORD_TRANSFER_PREPARATION_ERROR";
const      TypeNameType      NOTIFY_NEW_ALARM = "NOTIFY_NEW_ALARM";
const      TypeNameType      NOTIFY_ALARM_CHANGED = "NOTIFY_ALARM_CHANGED";
const      TypeNameType      NOTIFY_ALARM_CLEARED = "NOTIFY_ALARM_CLEARED";
const      TypeNameType      NOTIFY_OBJECT_CREATION = "NOTIFY_OBJECT_CREATION";
const      TypeNameType      NOTIFY_OBJECT_DELETION = "NOTIFY_OBJECT_DELETION";
const      TypeNameType      NOTIFY_CURRENT_ALARM_SUMMARY_REPORT =
"NOTIFY_CURRENT_ALARM_SUMMARY_REPORT";
const      TypeNameType      NOTIFY_HEARTBEAT = "NOTIFY_HEARTBEAT";
const      TypeNameType      NOTIFY_REQUEST_CM_SYNCHRONIZATION =
"NOTIFY_REQUEST_CM_SYNCHRONIZATION";

```

// The following constant declarations define the standard property name contained in
//filterable body. Including common property for each notification and special property
// for special notification.

//for common property

```

const string SystemName = "SystemName";
//SystemName takes a value of type string

```

// special property for alarm notification

```

const string AlarmId = "AlarmId";
//alarmId takes a value of type string.
const string MOInstance = "MOInstance";
//moInstance take a value of NmclIfComm.: ManagedObjectType

```

```

const string EventTime = "EventTime";
//eventTime takes a value of type string, and it should conform to what defined as
//GeneralizedTime in ASN.1
const string AlarmType = "AlarmType";
//AlarmType takes a value of type eAlarmType, defined as following.
enum eAlarmType
{
    CommunicationAlarm,
    QualityOfServiceAlarm,
    ProcessingErrorAlarm,
    EquipmentAlarm,
    EnvironmentAlarm,
    IntegrityViolence,
    OperationViolence,
    PhysicalViolence,
    SecurityServiceViolence,
    MechanismViolence,
    TimeDomainViolence
};

const string ProbableCause = "ProbableCause";
//ProbableCause takes a value of type string, and the value it can take should be contained in
//the list of value field in the tables of Appendix E.
const string PerceivedSeverity = "PerceivedServerty";
//PerceivedServerty takes a value of eServity
enum eSeverityType
{
    INDERTERMINATE,
    CRITICAL,
    MAJOR,
    MINOR,
    WARNING,
    CLEARED
};

const string SpecificProblem = "SpecificProblem";
//SpecificProblem takes a value of string, the format should be "domain, domain_specific_probable_cause".
const string CorrelatedNotifications = "CorrelatedNotifications";
//CorrelatedNotifications takes a value of AlarmIdentifierListTypeList, as is defined below
struct AlarmIdentifierType
{
    string systemName;
    string alarmId;
};

```

```

typedef sequence<AlarmIdentifierType> AlarmIdentifierListType;

const string AdditionalInfo = "AdditionalInfo";
// AdditionalInfo takes a value of AddiPropertyListType,
// which is a sequence of properties.
typedef CosNotification; : PropertySeq AddiPropertyListType;
//speical properties for transfer ready notification
const string TransferId = "TransferId";
//TransferId takes a value of string.
const string IPAddr = "IPAddr";
//IPAddr takes a value of string. It should conform to the format defined in TCP/IP
const string UserName = "UserName";
//UserName takes a value of string.
const string Password = "Password";
//Password takes a value of string.
const string JobFileInfoList = "JobFileInfoList";
//JobFileInfoList takes a value of NmcIfPMHandler;: FileInfoListType,
//which is defined in NmcIfPMHandler module

//special properties for transfer preparation error notification
//here also contains the "TransferId" property.
const string Reason = "Reason";
//Reason takes a value of type string.
//in this notification PerceivedServeity is also contained.
//SpecificProblem is also contained
const string AdditionalText = "AdditionalText";
//AdditioanalText takes a value of type string

//special properties for object deletion notification
//MOInstance is contained in this notification

//special properties for object creation notification
//moInstance is contained in this notification
const string AttributeList = "AttributeList";
//AttributeList takes a value of AttrInfoListType, as defined in NmcIfComm module, which
//is a list of sequence of attrName and attrValue.

//special properties for attribute change and state change notification
//MOInstance is contained in this notification
const string AttrChangeDef = "AttrChangeDef";
//AttrChangeDef takes a value of AttrChangeInfoListType which is a list of sequence of struct of
//attrName, newAttrValue and oldAttrValue, as defined below.
union OptionalAttrValueType switch (boolean)
{

```

```

    case TRUE : NmclComm:: AttrValueType value;
    // when the implementor doesn't support the old value,
    // set the discriminator value to FALSE.
};
struct AttrChangedInfoType
{
    NmclComm:: AttrNameType attrName;
    NmclComm:: AttrValueType newAttrValue;
    OptionalAttrValueType oldAttrValue;
};
typedef sequence<AttrChangedInfoType> AttrChangeInfoListType;
//special properties for current alarm summary report notification
const string AlarmSummary = "AlarmSummary";
//AlarmSummary takes a value of type AlarmSummaryDataType, which is defined below
typedef NmclMCommType:: eAlarmStatusType eAlarmStatusType;
typedef string ProbableCauseType;
// ProbableCauseType takes a value of type string, and the value it can take should
// be contained in the list of value field in the tables of Appendix E.

struct AlarmInfoType
{
    AlarmIdentifierType alarmIdentifier;
    string eventTime;
    eAlarmType eventType;
    eSeverityType alarmSeverity;
    ProbableCauseType probableCause;
    string specificProblem;
    AlarmIdentifierListType correlatedAlarms;
    AddiPropertyListType additionalInfo;
};

typedef sequence<AlarmInfoType> AlarmInfoListType;
struct ObjectAlarmInfoType
{
    NmclComm:: ManagedObjectType moInstance;
    eAlarmStatusType alarmStatus;
    AlarmInfoListType objectAlarms;
};
typedef sequence<ObjectAlarmInfoType> AlarmSummaryDataType;
// The AlarmSummaryDataType could be sent to NMC by a sequence of
// notificaitons, each contains part of the data, so as to avoid
// a very long notificaiton. The max length of alarms for each
// notificaiton is not defined here, it is decided by the implementor.
const string SubscriptionId = "SubscriptionId";

```



```
// SubscriptionId takes a value of SubscriptionIdType.
const string IsLast = "IsLast";
// IsLast takes a value of boolean.
const string FormerNotificationId = "FormerNotificationId";
// FormerNotificationId takes a value of string.
const string UnsuccessFlag = "UnsuccessFlag";
// UnsuccessFlag always takes the value TRUE of type boolean, when present.

// constant definitions for Heartbeat Notificaiton
const string ChannelId = "ChannelId";
// ChannelId takes type of string
const string Period = "Period";
// Period takes type of unsigned short
const string TimeStamp = "TimeStamp";
// Period takes type of TimeBase:: UtcT

// constant definitions for Request CM Synchronization Notifications.
const string BaseMOInstance = "BaseMOInstance";

// constant definitions for the "AdditionalInfo" field of Alarm Notifications
const string locatingInfo = "locatingInfo";
const string originalType = "originalType";
const string originalSeverity = "originalSeverity";
const string detailedContext = "detailedContext";
const string repairProposal = "repairProposal";
};

#endif
```

附录 C
(规范性附录)
文件格式定义

C.1 简介

本附录规定数据文件的格式以及 ASN.1 定义。

C.2 文件内容

C.2.1 性能数据测量文件

性能数据测量文件包括以下内容。

- a) **jobId** (必选): 测量任务的标识符, NMC 使用 **jobId** 将结果文件与相应的测量任务关联起来。
- b) **startingTime** (必选): 指测量任务的开始时间。
- c) **aggregationPeriod** (必选): 指测量任务的数据采集周期。
- d) **reportingPeriod** (必选): 指测量数据结果的上报周期。
- e) **measurementInfoList** (必选): 测量信息列表, 包括以下内容。
 - **moInstance** (必选): 标识一个管理对象实例。
 - **moDN** (必选): 对象的区分名, 其格式遵照 6.2.1.2 节的定义。
 - **moClass** (必选): 指该对象的管理对象类名。
 - **measurementAttrInfoList** (必选): 测量属性信息列表。
 - **measurementAttrName** (必选): 测量属性名。
 - **attrValue** (必选): 测量属性值。

f) **IncompleteScan** (条件可选): 如果数据采集没有完成则该域存在, 表示未完成的原因。其类型为整型, 值取 0 时表示“超过最大容量”, 取 1 时表示“超过了扫描时间”, 取 2 时表示“其他原因”。

C.2.2 配置同步文件

配置同步文件的格式如下。

- a) **transactionId** (必选): NMC 使用 **transactionId** 把将结果文件与同步配置信息的操作相关联。
- b) **moInfoList** (必选): 管理对象信息列表, 包括以下内容。
 - **moInstance** (必选): 标识一个管理对象实例。
 - **moDN** (必选): 对象的区分名, 其格式遵照 6.2.1.2 节的定义。
 - **moClass** (必选): 指该对象的管理对象类名。
 - **attrInfoList** (必选): 属性信息列表。
 - **attrName** (必选): 属性名。
 - **attrValue** (必选): 属性值。

C.2.3 拓扑文件

拓扑文件的内容如下。

- a) **transactionId** (必选): NMC 使用 **transactionId** 把将结果文件与获取拓扑的操作相关联。
- b) **moInstanceList** (必选): 管理对象列表。
 - **moDN** (必选): 对象的区分名, 其格式遵照 6.2.1.2 节的定义。
 - **moClass** (必选): 指该对象的管理对象类名。

C.2.4 日志记录文件

日志记录文件的内容如下。

- a) subscriptionId (必选): NMC 使用 subscriptionId 将结果文件与相应的获取日志记录操作相关联。
- b) logRecordList (必选): 告警记录信息列表。
- c) logRecordId (必选): 日志记录的标识符。
- d) loggingTime (必选): 日志记录生成的时间。
- e) recordInfo (必选): 所记录的事件信息。

C.2.5 告警同步文件

告警同步文件的内容如下。

- a) transactionId (必选): NMC 使用 transactionId 将结果文件与相应的告警同步操作相关联。
- b) alarmSummaryData (必选): 管理对象及其当前活跃告警信息列表。
 - moInstance (必选): 标识一个管理对象实例。
 - moDN (必选): 对象的区分名, 其格式遵照 6.2.1.2 节的定义。
 - moClass (必选): 指该对象的管理对象类名。
 - alarmStatus (必选): 该管理对象的告警状态。
 - objectAlarms (必选): 告警信息列表。
 - alarmIdentifier (必选): 告警标识符。
 - eventTime (必选): 告警产生时间。
 - eventType (必选): 告警类型。
 - alarmSeverity (必选): 告警级别。
 - probableCause (必选): 告警的可能原因。
 - specificProblem (条件可选): 当告警原因为厂商自定义时, 用该域来表示。
 - correlatedAlarms (条件可选): 如果该告警有相关联的告警时使用。
 - additionalInfo (可选): 告警的附件信息。

C.3 文件格式的 ASN.1 定义

-- DOCUMENT FILEFORMAT-NMCIF-TMN-ASN1Module

FILEFORMAT-NMCIF-TMN-ASN1Module

DEFINITIONS IMPLICIT TAGS:: =

BEGIN

MOClassNameType:: =GraphicString

-- The values of MOClassNameType should conform to the MOC names

-- defined in module NmcHfIMBaseType:: eClassIdType

DistinguishedNameType ::= GraphicString

DNListType ::= SET OF DistinguishedNameType

```
ManagedObjectType:: =SEQUENCE {
    className    MOClassNameType,
    moDN        DistinguishedNameType
}
```

ManagedObjectListType ::= SET OF ManagedObjectType

AttrNameType:: =GraphicString

- The values of AttrNameType should conform to the attribute names
- defined in module NmclfIMBaseType:: eAttrIdType

```
AttrInfoType:: =SEQUENCE {
    attrName      AttrNameType,
    attrValue ANY DEFINED BY attrName
}
```

JobIdType:: =INTEGER

AggregationPeriodType:: =INTEGER

ReportingPeriodType:: =INTEGER

MeasurementAttrNameType:: =GraphicString

- The names of MeasurementAttrName should confirm to the MOC names
- defined NmclfIMMeasurement:: mAttrIdType

```
MeasurementAttrInfoType:: =SEQUENCE {
    measurementAttrName MeasurementAttrNameType,
    measurementAttrValue ANY DEFINED BY measurementAttrName
}
```

MeasurementCategoryType :: = ENUMERATED

```
{
    circuitEndpointSubGroup (0),
    observedDestination (1),
    trafficFlow (2),
    bscHDOSTatistics (3),
    bscTrafficStatistics (4),
    bscProcessorLoad (5),
    bscGRPSSubLayerTraffic (6),
    btsTrafficData (7),
    powerLevelOfBS (8),
    btsHDOSTatistics (9),
    btsLocationUpdate (10),
    btsGPRSRadioResource (11),
    btsGPRSPacketService (12),
    eirPerformanceStatistics (13),
    eirProcessorLoad (14),
    handoverObservation (15),
    hlrPerformanceStatistics (16),
    hlrProcessorLoad (17),
    mscPerformance (18),
    mscQoS (19),
    mscProcessorLoad (20),
    ssfTraffic (21),
```

smsPerformanceStatistics (22),
 smsProcessorLoad (23),
 vlrPerformanceStatistics (24),
 vlrProcessorLoad (25),
 signallingLinkSetTPTraffic (26)

}

MeasurementInfoType ::= SEQUENCE {
 moInstance ManagedObjectType,
 measurementCategory MeasurementCategoryType,
 measurementAttrInfoList SET OF MeasurementAttrInfoType,
 additionalInfo PropertyListType OPTIONAL

}

IncompleteScanType ::= INTEGER {sizeExceeded (0), timeExceeded (1), other (2)}

PmMeasurementFileType ::= SEQUENCE

{

 jobId JobIdType,
 startingTime GeneralizedTime,
 aggregationPeriod AggregationPeriodType,
 reportingPeriod ReportingPeriodType,
 measurementInfoList SET OF MeasurementInfoType,
 incompleteScan IncompleteScanType OPTIONAL

}

TransactionIdType ::= INTEGER

AdditionalInfoType ::= GraphicString

TopologyFileType ::= SEQUENCE {

 tranId TransactionIdType,
 moList SET OF ManagedObjectType

}

MOInfoListType ::= SEQUENCE {

 moInstance ManagedObjectType,
 attrList SET OF AttrInfoType

}

CMFile ::= SEQUENCE {

 tranId TransactionIdType,
 moInfoList SET OF MOInfoListType

}

SubscriptionIdType ::= INTEGER

LogRecordIdType ::= INTEGER

LoggingTimeType ::= =GeneralizedTime

```
LogRecordType ::= =SEQUENCE {
    logRecordId  LogRecordIdType,
    loggingTime  LoggingTimeType,
    recordInfo   ANY
}
```

--the recordInfo element should be the event information logged.

```
LogFileType ::= =SEQUENCE {
    subId        SubscriptionIdType,
    logRecordList SET OF LogRecordType
}
```

--The following type definitions are for Performance Measurement Jobs

CountType ::= = INTEGER

GSMMeasurementType1 ::= = INTEGER

GSMMeasurementType2 ::= = REAL

```
GSMMeasurementType3 ::= = SET OF SEQUENCE {
    cause CauseType,
    value INTEGER
}
```

CauseType ::= = ENUMERATED {

--Internal and External Handover causes (GSM 08.08)

```
    uplinkQuality (0),
    uplinkStrength (1),
    downlinkQuality (2),
    downlinkStrength (3),
    distance (4),
    betterCell (5),
    operationAndMaintenanceIntervention (6),
    directedRetry (7),
```

-- valid for external handovers only:

```
    responseToMscInvocation (8),
```

-- Immediate assignment procedure causes (GSM 04.08 [2]):

```
    emergencyCall (9),
    callReEstablishment (10),
    answerToPaging (11),
    originatingCall (12),
    locationUpdating (13),
    otherProcedures (14),
    reservedEstablishmentCause (15)
```

}

```

GSMMeasurementType4 ::= SET OF SEQUENCE {
    ssOperation SSOperationType,
    value INTEGER
}
-- SS operation definitions (GSM 09.02):
SSOperationType ::= INTEGER {
    register (0),
    erase (1),
    activate (2),
    deactivate (3),
    registerPassword (4),
    interrogateSSOperation (5),
    processRequest (6)
}
GSMMeasurementType5 ::= SEQUENCE {
    nbrOfCallLostBecauseOfTrunkUnavailable INTEGER,
    nbrOfCallLostBecauseOfExchangeCongested INTEGER,
    nbrOfAbnormalCalls INTEGER
}
GSMMeasurementType6 ::= SET OF SEQUENCE {
    locationAreaCode LocationAreaCodeType,
    value INTEGER
}
PeakTimeType ::= GeneralizedTime

EndCodeType ::= ENUMERATED
{
    longCallNoAnswer (0),
    calledPartyBusy (1),
    incompleteAddress (2),
    emptyNumber (3),
    timeoutRelease (4),
    equipmentCongestion (5),
    circuitCongestion (6),
    pagingNoResponse (7)
}

GSMMeasurementType7 ::= SET OF SEQUENCE
{
    endCode EndCodeType,
    value INTEGER
}

```

GSMMeasurementType8 ::= SEQUENCE

```
{
    channelsPerInterferenceBand1 REAL,
    channelsPerInterferenceBand2 REAL,
    channelsPerInterferenceBand3 REAL,
    channelsPerInterferenceBand4 REAL,
    channelsPerInterferenceBand5 REAL
}
```

GSMMeasurementType9 ::= SEQUENCE

```
{
    numOfUpgrades INTEGER,
    numOfDowngrades INTEGER
}
```

GPRSMeasurementType3 ::= SET OF SEQUENCE

```
{
    nsvei INTEGER,
    value INTEGER
}
```

-- The following type definitions for CM part,
 -- which will be used to fill the ANY

AlarmStatusType ::= ENUMERATED

```
{
    cleared (0),
    activeReportable-Indeterminate (1),
    activeReportable-Warning (2),
    activeReportable-Minor (3),
    activeReportable-Major (4),
    activeReportable-Critical (5),
    activePending (6)
}
```

ObjectIdType ::= INTEGER

ObjectIdListType ::= SET OF ObjectIdType

NetworkDestCodeType ::= OCTET STRING (SIZE (5))

CountryCodeType ::= OCTET STRING (SIZE (4))

MobileCountryCodeType ::= OCTET STRING (SIZE (3))

MobileNetworkCodeType ::= OCTET STRING (SIZE (2))

ISDNAddrStringType ::= OCTET STRING (SIZE (1..20))

NetworkDestCodeSetType ::= SET OF NetworkDestCodeType

CountryCodeSetType ::= SET OF CountryCodeType

NetworkColourCodeType ::= INTEGER (0..7)

BTSColourCodeType ::= INTEGER (0..7)

BServiceIdType ::= ENUMERATED

```
{
    ts11 (0),
    ts21 (1),
    ts22 (2),
    ts61 (3),
    ts62 (4),
    bs21 (5),
    bs22 (6),
    bs23 (7),
    bs24 (8),
    bs25 (9),
    bs26 (10),
    bs31 (11),
    bs32 (12),
    bs33 (13),
    bs34 (14),
    bs41 (15),
    bs42 (16),
    bs43 (17),
    bs44 (18),
    bs45 (19),
    bs46 (20),
    bs51 (21),
    bs52 (22),
    bs53 (23),
    bs61A (24),
    bs61S (25),
    bs71 (26),
    bs81A (27),
    bs81S (28)
}
```

BServiceIdListType ::= SET OF BServiceIdType

SServiceIdType ::= ENUMERATED

```
{
    clip (0),
```

YD/T 1279-2003

clir (1),
colr (2),
colp (3),
cfu (4),
cfb (5),
cfny (6),
cfnc (7),
cw (8),
hold (9),
mpty (10),
cug (11),
aoci (12),
aocc (13),
baoc (14),
boic (15),
boicexhc (16),
baic (17),
bicroam (18)

}

SServiceIdListType ::= SET OF SServiceIdType

ListOfValidCUGInterlockCodesType ::= SET OF INTEGER

-- the following types are got from NmcIfMCM module

NetworkIndicatorType ::= ENUMERATED

{

international (0),
spare (1),
national (2),
reservedNationalUse (3)

}

SignallingPointLengthType ::= ENUMERATED

{

bits-24 (0),
bits-14 (1)

}

SignallingInfoType ::= SEQUENCE

{

signallingPointLength SignallingPointLengthType,
signallingPointCode INTEGER,
networkIndicator NetworkIndicatorType

}

SignallingInfoListType ::= SET SIZE (1..4) OF SignallingInfoType

SignallingPointType ::= INTEGER {
 noSTP (0),
 sTP (1)
 }

BSIdCodeType ::= SEQUENCE
 {
 ncc NetworkColourCodeType,
 bcc BTSColourCodeType
 }

AbsoluteRFChannelNoType ::= INTEGER (0..1023)

CellAllocationType ::= SEQUENCE SIZE (1..64) OF AbsoluteRFChannelNoType

GsmcdsIndicatorType ::= ENUMERATED
 {
 gsm (0),
 extendedGsm (1),
 dcs (2),
 multiband (3) -- which can support gsm and dcs1800
 }

CellIdentityType ::= INTEGER (0..65535)

LocationAreaCodeType ::= INTEGER (0..65535)

LocationAreaIdentityType ::= SEQUENCE
 {
 mcc MobileCountryCodeType,
 mnc MobileNetworkCodeType,
 lac LocationAreaCodeType
 }

CellGlobalIdType ::= SEQUENCE
 {
 lai LocationAreaIdentityType,
 ci CellIdentityType
 }

TimeUnitType ::= ENUMERATED
 {
 mSec (0),
 sec (1),
 }

```
    min (2),  
    noOfTDMAFrames (3),  
    noOfSlots (4),  
    factor (5)  
}
```

```
TimerDataType ::= SEQUENCE  
{  
    timeUnit TimeUnitType,  
    timeValue INTEGER  
}
```

```
NetworkColorCodeType ::= INTEGER (0..7)  
PlmnPermittedType ::= SET SIZE (0..8) OF NetworkColorCodeType
```

```
MaxRetranType ::= ENUMERATED  
{  
    one (0),  
    two (1),  
    four (2),  
    seven (3)  
}
```

```
DtxUplinkType ::= INTEGER {  
    msMayUseDtx (0),  
    msShallUseDtx (1),  
    msShallNotUseDtx (2)  
}
```

```
ClassNumberType ::= INTEGER (0..15)  
ClassNumberListType ::= SET SIZE (0..15) OF ClassNumberType
```

```
MsmProcessingModeType ::= ENUMERATED  
{  
    basicMeasurementReporting (0),  
    btsProcessingMeasurementReporting (1)  
}
```

```
PCMTTimeSlotType ::= SEQUENCE  
{  
    pcmCircuitId INTEGER (0..2047),  
    timeSlotNum INTEGER  
}
```

```
MultiTimeSlotType ::= SEQUENCE
```

```

{
    timeSlot PCMTTimeSlotType,
    subSlot INTEGER
}

AbisChannelType ::= SEQUENCE
{
    bscTimeSlot MultiTimeSlotType,
    btsTimeSlot MultiTimeSlotType
}

RelatedTrunkGroupTPType ::= SEQUENCE {
    priority INTEGER,
    trunkGroupTP DistinguishedNameType
}

RelatedTrunkGroupTPListType ::= SEQUENCE OF RelatedTrunkGroupTPType
RouteType ::= SEQUENCE
{
    nameOfRoute GraphicString,
    trunkGroupTPList RelatedTrunkGroupTPListType
}

SourceNumberType ::= CHOICE
{
    notUsed NULL,
    number DigitListType
}

RelatedRouteInfoType ::= SET OF SEQUENCE
{
    sourceNumber SourceNumberType,
    routeInfo DistinguishedNameType
}

DigitParticipationIndicatorType ::= INTEGER {
    seeNextCode (0),
    internationalCode (1),
    nationalPSTNCode (2),
    nationalPLMNCode (3),
    toLocalMSC (4),
    toOwnMSC (5),
    localPSTNCode (6),
    localSPXCode (7),
    analogPLMNCode (9),
}

```

others (10)
}

DigitElementType ::= GraphicString
(FROM ("1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" | "0" | "A" | "B" | "C" | "D" | "E" | "F" | "*" | "#") |
SIZE (1))

DigitListType ::= SEQUENCE OF DigitElementType

DestinationCodeType ::= DigitListType

TrunkGroupType ::= ENUMERATED

{
toInternationalToll (0),
toNationalToll (1),
toLocalTandem (2),
toLocalEnd (3),
toAnalogMobileTransit (4),
toGSMFirstTransit (6),
toGSMSecondTransit (7),
toGSM MSC (8),
toBSC (9),
toVoiceMailbox (10),
toOthers (11)
}

CircuitType ::= ENUMERATED

{
lowCallLoss (0),
highEfficiencyDirect (1)
}

CircuitDirectionalityType ::= ENUMERATED

{
onewayOut (0),
onewayIn (1),
twoway (2)
}

TransmissionCharacteristicsType ::= INTEGER {

opticalFiberCable (1),
coaxialCable (2),
analogMicrowave (3),
digitMicrowave (4),
satellite (5),

```

    mixedGroup (6),
    transOthers (7)
}
TransmissionCharacteristicsSetType ::= SET SIZE (1..7) OF TransmissionCharacteristicsType

DestType ::= ENUMERATED
{
    international-dest (0),
    national-dest (1),
    local-dest (2),
    other-dest (3)
}

NatureOfAddressType ::= BIT STRING (SIZE (7))
-- only 0 and 1 can occur in the string, it is of BIT STRING type.

DestinationTypeChoiceType ::= ENUMERATED
{
    natureOfAddressChoice (0),
    destTypeChoice (1)
}

DestinationType ::= CHOICE {
    natureOfAddress [0] NatureOfAddressType,
    destType [1] DestType
}

-- type definitions for alarmRecord
DomainNameType ::= GraphicString
TypeNameType ::= GraphicString
EventNameType ::= GraphicString

AlarmIdType ::= GraphicString
EventTimeType ::= GeneralizedTime
AlarmType ::= ENUMERATED
{
    communicationAlarm (0),
    qualityOfServiceAlarm (1),
    processingErrorAlarm (2),
    equipmentAlarm (3),
    environmentAlarm (4),
    intergrityViolence (5),
    operationViolence (6),
    physicalViolence (7),

```

```
securityServiceViolence (8),  
mechanismViolence (9),  
timeDomainViolence (10)
```

```
}
```

SeverityType ::= ENUMERATED

```
{  
    indeterminate (0),  
    critical (1),  
    major (2),  
    minor (3),  
    warning (4),  
    cleared (5)
```

```
}
```

SpecificProblem ::= GraphicString

AlarmIdentifierType ::= SEQUENCE

```
{  
    alarmId    GraphicString,  
    systemName GraphicString
```

```
}
```

AlarmIdentifierListType ::= SET OF AlarmIdentifierType

PropertyType ::= SEQUENCE

```
{  
    name GraphicString,  
    value ANY
```

```
}
```

PropertyListType ::= SET OF PropertyType

AttrChangedInfoType ::= SEQUENCE

```
{  
    attrName GraphicString,  
    newAttrValue AttrInfoType,  
    oldAttrValue AttrInfoType OPTIONAL
```

```
}
```

ProbableCauseType ::= GraphicString

AlarmInfoType ::= SEQUENCE

```
{  
    alrmIdentifier AlarmIdentifierType,  
    eventTime    GeneralizedTime,  
    eventType    AlarmType,
```



```

alarmSeverity SeverityType,
probableCause ProbableCauseType,
specificProblem GraphicString,
correlatedAlarms AlarmIdentifierListType,
additionalInfo PropertyListType
}

```

AlarmInfoListType ::= SET OF AlarmInfoType

```

ObjectAlarmInfoType ::= SEQUENCE
{
    moInstance    ManagedObjectType,
    alarmStatus   AlarmStatusType,
    objectAlarms  AlarmInfoListType
}

```

AlarmSummaryDataType ::= SET OF ObjectAlarmInfoType

```

AlarmSummaryFileType ::= SEQUENCE
{
    transactionId TransactionIdType,
    alarmSummaryData AlarmSummaryDataType
}

```

```

AdminStateType ::= ENUMERATED
{
    locked (0),
    unlocked (1),
    shuttingDown (2)
}

```

```

OpStateType ::= ENUMERATED
{
    disabled (0),
    enabled (1)
}

```

```

UsageStateType ::= ENUMERATED
{
    idle (0),
    active (1),
    busy (2)
}

```

UnknownStatusType ::= BOOLEAN

ProbableCauseType ::= GraphicString -- contains the value of integer,

YD/T 1279-2003

CurrentProblemType ::= SEQUENCE

```
{  
    problem ProbableCauseType,  
    alarmStatus AlarmStatusType  
}
```

CurrentProblemListType ::= SET OF CurrentProblemType

CurrentProblemListTypeOpt ::= CHOICE

```
{  
    notSupported [0] NULL,  
    problemList [1] CurrentProblemListType  
}
```

BssNsUserEntityMappingTableType ::= SET OF SEQUENCE

```
{  
    userEntity GraphicString,  
    bvci INTEGER,  
    nsei INTEGER  
}
```

BssNsEntityMappingTableType ::= SET OF SEQUENCE

```
{  
    nsei INTEGER,  
    bvci INTEGER,  
    nsvci INTEGER,  
    dlci INTEGER,  
    bearerChanne INTEGER,  
    loadShareMechanism REAL  
}
```

SLSType ::= INTEGER (0..15)

SLSCoListType ::= SET SIZE (0..16) OF SLSType

NECodeType ::= GraphicString

NECodeListType ::= SET OF NECodeType

OmcType ::= ENUMERATED

```
{  
    omc (0),  
    omc-s (1),  
    omc-r (2)  
}
```

SignallingLinkPriorityType ::= INTEGER

RoutingAreaIdType ::= INTEGER

ChannelEncodeModeType ::= ENUMERATED

```
{
  cs1 (0),
  cs2 (1),
  cs3 (2),
  cs4 (3)
}
```

ChannelEncodeModeListType ::= SET OF ChannelEncodeModeType

MediaAccessControlModeType ::= ENUMERATED

```
{
  dynamicAllocation (0),
  extendedDynamicAllocation (1),
  fixedAllocation (2)
}
```

CellReselectArithmType ::= ENUMERATED

```
{
  c1 (0),
  c2 (1),
  c31 (2),
  c32 (3)
}
```

CellReselectArithmListType ::= SET OF CellReselectArithmType

NEType ::= ENUMERATED

```
{
  neTypeMsc (0),
  neTypeVlr (1),
  neTypeHlr (2),
  neTypeAuc (3),
  neTypeEir (4),
  neTypeBss (5),
  neTypeOmc (6),
  neTypeSms (7)
}
```

NEListType ::= SEQUENCE OF NEType

TrxPowerType ::= REAL

MscType ::= ENUMERATED

```
{
```

YD/T 1279-2003

```
tmse1 (0),  
tmse2 (1),  
gmse (3),  
mse-vlr (4)  
}
```

MseUpgradeType ::= SEQUENCE

```
{  
    upgradeToSSP Boolean,  
    upgradeToPOI Boolean  
}
```

Integer ::= INTEGER

String ::= GraphicString

Boolean ::= BOOLEAN

END

注：所有类型定义为 IDL 中 “string” 的属性类型都映射为 ASN.1 格式中的 “GraphicString”。

附录 D
(规范性附录)
接口信息模型的 IDL 定义

D.1 简介

本附录给出基于 IDL 的 NMC-OMC 接口信息模型定义。本附录给出的信息模型 IDL 定义除类型定义外,其他不要求厂商在 NMC-OMC 接口中实现,其目的仅仅是供厂商在实现时予以参考。

D.2 配置管理信息定义

本节中给出管理对象类配置属性的详细信息,见表 D.1~表 D.25。

表 D.1 plmnSubnetwork 的信息

属 性	描 述
networkId	plmnSubnetwork 的标识符,是命名属性
userLabel	用户友好名,由 OMC 厂商自己指定
mcc	移动国家代码
mnc	移动网络代码
setOfCc	国家代码集合
setOfNdc	网络目的码集合
listOfSupportedBS	所支持的基本服务列表
listOfSupportedSS	所支持的附加服务列表

表 D.2 managedElement 的信息

属 性	描 述
managedElementId	managedElement 的标识符,命名属性
alarmStatus	告警状态

表 D.3 gsmEquipment 的信息

属 性	描 述
equipmentId	设备标识,是命名属性
userLabel	用户友好名,由 OMC 厂商自己指定,可以作为其内部的设备标识
vendorName	厂商名称
version	设备版本号
locationName	位置名称,应指明该设备所处的具体房间
relatedGSMFunction	关联属性,指该设备所实现的功能实体列表

表 D.3 (续)

属 性	描 述
adminState	管理状态
opState	运行状态
alarmStatus	告警状态
neCode	区分 NE 的代码

表 D.4 software 的信息

属 性	描 述
softwareId	软件标识符, 命名属性
adminState	管理状态
opState	运行状态
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识
vendorName	软件厂商名称
version	软件版本
currentProblemList	当前软件存在的问题列表

表 D.5 signallingLinkTP 的信息

属 性	描 述
signallingLinkTPId	signallingLinkTP 标识符, 是命名属性
slc	用于区分信令链路集中信令链路的信令链路代码
adminState	管理状态
opState	运行状态
usageState	使用状态
signallingLinkPriority	信令链路优先级
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识
slsCodeNormalList	信令链路选择码 (SLS), 该属性是因管理的需要而分配给 signallingLinkTP 的, 用于正常的操作
slsCodeCurrentList	该属性表示当前支配给 signalingLinkTP 的 SLS。必须保证所有的 SLS 都被覆盖, 并且在该属性中出现的 SLS 不能多于一次。该属性可能因系统内部原因而改变

表 D.6 signallingLinkSetTP 的信息

属 性	描 述
signallingLinkSetTPId	signallingLinkSetTP 标识符, 是命名属性
signallingLinkSetName	信令链路点下的信令链路集名称
numberOfSignallingLinks	所包含的信令链路数日

表 D.6 (续)

属 性	描 述
adjacentSignallingInfo	指与 MTP 信令集相关联的远端信令点编码和网络指示符
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识

表 D.7 signallingPoint 的信息

属 性	描 述
signallingPointId	signallingPoint 标识符, 是命名属性
signallingInfo	信令信息, 指与 MTP 相关的信令点编码和网络指示符
signallingPointType	信令点类型, 指明该信令点的模式是 STP, 或不在 MTP 中, 而是由网络指示符来识别。其值总是“noSTP”
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识

表 D.8 omcFunction 的信息

属 性 名	描 述
omcFunctionId	omcFunction 的标识符, 是命名属性
omcType	OMC 的类型, 可以是 omc, omc-r, 或者 omc-s
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识
relatedGSMEquipment	关联属性, 指实现该功能的 GSM 物理设备

表 D.9 aucFunction 的信息

属 性	描 述
aucFunctionId	aucFunction 标识
adminState	管理状态
opState	运行状态
maxNumberOfLogicalAuc	指该 AUC 对象中可以包含的最大逻辑 AUC 的个数
currentNumberOfLogicalAuc	指该 AUC 对象中当前包含的逻辑 AUC 的个数
maxNumberOfImsiInAuc	指该 AUC 中所能包含的 IMSI 的最大个数
currentNumberOfImsiInAuc	指该 AUC 中当前所包含的 IMSI 的个数
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识
relatedGSMEquipment	关联属性, 指实现该功能的 GSM 设备

表 D.10 bssFunction 的信息

属 性	描 述
bssFunctionId	bssFunction 标识符, 是命名属性
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识

表 D.11 eirFunction 的信息

属 性	描 述
eirFunctionId	cTelEirFunction 标识符, 是命名属性
adminState	管理状态
opState	运行状态
eirId	指由该 cTelEirFunction 表示的 EIR 的标识符
eirNumber	该 eirFunction 代表的 EIR 的 ISDN 号码, 用于通过信令来定位该 EIR
maxNumberOfWhiteListEntries	白名单的最大数目
maxNumberOfGreyListEntries	灰名单的最大数目
maxNumberOfBlackListEntries	黑名单的最大数目
currentNumberOfWhiteListEntries	白名单中的当前数目
currentNumberOfGreyListEntries	灰名单中的当前数目
currentNumberOfBlackListEntries	黑名单中的当前数目
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识
relatedGSMEquipment	关联属性, 指实现该功能的 GSM 设备

表 D.12 hlrFunction 的信息

属 性	描 述
hlrFunctionId	cTelHlrFunction 标识符, 是命名属性
adminState	管理状态
opState	运行状态
maxNumberOfLogicalHlr	指该 HLR 对象中可以包含的最大逻辑 HLR 的个数
currentNumberOfLogicalHlr	指该 HLR 对象中当前包含的逻辑 HLR 的个数
maxNumberOfimsiInHlr	指该 HLR 中所能包含的 IMSI 的最大个数
currentNumberOfimsiInHlr	指该 HLR 中当前所包含的 IMSI 的个数
maxNumberOfmsisdnInHlr	指该 HLR 中所能包含的 msisdnInHlr 对象的最大个数
currentNumberOfmsisdnInHlr	指该 HLR 中当前所包含的 msisdnInHlr 对象的个数
defaultPW	指要在 subscriberInHlr 对象中使用的缺省的口令
defaultAnnouncement	指要在 subscriberInHlr 对象中使用的缺省的声名
listOfValidCUGInterlockCodes	指所有有效的 CUG 互锁码列表
usageState	使用状态
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识
relatedGSMEquipment	关联属性, 指实现该功能的 GSM 设备
hlrNumber	该 hlrFunction 代表的 HLR 的 ISDN 号码, 用于通过信令来定位该 HLR
supportGPRS	该 HLR 是否支持 GPRS 功能

表 D.13 vlrFunction 的信息

属 性	描 述
vlrFunctionId	vlrFunction 标识符, 是命名属性
adminState	管理状态
opState	运行状态
vlrId	指由 vlrFunction 所表示的 VLR 的标识
vlrNumber	该 vlrFunction 代表的 VLR 的 ISDN 号码, 用于通过信令来定位该 VLR
maxNumberOfImsiInVlr	指该 VLR 可以存储的最大 IMSI 的最大数目
currentNumberOfImsiInVlr	指该 VLR 中当前存储的 IMSI 的数目
usageState	使用状态
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为其内部的设备标识
relatedGSMEquipment	关联属性, 指实现该功能的 GSM 物理设备

表 D.14 mscFunction 的信息

属 性	描 述
mscFunctionId	mscFunction 标识符
adminState	管理状态
opState	运行状态
mscId	MSC 的标识
mscNumber	该 mscFunction 代表的 MSC 的 ISDN 号码, 用于通过信令来定位该 MS
usageState	使用状态
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为其内部的设备标识
maxMscBHCA	MSC BHCA 的最大设计值
mscCapacity	MSC 容量
maxNumOfCPUConfigurable	MSC 可配置 CPU 的最大数目
mscType	MSC 类型, 可以是 TMSC1、TMSC2、GMSC 或 MSCVLR
mscUpgradeIndicator	MSC 升级标志, 表示 MSC 是否升级为 SSP 或 POI
numOf2MPort	MSC 的 2Mbit/s 端口数目
mscIWFDataBlockCapacity	MSC 互联功能数据块的容量 (单位时间可处理的用户数)
numOfIWFDataBlock	MSC 互联功能数据块的数目
relatedGSMEquipment	关联属性, 指实现该功能的 GSM 物理设备

表 D.15 circuitEndPointSubgroup 的信息

属 性	描 述
circuitEndPointSubgroupId	circuitEndPointSubgroup 标识符, 是命名属性
numberOfCircuits	电路群中的电路数目
circuitDirectionality	该电路群的方向
transmissionCharacteristics	该电路群所支持的传输特性, 如光纤、同轴电缆、模拟微波、数字微波、卫星和混合组, 由相应的比特位决定
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识
circuitType	指电路类型, 如: <i>lowCallLoss</i> , <i>highEfficiencyDirect</i>
trunkGroupType	中继群类型, 如 <i>toInternationalTol</i> 、 <i>toNationalToll</i> 、 <i>toLocalTandem</i> 、 <i>toLocalEnd</i> 、 <i>toAnalogMobileTransit</i> 、 <i>toGSMFirstTransit</i> 、 <i>toGSMSecondTransit</i> 、 <i>toGSMMSC</i> 、 <i>toBSC</i> 和 <i>toVoiceMailbox</i>
adminState	管理状态
opState	运行状态
alarmStatus	告警状态
maxNumOfCircuitsConfigurable	指电路群的最大可配置电路数目
signallingInfoOfFarEnd	指该中继群的远端的信令信息
signallingPointOfFarEnd	是指向该中继群的远端信令对象的指针

表 D.16 digitsSegment 的信息

属 性	描 述
digitsSegmentId	digitsSegment 标识
digitList	表示由 1~n 个数字组成的号码段, 用于路由选择
digitParticipationIndicator	表示该号码段的是否是下列元素或其一部分: <i>InternationalCode</i> 、 <i>NationalPSTNCode</i> 、 <i>NationalPLMNCode</i> 、 <i>toLocalMSC</i> 、 <i>toOwnMSC</i> 、 <i>LocalPSTNCode</i> 、 <i>LocalSPXCode</i> 、 <i>AnlogPLMNCode</i> 、 <i>others</i> 或 <i>seeNextCode</i>
labelOfFarEndExchange	远端交换机的用户友好名
relatedRouteInfo	关联属性, 表示该号码段所关联的路由信息 <i>routingInfo</i> 的对象 (有些相关路由与源号码相关, 由二者共同组成序偶, 可以构成相关路由的列表)

表 D.17 routingInfo 的信息

属 性	描 述
routingInfoId	routingInfo 标识符, 是命名属性
firstChoiceRoute	第一选择路由和相关中继群的信息
firstCircuitousRoute	第一迂回路由和相关中继群的信息
secondCircuitousRoute	第二迂回路由和相关中继群的信息
thirdCircuitousRoute	第三迂回路由和相关中继群的信息

表 D.18 pcu 的信息

属 性	描 述
pcuId	pcu 标识符, 是命名属性
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识
locationName	位置名称, 应指明该设备所处的具体房间
relatedGSMEquipment	关联属性, 指实现该功能的 GSM 设备
relatedSGSN	关联属性, 是指向相关链的 SGSN 的指针, 其取值为 SGSN 的网元编码
numOfPDCHSupported	PCU 所支持的 PDCH 的数目
adminState	管理状态
alarmStatus	告警状态
opState	运行状态
usageState	使用状态
unknownStatus	未知状态

表 D.19 bsc 的信息

属 性	描 述
bscId	bsc 标识符, 是命名属性
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识
locationName	位置名称, 应指明该设备所处的具体房间
handoverReqParam	切换请求参数, 定义了用于产生向 MSC 发送切换请求是所用的参数 'n', 该参数指明了切换请求参数中所要传送给的目标小区的个数
relatedGSMEquipment	关联属性, 指实现该功能的 GSM 物理设备
adminState	管理状态
alarmStatus	告警状态
opState	运行状态
usageState	使用状态
unknownStatus	未知状态
maxNumOfBscBHCA	BSC 中 BHCA 的最大设计值
bscCapacity	BSC 的容量
trunkPortNumber	BSC 的中继端口数目
signallingPortNumber	BSC 的信令端口数目
enableInternalInterCellHandover	该属性提供管理系统允许或禁止由该 BSC 控制的小区间切换的能力
enableInternalIntraCellHandover	该属性提供管理系统允许或禁止由该 BSC 控制的小区间切换的能力
relatedPCU	关联属性, 指与该 BSC 相关的 PCU 对象实例的指针
relatedSGSN	关联属性, 指与该 BSC 相关的 SGSN 对象实例的指针, 其取值为 neCode
bscNsEntityMappingTable	该属性是 BSS 侧的网络业务实体映射表
bscNsUserEntityMappingTable	该属性是 BSS 侧的网络业务用户实体映射表

表 D.20 btsSiteManager 的信息

属 性	描 述
btsSiteManagerId	btsSiteManager 标识符, 是命名属性
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识
locationName	位置名称, 应指明该设备所处的具体房间
relatedGSMEquipment	关联属性, 指实现该功能的 GSM 物理设备
relatedOAMLapdLink	关联属性, 是相关的用于操作维护的 LAPD 链路

表 D.21 bts 的信息

属 性	描 述
btsId	bts 标识符, 是命名属性
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识
locationName	位置名称, 应指明该设备所处的具体房间
bsIdentityCode	基站识别码 (BSIC), 该值将在 SCH 上发送来表示该 BTS。该码由网络色码 (NCC) 和基站色码 (NCC) 组成
cellAllocation	分配给小区的无线频率集
gsmDcsIndicator	指示该小区的类型: GSM 或 DCS1800
cellGlobalIdentity	小区全局标识, 包括小区标识和位置区标识, 位置区标识在一个 GSM PLMN 中是惟一的, 小区标识在一个位置区中是惟一的
cellReselectHysteresis	小区重选滞后值
ny1	指小区切换过程中, 在无线接口上重传“物理信息”(PHYSICAL INFORMATION) 消息的最大次数
relatedGSMEquipment	关联属性, 指实现该功能的 GSM 物理设备
periodCCCHLoadIndication	指 CCCH 负载指示信息上报给 BSC 的频率
plmnPermitted	标识允许接入该 BTS 的移动网手机手机, 其取值为这些 PLMN 的网络色码 (NCC), 这些值将在手机接入该 BTS 时使用
rACHBusyThreshold	RACH 突发期间接收信号级别的阈值, 超过该阈值的信号将被视为一个忙 RACH
rACHLoadAveragingSlots	RACH 测量期间 RACH 突发次数
radioLinkTimeout	检测出无线链路失败的定时器超时值
relatedOAMLapdLink	关联属性, 指相关的用于操作维护的 LAPD 链路
relatedTranscoder	关联属性, 指相关的码速适配器
rxLevAccessMin	移动台接入 BTS 系统是可允许的最小功率
thresholdCCCHLoadIndication	有 BTS 通知 BSC 关于 CCCH 负荷指示的门限值
adminState	管理状态
alarmStatus	告警状态

表 D.21 (续)

属 性	描 述
opState	运行状态
unknownStatus	未知状态
usageState	使用状态
maxNumberRetransmission	移动台可在 RACH 上执行的最大重传次数
mSTxPwrMaxCCH	移动台在接入小区时可在 CCCH 上使用的最大传输功率
numberOfSlotsSpreadTrans	移动台在发生随机接入冲突后, 可以等待的最大 RACH 时隙数
noOfBlocksForAccessGrant	在 51 个 TDMA 帧 (复帧) 周期中, 收到的含有 AGCH 的 TDMA 帧的数目
noOfMultiframeBetweenPaging	指在向同一个寻呼组发送同一条寻呼消息所应间隔的复帧 (51 个 TDMA 帧) 数
allowIMSIAttachDetach	小区是否允许 IMSI 的附着/分离
callReestablishmentAllowed	小区是否允许呼叫重新建立
cellBarred	小区是否禁止移动台停留
dtxDownlink	BTS 的下行 DTX (非连续性发信) 选项; 如果允许的话, 将由 MSC 来控制下行 DTX
dtxUplink	BTS 的上行 DTX 选项, 由移动台使用
emergencyCallRestricted	是否允许所有的移动台或访问级别为 11~15 的移动台使用紧急呼叫
notAllowedAccessClasses	禁止访问该 BTS 的移动台类型
timerPeriodicUpdateMS	移动台位置更新的时间间隔
noOfTransceivers	基站收发信机 (TRX) 的数目
noOfRadioCarriers	无线载波数目
noOfSDCCH	SDCCH 数目
noOfTCH	TCH 数目
trxPower	TRX 的功率
maxQueueLength	指 BTS 中允许呼叫或切换请求排队的最大队列长度, 其取值是该 BTS 中以所有工作 TCH 数目为基数的百分比。其中: 0: 不使用队列。 100: 最大队列长度等于可用的 TCH 数目。
msPriorityUsedInQueuing	指排队处理时是否考虑来自 MSC 的分配请求或切换请求消息的呼叫优先级
timeLimitCall	呼叫请求等待 TCH 可用的最大等待时长
timeLimitHandover	切换呼叫请求等待 TCH 可用的最大等待时长
hoMsmProcessingMode	指切换控制算法的处理模式。切换控制算法的无线链路测量可以直接传送给 BSC, 也只需要将处理结果和门限比较传给 BSC, 该属性指明了 BTS 所采用的模式。其缺省值是采用前者

表 D.21 (续)

属 性	描 述
pcMsmProcessingMode	指功率控制算法的处理模式。功率控制算法的无线链路测量由 BTS 来采集, 其结果可以直接传送给 BSC 处理, 也可以将在 BTS 本地进行处理和门限比较后的结果发送给 BSC, 该属性指明了 BTS 所采用的模式。其缺省值是采用前者
cellReselectionArithm	BTS 中 GPRS 的小区重选算法
rai	BTS 中 GPRS 的路由区标识
pBCCHIndicator	BTS 中是否配置了 PBCCH; 该值为真表示配置了
pCCCHIndicator	BTS 中是否配置了 PCCCH; 该值为真表示配置了
numOfSlotsForGPRS	用于 GPRS 的时隙数
numOfTrxForGPRS	用于 GPRS 的 TRX 数目
channelEncodeMode	BTS 的信道编码模式
mediaAccessControlMode	媒质访问控制模式
relatedPCU	管理属性, 表示指示相关联的 PCU; 如果该 BTS 没有相关的 PCU, 则该属性可以为空

表 D.22 lapdLink 的信息

属 性	描 述
lapdLinkId	lapdLink 标识符, 是命名属性
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识
locationName	位置名称, 应指明该设备所处的具体房间
abisSigChannel	标识在 Abis 接口上分配给 LAPD 链路的 PCM 时隙和可能的子时隙
sapi	指 lapdLink 对象的业务接入点标识
tei	指 lapdLink 对象的终端点标识
adminState	管理状态
alarmStatus	告警状态
opState	运行状态
unknownStatus	未知状态

表 D.23 pcmCircuit 的信息

属 性	描 述
pcmCircuitId	pcmCircuit 标识符, 是命名属性
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识
adminState	管理状态

表 D.23 (续)

属 性	描 述
alarmStatus	告警状态
opState	运行状态
unknownStatus	未知状态
usageState	使用状态

表 D.24 transcoder 的信息

属 性	描 述
transcoderId	transcoder 标识符, 是命名属性
userLabel	用户友好名, 由 OMC 厂商自己指定, 可以作为 NE 和 NR 的内部标识
locationName	位置名称, 应指明该设备所处的具体房间
relatedGSMEquipment	关联属性, 指实现该功能的 GSM 物理设备
adminState	管理状态
alarmStatus	告警状态
opState	运行状态
unknown Status	未知状态

表 D.25 observedDestination 的信息

Attributes	Description
observedDestinationId	observedDestination 标识符, 是命名属性
destinationCode	目的码, 表示国家码、地区码、交换机码, 或其他该对象所指向的位置的号码
destinationType	表示目的的类型。其可以是根据 Q.763 描述的由 7 比特组成的自然地址, 或目的地类型的枚举列表中的某个值

D.3 配置信息模型 IDL 定义

整个 IDL 定义分为通用的数据类型定义模块和被管理对象接口模块 (包含所有属性定义) 两个部分。

注: 尽管本附录中定义的管理对象对 NMC-OMC 接口而言是不可见的, 但所有相关的数据类型在实现该接口时都要使用, 这是因为所有管理对象的属性值都可能插入到 CORBA 的 ANY 类型中进行传送, 因此这些类型定义对实现而言是必须的。

D.3.1 通用类型模块

```
// File NmcIfIMCommType.idl
// The types defined in this file are not just for interfaces contained in
// module "NmcIfIMComm", but also for the type "AttrValueType" defined in
// module NmcIfComm, which will use these types to fill the any
// for attribute values of different managed objects.
```

```
#ifndef _NMC_IF_IM_COMM_TYPE_IDL_
```

```

#define _NMC_IF_IM_COMM_TYPE_IDL_

#include <NmclIfComm.idl>

module NmclIfIMCommType
{
    enum eAdminStateType
    {
        LOCKED,
        UNLOCKED,
        SHUTTING_DOWN
    };
    enum eOpStateType
    {
        DISABLED,
        ENABLED
    };
    enum eAlarmStatusType
    {
        CLEARED,
        ACTIVE_REPORTABLE_INDETERMINATE,
        ACTIVE_REPORTABLE_WARNING,
        ACTIVE_REPORTABLE_MINOR,
        ACTIVE_REPORTABLE_MAJOR,
        ACTIVE_REPORTABLE_CRITICAL,
        ACTIVE_PENDING
    };
    enum eUsageStateType
    {
        IDLE,
        ACTIVE,
        BUSY
    };
    typedef boolean eUnknownStatusType;
    typedef unsigned long ObjectIdType;
    typedef sequence<ObjectIdType> ObjectIdListType;

    typedef string<5> NetworkDestCodeType;
    typedef string<4> CountryCodeType;
    typedef string<3> MobileCountryCodeType;
    typedef string<2> MobileNetworkCodeType;
    typedef string<20> ISDNAddrStringType;
    typedef sequence<NetworkDestCodeType> NetworkDestCodeSetType;
    typedef sequence<CountryCodeType> CountryCodeSetType;

```



```
typedef unsigned long NetworkColourCodeType; // 0.7
typedef unsigned long BTSColourCodeType; // 0.7
```

```
enum eBServiceIdType
```

```
{
    TS11,
    TS21,
    TS22,
    TS61,
    TS62,
    BS21,
    BS22,
    BS23,
    BS24,
    BS25,
    BS26,
    BS31,
    BS32,
    BS33,
    BS34,
    BS41,
    BS42,
    BS43,
    BS44,
    BS45,
    BS46,
    BS51,
    BS52,
    BS53,
    BS61A,
    BS61S,
    BS71,
    BS81A,
    BS81S
};
typedef sequence<eBServiceIdType> eBServiceIdListType;
```

```
enum eSServiceIdType
```

```
{
    CLIP,
    CLIR,
    CoLR,
    CoLP,
```

```

    CFU,
    CFB,
    CFNRy,
    CFNRc,
    CW,
    HOLD,
    MPTY,
    CUG,
    AoCI,
    AoCC,
    BAOC,
    BOIC,
    BOICexHC,
    BAIC,
    BICRoAM
};
typedef sequence<eSServiceIdType> eSServiceIdListType;
typedef sequence<long> ListOfValidCUGInterlockCodesType;

// the following types are got from NmcIfIMCM module
enum eNetworkIndicatorType
{
    international,
    spare,
    national,
    reservedNationalUse
};
enum eSignallingPointLengthType
{
    bits_24, // 0
    bits_14 // 1
};
struct SignallingInfoType
{
    eSignallingPointLengthType signallingPointLength;
    unsigned long signallingPointCode;
    eNetworkIndicatorType networkIndicator;
};
typedef sequence<SignallingInfoType, 4> SignallingInfoListType;

typedef unsigned long SignallingPointType;
const SignallingPointType noSTP=0;
const SignallingPointType STP=1;
struct BSIdCodeType

```

```

{
    NetworkColourCodeType ncc;
    BTSColourCodeType bcc;
};
typedef unsigned long AbsoluteRFChannelNoType; //[0..1023]
typedef sequence<AbsoluteRFChannelNoType, 64> CellAllocationType;
enum eGsmcdcsIndicatorType
{
    gsm,
    extendedGsm,
    dcs,
    multiband
};
typedef unsigned long CellIdentityType; //[0...65535]
typedef unsigned long LocationAreaCodeType; //[0...65535]
struct LocationAreaIdentityType
{
    MobileCountryCodeType mcc;
    MobileNetworkCodeType mnc;
    LocationAreaCodeType lac;
};

struct CellGlobalIdType
{
    LocationAreaIdentityType lai;
    CellIdentityType ci;
};

enum eTimeUnitType
{
    mSec,
    sec,
    min,
    noOfTDMAFrames,
    noOfSlots,
    factor
};
struct TimerDataType
{
    eTimeUnitType timeUnit;
    unsigned long timeValue;
};

typedef unsigned long NetworkColorCodeType; //[0..7]

```

```

typedef sequence<NetworkColorCodeType, 8> PlmnPermittedType;
enum eMaxRetranType
{
    one,
    two,
    four,
    seven
};
typedef unsigned long DtxUplinkType;
const DtxUplinkType msMayUseDtx=0;
const DtxUplinkType msShallUseDtx=1;
const DtxUplinkType msShallNotUseDtx=2;
typedef unsigned long ClassNumberType; //[0..15]
typedef sequence<ClassNumberType, 15> ClassNumberListType;
enum eMsmProcessingModeType
{
    basicMeasurementReporting,
    btsProcessingMeasurementReporting
};
struct PCMTimeSlotType
{
    unsigned long pcmCircuitId; //[0..2047]
    unsigned long timeSlotNum;
};
struct MultiTimeSlotType
{
    PCMTimeSlotType timeSlot;
    unsigned long subSlot;
};
struct AbisChannelType
{
    MultiTimeSlotType bscTimeSlot;
    MultiTimeSlotType btsTimeSlot;
};
struct RelatedTrunkGroupTPType
{
    unsigned long priority;
    NmclComm:: DistinguishedNameType trunkGroupTP;
};
typedef sequence<RelatedTrunkGroupTPType> RelatedTrunkGroupTPLListType;
struct RouteType
{
    string nameOfRoute;
    RelatedTrunkGroupTPLListType trunkGroupTPLList;
};

```

```

};
typedef unsigned long DigitParticipationIndicatorType;
const DigitParticipationIndicatorType seeNextCode=0;
const DigitParticipationIndicatorType internationalCode=1;
const DigitParticipationIndicatorType nationalPSTNCode=2;
const DigitParticipationIndicatorType nationalPLMNCode=3;
const DigitParticipationIndicatorType toLocalMSC=4;
const DigitParticipationIndicatorType toOwnMSC=5;
const DigitParticipationIndicatorType localPSTNCode=6;
const DigitParticipationIndicatorType localSPXCode=7;
const DigitParticipationIndicatorType anlogPLMNCode=9;
const DigitParticipationIndicatorType others=10;
typedef char DigitElementType;
// the possible value for DigitElementType are:
// '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '0',
// 'A', 'B', 'C', 'D', 'E', 'F', '*', '#'
typedef string DigitListType; //a string type formed from DigitElementType

enum eTrunkGroupType
{
    toInternationalToll,
    toNationalToll,
    toLocalTandem,
    toLocalEnd,
    toAnalogMobileTransit,
    toGSMFirstTransit,
    toGSMSecondTransit,
    toGSMMSC,
    toBSC,
    toVoiceMailbox,
    toOthers
};

enum eCircuitType
{
    lowCallLoss,
    highEfficiencyDirect
};
enum eCircuitDirectionalityType
{
    onewayOut,
    onewayIn,
    twoway
};

```

```

typedef unsigned long TransmissionCharacteristicsType;
const TransmissionCharacteristicsType opticalFiberCable=1;
const TransmissionCharacteristicsType coaxialCable=2;
const TransmissionCharacteristicsType analogMicrowave=3;
const TransmissionCharacteristicsType digitMicrowave=4;
const TransmissionCharacteristicsType satellite=5;
const TransmissionCharacteristicsType mixedGroup=6;
const TransmissionCharacteristicsType transOthers=7;
typedef sequence<TransmissionCharacteristicsType, 7> TransmissionCharacteristicsSetType;
enum eDestType
{
    international_dest,
    national_dest,
    local_dest,
    other_dest
};
typedef sequence<octet, 7> NatureOfAddressType;
//only 0 and 1 can occur in the string, it is of BITSTRING type.
enum eDestinationTypeChoiceType
{
    NatureOfAddressChoice,
    DestTypeChoice
};

union DestinationType switch (eDestinationTypeChoiceType)
{
    case NatureOfAddressChoice : NatureOfAddressType natureOfAddress;
    case DestTypeChoice : eDestType destType;
};

typedef string NECodeType;
typedef sequence<NECodeType> NECodeListType;
typedef sequence<eChannelEncodeModeType> eChannelEncodeModeListType;
typedef sequence<eCellReselectArithmType> CellReselectArithmListType;

enum eOmcType
{
    omc, // 0
    omc_s, // 1
    omc_r // 2
};

typedef unsigned short SignallingLinkPriorityType;

```

```
typedef unsigned long RoutingAreaIdType;
```

```
enum eChannelEncodeModeType
```

```
{
    cs1, // 0
    cs2, // 1
    cs3, // 2
    cs4 // 3
};
```

```
enum eMediaAccessControlModeType
```

```
{
    dynamicAllocation, //0
    extendedDynamicAllocation, //1
    fixedAllocation //2
};
```

```
enum eCellReselectArithmType
```

```
{
    c1, // 0
    c2, // 1
    c31, // 2
    c32 // 3
};
```

```
enum eNEType
```

```
{
    neTypeMsc, // 0
    neTypeVlr, // 1
    neTypeHlr, // 2
    neTypeAuc, // 3
    neTypeEir, // 4
    neTypeBss, // 5
    neTypeOmc, // 6
    neTypeSms // 7
};
```

```
typedef sequence<eNEType> NEListType;
```

```
typedef float TrxPowerType;
```

```
enum eMscType
```

```
{
```

```

    mscTypeTMSC1,    // 0
    mscTypeTMSC2,    // 1
    mscTypeGMSC,     // 2
    mscTypeMSCVLR    // 3
};

struct MscUpgradeType
{
    boolean upgradeToSSP;
    boolean upgradeToPOI;
};

typedef string ProbableCauseType;
struct CurrentProblemType
{
    ProbableCauseType problem;
    eAlarmStatusType alarmStatus;
};
typedef sequence<CurrentProblemType> CurrentProblemListType;

union CurrentProblemListTypeOpt switch (boolean)
{
    case TRUE: CurrentProblemListType problemList;
    // when the implementor doesn't support currenProblemList attributes,
    // set the discriminator value to FALSE.
};

struct BssNsUserEntityMappingItemType
{
    string userEntity;
    unsigned long bvci;
    unsigned long nsei;
};
typedef sequence<BssNsUserEntityMappingItemType> BssNsUserEntityMappingTableType;

struct BssNsEntityMappingItemType
{
    unsigned long nsei;
    unsigned long bvci;
    unsigned long nsvci;
    unsigned long dlci;
    unsigned long bearerChannel;
    float loadShareMechanism;
};

```



```

typedef sequence<BssNsEntityMappingItemType> BssNsEntityMappingTableType;

typedef unsigned short SLSType; // 0..15
typedef sequence<SLSType, 16> SLSCoedListType;

union SourceNumberType switch (boolean)
{
    // when the routingInfo is corresponded to a source Number
    // the 'number' field shall be used , or else the discriminator
    // may take the value of FALSE.
    case TRUE:
        string number; //a string formed from DigitElement,
};
struct RoutePairType
{
    SourceNumberType sourceNumber;
    NmclComm:: DistinguishedNameType routeInfo;
};
typedef sequence<RoutePairType> RelatedRouteInfoType;

};

#endif

```

D.3.2 被管理对象接口模块

```

// File: NmcIfIMCM.idl
// This file contains object definition in CM part, and is also just for documentation.
// This file is derived from NMC_OMC Interface Specification for TDMA GSM Network
// Management——Information Model for Configuration Management

```

```

#ifndef _NMC_IF_IM_CM_IDL_
#define _NMC_IF_IM_CM_IDL_

#include <NmcIfIMCommType.idl>

module NmcIfIMCM
{
    interface network
    {
        attribute NmcIfIMCommType:: ObjectIdType networkId;
        attribute string userLabel; //conditional
    };

    interface plmnSubNetwork: network

```

```

{
    attribute NmcIfIMCommType:: MobileCountryCodeType mcc; // mobile country code of the network
    attribute NmcIfIMCommType:: MobileNetworkCodeType mnc; //mobile network code
    attribute NmcIfIMCommType:: CountryCodeSetType setOfCc; // set of country code
    attribute NmcIfIMCommType:: NetworkDestCodeSetType setOfNdc; //set of network destination code
    attribute NmcIfIMCommType:: eBServiceIdListType listOfSupportedBS; //list of supported basic
service
    attribute NmcIfIMCommType:: eSServiceIdListType listOfSupportedSS; //list of supported
supplementary service

    // The following notifications may be sent from this MO,
    // objectCreation
    // objectDeletion
    // attributeValueChange
    // stateChange
};

interface managedElement
{
    attribute NmcIfIMCommType:: ObjectIdType managedElementId;
    attribute NmcIfIMCommType:: eAlarmStatusType alarmStatus;

    // The following notifications may be sent from this MO,
    // objectCreation
    // objectDeletion
    // stateChange
};

interface software
{
    attribute NmcIfIMCommType:: ObjectIdType softwareId;
    attribute NmcIfIMCommType:: eAdminStateType adminState;
    attribute NmcIfIMCommType:: eOpStateType opState;
    attribute string version;
    attribute string vendorName;
    attribute NmcIfIMCommType:: CurrentProblemListType currentProblemList; //conditional

    // The following notifications may be sent from this MO,
    // objectCreation
    // objectDeletion
    // attributeValueChange
    // stateChange
    // processingErrorAlarm
};

```

```

interface gsmEquipment
{
    attribute NmclFIMCommType:: ObjectIdType equipmentId;
    attribute NmclFIMCommType:: eAdminStateType adminState;
    attribute NmclFIMCommType:: eOpStateType opState;
    attribute NmclFIMCommType:: eAlarmStatusType alarmStatus;
    attribute NmclFIMCommType:: ManagedObjectListType relatedGSMFunction;
    attribute string userLabel;
    attribute string vendorName;
    attribute string locationName;
    attribute string version;
    attribute NmclFIMCommType:: NECodeType neCode;

    // The following notifications may be sent from this MO,
    // objectCreation
    // objectDeletion
    // attributeValueChange
    // stateChange
    // communicationsAlarm
    // environmentalAlarm
    // equipmentAlarm
    // processingErrorAlarm
};

interface gsmManagedFunction
{
    attribute string userLabel;           //conditional
    attribute string locationName;       //conditional

    // The following notifications may be sent from the subclass of this MO
    // objectCreation
    // objectDeletion
    // attributeValueChange
};

interface signallingLinkTP
{
    attribute NmclFIMCommType:: ObjectIdType signallingLinkTPId;
    attribute NmclFIMCommType:: eAdminStateType adminState;
    attribute NmclFIMCommType:: eOpStateType opState;
    attribute unsigned long signallingLinkCode; //range [0, 15]
    attribute NmclFIMCommType:: eUsageStateType usageState;
    attribute NmclFIMCommType:: SignallingLinkPriorityType signallingLinkPriority;
    attribute string userLabel;

```

```

attribute NmcIfIMCommType:: SLSCodeListType slsCodeNormalList;
attribute NmcIfIMCommType:: SLSCodeListType slsCodeCurrentList;

```

```

// The following notifications may be sent from this MO,
// objectCreation
// objectDeletion
// attributeValueChange
// stateChange
// communicationsAlarm
// processingErrorAlarm

```

```
};
```

```
interface signallingLinkSetTP
```

```
{
```

```

attribute NmcIfIMCommType:: ObjectIdType signallingLinkSetTPId;
attribute string signallingLinkSetName;
attribute unsigned long numOfSignallingLinks;
attribute NmcIfIMCommType:: SignallingInfoListType adjacentSignallingInfo;
attribute string userLabel;

```

```

// The following notifications may be sent from this MO,
// objectCreation
// objectDeletion
// attributeValueChange
// processingErrorAlarm

```

```
};
```

```
interface signallingPoint
```

```
{
```

```

attribute NmcIfIMCommType:: ObjectIdType signallingPointId;
attribute NmcIfIMCommType:: SignallingInfoListType signallingInfo;
attribute NmcIfIMCommType:: SignallingPointType signallingPointType;
attribute string userLabel;

```

```

// The following notifications may be sent from this MO,
// objectCreation
// objectDeletion
// attributeValueChange

```

```
};
```

```
interface omcFunction
```

```
{
```

```

attribute string omcFunctionId;
attribute NmcIfComm:: DNListType relatedGSMEquipment;

```

```

attribute NmcIfIMCommType:: eOmcType omcType;
attribute string userLabel;

```

```

// The following notifications may be sent from this MO,
// objectCreation
// objectDeletion
// attributeValueChange
// processingErrorAlarm

```

```
};
```

```
interface aucFunction
```

```
{
```

```

attribute NmcIfIMCommType:: ObjectIdType aucFunctionId;
attribute NmcIfIMCommType:: eAdminStateType adminState;
attribute NmcIfIMCommType:: eOpStateType opState;
attribute unsigned long maxNumberOfLogicalAuc;
attribute unsigned long currentNumberOfLogicalAuc;
attribute unsigned long maxNumberOfImsiInAuc;
attribute unsigned long currentNumberOfImsiInAuc;
attribute string userLabel;
attribute NmcIfComm:: DNListType relatedGSMEquipment;

```

```

// The following notifications may be sent from this MO,
// objectCreation
// objectDeletion
// attributeValueChange
// stateChange

```

```
};
```

```
interface eirFunction
```

```
{
```

```

attribute NmcIfIMCommType:: ObjectIdType eirFunctionId;
attribute NmcIfIMCommType:: eAdminStateType adminState;
attribute NmcIfIMCommType:: eOpStateType opState;
attribute string eirId;
attribute NmcIfIMCommType:: ISDNAddrStringType eirNumber;
attribute unsigned long maxNumOfWhiteListEntries;
attribute unsigned long maxNumOfGreyListEntries;
attribute unsigned long maxNumOfBlackListEntries;
attribute unsigned long curNumOfWhiteListEntries;
attribute unsigned long curNumOfGreyListEntries;
attribute unsigned long curNumOfBlackListEntries;
attribute string userLabel;
attribute NmcIfComm:: DNListType relatedGSMEquipment;

```

```

    // The following notifications may be sent from this MO,
    // objectCreation
    // objectDeletion
    // attributeValueChange
    // stateChange
};
interface hlrFunction
{
    attribute NmcIfIMCommType:: ObjectIdType hlrFunctionId;
    attribute NmcIfIMCommType:: eAdminStateType adminState;
    attribute NmcIfIMCommType:: eOpStateType opState;
    attribute unsigned long maxNumOfLogicalHlr;
    attribute unsigned long curNumOfLogicalHlr;
    attribute unsigned long maxNumOfImsiInHlr;
    attribute unsigned long curNumOfImsiInHlr;
    attribute unsigned long maxNumOfMsisdnInHlr;
    attribute unsigned long curNumOfMsisdnInHlr;
    attribute string defaultPW;
    attribute NmcIfIMCommType:: ISDNAddrStringType defAnnouncement;
    attribute NmcIfIMCommType:: ListOfValidCUGInterlockCodesType listOfValidCUGInterlockCodes;
    attribute NmcIfIMCommType:: eUsageStateType usageState;
    attribute string userLabel;
    attribute NmcIfComm:: DNListType relatedGSMEquipment;
    attribute NmcIfIMCommType:: ISDNAddrStringType hlrNumber;
    attribute boolean supportGPRS; //conditional

    // The following notifications may be sent from this MO,
    // objectCreation
    // objectDeletion
    // attributeValueChange
    // stateChange
};

interface smsGIWFunction
{
    attribute NmcIfIMCommType:: ObjectIdType smsGIWFunctionId;
    attribute NmcIfIMCommType:: eAdminStateType adminState;
    attribute NmcIfIMCommType:: eOpStateType opState;
    attribute string userLabel;
    attribute NmcIfComm:: DNListType relatedGSMEquipment;

    // The following notifications may be sent from this MO,
    // objectCreation
    // objectDeletion

```

```

    // attributeValueChange
    // stateChange
};

interface vlrFunction
{
    attribute NmclFIMCommType:: ObjectIdType vlrFunctionId;
    attribute NmclFIMCommType:: eAdminStateType adminState;
    attribute NmclFIMCommType:: eOpStateType opState;
    attribute string vlrId;
    attribute NmclFIMCommType:: ISDNAddrStringType vlrNumber;
    attribute unsigned long maxNumOfImsiInVlr;
    attribute unsigned long curNumOfImsiInVlr;
    attribute string userLabel;
    attribute NmclFComm:: DNListType relatedGSMEquipment;
    attribute NmclFIMCommType:: eUsageStateType usageState;

    // The following notifications may be sent from this MO,
    // objectCreation
    // objectDeletion
    // attributeValueChange
    // stateChange
};

interface ssFunction
{
    attribute NmclFIMCommType:: ObjectIdType ssFunctionId;
    attribute NmclFIMCommType:: eAdminStateType adminState;
    attribute NmclFIMCommType:: eOpStateType opState;
    attribute NmclFIMCommType:: eUsageStateType usageState;
    attribute string userLabel;
    attribute NmclFComm:: DNListType relatedGSMEquipment;

    // The following notifications may be sent from this MO,
    // objectCreation
    // objectDeletion
    // attributeValueChange
    // stateChange
};

interface mscFunction
{
    attribute NmclFIMCommType:: ObjectIdType mscFunctionId;
    attribute NmclFIMCommType:: eAdminStateType adminState;

```

```

attribute NmcIfIMCommType:: eOpStateType opState;
attribute string mscId;
attribute NmcIfIMCommType:: ISDNAddrStringType mscNumber;
attribute string userLabel;
attribute NmcIfComm:: DNListType relatedGSMEquipment;
attribute NmcIfIMCommType:: eUsageStateType usageState;
attribute unsigned long maxMscBHCA;
attribute unsigned long mscCapacity;
attribute unsigned long maxNumOfCPUConfigurable;
attribute NmcIfIMCommType:: eMscType mscType;
attribute NmcIfIMCommType:: MscUpgradeType mscUpgradeIndicator;
attribute unsigned long numOf2MPort;
attribute unsigned long mscIWFDataBlockCapacity;
attribute unsigned long numOfIWFDataBlock;

```

```

// The following notifications may be sent from this MO,
// objectCreation
// objectDeletion
// attributeValueChange
// stateChange

```

```

};

```

```

interface circuitEndPointSubgroup

```

```

{

```

```

attribute NmcIfIMCommType:: ObjectIdType circuitEndPointSubgroupId;
attribute unsigned long numOfCircuits;
attribute NmcIfIMCommType:: eCircuitDirectionalityType circuitDirectionality;
attribute NmcIfIMCommType:: TransmissionCharacteristicsSetType transmissionCharacteristics;
attribute string userLabel;
attribute NmcIfIMCommType:: eCircuitType circuitType;
attribute NmcIfIMCommType:: eTrunkGroupType trunkGroupType;
attribute NmcIfIMCommType:: eAdminStateType adminState;
attribute NmcIfIMCommType:: eOpStateType opState;
attribute NmcIfIMCommType:: eAlarmStatusType alarmStatus;
attribute unsigned long maxNumOfCircuitsConfigurable;
attribute NmcIfIMCommType:: SignallingInfoType signallingInfoOfFarEnd;
attribute NmcIfComm:: DistinguishedNameType signallingPointOfFarEnd;

```

```

// The following notifications may be sent from this MO,
// objectCreation
// objectDeletion
// attributeValueChange
// stateChange
// communicationsAlarm

```



```

    // processingErrorAlarm
};

interface digitsSegment
{
    attribute NmcfIMCommType:: ObjectIdType digitsSegmentId;
    attribute NmcfIMCommType:: DigitListType digitList;
    attribute NmcfIMCommType:: DigitParticipationIndicatorType digitParticipationIndicator;
    attribute string labelOfFarEndExchange;
    attribute NmcfIMCommType:: RelatedRouteInfoType relatedRouteInfo;

    // The following notifications may be sent from this MO,
    // objectCreation
    // objectDeletion
    // attributeValueChange
};

interface routingInfo
{
    attribute NmcfIMCommType:: ObjectIdType routingInfoId;
    attribute NmcfIMCommType:: RouteType firstChoiceRoute;
    attribute NmcfIMCommType:: RouteType firstCircuitousRoute;
    attribute NmcfIMCommType:: RouteType secondCircuitousRoute;
    attribute NmcfIMCommType:: RouteType thirdCircuitousRoute;

    // The following notifications may be sent from this MO,
    // objectCreation
    // objectDeletion
    // attributeValueChange
};

interface observedDestination
{
    attribute NmcfIMCommType:: ObjectIdType observedDestinationId;
    attribute string destinationCode;
    attribute NmcfIMCommType:: DestinationType destinationType;

    // no notification is required to be sent by this MO ??
};

interface bssFunction
{
    attribute NmcfIMCommType:: ObjectIdType bssFunctionId;
    attribute string userLabel;

```

```

    // no notification is required to be sent by this MO
};

interface bsc: gsmManagedFunction
{
    attribute NmcIfIMCommType:: ObjectIdType bscId;
    attribute unsigned long handoverReqParam;
    attribute NmcIfComm:: DNListType relatedGSMEquipment;
    attribute NmcIfIMCommType:: eAdminStateType adminState;
    attribute NmcIfIMCommType:: eAlarmStatusType alarmStatus;
    attribute NmcIfIMCommType:: eOpStateType opState;
    attribute NmcIfIMCommType:: eUnknownStatusType unknownStatus;
    attribute NmcIfIMCommType:: eUsageStateType usageState;
    attribute unsigned long maxNumOfBscBHCA;
    attribute unsigned long bscCapacity;
    attribute unsigned long trunkPortNumber;
    attribute unsigned long signallingPortNumber;
    attribute boolean enableInternalInterCellHandover; //conditional
    attribute boolean enableInternalIntraCellHandover; //conditional
    attribute NmcIfComm:: DNListType relatedPCU; //conditional
    attribute NmcIfIMCommType:: NECodeType relatedSGSN; //conditional
    attribute NmcIfIMCommType:: BssNsEntityMappingTableType bssNsEntityMappingTable; //conditional
    attribute NmcIfIMCommType:: BssNsUserEntityMappingTableType bssNsUserEntityMappingTable;
//conditional

    // The following notifications may be sent from this MO,
    // objectCreation
    // objectDeletion
    // attributeValueChange
    // stateChange
    // communicationsAlarm
    // processingErrorAlarm
    // qualityofServiceAlarm    --optional
    // equipmentAlarm,         --conditional
    // environmentalAlarm,     --conditional
};

interface btsSiteManager: gsmManagedFunction
{
    attribute NmcIfIMCommType:: ObjectIdType btsSiteManagerId;
    attribute NmcIfComm:: DNListType relatedGSMEquipment;
    attribute NmcIfComm:: DistinguishedNameType relatedOAMLapdLink;

    // The following notifications may be sent from this MO,

```

```

// objectCreation
// objectDeletion
// attributeValueChange
// equipmentAlarm,          --conditional
// environmentalAlarm,     --conditional
};

interface bts: gsmManagedFunction
{
    attribute NmcIfIMCommType:: ObjectIdType btsId;
    attribute NmcIfIMCommType:: BSIdCodeType bsIdCode;
    attribute NmcIfIMCommType:: CellAllocationType cellAllocation;
    attribute NmcIfIMCommType:: eGsmcdcsIndicatorType gsmcdcsIndicator;
    attribute NmcIfIMCommType:: CellGlobalIdType cellGlobalIdentity;
    attribute unsigned long cellReselectHysteresis; //[0..7]
    attribute unsigned long ny1;
    attribute NmcIfComm:: DNListType relatedGSMEquipment;
    attribute NmcIfIMCommType:: TimerDataType periodCCCHLoadIndication;
    attribute NmcIfIMCommType:: PlmnPermittedType plmnPermitted;
    attribute unsigned long rACHBusyThreshold; //[0..63]
    attribute unsigned long rACHLoadAveragingSlots; //[0..65535]
    attribute unsigned long radioLinkTimeout; //[0..15]
    attribute NmcIfComm:: DistinguishedNameType relatedOAMLapdLink;
    attribute NmcIfComm:: DistinguishedNameType relatedTranscoder;
    attribute unsigned long rxLevAccessMin; //[0..63]
    attribute long thresholdCCCHLoadIndication;
    // state attribute
    attribute NmcIfIMCommType:: eAdminStateType adminState;
    attribute NmcIfIMCommType:: eAlarmStatusType alarmStatus;
    attribute NmcIfIMCommType:: eOpStateType opState;
    attribute NmcIfIMCommType:: eUnknownStatusType unknownStatus;
    attribute NmcIfIMCommType:: eUsageStateType usageState;
    // bts common control channel configuration
    attribute NmcIfIMCommType:: eMaxRetranType maxNumberRetransmission;
    attribute unsigned long mSTxPwrMaxCCH;
    attribute unsigned long numOfSlotsSpreadTrans; //[0..15]
    attribute unsigned long noOfBlocksForAccessGrant; //[0..7]
    attribute unsigned long noOfMultiframesBetweenPaging; //[2..9]
    // bts options
    attribute boolean allowIMSIAttachDetach;
    attribute boolean callReestablishmentAllowed;
    attribute boolean cellBarred;
    attribute boolean dtxDownlink;
    attribute NmcIfIMCommType:: DtxUplinkType dtxUplink;

```

```

attribute boolean emergencyCallRestricted;
attribute NmcIfIMCommType:: ClassNumberListType notAllowedAccessClasses;
attribute unsigned long timerPeriodicUpdateMS; //[0..255]
//ctelbts package
attribute unsigned long noOfTransceivers;
attribute unsigned long noOfRadioCarriers;
attribute unsigned long noOfSDCCH;
attribute unsigned long noOfTCH;
attribute NmcIfIMCommType:: TrxPowerType trxPower;
attribute NmcIfComm:: DNListType adjacentCellList;
// bts queuing
attribute unsigned long maxQueuingLength; //[0..100] //conditional
attribute boolean msPriorityUsedInQueuing; //conditional
attribute unsigned long timeLimitHandover; //in seconds //conditional
attribute unsigned long timeLimitCall; //in seconds //conditional
//hoMsmtProcessingMode and pcMsmtProcessingMode
attribute NmcIfIMCommType:: eMsmtProcessingModeType hoMsmtProcessingMode; //conditional
attribute NmcIfIMCommType:: eMsmtProcessingModeType pcMsmtProcessingMode; //conditional
// bts GPRS related attributes
attribute NmcIfIMCommType:: eCellReselectArithmListType cellReselectArithm; //conditional
attribute NmcIfIMCommType:: RoutingAreaIdType rai; //conditional
attribute boolean pBCCHIndicator; //conditional
attribute boolean pCCCHIndicator; //conditional
attribute unsigned long numOfSlotsForGPRS; //conditional
attribute unsigned long numOfTrxForGPRS; //conditional
attribute NmcIfIMCommType:: eChannelEncodeModeListType channelEncodeMode; //conditional
attribute NmcIfIMCommType:: eMediaAccessControlModeType mediaAccessControlMode; //conditional
attribute NmcIfComm:: DNListType relatedPCU; //conditional

// The following notifications may be sent from this MO,
// objectCreation
// objectDeletion
// attributeValueChange
// stateChange
// communicationsAlarm
// processingErrorAlarm
// qualityofServiceAlarm --optional
// equipmentAlarm, --conditional
// evironmentalAlarm, --conditional
);

interface lapdLink: gsmManagedFunction
{
attribute NmcIfIMCommType:: ObjectIdType lapdLinkId;

```

```

attribute NmcIfIMCommType:: AbisChannelType abisSigChannel;
attribute unsigned long sapi; /[0..63]
attribute unsigned long tei; /[0..127]
attribute NmcIfIMCommType:: eAdminStateType adminState;
attribute NmcIfIMCommType:: eAlarmStatusType alarmStatus;
attribute NmcIfIMCommType:: eOpStateType opState;
attribute NmcIfIMCommType:: eUnknownStatusType unknownStatus;

```

```

// The following notifications may be sent from this MO,
// objectCreation
// objectDeletion
// attributeValueChange
// stateChange
// communicationsAlarm
// processingErrorAlarm
// qualityofServiceAlarm --optional

```

```
};
```

```
interface pcmCircuit: gsmManagedFunction
```

```
{
```

```

attribute NmcIfIMCommType:: ObjectIdType pcmCircuitId;
attribute NmcIfIMCommType:: eAdminStateType adminState;
attribute NmcIfIMCommType:: eAlarmStatusType alarmStatus;
attribute NmcIfIMCommType:: eOpStateType opState;
attribute NmcIfIMCommType:: eUnknownStatusType unknownStatus;
attribute NmcIfIMCommType:: eUsageStateType usageState;
// this MOC does need locationName

```

```

// The following notifications may be sent from this MO,
// objectCreation
// objectDeletion
// attributeValueChange
// stateChange
// communicationsAlarm
// processingErrorAlarm
// qualityofServiceAlarm --optional
// equipmentalarm --conditional
// environmentalAlarm --conditional

```

```
};
```

```
interface pcu: gsmManagedFunction
```

```
{
```

```

attribute NmcIfIMCommType:: ObjectIdType pcuId;
attribute NmcIfComm:: DNListType relatedGSMEquipment;

```

```

attribute NmcIfIMCommType:: NECodeType relatedSGSN;
attribute unsigned long numOfPDCHSupported;
attribute NmcIfIMCommType:: eAdminStateType adminState;
attribute NmcIfIMCommType:: eAlarmStatusType alarmStatus;
attribute NmcIfIMCommType:: eOpStateType opState;
attribute NmcIfIMCommType:: eUsageStateType usageState;
attribute NmcIfIMCommType:: eUnknownStatusType unknownStatus;

```

```

// The following notifications may be sent from this MO,
// objectCreation
// objectDeletion
// attributeValueChange
// stateChange
// communicationsAlarm
// processingErrorAlarm
// qualityofServiceAlarm    --optional
// equipmentalarm           --conditional
// environmentalAlarm       --conditional

```

```
};
```

```
interface transcoder: gsmManagedFunction
```

```
{
```

```

attribute NmcIfIMCommType:: ObjectIdType transcoderId;
attribute NmcIfComm:: DNListType relatedGSMEquipment;
attribute NmcIfIMCommType:: eAdminStateType adminState;
attribute NmcIfIMCommType:: eAlarmStatusType alarmStatus;
attribute NmcIfIMCommType:: eOpStateType opState;
attribute NmcIfIMCommType:: eUnknownStatusType unknownStatus;

```

```

// The following notifications may be sent from this MO,
// objectCreation
// objectDeletion
// attributeValueChange
// stateChange
// communicationsAlarm
// processingErrorAlarm
// qualityofServiceAlarm    --optional
// equipmentalarm           --conditional
// environmentalAlarm       --conditional

```

```
};
```

```
};
```

```
#endif
```

D.4 配置管理对象包含树

配置管理对象包含树如图 D.1 所示。

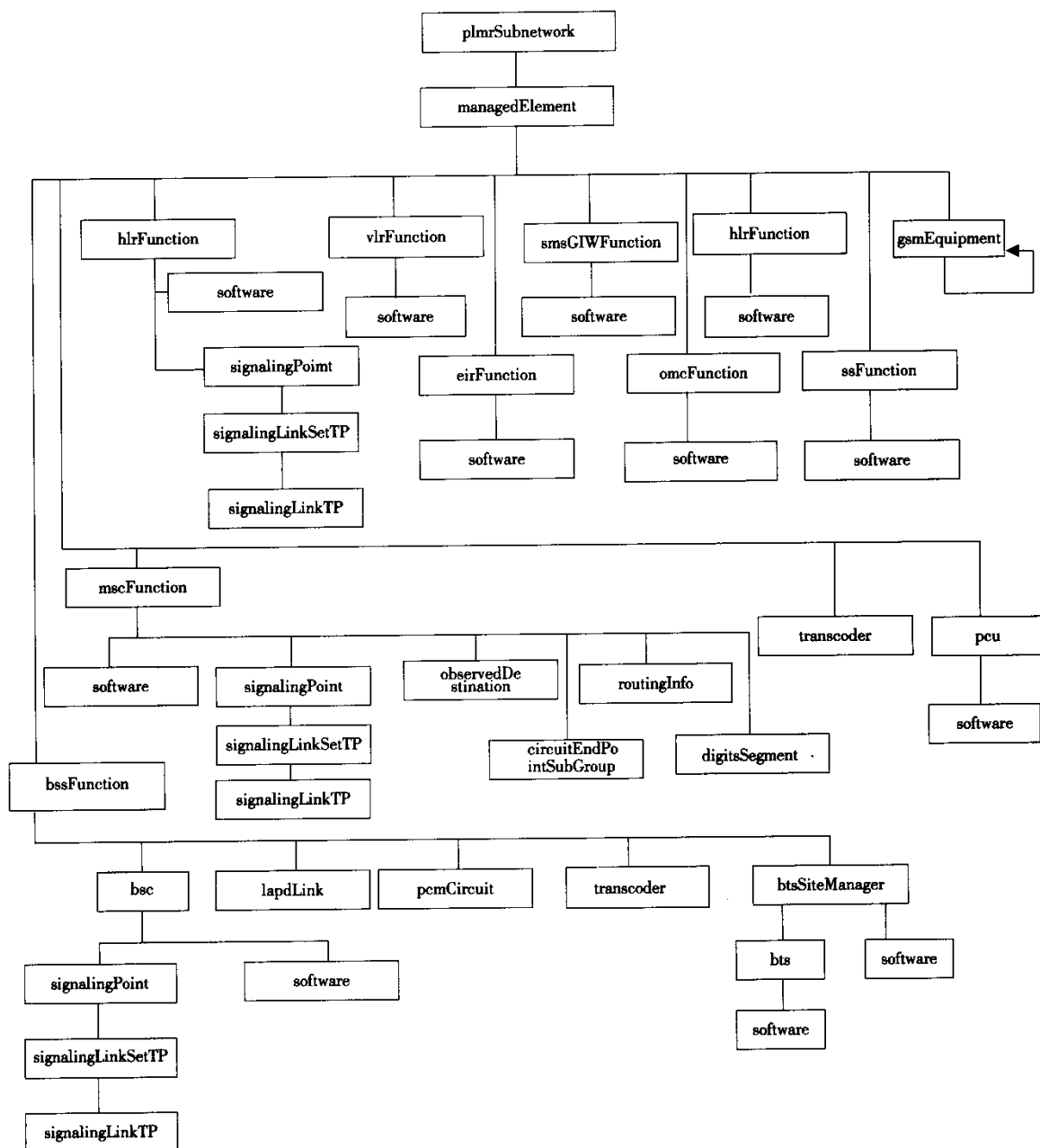


图 D.1 配置管理对象包含树

D.5 接口中使用的对象类名和属性名

本节给出了 NMC-OMC 接口中所用到的管理对象类名和属性名的详细定义。

```
// File: NmclFIMBaseType.idl
// This file contains the class names and attribute names .
```

```
#ifndef _NMC_IF_IM_BASE_TYPE_IDL_
#define _NMC_IF_IM_BASE_TYPE_IDL_
```

```
module NmclFIMBaseType
{
    enum eClassIdType
    {
        Network,
        PLMNSubNetwork,
        ManagedElement,
        GSMEquipment,
        Software,
        SignallingPoint,
        SignallingLinkSetTP,
        SignallingLinkTP,
        OMCFunction,
        MSCFunction,
        HLRFunction,
        VLRFunction,
        AUCFunction,
        BSSFunction,
        SMSGIWFunction,
        SSFunction,
        EIRFunction,
        CircuitEndPointSubgroup,
        RoutingInfo,
        DigitsSegment,
        ObservedDestination,
        BSC,
        BTSSiteManager,
        BTS,
        PCMCircuit,
        Transcoder,
        LAPDLink,
        PCU
    };

    enum eAttrIdType
    {
        //common attributeId
        userLabel,
```



```
version,  
locationName,  
vendorName,  
relatedGSMEquipment,  
//state attributes  
adminState,  
alarmStatus,  
opState,  
usageState,  
unknownStatus,  
  
//class network  
networkId,  
//class plmnNetwork: network  
mcc, //mobile country code of the network  
mnc, //mobile network code  
setOfCc, //set of country code  
setOfNdc, //set of network destination code  
listOfSupportedBS, //list of supported basic service  
listOfSupportedSS, //list of supported supplementary service  
//class plmnSubNetwork: network  
  
//class managedElement  
managedElementId,  
//class software  
softwareId,  
currentProblemList,  
//class gsmEquipment  
equipmentId,  
relatedGSMFunction,  
neCode,  
//class gsmManagedFunction  
  
//class signallingLink  
signallingLinkTPIId,  
signallingLinkCode,  
signallingLinkPriority,  
slsCodeNormalList,  
slsCodeCurrentList,  
//class signallingLinkSet  
signallingLinkSetTPIId,  
signallingLinkSetName,  
numOfSignallingLinks,  
adjacentSignallingInfo,
```

```

//class signallingPoint
signallingPointId,
signallingInfo,
signallingPointType,

//class omcFunction
omcFunctionId,
omcType,
//class aucFunction
aucFunctionId,
maxNumberOfLogicalAuc,
currentNumberOfLogicalAuc,
maxNumberOfImsiInAuc,
currentNumberOfImsiInAuc,
//class eirFunction
eirFunctionId,
eirId,
eirNumber,
maxNumOfWhiteListEntries,
maxNumOfGreyListEntries,
maxNumOfBlackListEntries,
curNumOfWhiteListEntries,
curNumOfGreyListEntries,
curNumOfBlackListEntries,
//class hlrFunction
hlrFunctionId,
maxNumOfLogicalHlr,
curNumOfLogicalHlr,
maxNumOfImsiInHlr,
curNumOfImsiInHlr,
maxNumOfMsisdnInHlr,
curNumOfMsisdnInHlr,
defaultPW,
defAnnouncement,
listOfValidCUGInterlockCodes,
hlrNumber,
supportGPRS,
//class smsGIWFunction
smsGIWFunctionId,
//class vlrFunction
vlrFunctionId,
vlrId,
vlrNumber,
maxNumOfImsiInVlr,
curNumOfImsiInVlr,

```

```
//class ssFunction
ssFunctionId,

//class mscFunction
mscFunctionId,
mscId,
mscNumber, //ISDN address string
maxMscBHCA,
mscCapacity,
maxNumOfCPUConfigurable,
mscType,
mscUpgradeIndicator,
numOf2MPort,
mscIWFDatablockCapacity,
numOfIWFDatablock,
// class circuitEndPointSubgroup
circuitEndPointSubgroupId,
numOfCircuits,
circuitDirectionality,
transmissionCharacteristics,
circuitType,
trunkGroupType,
maxNumOfCircuitsConfigurable,
signallingInfoOffFarEnd,
signallingPointOffFarEnd,
//class digitsSegment
digitsSegmentId,
digitList,
digitParticipationIndicator,
labelOffFarEndExchange,
relatedRouteInfo,
//class routingInfo
routingInfoId,
firstChoiceRoute,
firstCircuitousRoute,
secondCircuitousRoute,
thirdCircuitousRoute,
//class observedDestination
observedDestinationId,
destinationCode,
destinationType,

//class bssFunction
bssFunctionId,
//class bsc
bscId,
```

```

handoverReqParam,
maxNumOfBscBHCA,
bscCapacity,
trunkPortNumber,
signallingPortNumber,
relatedPCU,
relatedSCSN,
enableInternalInterCellHandover,
enableInternalIntraCellHandover,
bssNsEntityMappingTable,
bssNsUserEntityMappingTable,
//class btsSiteManager
btsSiteManagerId,
relatedOAMLapdLink,

// class bts
btsId,
bsIdCode, // base station identity code NCC+BCC gsm1220
cellAllocation,
gsmdcsIndicator,
cellGlobalIdentity,
cellReselectHysteresis,
ny1,
periodCCCHLoadIndication,
plmnPermitted,
rACHBusyThreshold,
rACHLoadAveragingSlots,
radioLinkTimeout,
relatedTranscoder,
rxLevAccessMin,
thresholdCCCHLoadIndication,
//bts common control channel configuration
maxNumberRetransmission,
mSTxPwrMaxCCH,
numOfSlotsSpreadTrans,
noOfBlocksForAccessGrant,
noOfMultiframesBetweenPaging,
//bts options
allowIMSIAttachDetach,
callReestablishmentAllowed,
cellBarred,
dtxDownlink,
dtxUplink,
emergencyCallRestricted,
notAllowedAccessClasses,

```

```

timerPeriodicUpdateMS,
//ctelBts package
noOfTransceivers,
noOfRadioCarriers,
noOfSDCCH,
noOfTCH,
trxPower,
adjacentCellList,
// bts queuing
maxQueuingLength, //conditional
msPriorityUsedInQueuing, //conditional
timeLimitHandover, //conditional
timeLimitCall, //conditional
// bts hoMsmProcessingMode
hoMsmProcessingMode, //conditional
pcMsmProcessingMode, //conditional
// bts GPRS related attributes
cellReselectArithm, //conditional
rai, //conditional
pBCCHIndicator, //conditional
pCCCHIndicator, //conditional
numOfSlotsForGPRS, //conditional
numOfTrxForGPRS, //conditional
channelEncodeMode, //conditional
mediaAccessControlMode, //conditional

//class lapdLink
lapdLinkId,
abisSigChannel,
sapi,
tei,
//class pcmCircuit
pcmCircuitId,
//class pcu
pcuId,
numOfPDCHSupported,
//class transcoder
transcoderId
};
};

#endif

```

附录 E
(规范性附录)
故障原因定义

E.1 简介

本附录列出了定义在不同 ITU-T 建议和 ETSI 建议中的可能故障原因。对于每一个故障原因，本标准分配了唯一的值来进行标识（其中值 1000000 留作厂家自定义的故障原因的标识），即表格中“取值”域的值。告警通知中的 ProbableCause 域将使用本附录中所分配的故障原因的标识值。

对于在多个建议中重复定义的故障原因，E.3 节中提出了解决的方法。

E.2 故障原因定义**E.2.1 M.3100 中的故障原因定义**

本节以表格形式给出了定义在 M.3100 中的可能故障原因。从 0 到 999999 的所有值都被保留作 M.3100 定义的可能故障原因使用，以备将来扩展。

表 E.1 M.3100 中的可能故障原因

可能原因	取值	事件类型
Indeterminate	0	Unknown
Alarm Indication Signal (AIS)	1	Communications
Call Setup Failure	2	Communications
Degraded Signal	3	Communications
Far End Receiver Failure (FERF)	4	Communications
Framing Error	5	Communications
Loss Of Frame (LOF)	6	Communications
Loss Of Pointer (LOP)	7	Communications
Loss Of Signal (LOS)	8	Communications
Payload Type Mismatch	9	Communications
Transmission Error	10	Communications
Remote Alarm Interface	11	Communications
Excessive Bit Error Rate (EBER)	12	Communications
Path Trace Mismatch	13	Communications
Unavailable	14	Communications
Signal Label Mismatch	15	Communications
Loss Of Multi Frame	16	Communications
Back Plane Failure	51	Equipment
Data Set Problem	52	Equipment

表 E.1 (续)

可能原因	取值	事件类型
Equipment Identifier Duplication	53	Equipment
External IF Device Problem	54	Equipment
Line Card Problem	55	Equipment
Multiplexer Problem	56	Equipment
NE Identifier Duplication	57	Equipment
Power Problem	58	Equipment
Processor Problem	59	Equipment
Protection Path Failure	60	Equipment
Receiver Failure	61	Equipment
Replaceable Unit Missing	62	Equipment
Replaceable Unit Type Mismatch	63	Equipment
Synchronization Source Mismatch	64	Equipment
Terminal Problem	65	Equipment
Timing Problem	66	Equipment
Transmitter Failure	67	Equipment
Trunk Card Problem	68	Equipment
Replaceable Unit Problem	69	Equipment
Air Compressor Failure	101	Environmental
Air Conditioning Failure	102	Environmental
Air Dryer Failure	103	Environmental
Battery Discharging	104	Environmental
Battery Failure	105	Environmental
Commercial Power Failure	106	Environmental
Cooling Fan Failure	107	Environmental
Engine Failure	108	Environmental
Fire Detector Failure	109	Environmental
Fuse Failure	110	Environmental
Generator Failure	111	Environmental
Low Battery Threshold	112	Environmental
Pump Failure	113	Environmental
Rectifier Failure	114	Environmental
Rectifier High Voltage	115	Environmental
Rectifier Low F Voltage	116	Environmental
Ventilation System Failure	117	Environmental

表 E.1 (续)

可能原因	取值	事件类型
Enclosure Door Open	118	Environmental
Explosive Gas	119	Environmental
Fire	120	Environmental
Flood	121	Environmental
High Humidity	122	Environmental
High Temperature	123	Environmental
High Wind	124	Environmental
Ice Build Up	125	Environmental
Intrusion Detection	126	Environmental
Low Fuel	127	Environmental
Low Humidity	128	Environmental
Low Cable Pressure	129	Environmental
Low Temperature	130	Environmental
Low Water	131	Environmental
Smoke	132	Environmental
Toxic Gas	133	Environmental
Storage Capacity Problem	151	Processing error
Memory Mismatch	152	Processing error
Corrupt Data	153	Processing error
Out Of CPU Cycles	154	Processing error
Software Environment Problem	155	Processing error
Software Download Failure	156	Processing error

E.2.2 X.721/X.733 中的故障原因定义

本节以表格形式给出了定义在 X.721 和 X.733 中的可能故障原因。从 1 000 001 到 2 000 000 的所有值都被保留作为 X.721 和 X.733 中定义的可能故障原因使用，以备将来扩展。具体内容详见表 E.2。

表 E.2 X.721 和 X.733 中的可能故障原因

可能原因	取值	事件类型
Adapter Error	1 000 001	Equipment
Application Subsystem Failure	1 000 002	Processing error
Bandwidth Reduction	1 000 003	Quality of service
Call Establishment Error	—	Communications
Communication Protocol Error	1 000 005	Communications
Communication Subsystem Failure	1 000 006	Communications

表 E.2 (续)

可能原因	取值	事件类型
Configuration or Customizing Error	1 000 007	Processing error
Congestion	1 000 008	Quality of service
Corrupt Data	—	Processing error
CPU Cycles Limit Exceeded	1 000 010	Processing error
Data Set or Modem Error	1 000 011	Equipment
Degraded Signal	—	Communications
DTE-DCE Interface Error	1 000 013	Communications
Enclosure Door Open	—	Environmental
Equipment Malfunction	1 000 015	Equipment
Excessive Vibration	1 000 016	Environmental
File Error	1 000 017	Processing error
Fire Detected	—	Environmental
Flood Detected	—	Environmental
Framing Error	—	Communications
Heating or Ventilation or Cooling System Problem	1 000 021	Environmental
Humidity Unacceptable	1 000 022	Environmental
Input/Output Device Error	1 000 023	Equipment
Input Device Error	1 000 024	Equipment
LAN Error	1 000 025	Communications
Leak Detection	1 000 026	Environmental
Local Node Transmission Error	1 000 027	Communications
Loss of Frame	—	Communications
Loss of Signal	—	Communications
Material Supply Exhausted	1 000 030	Environmental
Multiplexer Problem	—	Equipment
Out of Memory	1 000 032	Processing error
Output Device Error	1 000 033	Equipment
Performance Degraded	1 000 034	Environmental
Power Problem	—	Equipment
Pressure Unacceptable	1 000 036	Environmental
Processor Problem	—	Equipment
Pump Failure	—	Environmental
Queue Size Exceeded	1 000 039	Quality of service

表 E.2 (续)

可能原因	取值	事件类型
Receive Failure	1 000 040	Equipment
Receiver Failure	—	Equipment
Remote Node Transmission Error	1 000 042	Communications
Resource at or Nearing Capacity	1 000 043	Quality of service
Response Time Excessive	1 000 044	Quality of service
Re-transmission Rate Excessive	1 000 045	Quality of service
Software Error	1 000 046	Processing error
Software Program Abnormally Terminate	1 000 047	Processing error
Software Program Error	1 000 048	Processing error
Storage Capacity Problem	—	Processing error
Temperature Unacceptable	1 000 050	Environmental
Threshold Crossed	1 000 051	Quality of service
Timing Problem	—	Equipment
Toxic Leak Detected	1 000 053	Environmental
Transmit Failure	1 000 054	Equipment
Transmitter Failure	—	Equipment
Underlying Resource Unavailable	1 000 056	Processing error
Version Mismatch	1 000 057	Processing error

E.2.3 GSM 12.11 中的故障原因定义

本节以表格形式给出了定义在 ETSI GSM 12.11 中的可能故障原因。从 3 000 001 到 4 000 000 的所有值都被保留作为 ETSI GSM 12.11 中定义的可能故障原因使用，以备将来扩展。具体内容详见表 E.3。

表 E.3 GSM12.11 中的可能故障原因

可能原因	取值	事件类型
A-bis to BTS interface failure	3 000 001	Equipment
A-bis to TRX interface failure	3 000 002	Equipment
Antenna problem	3 000 003	Equipment
Battery breakdown	3 000 004	Equipment
Battery charging fault	3 000 005	Equipment
Clock synchronization problem	3 000 006	Equipment
Combiner problem	3 000 007	Equipment
Disk problem	—	Equipment
Equipment failure	3 000 009	Equipment
Excessive receiver temperature	3 000 010	Equipment

表 E.3 (续)

可能原因	取值	事件类型
Excessive transmitter output power	3 000 011	Equipment
Excessive transmitter temperature	3 000 012	Equipment
Frequency hopping degraded	3 000 013	Equipment
Frequency hopping failure	3 000 014	Equipment
Frequency redefinition failed	3 000 015	Equipment
Line interface failure	3 000 016	Equipment
Link failure	3 000 017	Equipment
Loss of synchronization	3 000 018	Equipment
Lost redundancy	3 000 019	Equipment
Mains breakdown with battery back-up	3 000 020	Equipment
Mains breakdown without battery back-up	3 000 021	Equipment
Power supply failure	3 000 022	Equipment
Receiver antenna fault	3 000 023	Equipment
Receiver Failure	—	Equipment
Receiver multicoupler failure	3 000 025	Equipment
Reduced transmitter output power	3 000 026	Equipment
Signal quality evaluation fault	3 000 027	Equipment
Timeslot hardware failure	3 000 028	Equipment
Transceiver problem	3 000 029	Equipment
Transcoder problem	3 000 030	Equipment
Transcoder or rate adapter problem	3 000 031	Equipment
Transmitter antenna failure	3 000 032	Equipment
Transmitter antenna not adjusted	3 000 033	Equipment
Transmitter failure	—	Equipment
Transmitter low voltage or current	3 000 035	Equipment
Transmitter off frequency	3 000 036	Equipment
Database inconsistency	3 000 037	Processing error
File system call unsuccessful	3 000 038	Processing error
Input parameter out of range	3 000 039	Processing error
Invalid parameter	3 000 040	Processing error
Invalid pointer	3 000 041	Processing error
Message not expected	3 000 042	Processing error

表 E.3 (续)

可能原因	取值	事件类型
Message not initialised	3 000 043	Processing error
Message out of sequence	3 000 044	Processing error
System call unsuccessful	3 000 045	Processing error
Timeout expired	3 000 046	Processing error
Variable out of range	3 000 047	Processing error
Watch dog timer expired	3 000 048	Processing error
Cooling system failure	3 000 049	Environmental
External equipment failure	3 000 050	Environmental
External power supply failure	3 000 051	Environmental
External transmission device failure	3 000 052	Environmental
Fan failure	—	Environmental
High humidity	—	Environmental
High temperature	—	Environmental
Intrusion detected a	—	Environmental
Low humidity	—	Environmental
Low temperature	—	Environmental
Smoke detected	—	Environmental
Excessive Error Rate	3 000 060	Quality of service
Reduced alarm reporting	3 000 061	Quality of service
Reduced event reporting	3 000 062	Quality of service
Reduced logging capability	3 000 063	Quality of service
System resources overload	3 000 064	Quality of service
Broadcast channel failure	3 000 065	Communications
Connection establishment error	3 000 066	Communications
Invalid message received	3 000 067	Communications
Invalid MSU received	3 000 068	Communications
LAPD link protocol failure	3 000 069	Communications
Local alarm indication	3 000 070	Communications
Remote alarm indication	3 000 071	Communications
Routing failure	3 000 072	Communications
SS7 protocol failure	3 000 073	Communications
Transmission error	3 000 074	Communications

E.3 故障原因重复定义的解决方法

由于某些故障原因在多个建议中都有定义，为通过取值惟一标识一个故障原因，本标准中只对其分配了一个值。表 E.4 列出了重复定义的故障原因的值的分配。

表 E.4 重复定义的故障原因所使用的值

Probable cause	GSM12.11	X.721X.73	M.3100	value	Event type
Call Establishment Failure (X.721/X.733) Call Setup Failure (M.3100)		Y	Y	2	Communications
Degraded Signal		Y	Y	3	Communications
Framing Error		Y	Y	5	Communications
Loss of Frame		Y	Y	6	Communications
Loss of Signal		Y	Y	8	Communications
Equipment Failure (GSM12.11) Equipment Malfunction (X.721/X.733)	Y	Y		1 000 015	Equipment
Multiplexer Problem		Y	Y	56	Equipment
Power Problem		Y	Y	58	Equipment
Processor Problem		Y	Y	59	Equipment
Receiver Failure	X	X	X	61	Equipment
Timing Problem		X	X	66	Equipment
Transmitter Failure	X	X	X	67	Equipment
Enclosure Door Open		X	X	118	Environmental
Fan Failure (GSM 12.11) Cooling Fan Failure (M.3100)	X		X	107	Environmental
Fire Detected (X.721/X.733) Fire (M.3100)		X	X	120	Environmental
Flood Detected (X.721/X.33) Flood (M.3100)		X	X	121	Environmental
High Humidity	X		X	122	Environmental
High Temperature	X		X	123	Environmental
Intrusion Detected (GSM12.11) Intrusion Detection (M.3100)	X		X	126	Environmental
Low Humidity	X		X	128	Environmental
Low Temperature	X		X	129	Environmental
Pump Failure		X	X	113	Environmental

表 E.4 (续)

Probable cause	GSM12.11	X.721X.73	M.3100	value	Event type
Smoke Detected (GSM 12.11) Smoke (M.3100)	Y		Y	132	Environmental
Storage Capacity Problem		Y	Y	151	Processing error
Corrupt Data		X	X	153	Processing error

附录 F
(规范性附录)
测量任务定义

F.1 简介

该附录根据不同的测量任务，描述了详细的测量数据信息，并给出了相应的 IDL 定义。

F.2 详细信息

F.2.1 MSC 测量

该任务所属的管理对象类为 *mscFunction*。

该任务采集与 MSC 性能统计相关的信息，包括以下的统计：

- HLR 查询；
- MSC 中发生的切换；
- 结束码的测量；
- 服务质量；
- MSC 处理器的负荷。

F.2.1.1 MSC 性能统计

测量类型 *mscPerformance* 的测量属性见表 F.1。

表 F.1 测量类型 *mscPerformance* 的测量属性

属性名	说明	触发点	类型
因入呼路由选择的 HLR 查询尝试次数 (attInterrogationOfHLRsForRouting)	指 MSC 为入呼叫选择路由而对 HLR 进行查询的尝试次数	发送“MAP_SEND_ROUTING_INFORMATION”服务请求 (GSM 09.02)	GSMMeasurementType1
因获得 MSRN 的 HLR 查询成功次数 (succInterrogationOfHLRsMSRNObtained)	指获得 MSRN 的 HLR 查询成功次数	收到含有“MSRN”参数值的“MAP_SEND_ROUTING_INFORMATION”服务确认 (GSM 09.02)	GSMMeasurementType1
因呼叫转移的 HLR 查询成功次数 (succInterrogationOfHLRsCallForwarding)	指由于呼叫转移的 HLR 查询成功次数	收到包含“Forwarding Data”参数值的“MAP_SEND_ROUTING_INFORMATION”服务证实 (GSM 09.02)	GSMMeasurementType1
切入本 MSC 小区的 MSC 间切换尝试次数 (attIncomingInterMSCHDOs)	指在不同的 MSC 之间，所有从另一个 MSC 控制的小区切入到该 MSC 所控制的小区的切换尝试次数	收到 MAP_PREPARE_HANDOVER”服务指示，其中参数里的目的小区标识符为被观测小区 (GSM 09.02)	GSMMeasurementType1
切入本 MSC 小区的 MSC 间切换成功次数 (succIncomingInterMSCHDOs)	指在不同的 MSC 之间，所有从另一个 MSC 控制的小区切入到该 MSC 所控制的小区的切换成功次数	收到“MAP_SEND_END_SIGNAL”服务确认 (GSM 09.02)	GSMMeasurementType1

表 F.1 (续)

属性名	说明	触发点	类型
切出本 MSC 小区的 MSC 间切换尝试次数 (attOutgoingInterMSCHDOs)	指在不同的 MSC 之间, 所有从该 MSC 所控制的小区切出到另一个 MSC 控制的小区的切换尝试次数	发送 MAP_PREPARE_HANDOVER” 服务请求 (GSM 09.02)	GSMMeasurementType1
切出本 MSC 小区的 MSC 间切换成功次数 (succOutgoingInterMSCHDOs)	指在不同的 MSC 之间, 所有从该 MSC 所控制的小区切出到另一个 MSC 控制的小区的切换成功次数	发送 MAP_SEND_END_SIGNAL” 服务响应 (GSM 09.02)	GSMMeasurementType1
后继为 MSCa 的 MSC 间小区切换尝试次数 (attSubsequentInterMSCHDOsMSCa)	指后继为 MSCa 的 MSC 间小区切换的尝试次数, 其中通话又切换回源 MSC (MSCa)。如首先由 MSCa 切换到 MSCb, 然后又由于某些原因要切换回 MSCa	收到 “MAP_PREPARE_SUBSEQUENT_HANDOVER” 服务指示, 其中参数里的目的 MSC 标识为 MSCa (GSM 09.02)	GSMMeasurementType1
后继为 MSCa 的 MSC 间小区切换成功次数 (succSubsequentInterMSCHDOsMSCa)	指后继为 MSCa 的 MSC 间小区切换的成功次数, 其中通话又切换回源 MSC (MSCa)。如首先由 MSCa 切换到 MSCb, 然后又由于某些原因要切换回 MSCa	对于后继为 MSCa 的 MSC 间小区切换, 发送 “MAP_SEND_END_SIGNAL” 服务响应	GSMMeasurementType1
后继为 MSCc 的 MSC 间小区切换尝试次数 (attSubsequentInterMSCHDOsMSCc)	指后继为 MSCc 的 MSC 间小区切换的尝试次数, 其中通话切换到了 MSCc。如首先由 MSCa 切换到 MSCb, 然后又由 MSCb 切换到 MSCc	收到 “MAP_PREPARE_SUBSEQUENT_HANDOVER” 服务指示, 其中参数里的目的 MSC 标识为 MSCc (GSM 09.02)	GSMMeasurementType1
后继为 MSCc 的 MSC 间小区切换成功次数 (succSubsequentInterMSCHDOsMSCc)	指后继为 MSCc 的 MSC 间小区切换的成功次数, 其中通话切换到了 MSCc。如首先由 MSCa 切换到 MSCb, 然后又由 MSCb 切换到 MSCc	对于后继为 MSCc 的 MSC 间小区切换, 发送 “MAP_SEND_END_SIGNAL” 服务响应 (切换到 MSCc, GSM 09.02)	GSMMeasurementType1
MSC 控制的小区间切换的尝试总次数 (externalHDOs)	指所有的由该 MSC 控制的小区切换的尝试总次数	收到 “HANDOVER REQUIRED” 消息 (GSM 08.08)	GSMMeasurementType1
MSC 控制的小区间切换各原因统计的尝试总次数 (externalHDOsPerCause)	指按照每种原因来统计的 MSC 由该 MSC 控制的小区切换的尝试总次数	收到含不同原因的 “HANDOVER REQUIRED” 消息 (GSM 08.08)	GSMMeasurementType3
MSC 内切换不成功但未掉话的切换次数 (unsuccExternHDOsWithReconnectionPerMSC)	指 MSC 中所有切换不成功, 但又返回原信道 (TCH 和 SDCCH) 通话的切换次数	对所请求的切换收到 “HANDOVER FAILURE” 消息 (GSM08.08)	GSMMeasurementType1
MSC 内切换不成功而掉话的切换次数 (unsuccExternHDOsWithLossOfConnectionPerMSC)	指 MSC 中所有切换不成功而造成掉话的切换次数	对所请求的切换收到 “CLEAR REQUEST” 消息 (GSM08.08)	GSMMeasurementType1

表 F.1 (续)

属性名	说明	触发点	类型
系统寻呼总次数 (attSystemPaging)	指在测量周期中, 该 MSC 系统中所有寻呼请求的尝试次数		GSMMeasurementType1
交换设备出现故障总次数 (numOfMSCFault)	指测量周期中, 该 MSC 所发生故障的次数	无	GSMMeasurementType1
交换设备出现故障总时长 (falutTimeOfMSC)	指测量周期中, MSC 设备所发生故障的累计总时间 (单位: s)	无	GSMMeasurementType1
结束码按各种原因的出现次数 (numOfEndCode)	指在测量周期中, 根据不同的结束码原因统计的结束码出现次数。必要的结束码包括久叫不应、被叫忙、号码不全、空号、超时释放、设备拥塞、电路拥塞和寻呼无响应		GSMMeasurementType7
IMSI 的分离次数 (imsiDetachProcs)	指 IMSI 的分离次数	从 MS 收到 “IMSI DETACH INDICATION” 消息	GSMMeasurementType1
IMSI 的附着次数 (imsiAttachProcs)	指 IMSI 的附着次数	从 MS 收到 “LOCATION UPDATING REQUEST” 消息	GSMMeasurementType1

F.2.1.2 服务质量

测量类型 mscQoS 的测量属性见表 F.2。

表 F.2 测量类型 mscQoS 的测量属性

属性名	说明	触发点	类型
平均呼叫建立时长 (meanTimeToCallSetup Service)	指对 MS 请求的呼叫服务的建立时间的算术平均值 (单位: s)	累计在一个测量周期中, 从 “建立” (“SETUP”) 到相应的 “振铃” (“ALERTING”) 消息之间的时间间隔。将这段时间除以所观察到的呼叫建立次数, 即得该算术平均值 (GSM 04.08, GSM 08.08)	GSMMeasurementType2
平均位置更新时长 (meanTimeToLocation UpdateService)	指对于 MS 改变地理位置所需的位置更新服务时间的算术平均值 (单位: s)	累计在一个测量周期中, 从 “位置更新请求” (“LOCATION UPDATE REQUEST”) 到相应的 “位置更新接受” (“LOCATION UPDATING ACCEPT”) 消息之间的时间间隔。将这段时间除以位置更新请求次数, 即得该算术平均值 (GSM 04.08, GSM 08.08)	GSMMeasurementType2
平均通话时长 (meanCallDuration)	指平均通话时长 (单位: s)	累计在一个测量周期内, 从 “连接” (“CONNECT”) 到相应的呼叫清除之间的时间间隔, 将这段时间除以所观察到的呼叫应答次数, 即得该算术平均值	GSMMeasurementType2
平均中继占用时长 (meanTrunkSeizure Duration)	指中继线的平均占用时长 (单位: s)	累计在一个测量周期内, 从 “地址完成消息” (“Address Complete Message”) 到相应的呼叫清除之间的时间间隔, 将这段时间除以所观察到的呼叫建立次数, 即得该算术平均值	GSMMeasurementType2

F.2.1.3 处理器负荷

测量类型 `mscProcessorLoad` 的测量属性见表 F.3。

表 F.3 测量类型 `mscProcessorLoad` 的测量属性

属性名	说明	触发点	类型
平均占用率 (<code>meanUsage</code>)	指测量周期中, 主处理器的平均使用率	无	<code>GSMMeasurementType2</code>
峰值占用率 (<code>peakUsage</code>)	指测量周期中, 主处理器的峰值使用率	无	<code>GSMMeasurementType2</code>
峰值出现时刻 (<code>peakTime</code>)	指测量周期中, 主处理器峰值出现的最后时刻	无	<code>PeakTimeType</code>
峰值时长 (<code>peakDuration</code>)	主处理器处于峰值期间的持续时间 (单位为 s, 且缺省的 CPU 峰值使用率的门限值为 80%)	无	<code>GSMMeasurementType1</code>

F.2.2 智能业务性能测量

该任务所属的管理对象类为 `ssFunction`。

该任务采集与智能业务性能统计相关的信息, 包括智能业务的话务量统计等, 详见表 F.4。

表 F.4 测量类型 `ssfTraffic` 的测量属性

属性名	说明	触发点	类型
入 SSF 的智能呼叫次数 (<code>attIncomingCamelCallsToSSF</code>)	指在一个测量周期中, SSF 接收到的智能呼叫次数	本地 MSC 从 BSC/GMSC/其他 MSC/ISDN/PSTN 收到 "SETUP/IAM/IAI" 消息, 且号码分析的结果表明主叫方或被叫方为 CAMEL 用户	<code>GSMMeasurementType1</code>
出 SSF 的智能呼叫次数 (<code>attOutgoingCamelCallsFromSSF</code>)	指在一个测量周期中, SSF 提交给 SCF 的智能呼叫次数	从本 SSF 发送 "InitialDP" 消息给 SCF (DP2+DP12), 其中主叫或被叫为 CAMEL 用户	<code>GSMMeasurementType1</code>
移动智能业务用户呼叫移动智能业务用户次数 (<code>attCamelToCamelCalls</code>)	指在一个测量周期中, SSF 接收到的移动智能业务用户呼叫移动智能业务用户的试呼次数	本地 MSC 从 BSC/GMSC/其他 MSC/PSTN 收到 "SETUP/IAM/IAI", 其中主叫和被叫均为 CAMEL 用户	<code>GSMMeasurementType1</code>
移动智能业务用户呼叫移动智能业务用户应答次数 (<code>answeredCamelToCamelCalls</code>)	指在一个测量周期中, SSF 接收到的移动智能业务用户呼叫移动智能业务用户的呼叫中应答的次数	从 BSC/GMSC/其他 MSC/PSTN 收到 "ANC/ANM/CON" 消息, 其中主叫和被叫均为 CAMEL 用户	<code>GSMMeasurementType1</code>
由于系统原因而不成功的智能呼叫次数 (<code>unsuccCamelCallsForSystemReason</code>)	指在一个测量周期中, 在提交给 SCF 的智能呼叫中, 由于系统的原因, 例如智能业务逻辑操作超时、被 SCF 丢弃, 而引起智能呼叫处理失败的智能呼叫次数	在 CAMEL 呼叫的建立阶段, 业务逻辑操作超时, 或从 SSF 收到由于系统原因的 "ReleaseCall" 消息	<code>GSMMeasurementType1</code>

表 F.4 (续)

属性名	说明	触发点	类型
由于用户原因而不成功的智能呼叫次数 (unsuccCamelCallsForSubscriberReason)	指在一个测量周期中, 在提交给 SCF 的智能呼叫中, 由于用户的原因, 例如用户输入非法、用户输入超时、用户输入错误、用户资费不够等, 而引起智能呼叫处理失败的智能呼叫次数	在 CAMEL 呼叫建立阶段, 从 SCF 收到“ReleaseCall”消息, 其中的用户原因为不成功呼叫	GSMMeasurementType
主叫话务量 (camelOriginatingTrafficVolume)	指在一个测量周期中, 由于移动智能业务用户作主叫而引起的应答话务量	累计在一个测量周期中, 从“ANC/ANM/CON”到相应的呼叫清除之间的时间间隔, 其中主叫为 CAMEL 用户, 单位: Erl/s	GSMMeasurementType2
被叫话务量 (camelTerminatingTrafficVolume)	指在一个测量周期中, 由于移动智能业务用户作被叫而引起的应答话务量	累计在一个测量周期中, 从“ANC/ANM/CON”到相应的呼叫清除的时间间隔, 其中被叫为 CAMEL 用户, 单位: Erl/s	GSMMeasurementType2
主叫通话平均时长 (meanDurationOfCamelOriginatingCalls)	指在一个测量周期中, 主叫为移动智能业务用户的通话的平均时长 (单位: s)	累计在一个测量周期中, 从“ANC/ANM/CON”到相应的呼叫清除的时间间隔, 其中主叫为 CAMEL 用户。再将这段时间的最终结果除以应答的 CAMEL 始发呼叫的次数, 得到呼叫平均时长	GSMMeasurementType2
被叫通话平均时长 (meanDurationOfCamelTerminatingCalls)	指在一个测量周期中, 被叫为移动智能业务用户的通话的平均时长 (单位: s)	累计在一个测量周期中, 从“ANC/ANM/CON”到相应的呼叫清除的时间间隔, 其中被叫为 CAMEL 用户。再将这段时间的最终结果除以应答的 CAMEL 终结呼叫的次数, 得到呼叫平均时长	GSMMeasurementType2
主叫试呼次数 (attCamelOriginatingCalls)	指在一个测量周期中, 主叫为移动智能业务用户的试呼次数	本地 MSC 从 BSC/GMSC/其他 MSC/PSTN (MO) 收到“SETUP/IAM/IAI”消息, 其中主叫为 CAMEL 用户	GSMMeasurementType1
主叫占用次数 (seizuredCamelOriginatingCalls)	指在一个测量周期中, 主叫为移动智能业务用户的占用次数	本地 MSC 向 BSC (局内呼叫)/GMSC/其他 MSC/PSTN (MT) 发送“SETUP/IAM/IAI”消息, 其中主叫为 CAMEL 用户	GSMMeasurementType1
主叫接通次数 (connectedCamelOriginatingCalls)	指在一个测量周期中, 主叫为移动智能业务用户的接通次数	本地 MSC 从 BSC (局内呼叫)/GMSC/其他 MSC/PSTN (MT) 收到“Alerting/ACM”消息, 其中主叫为 CAMEL 用户	GSMMeasurementType1
主叫应答次数 (answeredCamelOriginatingCalls)	指在一个测量周期中, 主叫为移动智能业务用户的应答次数	收到“ANC/ANM/CON”消息, 其中主叫为 CAMEL 用户	GSMMeasurementType1
被叫试呼次数 (attCamelTerminatingCalls)	指在一个测量周期中, 被叫为移动智能业务用户的试呼次数	本地 MSC 从 BSC/GMSC/other MSC/PSTN (MO) 收到“SETUP/IAM/IAI”消息, 其中被叫为 CAMEL 用户	GSMMeasurementType1

表 F.4 (续)

属性名	说明	触发点	类型
被叫占用次数 (seizedCamelTerminatingCalls)	指在一个测量周期中, 被叫为移动智能业务用户的占用次数	本地 MSC 向 BSC (局内呼叫) / GMSC/其他 MSC/PSTN (MT) 发送“SETUP/IAL/IAM”消息, 其中被叫为 CAMEL 用户	GSMMeasurementType1
被叫接通次数 (connectedCamelTerminatingCalls)	指在一个测量周期中, 被叫为移动智能业务用户的接通次	本地 MSC 从 BSC (局内呼叫) /GMSC/其他 MSC/PSTN (MT) 收到“Alerting/ACM”消息, 其中被叫为 CAMEL 用户	GSMMeasurementType1
被叫应答次数 (answeredCamelTerminatingCalls)	指在一个测量周期中, 被叫为移动智能业务用户的应答次数	收到“ANC/ANM/CON”消息, 其中被叫为 CAMEL 用户	GSMMeasurementType1
智能呼叫的占用次数 (seizedCamelCalls)	指在一个测量周期中, 移动智能呼叫的占用次数 (当主被叫均为移动智能业务用户时按一次占用次数计算)	本地 MSC 向 BSC/GMSC/其他 MSC/PSTN 发送“SETUP/IAL/IAM”, 其中主叫或被叫为 CAMEL 用户 (seizedCamelCalls = seizedCamelOriginatingCalls + seizedCamelTerminatingCalls - 主叫和被叫均为 CAMEL 用户的呼叫占用次数)	GSMMeasurementType1
智能呼叫的接通次数 (connectedCamelCalls)	指在一个测量周期中, 移动智能呼叫的接通次数 (当主被叫均为移动智能业务用户时按一次接通次数计算)	本地 MSC 从 BSC/GMSC/其他 MSC/PSTN 收到“Alerting/ACM”消息, 其中主叫或被叫为 CAMEL 用户 (connectedCamelCalls=connectedCamelOriginatingCalls + connectedCamelTerminatingCalls - 主叫和被叫均为 CAMEL 用户的呼叫接通次数)	GSMMeasurementType1
智能呼叫的应答次数 (answeredCamelCalls)	指在一个测量周期中, 移动智能呼叫的应答次数 (当主被叫均为移动智能业务用户时按一次应答次数计算)	收到“ANC/ANM/CON”消息, 其中主叫或被叫为 CAMEL 用户 (answeredCamelCalls = answeredCamelOriginatingCalls + answeredCamelTerminatingCalls - 主叫和被叫均为 CAMEL 用户的呼叫应答次数)	GSMMeasurementType1
智能呼叫的应答话务量 (answeredCamelCallsTraffic Volume)	指在一个测量周期中, 移动智能呼叫的应答话务量	累计在一个测量周期中, 从“ANC/ANM/CON”到相应的呼叫清除的时间间隔, 其中主叫或被叫为 CAMEL 用户, 单位: Erl/s	GSMMeasurementType2
智能呼叫的接通话务量 (connectedCamelCallsTraffic Volume)	指在一个测量周期中, 移动智能呼叫的接通话务量	累计在一个测量周期中, 从“Alerting/ACM”到相应的呼叫清除的时间间隔, 其中主叫或被叫为 CAMEL 用户, 单位: Erl/s	GSMMeasurementType2

表 F.4 (续)

属性名	说明	触发点	类型
智能呼叫的占用话务量 (seizedCamelCallsTraffic Volume)	指在一个测量周期中, 移动智能呼叫的占用话务量	累计在一个测量周期中, 从“SETUP/IAI/ IAM”到相应的呼叫清除的时间间隔, 其中主叫或被叫为 CAMEL 用户, 单位: Erl/s	GSMMeasurementType2
智能呼叫的用户早释次数 (callingPartyEarlyReleaseCamel Calls)	指在一个测量周期中, 移动智能呼叫的主叫用户早释次数	在呼叫建立阶段, 在被叫振铃之前就从主叫方收到“Clear Request/Clear Forward/REL”消息, 其中主叫或被叫为 CAMEL 用户。当主叫和被叫均为 CAMEL 用户时, 该计数器只能统计一次	GSMMeasurementType1
智能呼叫的振铃早释次数 (callingPartyRingRelease CamelCalls)	指在一个测量周期中, 移动智能呼叫的被叫振铃早释次数	在呼叫建立阶段, 在被叫振铃之后而且是应答之前, 就从主叫方收到“Clear Request/Clear Forward/REL”消息, 其中主叫或被叫为 CAMEL 用户。当主叫和被叫均为 CAMEL 用户时, 该计数器只能统计一次	GSMMeasurementType1
智能呼叫的被叫忙次数 (calledPartyBusyCamelCalls)	指在一个测量周期中, 智能呼叫的被叫忙次数	DP5 + DP13 - (DP5 & DP13) (注: DP5 & DP13 表示在一次呼叫中 DP5 和 DP13 同时计数的情况, 即被叫忙, 且主叫和被叫均为 CAMEL 用户。)	GSMMeasurementType1
智能呼叫的久叫不应次数 (calledPartyNoAnswerCamelCalls)	指在一个测量周期中, 移动智能呼叫的被叫久叫不应次数	DP6 + DP14 - (DP6 & DP14) (注: DP6 & DP14 表示在一次计数中 DP6 和 DP14 同时计数的情况, 即, 被叫久叫不应, 且主叫和被叫均为 CAMEL 用户。)	GSMMeasurementType1
DP2 检测点的统计次数 (dp2)	指“收集_信息” (Collected_Info) 事件的出现次数		GSMMeasurementType1
DP4 检测点的统计次数 (dp4)	指“路由_选择_故障” (Route_select_Failure) 事件的出现次数		GSMMeasurementType1
DP5 检测点的统计次数 (dp5)	指“发端_遇忙”(O_Busy) 事件的出现次数		GSMMeasurementType1
DP6 检测点的统计次数 (dp6)	指“发端_无应答”(O_No_ Answer) 事件的出现次数		GSMMeasurementType1
DP7 检测点的统计次数 (dp7)	指“发端_应答” (O_Answer) 事件的出现次数		GSMMeasurementType1
DP9 检测点的统计次数 (dp9)	指“发端_拆线” (O_Disconnect) 事件的出现次数		GSMMeasurementType1

表 F.4 (续)

属性名	说明	触发点	类型
DP10 检测点的统计次数 (dp10)	指“发端_放弃”(O_Abandon)事件的出现次数		GSMMeasurementType1
DP12 检测点的统计次数 (dp12)	指“终端_试呼”_鉴权(Terminating_Attemptin_Authorised)事件的出现次数		GSMMeasurementType1
DP13 检测点的统计次数 (dp13)	指“终端_遇忙”(T_Busy)事件的出现次数		GSMMeasurementType1
DP14 检测点的统计次数 (dp14)	指“终端_无应答”(T_No_Answer)事件的出现次数		GSMMeasurementType1
DP15 检测点的统计次数 (dp15)	指“终端_应答”(T_Answer)事件的出现次数		GSMMeasurementType1
DP17 检测点的统计次数 (dp17)	指“终端_拆线”(T_Disconnect)事件的出现次数		GSMMeasurementType1
DP18 检测点的统计次数 (dp18)	指“终端_放弃”(T_Abandon)事件的出现次数		GSMMeasurementType1

F.2.3 交换机业务量统计

该任务所属的管理对象类为 *mscFunction*。

该测量任务以收集某个交换机的全部业务量性能数据。根据基于 ITU-T E.502 建议中的图 4/E.502, 可得本标准的交换机话务量测量所采用移动话务流量, 如图 F.1 所示。其测量属性见表 F.5。

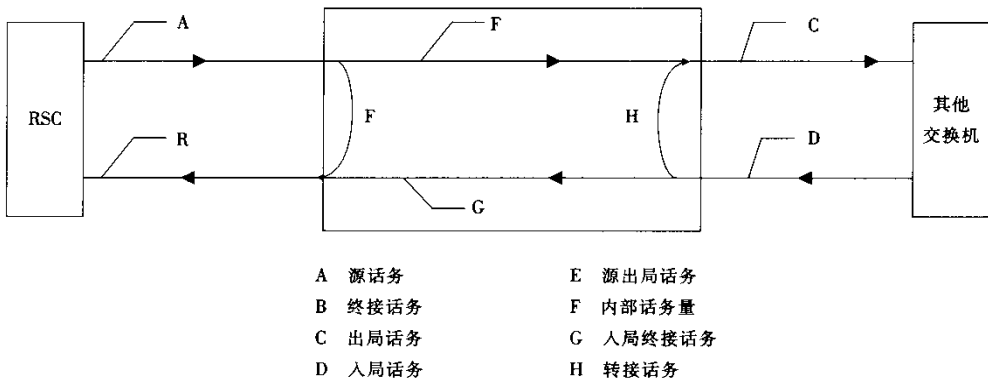


图 F.1 移动话务量流量

从图 F.1 中可以看出, 各方向话务流量应有下面的关系:

$$\begin{aligned}
 A &= E + F + Z1 & B &= F + G - d1 \\
 D &= G + H + Z2 & C &= E + H - d2
 \end{aligned}$$

其中, Z1 和 Z2 统计未完成的呼叫或含有非法拨号信息的呼叫所产生的话务量, d1 和 d2 统计由于以下原因导致在交换机中呼叫失败而产生的话务量:

- 所有的电路均处于忙状态或不可用；
- 交换机内部阻塞；
- 拨号不完整；
- 非法的目的码；
- 服务被禁用或/和阻塞。

在该测量任务中，将监视 A~H 类型的话务量、交换机内部阻塞以及由于所有电路均忙而产生的阻塞。

表 F.5 测量类型 trafficFlow 的测量属性

属性名		描述	触发点	类型
始 发 呼 叫	始发试呼次数 (attMobileOriginatingCalls)	指交换机中始发呼叫的试呼次数	从 BSC 收到“SETUP”消息	GSMMeasurementType1
	始发呼通次数 (succMobileOriginatingCalls)	指交换机中成功的始发呼叫次数	向 BSC (MO) 发送“ALERTING”消息	GSMMeasurementType1
	始发应答次数 (ansMobileOriginatingCalls)	指有应答的交换机始发呼叫次数	向 BSC (MO) 发送“CONNECT”消息	GSMMeasurementType1
	始发呼叫各失败原因统计次数 (failReasonForMobileOriginatingCalls)	指由各种原因分别产生的始发呼叫失败次数 ^{注*}		GSMMeasurementType5
	始发试呼话务量 (seizureTrafficVolumeForMobileOriginatingCalls)	指交换机始发试呼所产生的话务量	累计在一个测量周期中，统计从收到 MO 的“SETUP”消息到呼叫清除的时间间隔，单位：Erl/s (GSM 04.08, GSM 08.08)	GSMMeasurementType2
	始发呼通话务量 (succTrafficVolumeForMobileOriginatingCalls)	指交换机始发呼叫成功所产生的话务量	累计在一测量周期中，统计从向 MO 发送“ALERTING”消息到呼叫释放的时间间隔，单位：Erl/s (GSM 04.08, GSM 08.08)	GSMMeasurementType2
	始发应答话务量 ansTrafficVolumeForMobileOriginatingCalls	指由应答交换机始发呼叫所产生的话务量	累计在一个测量周期中，统计从向 MO 发送“CONNECT”到呼叫释放的时间间隔，单位：Erl/s (GSM 04.08, GSM 08.08)	GSMMeasurementType2
内 部 呼 叫	本 MSC 试呼次数 (attLocalCalls)	指交换机内部呼叫的试呼次数	从 BSC 收到“SETUP”消息后，在向 HLR 取路由信息后判定该呼叫为本 MSC 局内呼叫	GSMMeasurementType1
	本 MSC 呼通次数 (succLocalCalls)	指交换机内部呼叫的成功次数	当呼叫为本 MSC 局内呼叫时，向 BSC (MO) 发送“ALTERING”消息	GSMMeasurementType1
	本 MSC 应答次数 (ansLocalCalls)	指交换机内部呼叫的中有应答的次数	当呼叫为本 MSC 局内呼叫时，向 BSC (MO) 发送“CONNECT”消息	GSMMeasurementType1
	本 MSC 呼叫各失败原因统计次数 (failReasonForLocalCalls)	指由各种原因分别产生的内部呼叫失败次数 ^{注*}		GSMMeasurementType5

表 F.5 (续)

属性名		描述	触发点	类型
内 部 呼 叫	本 MSC 试呼话务量 (seizureTrafficVolumeForLocalCalls)	指交换机内部试呼所产生的话务量	累计在一个测量周期中, 对于本 MSC 的局内呼叫, 从收到 MO 的“SETUP”消息到呼叫释放的时间间隔, 单位: Erl/s (GSM 04.08, GSM 08.08)	GSMMeasurementType2
	本 MSC 呼通话务量 (succTrafficVolumeForLocalCalls)	指交换机内部呼叫成功所产生的话务量	累计在一个测量周期中, 对于本 MSC 的局内呼叫, 从向 MO 发送“ALERTING”消息到呼叫释放的时间间隔, 单位: Erl/s (GSM 04.08, GSM 08.08)	GSMMeasurementType2
	本 MSC 应答应话务量 (ansTrafficVolumeForLocalCalls)	指由应答交换机内部呼叫所产生的话务量	累计在一个测量周期中, 对于本 MSC 的局内呼叫, 从向 MO 发送“CONNECT”到呼叫释放的时间间隔, 单位: Erl/s (GSM 04.08, GSM 08.08)	GSMMeasurementType2
终 接 呼 叫	终接试呼次数 (attMobileTerminatingCalls)	指交换机中终接呼叫的试呼次数	向 BSC (MT) 发送“SETUP”消息	GSMMeasurementType1
	终接呼通次数 (succMobileTerminatingCalls)	指交换机中终接呼叫的成功次数	从 BSC (MT) 收到“ALERTING”消息	GSMMeasurementType1
	终接应答次数 (ansMobileTerminatingCalls)	指交换机中终接呼叫的应答次数	从 BSC (MT) 收到“CONNECT”消息	GSMMeasurementType1
	终接呼叫各失败原因统计次数 (failReasonForMobileTerminatingCalls)	指由各种原因分别产生的终接呼叫失败次数*		GSMMeasurementType5
	终接试呼话务量 (seizureTrafficVolumeForMobileTerminatingCalls)	指交换机终接试呼所产生的话务量	累计在一个测量周期中, 从“向 MT 发送“SETUP”到呼叫释放的时间间隔, 单位: Erl/s (GSM 04.08, GSM 08.08)	GSMMeasurementType2
	终接呼通话务量 (succTrafficVolumeForMobileTerminatingCalls)	指交换机终接呼叫成功所产生的话务量	累计在一个测量周期中, 从 MT 接收“ALERTING”消息到呼叫释放的时间间隔, 单位: Erl/s (GSM 04.08, GSM 08.08)	GSMMeasurementType2
	终接应答应话务量 (ansTrafficVolumeForMobileTerminatingCalls)	指由应答交换机终接呼叫所产生的话务量	累计在一个测量周期中, 从 MT 收到“CONNECT”消息到呼叫释放的时间间隔, 单位: Erl/s (GSM 04.08, GSM 08.08)	GSMMeasurementType2

表 F.5 (续)

属性名		描述	触发点	类型
入 局 呼 叫	入局试呼次数 (attIncomingCalls)	指交换机的入局试呼次数	从始发局(或转接局)收到 IAI 或 IAM 消息	GSMMeasurementType1
	入局呼通次数 (succIncomingCalls)	指交换机的入局呼叫成功次数	向始发局发送 ACM 消息	GSMMeasurementType1
	入局应答次数 (ansIncomingCalls)	指有应答的交换机入局呼叫次数	向始发局发送应答信号 (ANM/ANC/ANN)	GSMMeasurementType1
	入局呼叫各失败原因统计次数 (failReasonForIncomingCalls)	指由各种原因分别产生的入局呼叫失败次数 ^{注*}		GSMMeasurementType5
	入局试呼话务量 (seizureTrafficVolumeForIncomingCalls)	指交换机入局试呼所产生的话务量	累计在一个测量周期中, 从收到 IAI/IAM 消息到呼叫释放的时间间隔, 单位: Erl/s	GSMMeasurementType2
	入局呼通话务量 (succTrafficVolumeForIncomingCalls)	指交换机入局呼叫成功所产生的话务量	累计在一个测量周期中, 从发送 ACM 消息到呼叫释放的时间间隔, 单位: Erl/s	GSMMeasurementType2
	入局应答应话务量 (ansTrafficVolumeForIncomingCalls)	指由应答交换机入局呼叫所产生的话务量	累计在一个测量周期中, 从发送应答信号到呼叫释放的时间间隔, 单位: Erl/s	GSMMeasurementType2
出 局 呼 叫	出局试呼次数 (attOutgoingCalls)	指交换机的出局试呼次数	发送 IAI 或 IAM 消息	GSMMeasurementType1
	出局呼通次数 (succOutgoingCalls)	指交换机的出局呼叫成功次数	收到 ACM 消息	GSMMeasurementType1
	出局应答次数 (ansOutgoingCalls)	指有应答的交换机出局呼叫次数	收到应答信号 (ANM/ANC/ANN)	GSMMeasurementType1
	出局呼叫各失败原因统计次数 (failReasonForOutgoingCalls)	指由各种原因分别产生的出局呼叫失败次数 ^{注*}		GSMMeasurementType5
	出局试呼话务量 (seizureTrafficVolumeForOutgoingCalls)	指交换机出局试呼所产生的话务量	累计在一个测量周期中, 从发送 IAI/IAM 到呼叫释放的时间间隔, 单位: Erl/s	GSMMeasurementType2
	出局呼通话务量 (succTrafficVolumeForOutgoingCalls)	指交换机出局呼叫成功所产生的话务量	累计在一个测量周期中, 从收到 ACM 到呼叫释放的时间间隔, 单位: Erl/s	GSMMeasurementType2
	出局应答应话务量 (ansTrafficVolumeForOutgoingCalls)	指由应答交换机出局呼叫所产生的话务量	累计在一个测量周期中, 从收到应答信号到呼叫释放的时间间隔, 单位: Erl/s	GSMMeasurementType2
转 接 呼 叫	转接试呼次数 (attTransitCalls)	指交换机中的转接呼叫的试呼次数	在收到 IAI/IAM 入局试呼消息中, 经号码分析后为转接呼叫	GSMMeasurementType1
	转接呼通次数 (succTransitCalls)	指交换机中的转接呼叫的成功次数 (成功占用)	MSC 收到落地局的 ACM 消息后, 向始发局转发 ACM 消息	GSMMeasurementType1

表 F.5 (续)

属性名		描述	触发点	类型
转 接 呼 叫	转接应答次数 (ansTransitCalls)	指有应答的交换机中转移呼叫的次数	MSC 收到落地局的应答信号后, 向始发局转发应答消息	GSMMeasurementType1
	转接呼叫各失败原因统计次数 (failReasonForTransitCalls)	指由各种原因分别产生的转接呼叫失败次数 ^{注*}		GSMMeasurementType5
	转接试呼话务量 (seizureTrafficVolumeForTransitCalls)	指交换机转接试呼所产生的话务量		GSMMeasurementType2
	转接呼通话务量 (succTrafficVolumeForTransitCalls)	指交换机转接呼叫成功所产生的话务量		GSMMeasurementType2
	转接应答话务量 (ansTrafficVolumeForTransitCalls)	指由应答交换机转接呼叫所产生的话务量		GSMMeasurementType2
始 发 出 局 呼 叫	始发出局试呼次数 (attOriginatingOutgoingCalls)	指交换机中的始发出局呼叫的试呼次数	从 MO 收到 SETUP 消息后, 在向 HLR 取路由信息后判定该呼叫为出局呼叫	GSMMeasurementType1
	始发出局呼通次数 (succOriginatingOutgoingCalls)	指交换机中的始发出局呼叫的成功次数 (成功占用)	当呼叫为始发出局呼叫时, 向 BSC (MO) 发送 "ALTERING" 消息	GSMMeasurementType1
	始发出局应答次数 (ansOriginatingOutgoingCalls)	指有应答的交换机中始发出局呼叫的次数	当呼叫为始发出局呼叫时, 向 BSC (MO) 发送 "CONNECT" 消息	GSMMeasurementType1
	始发出局各失败原因统计次数 (failReasonForOriginatingOutgoingCalls)	指由各种原因分别产生的始发出局呼叫失败次数 ^{注*}		GSMMeasurementType5
	始发出局试呼话务量 (seizureTrafficVolumeForOriginatingOutgoingCalls)	指交换机始发出局试呼所产生的话务量	累计在一个测量周期中, 从 MO 收到 SETUP 消息到呼叫清除的时间间隔, 单位: Erl/s	GSMMeasurementType2
	始发出局呼通话务量 (succTrafficVolumeForOriginatingOutgoingCalls)	指交换机始发出局呼叫成功所产生的话务量	累计在一个测量周期中, 从向 MO 发送 ALTERING 消息到呼叫清除的时间间隔, 单位: Erl/s	GSMMeasurementType2
	始发出局应答话务量 (ansTrafficVolumeForOriginatingOutgoingCalls)	指由应答交换机始发出局呼叫所产生的话务量	累计在一个测量周期中, 从向 MO 发送 CONNECT 到呼叫清除的时间间隔, 单位: Erl/s	GSMMeasurementType2
终 接 入 局 呼 叫	终接入局试呼次数 (attIncomingTerminatingCalls)	指交换机中的终接入局呼叫的试呼次数	MSC 收到 IAL/IAM 消息, 做被叫号码分析后, 确定为终接呼叫	GSMMeasurementType1
	终接入局呼通次数 (succIncomingTerminatingCalls)	指交换机中的终接入局呼叫的成功次数 (成功占用)	(当呼叫为终结入局呼叫时) 向始发局发送 ACM 消息	GSMMeasurementType1
	终接入局应答次数 (ansIncomingTerminatingCalls)	指有应答的交换机中终接入局呼叫的次数	(当呼叫为终结入局呼叫时) 向始发局发送应答信号 (ANM/ANC/ANN)	GSMMeasurementType1
	终接入局各失败原因统计次数 (failReasonForIncomingTerminatingCalls)	指由各种原因分别产生的终接入局呼叫失败次数 ^{注*}		GSMMeasurementType5

表 F.5 (续)

属性名		描述	触发点	类型
终 接 入 局 呼 叫	终接入局试呼话务量 (seizureTrafficVolumeForIncomingTerminatingCalls)	指交换机终接入局试呼所产生的话务量	累计在一个测量周期中, 从接收到 IAL/IAM 消息到呼叫释放的时间间隔, 单位: Erl/s	GSMMeasurementType2
	终接入局呼通话务量 (succTrafficVolumeForIncomingTerminatingCalls)	指交换机终接入局呼成功所产生的话务量	累计在一个测量周期中, 从发送 ACM 消息到呼叫释放的时间间隔, 单位: Erl/s	GSMMeasurementType2
	终接入局应答话务量 (ansTrafficVolumeForIncomingTerminatingCalls)	指由应答交换机终接入局呼叫所产生的话务量	累计在一个测量周期中, 从发送应答信号到呼叫释放的时间间隔, 单位: Erl/s	GSMMeasurementType2
	内部保护阻塞次数 (callsBlockedByLoadShedding)	统计由于交换机内部保护机制而阻塞的试呼次数		GSMMeasurementType1
	内部阻塞话务量 (internalCongestionTrafficVolume)	由于交换机内部拥塞而被阻塞呼叫所产生的话务量		GSMMeasurementType2
	中继阻塞话务量 (blockedTrafficVolumeBecauseOfTrunkBusy)	由出局电路忙而阻塞的呼叫所产生的话务量		GSMMeasurementType2
注 a: 必须计算由以下原因引起的失败呼叫 (被叫应答前): 一无中继线所引起的呼叫阻塞; 一交换机内部阻塞引起的呼叫阻塞; 一异常释放。				

F.2.4 被观测小区切换的测量

该任务所属的管理对象类为 *mscFunction*。

该任务对每一个被观测的小区, 收集与另一个相邻的源小区或目的小区的切换信息。注意: 在测量工作初始化时, 需给出被观测的小区 and 相邻小区 (源小区或目的小区) 的标识。其测量属性见表 F.6。

表 F.6 测量类型 handoverObservation 的测量属性

属性名	说明	触发点	类型
MSC 内从指定相邻小区切换到被观测小区的尝试次数 (attIncomingExternalIntraMSCHDOsPerOriginatingCell)	指在同一 MSC 范围内, 由该 MSC 控制的某个指定相邻小区切换到被观测小区的尝试次数	向 BSC 发送切换请求 "HANDOVER REQUEST" 消息 (GSM 08.08)	GSMMeasurementType1
MSC 内从指定相邻小区切换到被观测小区的成功次数 (succIncomingExternalIntraMSCHDOsPerOriginatingCell)	指在同一 MSC 范围内, 由该 MSC 控制的某个指定相邻小区切换到被观测小区的成功次数	从 BSC 收到切换完成 "HANDOVER COMPLETE" 消息 (GSM 08.08)	GSMMeasurementType1
MSC 内从被观测小区切换到指定相邻小区的尝试次数 (attOutgoingExternalIntraMSCHDOsPerTargetCell)	指在同一 MSC 范围内, 从被观测小区切换到由该 MSC 控制的某个指定相邻小区的尝试次数	从 BSC 收到 "HANDOVER REQUIRED" 消息 (GSM 08.08)	GSMMeasurementType1

表 F.6 (续)

属性名	说明	触发点	类型
MSC 内从被观测小区切换到指定相邻小区的成功次数 (succOutgoingExternalIntraMSC HDOsPerTargetCell)	指在同一 MSC 范围内, 从被观测小区切换到由该 MSC 控制的某个指定相邻小区的成功次数, 详细的信息见 GSM 04.08、08.08 和 08.58	完成对所涉及信道的释放过程	GSMMeasurementType1
MSC 间从指定相邻小区切换到被观测小区的尝试次数 (attIncomingInterMSCHDOsPer OriginatingCell)	指在不同的 MSC 之间, 由另一个 MSC 控制的某个指定相邻小区切换到被观测小区的尝试次数	收到“MAP_PREPARE_HANDOVER”服务指示, 其中的目的小区标识符为被观测的小区 (GSM 09.02)	GSMMeasurementType1
MSC 间从指定相邻小区切换到被观测小区的尝试次数 (succIncomingInterMSCHDOsPer OriginatingCell)	指在不同的 MSC 之间, 由另一个 MSC 控制的某个指定相邻小区切换到被观测小区的成功次数	收到“MAP_SEND_END_SIGNAL”服务确认 (GSM 09.02)	GSMMeasurementType1
MSC 间从指定相邻小区切换到被观测小区的尝试次数 (attOutgoingInterMSCHDOsPer TargetCell)	指在不同的 MSC 之间, 从被观测小区切换到由另一个 MSC 控制的某个指定相邻小区的尝试次数	发送“MAP_PREPARE_HANDOVER”)服务请求 (GSM 09.02)	GSMMeasurementType1
MSC 间从指定相邻小区切换到被观测小区的尝试次数 (succOutgoingInterMSCHDOsPer TargetCell)	指在不同的 MSC 之间, 从被观测小区切换到由另一个 MSC 控制的某个指定相邻小区的成功次数	发送“MAP_SEND_END_SIGNAL”)服务响应 (GSM 09.02)	GSMMeasurementType1

F.2.5 目的地测量

该任务所属的管理对象类为 *observedDestination*。

该任务收集与所观测的目的地相关的话务量测量数据。其测量属性见表 F.7。

表 F.7 测量类型 *observedDestination* 的测量属性

属性名	描述	触发点	类型
去话试呼次数 (bids)	指对所观察目的码的试呼次数, 受网络管理控制影响的呼叫不包括在内	收到 BSC 发来的“SETUP”消息	GSMMeasurementType1
去话占用次数 (outgoingSeizures)	指对所观测目的码的成功占用次数	向 BSC 发送“ALERTING”消息	GSMMeasurementType1
去话应答次数 (answeredOutgoingSeizures)	指在从本局到所观测目的码的占用中, 收到应答信号的次数	向 BSC 发送“CONNECT”消息	GSMMeasurementType1
无可用电路次数 (noCircuitsAvailable)	指由于无空闲电路而导致被观察目的码不可达的不成功次数。比如, 到目的地的最后一个电路子群溢出。一般来说, noCircuitsAvailable 的值小于等于 bids 减去 outgoingSeizures 的值	无	GSMMeasurementType1

表 F.7 (续)

属性名	描述	触发点	类型
占用话务量 (seizureTrafficVolume)	指对被观测地点的占用业务量	累计在一个测量周期段内, 从发送“ALERTING”到相应的呼叫清除之间的时间间隔, 单位: Erl/s (GSM 04.08, GSM 08.08)	GSMMeasurementType2
应答话务量 (answerTrafficVolume)	指对被观测地点的应答业务量	累计在一个测量周期段内, 从发送“CONNECT”到相应的呼叫释放之间的时间间隔, 单位: Erl/s (GSM 04.08, GSM 08.08)	GSMMeasurementType2

F.2.6 电路端点子群的测量

该任务所属的管理对象类为 *circuitEndPointSubGroup*。

该任务收集一个被观察的电路端点子群的话务量性能数据。它实现了在 ITU-T E.502 建议中定义的测量类型 21 的要求, 但不包括实体 *g* 和 *h*。电路子群的性能监视属性基于电路子群的方向特性, 包括单向入、单向出或双向。其测量属性见表 F.8。

表 F.8 测量种类 *circuitEndPointSubgroup* 的测量属性

属性名	说明	触发点	类型
来话试呼次数 (incomingBids)	指对电路子群中电路的尝试占用次数, 该值对于单向入或双向电路子群是有效的, 它包括实施网络管理控制之前的试呼, 被网络管理控制或类型控制所禁止的试呼不包括在内	接收到 IAI/IAM 消息	GSMMeasurementType1
去话试呼次数 (outgoingBids)	指对电路子群中电路的尝试占用次数, 该值对于单向出或双向电路子群有效, 它包括实施网络管理控制之前的试呼, 被网络管理控制或类型控制所禁止的试呼不包括在内	发送 IAI/IAM 消息	GSMMeasurementType1
来话占用次数 (incomingSeizures)	指对电路子群的入局呼叫的成功占用次数, 该值对于单向入和双向电路子群有效	发送 ACM 消息	GSMMeasurementType1
去话占用次数 (outgoingSeizures)	指对电路子群的出局呼叫的成功占用次数, 该值对于单向出和双向电路子群有效	接收 ACM 消息	GSMMeasurementType1
去话应答次数 (answeredOutgoingSeizures)	指出局占用中收到应答信号的次数, 该值对于单向出或双向电路子群有效	接收应答信号 (ANM/ANC/ANN)	GSMMeasurementType1
来话应答次数 (answeredIncomingSeizures)	指入局占用中收到应答信号的次数, 该值对于单向入或双向电路子群有效	发送应答信号 (ANM/ANC/ANN)	GSMMeasurementType1

表 F.8 (续)

属性名	说明	触发点	类型
来话溢出次数 (incomingBidsOverflow)	指从该电路子群溢出的入局呼叫的次数, 该值对于单向入或双向电路子群有效	无	GSMMeasurementType1
去话溢出次数 (outgoingBidsOverflow)	指从该电路子群溢出的出局呼叫的次数, 该值对于单向出或双向电路子群有效	无	GSMMeasurementType1
来话总话务量 (incomingTrafficVolume)	以 Erl/s 为单位统计入局话务量, 该值对于单向入或双向电路子群有效, 一般通过采样方法获得	累计在一个测量周期内, 从发送被叫应答消息到相应的呼叫释放之间的时间间隔, 单位: Erl/s	GSMMeasurementType2
去话总话务量 (outgoingTrafficVolume)	以 Erl/s 为单位统计出局话务量, 该值对于单向入或双向电路子群有效, 一般通过采样方法获得	累计在一个测量周期内, 从接收被叫应答消息到相应的呼叫释放之间的时间间隔, 单位: Erl/s	GSMMeasurementType2
可用电路数 (numberOfAvailCircuits)	指可以承载话务的电路数, 包括当前正在承载话务的电路, 该值对于单向出、单向入或双向电路子群均有效, 在实现时可以根据电路的频率改变快慢来决定采用快照法或平均值, 两种方法是等效的	无	GSMMeasurementType1

F.2.7 信令链路集终端点的测量

该任务所属的管理对象类为 *signallingLinkSetTP*。

该任务采集与信令链路集终端点性能统计相关的信息, 包括业务流量的统计等。其测量属性见表 F.9。

表 F.9 测量类型 signallingLinkSetTPTraffic 的测量属性

属性名	说明	触发点	类型
闭塞信令链路总数 (blockedSignallingLinks)	指不能用于业务流的信令链路数, 在实现时可以根据电路的频率改变快慢来决定采用快照法或平均值, 两种方法是等效的	无	GSMMeasurementType1
信令链路闭塞次数 (signallingLinkBlockedTimes)	指在测量周期内, 该信令链路组中信令链路的阻塞次数		GSMMeasurementType1
信令链路总话务量 (signallingLinkSetTPTrafficVolume)	指信令链路组所承载的业务量		GSMMeasurementType2

F.2.8 VLR 测量

该任务所属的管理对象类为 *vlrFunction*。

该任务采集与 VLR 相关的性能统计信息, 包括下面的统计:

一寻呼;

- 鉴权；
- 位置更新；
- 用户数；
- VLR 处理器的负荷。

F.2.8.1 VLR 性能统计

VLR 性能统计的测量属性见表 F.10。

表 F.10 测量类型 vlrPerformanceStatistics 的测量属性

属性名	说 明	触发点	类 型
向 PVLR 请求身份识别的尝试次数 (attIdentificationReqToPVLRs)	指当用户重新在该 VLR 登记时向前一 VLR 请求身份识别信息的尝试次数	发送“MAP 发送身份识别”(“MAP_SEND_IDENTIFICATION”)服务请求 (GSM 09.02)	GSMMeasurementType1
向 PVLR 请求身份识别的成功次数 (succIdentificationReqToPVLRs)	指当用户重新在该 VLR 登记时向前一 VLR 请求身份识别信息成功次数	收到不含“用户错误”(“user error”)参数值的“MAP 发送身份识别”(“MAP_SEND_IDENTIFICATION”)服务证实 (GSM 09.02)	GSMMeasurementType1
寻呼请求尝试次数 (attPageReqs)	指寻呼请求的尝试次数	发送“MAP 寻呼”(“MAP_PAGE”)服务请求 (GSM 09.02)	GSMMeasurementType1
寻呼请求成功次数 (succPageReqs)	指寻呼请求的成功次数	收到不含“用户错误”(“user error”)参数值的“MAP 寻呼”(“MAP_PAGE”)服务证实 (GSM 09.02)	GSMMeasurementType1
每位置区的寻呼请求尝试次数 (attPageReqsPerLocationArea)	Provides the number of page requests per Location Area (these are counted as attempts) 指按每个位置区所统计的寻呼请求的尝试次数	发送“MAP 寻呼”(“MAP_PAGE”)服务请求 (GSM 09.02)	GSMMeasurementType6
每位置区的寻呼请求成功次数 (succPageReqsPerLocationArea)	指按每个位置区所统计的寻呼请求的成功次数	收到不含“用户错误”(“user error”)参数值的“MAP 寻呼”(“MAP_PAGE”)服务证实 (GSM 09.02)	GSMMeasurementType6
身份验证请求 VLR-HLR 的尝试次数 (attReqForAuthSetsSentToHLR)	指 VLR 发给 HLR 的鉴权集请求的尝试次数	发送“MAP 发送鉴权信息”(“MAP_SEND_AUTHENTICATION_INFO”)服务请求, 请求鉴权集。参数中含有“鉴权集种类”(“AuthenticationSetKind”, GSM 09.02)	GSMMeasurementType1
身份验证请求 VLR-HLR 的成功次数 (succReceivedAuthSetsFromHLR)	指 VLR 的鉴权集请求得到 HLR 响应的成功次数	收到“MAP 发送鉴权信息”(“MAP_SEND_AUTHENTICATION_INFO”)服务确认, 其中含有所请求的鉴权集参数, 包含“鉴权集列表”(“AuthenticationSetList”, GSM 09.02)	GSMMeasurementType1

表 F.10 (续)

属性名	说明	触发点	类型
身份验证请求 VLR-HLR 的空响应次数 (emptyResponsesForAuthFromHLR)	指 VLR 的鉴权集请求得到 HLR 空响应的次数 (即 VLR 必须使用旧的鉴权集信息)	收到“MAP 发送鉴权信息”(“MAP_SEND_AUTHENTICATIO N_INFO”)服务确认, 其中不含鉴权集	GSMMeasurementType1
发给 MSC 的鉴权请求尝试次数 (attAuthProcsInVLR)	指 VLR 发给 MSC 的鉴权请求的尝试次数	发送“MAP 鉴权”(“MAP_ AUTHENTICATE”)服务请求 (GSM 09.02)	GSMMeasurementType1
发给 MSC 的鉴权请求成功次数 (succAuthProcsInVLR)	指 VLR 发给 MSC 的鉴权请求的成功次数	收到“MAP 鉴权”(“MAP_ AUTHENTICATE”)服务确认, 其中收到的 SRES 参数值和存储在位置寄存器中的值一致 (GSM 09.02)	GSMMeasurementType1
VLR 内的位置更新尝试次数 (attIntraVLRLocationUpdates)	指在同一 VLR 范围内的不同位置区之间进行位置更新的尝试次数	收到“MAP 更新位置区”(“MAP_ UPDATE_LOCATION_AREA”)服务指示, 其中前一个位置区标识符参数所指位置区与现在的在同一个 VLR 中 (GSM 09.02)	GSMMeasurementType1
VLR 内的位置更新成功次数 (succIntraVLRLocationUpdates)	指在同一 VLR 范围内的不同位置区之间进行位置更新的成功次数	对于 VLR 内部的位置更新尝试请求, 发送不含“用户错误”(“user error”)参数值的“MAP 更新位置区”(“MAP_UPDATE_LOCATION_AREA”)服务响应 (GSM 09.02)	GSMMeasurementType1
VLR 间的位置更新尝试次数 (attInterVLRLocationUpdates)	指在不同的 VLR 范围内的位置区之间进行位置更新的尝试次数	收到“MAP 更新位置区”(“MAP_UPDATE_LOCATION_ AREA”)服务指示, 其中前一个位置区标识符参数所指位置区与现在的不在同一个 VLR 中 (GSM 09.02)	GSMMeasurementType1
VLR 间的位置更新成功次数 (succInterVLRLocationUpdates)	指在不同的 VLR 范围内的位置区之间进行位置更新的成功次数	对于 VLR 间的位置更新尝试请求, 发送不含“用户错误”(“user error”)参数值的“MAP 更新位置区”(“MAP_UPDATE_LOCATION_ AREA”)服务响应 (GSM 09.02)	GSMMeasurementType1
VLR 漫游用户总数 (nbrOfVisitorsInVLR)	从其他地区漫游来的用户总数		GSMMeasurementType1
VLR 用户总数 (nbrOfCurrentSubscriberInVLR)	指当前 VLR 中的用户数	无	GSMMeasurementType1
VLR 申请 MSRN 号码的次数 (attReqsForMSRNFromHLR)	指来自 HLR 的请求 MSRN 的尝试次数	收到“MAP 提供漫游号码”(“MAP_ PROVIDE_ROAMING_NUMBER”)服务请求 (GSM 09.02)	GSMMeasurementType1

表 F.10 (续)

属性名	说明	触发点	类型
VLR 申请 MSRN 号码的失败次数 (unsuccRepsForMSRNToHLR)	指向 HLR 发送的分配 MSRN 的不成功响应次数	发送含有“用户错误”(“user error”)参数值的“MAP 提供漫游号码”(“MAP_PROVIDE_ROAMING_NUMBER”)的业务确认(GSM 09.02)	GSMMeasurementType1
HLR 请求加入用户数据的次数 (attInsertSubDataServiceFromHLR)	指从 HLR 发来的插入用户数据请求的尝试次数	收到“MAP 插入用户数据”(“MAP_INSERT_SUBSCRIBER_DATA”)服务请求(GSM 09.02)	GSMMeasurementType1
HLR 请求加入用户数据的失败次数 (unsuccInsertSubDataServiceToHLR)	指从 HLR 发来的插入用户数据请求的不成功次数	发送含有“用户错误”(“user error”)参数值的“MAP 插入用户数据”(“MAP_INSERT_SUBSCRIBER_DATA”)服务指示(GSM 09.02)	GSMMeasurementType1

F.2.8.2 处理器负荷

VLR 处理器负荷的测量属性见表 F.11。

表 F.11 测量类型 vlrProcessorLoad 的测量属性

属性名	说明	触发点	类型
平均占用率 (meanUsage)	指测量周期中, 主处理器的平均使用率	无	GSMMeasurementType2
峰值占用率 (peakUsage)	指测量周期中, 主处理器的峰值使用率	无	GSMMeasurementType2
峰值出现时刻 (peakTime)	指测量周期中, 主处理器峰值出现的最后时刻	无	PeakTimeType
峰值时长 (peakDuration)	主处理器处于峰值期间的持续时间(单位为 s, 且缺省的 CPU 峰值使用率的门限值为 80%)	无	GSMMeasurementType1

F.2.9 HLR 测量

该任务所属的管理对象类为 *hlrFuction*。

该任务采集与 HLR 性能统计相关的信息, 包括以下的统计:

- 漫游用户数;
- 鉴权;
- 位置更新;
- 附加业务操作;
- 对 MSC 的查询;
- HLR 处理器的负荷。

F.2.9.1 HLR 性能统计

HLR 性能统计的测量属性见表 F.12。

表 F.12 测量类型 hrPerformanceStatistics 的测量属性

属性名	说明	触发点	类型
发送警告尝试次数 (attNbrOfSendAlerts)	指发送“发送警告”的尝试次数	发送“MAP 告警服务中心” (“MAP_ALERT_SERVICE_CENTRE”) 服务请求 (GSM 09.02)	GSMMeasurementType1
发送警告成功次数 (succNbrOfSendAlerts)	指发送“发送警告”的成功次数	收到“MAP 告警服务中心” (“MAP_ALERT_SERVICE_CENTRE”) 服务确认, 其中不含有“用户错误” (“user error”) 参数值 (GSM 09.02)	GSMMeasurementType1
漫游出本 HLR 的用户数 (nbrOfCurrentMSsRoamingOutsideHLR)	指漫游出本 HLR 的用户数		GSMMeasurementType1
从 VLR 收到的身份验证请求的尝试次数 attReqForAuthSetsReceivedByHLRFromVLRs)	指 HLR 收到的从 VLR 发出的鉴权集请求的尝试次数	收到请求鉴权集的“MAP 发送鉴权信息” (“MAP_SEND_AUTHENTICATION_INFO”) 服务指示。其中含有参数“鉴权集种类” (“AuthenticationSetKind”, GSM 09.02)	GSMMeasurementType1
成功返回给 VLR 的身份验证请求次数 (succReturnedAuthSetsFromHLRToVLRs)	指从 HLR 发送给 VLR 的鉴权请求的成功返回次数	发送包含鉴权集信息的“MAP 发送鉴权信息” (“MAP_SEND_AUTHENTICATION_INFO”) 服务响应 (其中含有参数“鉴权集列表” (“AuthenticationSetList”, GSM 09.02)	GSMMeasurementType1
对 VLR 的身份验证请求返回空响应的次数 (emptyResponsesForAuthSetsFromHLRToVLRs)	指对 VLR 的鉴权集请求进行空响应的次数 (即 VLR 仍需使用旧的鉴权集)	发送不含鉴权集信息的“MAP 发送鉴权信息” (“MAP_SEND_AUTHENTICATION_INFO”) 服务响应 (GSM 09.02)	GSMMeasurementType1
向 VLR 发送插入用户请求的次数 (attInsertSubDataService)	指向 VLR 发出插入用户数据请求的尝试次数	发送“MAP 插入用户数据” (“MAP_INSERT_SUBSCRIBER_DATA”) 服务请求 (GSM 09.02)	GSMMeasurementType1
向 VLR 发送插入用户请求的成功次数 (succInsertSubDataService)	指向 VLR 发出插入用户数据请求的成功次数	收到不含“用户错误” (“user error”) 参数值的“MAP 插入用户数据” (“MAP_INSERT_SUBSCRIBER_DATA”) 服务响应 (GSM 09.02)	GSMMeasurementType1
位置更新的尝试次数 (attLocationUpdate)	指要由 HLR 执行的位置更新的尝试次数, 如 VLR 发生变化	收到“MAP 更新位置” (“MAP_UPDATE_LOCATION”) 服务指示 (GSM 09.02)	GSMMeasurementType1
位置更新的成功次数 (succLocationUpdate)	指由 HLR 执行的位置更新的成功次数, 如 VLR 发生变化	发送不含“用户错误” (“user error”) 参数值的“MAP 更新位置” (“MAP_UPDATE_LOCATION”) 服务响应 (GSM 09.02)	GSMMeasurementType1

表 F.12 (续)

属性名	说明	触发点	类型
与附加业务相关的操作尝试次数 (attSSRelatedOperationsInHLR)	指 HLR 中与附加业务相关的操作的尝试次数	收到 HLR 中下列操作的 MAP 服务指示: 注册附加服务, 删除附加服务, 激活附加服务, 去活附加服务, 注册口令, 查询附加服务, 处理附加请求 (registerSS, eraseSS, activateSS, deactivateSS, registerPassword, interrogateSS 或 processSSrequest, GSM 09.02)	GSMMeasurementType4
与附加业务相关的操作成功次数 (succSSRelatedOperationsInHLR)	指 HLR 中与附加业务相关的操作的成功次数	发送参数值中不含“用户错误”(“user error”)的下列 MAP 服务操作的响应: 注册附加服务, 删除附加服务, 激活附加服务, 去活附加服务, 注册口令, 查询附加服务, 处理附加请求 (registerSS, eraseSS, activateSS, deactivateSS, registerPassword, interrogateSS 或 processSSrequest, GSM 09.02)	GSMMeasurementType4
向 VLR 发送 MSRN 请求的次数 (attReqForMSRN)	指发送给 VLR 的请求 MSRN 的尝试次数。	发送“MAP 提供漫游号码”(“MAP_PROVIDE_ROAMING_NUMBER”)服务请求 (GSM 09.02)	GSMMeasurementType1
向 VLR 发送 MSRN 请求的成功次数 (succReqForMSRN)	指从 VLR 收到的分配 MSRN 的成功响应次数	收到不含“用户错误”(“user error”)参数值的“MAP 提供漫游号码”(“MAP_PROVIDE_ROAMING_NUMBER”)的服务确认 (GSM 09.02)	GSMMeasurementType1
同时申请 GSM 服务 GPRS 服务的用户数 (numOfGSMGPRSUsers)	指既申请了 GSM 服务, 又申请了 GPRS 服务的当前用户数	无	GSMMeasurementType1
仅仅申请 GPRS 服务的用户数 (numOfGPRSOnlyUsers) 指仅申请了 GPRS 服务的当	前用户数	无	GSMMeasurementType1
漫游的 GPRS 用户数 (numOfRoamingGPRSUsers)	指漫游出本 HLR 的当前 GPRS 用户数	无	GSMMeasurementType1
GPRS 路由请求信息的尝试次数 (attGPRSRoutingReqsInHLR)	指 GGSN 向 HLR 请求 GPRS 路由信息的尝试次数	从 GGSN 接收到“MAP 发送 GPRS 路由请求”(“MAP_SEND_ROUTING_INFO_FOR_GPRS”)服务请求	GSMMeasurementType1
GPRS 路由请求信息的成功次数 (succGPRSRoutingReqsInHLR)	指 GGSN 向 HLR 请求 GPRS 路由信息的成功次数	发送不含“用户错误”(“user error”)的“MAP 发送 GPRS 路由请求”(“MAP_SEND_ROUTING_INFO_FOR_GPRS”)服务确认	GSMMeasurementType1

F.2.9.2 处理器负荷

HLR 处理器负荷的测量属性见表 F.13。

表 F.13 测量类型 hlrProcessorLoad 的测量属性

属性名	说明	触发点	类型
平均占用率 (meanUsage)	指测量周期中, 主处理器的平均使用率	无	GSMMeasurementType2
峰值占用率 (peakUsage)	指测量周期中, 主处理器的峰值使用率	无	GSMMeasurementType2
峰值出现时刻 (peakTime)	指测量周期中, 主处理器峰值出现的最后时刻	无	PeakTimeType
峰值时长 (peakDuration)	主处理器处于峰值期间的持续时间 (单位为 s, 且缺省的 CPU 峰值使用率的门限值为 80%)	无	GSMMeasurementType2

F.2.10 EIR 测量

该任务所属的管理对象类为 *eirFunction*。

该任务收集与 EIR 相关的性能数据, 包括下面的统计:

- IMEI 监测;
- EIR 处理器的负荷。

F.2.10.1 EIR 性能统计

EIR 性能统计的测量属性见表 F.14。

表 F.14 测量类型 eirPerformanceStatistics 的测量属性

属性名	说明	触发点	类型
收到 IMEI 检查次数 (nbrOfReceivedIMEICheckReqs)	指 EIR 收到的 IMEI 检查的请求次数	收到 “MAP 检查 IMEI” (“MAP_CHECK_IMEI”) 服务指示	GSMMeasurementType1
发送白名单应答次数 (nbrOfWhiteAnsInEIR)	指 EIR 发送的白名单应答的次数	发送 “MAP 检查 IMEI” (“MAP_CHECK_IMEI”) 服务响应, 其中包含的设备状态 (“equipment status”) 参数所指的是白名单中的设备 (GSM 09.02 和 GSM 02.16)	GSMMeasurementType1
发送灰名单应答次数 (nbrOfGreyAnsInEIR)	指 EIR 发送的灰名单应答的次数	发送 “MAP 检查 IMEI” (“MAP_CHECK_IMEI”) 服务响应, 其中包含的设备状态 (“equipment status”) 参数所指的是灰名单中的设备 (GSM 09.02 和 GSM 02.16)	GSMMeasurementType1
发送黑名单应答次数 (nbrOfBlackAnsInEIR)	指 EIR 发送的黑名单应答的次数	发送 “MAP 检查 IMEI” (“MAP_CHECK_IMEI”) 服务响应, 其中包含的设备状态 (“equipment status”) 参数所指的是黑名单中的设备 (GSM 09.02 和 GSM 02.16)	GSMMeasurementType1

表 F.14 (续)

属性名	说明	触发点	类型
发送未知 IMEI 应答次数 (nbrOfUnknownIMEIAnsInEIR)	指 EIR 发送的未知 IMEI 应答的次数	发送“MAP 检查 IMEI” (“MAP_CHECK_IMEI”) 服务响应, 其中包含的设备状态 (“equipment status”) 参数所指的未知的设备 (GSM 09.02 和 GSM 02.16)	GSMMeasurementType1

F.2.10.2 处理器负荷

EIR 的处理器负荷的测量属性见表 F.15。

表 F.15 测量类型 airProcessorLoad 的测量属性

属性名	说明	触发点	类型
平均占用率 (meanUsage)	指测量周期中, 主处理器的平均使用率	无	GSMMeasurementType2
峰值占用率 (peakUsage)	指测量周期中, 主处理器的峰值使用率	无	GSMMeasurementType2
峰值出现时刻 (peakTime)	指测量周期中, 主处理器峰值出现的最后时刻	无	PeakTimeType
峰值时长 (peakDuration)	主处理器处于峰值期间的持续时间 (单位为 s, 且缺省的 CPU 峰值使用率的门限值为 80%)	无	GSMMeasurementType1

F.2.11 SMS 测量

该任务所属的管理对象类为 *smsGIWFunction*。

该任务采集与 SMSC 相关的性能统计信息, 包括下面的统计:

- 短消息的转发;
- SMS 处理器的负荷。

F.2.11.1 SMS 性能统计

SMS 性能统计的测量属性见表 F.16。

表 F.16 测量类型 smsPerformanceStatistics 的测量属性

属性名	说明	触发点	类型
始发短消息尝试次数 (attMobileOriginatingSMForwards)	指从服务 MSC 向互操作网相关 MSC 转发短消息的尝试次数	收到“MAP 转发短消息” (“MAP_FORWARD_SHORT_MESSAGE”) 服务指示 (GSM 09.02)	GSMMeasurementType1
始发短消息成功次数 (succMobileOriginatingSMForwards)	指从服务 MSC 向互操作 MSC 转发短消息的成功次数	发送不含“用户错误” (“user error”) 参数值的“MAP 转发短消息” (“MAP_FORWARD_SHORT_MESSAGE”) 服务响应 (GSM 09.02)	GSMMeasurementType1

表 F.16 (续)

属性名	说明	触发点	类型
终接短消息尝试次数 (attMobileTerminatingSMForwards)	指从汇接 MSC 向服务 MSC 转发短消息的尝试次数	发送“MAP 转发短消息” (“MAP_FORWARD_SHORT_MESSAGE”) 服务请求 (GSM 09.02)	GSMMeasurementType1
终接短消息成功次数 (succMobileTerminatingSMForwards)	指从汇接 MSC 向服务 MSC 转发短消息的成功次数	收到不含“用户错误” (“user error”) 参数值的“MAP 转发短消息” (“MAP_FORWARD_SHORT_MESSAGE”) 服务响应 (GSM 09.02)	GSMMeasurementType1

F.2.11.2 处理器负荷

SMS 处理器负荷的测量属性见表 F.17。

表 F.17 测量类型 smsProcessorLoad 的测量属性

属性名	说明	触发点	类型
平均占用率 (meanUsage)	指测量周期中, 主处理器的平均使用率	无	GSMMeasurementType2
峰值占用率 (peakUsage)	指测量周期中, 主处理器的峰值使用率	无	GSMMeasurementType2
峰值出现时刻 (peakTime)	指测量周期中, 主处理器峰值出现的最后时刻	无	PeakTimeType
峰值时长 (peakDuration)	主处理器处于峰值期间的持续时间 (单位为 s, 且缺省的 CPU 峰值使用率的门限值为 80%)	无	GSMMeasurementType1

F.2.12 BSC 测量

该任务所属的管理对象类为 bsc。

该任务采集与 BSC 性能统计相关的信息, 包括以下的统计:

- 由该 BSC 所控制的小区切换;
- 话务量;
- BSC 处理器的负荷;
- GPRS 子层业务量。

F.2.12.1 切换统计

表 F.18 测量类型 bschDOSStatistics 的测量属性

属性名	说明	触发点	类型
BSC 小区内切换成功次数 (succInternalHDOsIntraCellInBSC)	指在通话过程中, 在 BSC 范围内同一个小区信道之间切换的成功次数。如果被观测的 BSC 包含多个小区, 该属性值应当是该 BSC 中所有的小区成功切换次数的总和	向 MSC 发送“切换已执行” (“HANDOVER PERFORMED”) 消息 (GSM 08.08)	GSMMeasurementType1

表 F.18 (续)

属性名	说明	触发点	类型
BSC 小区内切换失败次数 (unsuccInternalHDOsIntraCellInBSC)	指在通话过程中, 在 BSC 范围内同一个小区信道之间切换的成功次数。如果被观测的 BSC 包含多个小区, 该属性值应当是该 BSC 中所有的小区内成功切换次数的总和	对于进行尝试的小区内切换, 收到“分配失败” (“ASSIGNMENT FAILURE”) 消息 (GSM 04.08)	GSMMeasurementType1
BSC 小区间切换成功次数 (succInternalHDOsInBSC)	指在指定的 BSC 范围内成功的小区间切换的总数	向 MSC 发送“切换已执行” (“HANDOVER PERFORMED”) 消息 (GSM 08.08)	GSMMeasurementType1
BSC 小区间成功切换各种原因统计次数 (succInternalHDOsPerCause)	按照每种原因, 来统计在指定的 BSC 范围内成功的小区间切换的次数, 这些原因包括: 上行链路质量, 上行信号强度, 下行链路质量, 下行信号强度, 距离, 更好的小区, 操作维护干涉, 定向重试, 响应 MSC 调用	按照不同的原因, 向 MSC 发送“切换已执行” (“HANDOVER PERFORMED”) 消息 (GSM 08.08)	GSMMeasurementType3
BSC 小区切换不成功但未掉话次数 (unsuccInternalHDOsWithReconnectionInBSC)	指在给定的 BSC 中, 切换不成功, 但仍回到原信道通话的次数	对于进行尝试的小区内切换, 收到“切换失败” (“HANDOVER FAILURE”) 消息 (GSM 04.08)	GSMMeasurementType1
BSC 小区切换不成功而掉话次数 (unsuccInternalHDOsWithLossOfConnectionInBSC)	指在给定的 BSC 中, 切换不成功导致掉话的次数	小区间切换时, 计时器 T3103 超时; 小区内切换时, 计时器 T3107 超时 (GSM 04.08)	GSMMeasurementType1

F.2.12.2 话务量统计

BSC 话务量统计的测量属性见表 F.19。

表 F.19 测量类型 bscTrafficStatistics 的测量属性

属性名	说明	触发点	类型
BSC 服务请求失败次数 (unsuccReqsForService)	指 MS 发起的不成功的 BSC 业务请求次数 注: 参数“拒绝原因” (“reject cause”) 指明了业务请求被拒绝的原因。可能的原因包括 B 类用户忙、网络故障、协议错误等	向请求服务的手机发送“CM 服务拒绝” (“CM -SERVICE REJECT”) 消息 (GSM 04.08)	GSMMeasurementType1
寻呼信息的尝试发送次数 (attTransOfPagingMessagesInBSC)	指 BSC 发送的寻呼消息的试呼次数	发送“寻呼” (“PAGING”) 消息 (GSM 04.08)	GSMMeasurementType1

表 F.19 (续)

属性名	说明	触发点	类型
寻呼信息的不成功发送次数 (unsuccTransOfPagingMessages InBSC)	指 BSC 发送的不成功的寻呼消息的次数	没有收到因“寻呼请求” (“PAGING REQUEST”) 引起的“寻呼响应” (“PAGING RESPONSE”) 消息, T3113 计时器超时 (GSM04.08)	GSMMeasurementType1
BSC 立即分配尝试次数 (attImmediateAssignProcsInBSC)	指 BSC 的立即分配过程的尝试次数	收到“信道请求” (“CHANNEL request”) 消息 注: 建立的原因定义在 GSM 04.08 中, 包括紧急呼叫、呼叫重建、寻呼响应、始发呼叫、位置更新和其他过程	GSMMeasurementType1
BSC 立即分配成功次数 (succImmediateAssignProcsInBSC)	指 BSC 的立即分配过程的成功次数	发送“立即分配指令” (“IMMEDIATE ASSIGN COMMAND”) 消息。该消息包含“立即分配” (“IMMEDIATE ASSIGNMENT”) 消息或“扩展的立即分配” (“IMMEDIATE ASSIGNMENT EXTENDED”) 消息其中之一。如果发送的是“扩展的立即分配”, 由于消息中包含两个手机的分配信息, 计数器的值应加 2 (GSM 04.08)	GSMMeasurementType1
BSC 总话务量 (bscTotalTrafficVolume)	指以 Erl/s 为单位的 BSC 所承载的总话务量, 包含各种呼叫的成功和失败的时间段, 该值通过采样方式获得		GSMMeasurementType2
分组信道接入请求发生的平均时间间隔 (meanPSInterArrivalTime)	指移动台连续发起的分组信道接入请求消息间隔的算术平均值。将连续接收到 ‘PACKET CHANNEL REQUEST’ 之间的时间长度累加, 然后进行算术平均	在一个测量周期中, 累计连续接收到“分组信道请求” (“PACKET CHANNEL REQUEST”) 之间的时间间隔, 然后进行算术平均	GSMMeasurementType2
接收到的 flush 请求消息 (flushRequestReceived)	该测量值是从 SGSN 接收的 FLUSH 请求信息总数, 该信息请求冲掉某指定小区的 PDU	从 SGSN 接收到“PDU 清空请求” (“PDU FLUSH REQUEST”) 消息	GSMMeasurementType1

F.2.12.3 处理器负荷

BSC 处理器负荷的测量属性见表 F.20。

表 F.20 测量类型 bscProcessorLoad 的测量属性

属性名	说明	触发点	类型
平均占用率 (meanUsage)	指测量周期中, 主处理器的平均使用率	无	GSMMeasurementType2
峰值占用率 (peakUsage)	指测量周期中, 主处理器的峰值使用率	无	GSMMeasurementType2
峰值出现时刻 (peakTime)	指测量周期中, 主处理器峰值出现的最后时刻	无	PeakTimeType
峰值时长 (peakDuration)	主处理器处于峰值期间的持续时间 (单位为 s, 且缺省的 CPU 峰值使用率的门限值为 80%)	无	GSMMeasurementType1

F.2.12.4 GPRS 子层业务量

BSCGPRS 子层业务量的测量属性见表 F.21。

表 F.21 测量类型 bscGPRSSubLayerTraffic 的测量属性

属性名		说明	触发点	类型
无线链路控制 (RLC) 子层	PCU 的 Um 口的上行负荷 (nbrUmULBlock)	指 PCU 在 RLC 层的上行数据块数	在 RLC 子层上收到上行的数据块	GSMMeasurementType1
	PCU 的 Um 口的下行负荷 (nbrUmDLBlock)	指 PCU 在 RLC 层的下行数据块数	在 RLC 子层上收到下行的数据块	GSMMeasurementType1
网络业务 (NS) 子层	NS 子层发送的字节数 (nbrNsOctetSentBsc)	指 NS 子层成功发送到 Gb 接口上的 BSSGP 的数据量	在 NS 子层的 Gb 接口上发送 BSSGP, 并最终累计总字节数	GPRSMMeasurementType3
	NS 子层接收的字节数 (nbrNsOctetRvdBsc)	指 NS 子层从对端接收到的 BSSGP 的数据量	在 NS 子层的 Gb 接口上接收到 BSSGP, 并最终累计总字节数	GPRSMMeasurementType3
	NS 子层发送的数据 PDU 数量 (nbrNsPduSentBsc)	指 NS 子层成功发送到 Gb 接口上的 BSSGP 的 PDU 数量	在 NS 子层的 Gb 接口上发送 BSSGP PDU	GPRSMMeasurementType3
	NS 子层接收的数据 PDU 数量 (nbrNsPduRvdBsc)	指 NS 子层从对端接收到的 BSSGP 的 PDU 数量	在 NS 子层的 Gb 接口上接收到 BSSGP PDU	GPRSMMeasurementType3
	NS 子层丢弃的数据 PDU 数量 (nbrNsPduDiscardedBsc)	指 NS 没有成功发送到 Gb 接口的 BSSGP 的数据包的数目	在 NS 子层的 Gb 接口上丢弃 BSSGP PDU	GPRSMMeasurementType3
	NS 子层向对端发送复位消息的数量 (nbrNsResetMegSendBsc)	指 NS 为了复位 NS-VC 发送到对端的 NS-RESET 消息的数量	在 NS 子层向对端发送用于将 NS-VC 复位的“NS-RESET”消息	GPRSMMeasurementType3
	接收到对端发送的复位消息的数量 (nbrNsResetMegRvdBsc)	指 NS 接收到对端发送的 NS-RESET 消息的数量	在 NS 子层从对端接收到用于复位 NS-VC 的“NS-RESET”消息	GPRSMMeasurementType3

表 F.21 (续)

属性名	说明	触发点	类型
NS 子层向对端回复复位确认消息的数量 (nbrNsResetAckMegSentBsc)	NS 子层向对端回复 RESET ACK 消息时统计	在 NS 子层向对端发送“RESETACK”消息	GPRSMeasurementType3
接收到对端回复复位确认消息的数量 (nbrNsResetAckMegRvdBsc)	NS 子层收到对端回复 RESET ACK 消息时统计	在 NS 子层从对端接收到“RESETACK”消息	GPRSMeasurementType3
NS 子层向对端发送闭塞消息的数量 (nbrNsBlockMegSentBsc)	指 NS 为了闭塞 NS-VC 发送到对端的 NS-BLOCK 消息的数量	在 NS 子层向对端发送用于将闭塞 NS_VC 的“RESETACK”消息	GPRSMeasurementType3
接收到对端闭塞消息的数量 (nbrNsBlockMegRvdBsc)	指 NS 接收到对端发送的 NS-BLOCK 消息的数量	在 NS 子层向对端接收到用于将闭塞 NS_VC 的“RESETACK”消息	GPRSMeasurementType3
NS 子层向对端发送解闭消息的数量 (nbrNsUnblockMegSentBsc)	指 NS 为了解闭 NS-VC 发送到对端的 NS-UNBLOCK 消息的数量	在 NS 子层向对端发送用于 NS_VC 解闭的“RESETACK”消息	GPRSMeasurementType3
NS 子层向对端回复闭塞确认消息的数量 (nbrNsBlockAckMegSentBsc)	NS 子层向对端回复的 BLOCK PDU ACK 消息的数量	在 NS 子层向对端发送“BLOCKPDUACK”消息	GPRSMeasurementType3
接收到对端回复的闭塞确认消息数量 (nbrNsBlockAckMegRvdBsc)	NS 子层收到对端回复的 BLOCK PDU ACK 消息的数量	在 NS 子层向对端收到“BLOCKPDUACK”消息	GPRSMeasurementType3
NS 子层向对端回复解闭确认消息的数量 (nbrNsUnblockAckMegSentBsc)	NS 子层向对端回复的 UNBLOCK PDU ACK 消息的数量	在 NS 子层向对端发送“UNBLOCKPDUACK”消息	GPRSMeasurementType3
接收到对端回复的解闭确认消息数量 (nbrNsUnblockAckMegRvdBsc)	NS 子层收到对端回复的 UNBLOCK PDU ACK 消息的数量	在 NS 子层向对端收到“UNBLOCKPDUACK”消息	GPRSMeasurementType3
接收到对端解闭消息的数量 (nbrNsUnblockMegRvdBsc)	指 NS 接收到对端发送的 NS_UNBLOCK 消息的数量	在 NS 子层向对端收到“NS_UNBLOCK”消息	GPRSMeasurementType3

F.2.13 BTS 测量

该任务所属的管理对象类为 *bts*。

该任务采集与 BTS 性能统计相关的信息，包括以下的统计：

- BTS 的话务量；
- 功率电平；
- BTS 的小区切换；
- BTS 的位置更新；
- GPRS 的无线资源；
- GPRS 的分组业务。

F.2.13.1 BTS 业务量数据

BTS 业务量数据的测量属性见表 F.22。

表 F.22 测量类型 btsTrafficData 的测量属性

属性名	说明	触发点	类型
SDCCH 总数 (totalNumberOfSDCCH)	指 BTS 中 SDCCH 的总数	无	GSMMeasurementType1
可用 SDCCH 数 (availableSDCCH)	指 BTS 中可用的 SDCCH 数目 (不包括目前正在使用的)	无	GSMMeasurementType1
SDCCH 试呼次数 (attSDCCHSeizures)	指 SDCCH 的占用尝试次数	接收到“信道请求”(“CHANNEL REQUEST”)消息	GSMMeasurementType1
SDCCH 溢出次数 (blockedSDCCH)	指阻塞的 SDCCH 占用次数 (即所有 SDCCH 忙时发生的占用)	发送“立即分配拒绝”(“Immediate Assignment Reject”)消息(不包括由切换所引起的)	GSMMeasurementType1
SDCCH 掉话次数 (droppedSDCCH)	指丢弃的 SDCCH 占用次数		GSMMeasurementType1
TCH 阻塞次数 (blockedTCH)	指阻塞的 TCH 占用次数	发送“分配失败”(“Assignment Failure”)消息, 其原因为“没有无线资源”	GSMMeasurementType1
TCH 试呼次数 (attTCHSeizures)	指 TCH 的占用尝试次数	向 MS 发送“分配指令”消息(“ASSIGNMENT COMMAND”, GSM 04.08)	GSMMeasurementType1
TCH 成功占用次数 (succTCHSeizures)	指成功的 TCH 占用次数	从 MS 接收到“分配完成”消息 (“Assignment Complete”, GSM 04.08)	GSMMeasurementType1
TCH 掉话次数 (droppedTCH)	指丢弃的 TCH 占用次数	发送“清除请求”消息(“Clear Request”)	GSMMeasurementType1
SDCCH 总话务量 (totalTrafficVolumeOfSDCCH)	指以 Erl/s 为单位的 SDCCH 所承载的总话务量, 该值通过采样方式获得		GSMMeasurementType2
SDCCH 射频掉话次数 (droppedSDCCHOnRF)	指在无线载波上掉话的 SDCCH 占用次数		GSMMeasurementType1
SDCCH 平均占用时长 (averageHoldTimeOfSDCCH)	指 SDCCH 占用的平均占用时长(单位: s)	无	GSMMeasurementType2
SDCCH 信道全忙时长 (cellBusyTimeOfSDCCH)	指 BTS 中所有 SDCCH 全忙的时间段的累计合(单位: s)	无	GSMMeasurementType1
SDCCH 平均信道分配时长 (averageTimeAssignOfSDCCH)	指 BTS 中 SDCCH 的平均分配时长(单位: s)		GSMMeasurementType2
可用 TCH 数 (availableTCH)	可用的 TCH 数目(除去那些当前正在使用的)	无	GSMMeasurementType1
TCH 总话务量 (totalTrafficVolumeOfTCH)	指以 Erl/s 为单位的 TCH 所承载的总话务量, 该值通过采样方式获得		GSMMeasurementType2

表 F.22 (续)

属性名	说明	触发点	类型
TCH 分配成功未占用次数 (numOfSuccAssignUnSeizure)	指 BTS 中成功分配 TCH 但未占用的次数		GSMMeasurementType1
TCH 射频掉话次数 (droppedTCHOnRF)	指 TCH 的射频掉话次数		GSMMeasurementType1
TCH 平均占用时长 (averageHoldTimeOfTCH)	指 TCH 的平均占用时长 (单位: s)	无	GSMMeasurementType2
TCH 信道全忙时长 (cellBusyTimeOfTCH)	指 BTS 中 TCH 全忙的时长 (单位: s)	无	GSMMeasurementType1
TCH 平均信道分配时长 (averageTimeAssignOfTCH)	指 BTS 中 TCH 的平均分配 时长 (单位: s)		GSMMeasurementType2
最大 TCH 占用数量 (maxNumOfSeizuredTCH)	指在测量周期中, BTS 中同 时占用的最大 TCH 数	无	GSMMeasurementType1
可用 PCH 数 (availablePCH)	指可用的 PCH 数 (除去那 些正在使用的)	无	GSMMeasurementType1
寻呼试呼次数 (attTransOfPagingMessagesThe PCH)	指 BTS 中的寻呼的试呼次数	发送“寻呼”消息 (“PAGING”, GSM 04.08)	GSMMeasurementType1
寻呼信息的不成功发送次数 (unsuccTransOfPagingMessages ThePCH)	指 BTS 中不成功的寻呼次数	没有收到与“寻呼请求” (“PAGING REQUEST”) 相对应的“寻呼响应” 消息 (“PAGING RESPONSE”), 定时 器 T3113 超时 (GSM 04.08) 注: 当提供 PCCCH 时该属性只对电路 方式有效	GSMMeasurementType1
立即信道分配的尝试次数 (attImmediateAssignProcsInBTS)	指 BTS 的立即分配规程的试 呼次数	收到“信道请求” (“CHANNEL request”) 注: 建立的原因定义在 GSM 04.08 中, 包括紧急呼叫、呼叫重建、寻呼响应、 始发呼叫、位置更新和其他过程	GSMMeasurementType1
立即信道分配的成功次数 (succImmediateAssignProcsInBTS)	指 BTS 中成功的立即分配规 程的次数	发送“立即分配指令” (“IMMEDIATE ASSIGN COMMAND”) 消息。该消息 包含“立即分配” (“IMMEDIATE ASSIGNMENT”) 消息或“扩展的立即 分配” (“IMMEDIATE ASSIGNMENT EXTENDED”) 消息其中之一。如果发 送的是“扩展的立即分配”, 由于消息 中包含两个手机的分配信息, 计数器 的值应加 2 (GSM 04.08)	GSMMeasurementType1
可用载波数 (numOfAvailRadioCarriers)	指 BTS 中当前可用的载波数	无	GSMMeasurementType1
可用收发信机数目 (numOfAvailTrx)	指 BTS 中当前可用的收发信 机的数目	无	GSMMeasurementType1

表 F.22 (续)

属性名	说明	触发点	类型
相干频带内的平均空闲时隙数 (meanNbrOfIdleTCHsPerInterferenceBand)	指各相干频带内空闲时隙的算术平均值, 空闲时隙是指那些可被分配给一个请求的时隙, 空闲时隙根据测量到的干扰程度分为五类 (GSM 05.01)	按照预先定义的时间间隔 (系统定义) 进行的各相干频带内空闲时隙数的抽样值的算术平均 (GSM 04.08)	GSMMeasurementType8
从 PCH 队列中丢弃的寻呼消息数 (nbrOfPagesDiscardedFromPCHQueue)	该测量值是在传送之前被从 PCH 队列中删除的寻呼消息数目	寻呼消息可由于一些原因 (例如队列溢出、优先插入导致的队列溢出或排队计数器超时等) 从队列中被删除 (在采用队列机制的情况下, GSM 04.08)。该测量值在提供了 PCCCH 时只对电路方式有效	GSMMeasurementType1
成功的寻呼过程的平均时长 (meanDurationOfSuccPaging-Pros)	指一个成功寻呼过程持续时间的算术平均值, 即一个粒度期间内有关时间 (每次寻呼的持续时间) 的加权的算术平均	在测量周期内, 累计从向 MS 发送“寻呼请求” (“PAGING REQUEST”) 到接收到 MS 的“寻呼响应” (“PAGING RESPONSE”) 服务确认消息之间的时间, 最后求出平均值 (GSM 04.08)	GSMMeasurementType2
PCH-AGCH 平均队列长度 (meanPCHAGCHQueueLength)	该测量值是在 PCH-AGCH 信道上等待传送的消息数目的算术平均值 注: 该测量值当提供了 PC-CCH 信道的条件下只对电路方式有效	按照预先定义的间隔 (系统定义) 对 PCH-AGCH 队列的长度进行抽样, 然后进行算术平均 (GSM 04.08)	GSMMeasurementType2

F.2.13.2 基站的功率电平

基站的功率电平的测量属性见表 F.23。

表 F.23 测量类型 powerLevelOfBS 的测量属性

属性名	说明	触发点	类型
平均移动终端功率 (powerOfMobileStation)	指测量周期中, BTS 里所有的活跃的移动台功率的算术平均值 (单位: dBm)	无	GSMMeasurementType1
平均 BTS 发射功率 (powerOfBTS)	指测量周期中, BTS 功率的算术平均值 (单位: dBm)	无	GSMMeasurementType1
平均上行信号强度 (strengthOfUplink)	指测量周期中, BTS 里所有活跃的上行链路信号强度的算术平均值	无	GSMMeasurementType1
平均下行信号强度 (strengthOfDownlink)	指测量周期中, BTS 里所有活跃的下行链路信号强度的算术平均值	无	GSMMeasurementType1

表 F.23 (续)

属性名	说明	触发点	类型
平均上行质量 (qualityOfUplink)	指测量周期中, BTS 内所有活跃的上行链路信号质量的算术平均值	无	GSMMeasurementType1
平均下行质量 (qualityOfDownlink)	指测量周期中, BTS 内所有活跃的下行链路信号质量的算术平均值	无	GSMMeasurementType1
平均呼叫记录 (meanCallDistance)	指测量周期中, 所有呼叫距离的算术平均值, 即被观测的 BTS 与发起或接收呼叫的 MS 之间的距离 (单位: m)	无	GSMMeasurementType1
从 MS 来的加大功率请求次数 (attReqsForPowerIncreaseFromMS)	指在测量周期中, 由 MS 发出的功率增加请求的尝试次数		GSMMeasurementType1
从 MS 来的减小功率请求次数 (attReqsForPowerReduceFromMS)	指在测量周期中, 由 MS 发出的功率减少请求的尝试次数		GSMMeasurementType1
从 BSS 来的增大功率指令数 (attInstructionsForPowerIncreaseFromBSC)	指在测量周期中, 由 BSC 发出的功率增加请求的尝试次数		GSMMeasurementType1
从 BSS 来的减小功率指令数 (attInstructionsForPowerReduceFromBSC)	指在测量周期中, 由 BSC 发出的功率增加请求的尝试次数		GSMMeasurementType1

F.2.13.3 BTS 切换的测量

BTS 切换的测量的属性见表 F.24。

表 F.24 测量类型 btsHDOSStatistics 的测量属性

属性名	说明	触发点	类型
小区内切换成功次数 (succInternalHDOsIntraCellInBTS)	指在通话过程中, 在一个 BTS 的小区内的信道之间切换的成功次数		GSMMeasurementType1
小区内切换失败次数 (unsuccInternalHDOsIntraCellInBTS)	指在通话过程中, 在一个 BTS 的小区内的信道之间切换的不成功次数		GSMMeasurementType1
无资源小区内切换失败次数 (unsuccInternalHDOsForNoCHsIntraCellInBTS)	指一个 BTS 的小区内切换因没有可用的信道而失败的次数		GSMMeasurementType1

表 F.24 (续)

属性名	说明	触发点	类型
占用新信道引起的小区切换失败次数 (unsuccInternalHDOsForSeizingNewCHIntraCellInBTS)	指一个 BTS 中, 因占用新信道引起的小区切换失败次数		GSMMeasurementType1
切回老信道引起的小区切换失败次数 (unsuccInternalHDOsWithReconnectionIntraCellInBTS)	指一个 BTS 中, 小区内切换不成功, 但仍回到原信道通话的次数		GSMMeasurementType1
由 BSC 控制的切入小区的成功切换次数 (succIncomingInternalHDOsIntraBSCInBTS)	指在同一个 BSC 范围内, 切入该小区的成功切换次数		GSMMeasurementType1
由 BSC 控制的切入小区的不成功切换次数 (unsuccIncomingInternalHDOsIntraBSCInBTS)	指在同一个 BSC 范围内, 切入该小区的不成功切换次数		GSMMeasurementType1
由于小区内没有信道而引起的切入 BSC 小区失败次数 (unsuccIncomingInternalHDOsForNoCHsIntraBSCInBTS)	指在同一个 BSC 范围内, 由于小区内没有可用信道而引起的切入该小区的失败次数		GSMMeasurementType1
切出 BSC 小区的成功次数 (succOutgoingInternalHDOsIntraBscInBTS)	指在同一个 BSC 范围内, 切出该小区的成功次数		GSMMeasurementType1
切出 BSC 小区的不成功次数 (unsuccOutgoingInternalHDOsIntraBSCInBTS)	指在同一个 BSC 范围内, 切出该小区的不成功次数		GSMMeasurementType1
由于手机返回到老信道上而造成切出 BSC 小区失败 (unsuccOutgoingInternalHDOsWithReconnectionIntraBSCInBTS)	指在同一个 BSC 范围内, 切出小区不成功, 但仍回到原信道通话的次数		GSMMeasurementType1
切入 MSC 小区的成功次数 (succIncomingExternalHDOsIntraMSCInBTS)	指在同一 MSC 范围内, 切入该小区的成功次数 (除去那些在同一 BSC 范围内的小区切换)		GSMMeasurementType1
切入 MSC 小区的不成功次数 (unsuccIncomingExternalHDOsIntraMSCInBTS)	指在同一 MSC 范围内, 切入该小区的成功次数 (除去那些在同一 BSC 范围内的小区切换)		GSMMeasurementType1
由于无信道而造成的切入 MSC 小区失败次数 (unsuccIncomingExternalHDOsForNoCHsIntraMSCInBTS)	指在同一 MSC 范围内, 因没有信道而引起的小区切换失败次数, (除去那些在同一 BSC 范围内的小区切换)		GSMMeasurementType1

表 F.24 (续)

属性名	说明	触发点	类型
由于手机未能占上新的信道而造成的切入 MSC 小区的失败次数 (unsuccIncomingHDOforsNotSeizingNewCHIIntraMSCInBTS)	指在同一 MSC 范围内, 因没有占用到新信道而引起的小区切换失败次数 (除去那些在同一 BSC 范围内的小区切换)		GSMMeasurementType1
切出 MSC 小区的成功次数, 即 MSC 控制的小区切换次数 (succOutgoingExternalHDOforsIntraMSCInBTS)	指在同一 MSC 范围内, 切出该小区的成功切换次数 (除去那些在同一 BSC 范围内的小区切换)		GSMMeasurementType1
切出 MSC 小区的不成功次数 (unsuccOutgoingExternalHDOforsIntraMSCInBTS)	指在同一 MSC 范围内, 切出该小区的不成功切换次数 (除去那些在同一 BSC 范围内的小区切换)		GSMMeasurementType1
由于手机返回到老信道而造成的切出 MSC 小区失败次数 (unsuccOutgoingExternalHDOforsWithReconnectionIntraMSCInBTS)	指在同一 MSC 范围内, 切出该小区的不成功, 但仍回到原信道通话的次数 (除去那些在同一 BSC 范围内的小区切换)		GSMMeasurementType1
由于上行链路质量不足引起的切换请求次数 (attHDOforsReqsForLowUplinkQuality)	指由于上行链路质量不足引起的切换请求次数		GSMMeasurementType1
由于下行链路质量不足引起的切换请求次数 (attHDOforsReqsForLowDownlinkQuality)	指由于下行链路质量不足引起的切换请求次数		GSMMeasurementType1
由上行链路功率电平过低引起的切换请求次数 (attHDOforsReqsForLowUplinkPowerLevel)	指由于上行链路信号电平过低引起的切换请求次数		GSMMeasurementType1
由下行链路信号功率电平过低引起的切换请求次数 (attHDOforsReqsForLowDownlinkPowerLevel)	指由于下行链路信号电平过低引起的切换请求次数		GSMMeasurementType1
由于手机距 BTS 过远而引起的切换请求次数 (attHDOforsReqsForDistanceTooLongFromBTS)	指由于 MS 距 BTS 过远而引起的切换请求次数		GSMMeasurementType1
由 MSC 发出的切换请求 (找到更好的小区) 次数 (attHDOforsReqsForBetterCellFromMSC)	指由 MSC 发出的, 找到更好的小区的切换请求次数		GSMMeasurementType1

表 F.24 (续)

属性名	说明	触发点	类型
由于上行链路干扰而引起的小区切换请求次数 (attHDOReqsForUplinkDisturbance)	指由于上行链路干扰而引起的小区切换请求次数		GSMMeasurementType1
由于下行链路干扰而引起的小区切换请求次数 (attHDOReqsForDownlinkDisturbance)	指由于下行链路干扰而引起的小区切换请求次数		GSMMeasurementType1
由于功率算法选择更好的小区引起的切换请求次数 (attHDOReqsForBetterPowerSelectedCell)	指由于功率算法选择更好的小区引起的切换请求次数		GSMMeasurementType1
由于切换到雨伞小区而造成的切换请求次数 (attHDOReqsForUmbrellaCell)	由于切换到“雨伞”小区而造成的切换请求次数		GSMMeasurementType1
MSC 间切入尝试次数 (attIncomingExternalHDOsInterMSCInBTS)	指从不同的 MSC 小区切入该 MSC 小区的切换尝试次数		GSMMeasurementType1
MSC 间切入成功次数 (succIncomingExternalHDOsInterMSCInBTS)	指从不同的 MSC 小区切入该 MSC 小区的切换成功次数		GSMMeasurementType1
MSC 间切出尝试次数 (attOutgoingExternalHDOsInterMSCInBTS)	指从该 MSC 小区切出到不同的 MSC 小区的切换尝试次数		GSMMeasurementType1
MSC 间切出成功次数 (succOutgoingExternalHDOsInterMSCInBTS)	指从该 MSC 小区切出到不同的 MSC 小区的切换成功次数		GSMMeasurementType1
双频切换试呼总次数 (attDoubleFrequencyHDOs)	指双频切换的尝试次数 (在 900MHz 和 1800MHz 之间的切换)		GSMMeasurementType1
双频切换成功总次数 (succDoubleFrequencyHDOs)	指双频切换的成功次数 (在 900MHz 和 1800MHz 之间的切换)		GSMMeasurementType1

F.2.13.4 BTS 位置更新测量

BTS 位置更新测量的测量属性见表 F.25。

表 F.25 测量类型 btsLocationUpdate 的测量属性

属性名	说明	触发点	类型
再次寻呼的次数 (首次寻呼未应) (repagingReqsForMS)	指在 BTS 中, 对 MS 的再次寻呼次数 (首次寻呼未应)		GSMMeasurementType1
位置数据更新的次数 (attLocationUpdateInBTS)	指 BTS 中地址位置数据更新的尝试次数		GSMMeasurementType1
位置数据更新失败次数 (unsuccLocationUpdate)	指 BTS 中地址位置数据更新的成功次数		GSMMeasurementType1
周期性更新次数 (periodicLocationUpdate)	指 BTS 中, 位置数据的周期性更新次数		GSMMeasurementType1
IMSI 附着次数 (iMSIAttachments)	指 BTS 中 IMSI 的附着次数		GSMMeasurementType1

F.2.13.5 GPRS 无线资源测量

GPRS 无线资源测量的测量属性见表 F.26。

表 F.26 测量类型 btsGPRSRadioResource 的测量属性

属性名	说明	触发点	类型
成功抢占的 PDTCH 数 (succPDTCHSeizures)	指成功占用 PDTCH 的数目	收到 MS 从 PDTCH 发来的第一个 RLC 块 (PDU) 时计数 (GSM 04.60)	GSMMeasurementType1
可用的 PDCH 的平均数目 (meanNbrAvailablePDCH)	指一个测量时间内可用的 PDCH 数目 (包括实际正在使用) 的算术平均值	该测量项当 PDCH 可用时增加, 当 PDCH 不可用时, 该测量项减少	GSMMeasurementType1
可用的 PDCH 的最大数目 (maxNbrAvailablePDCH)	指一个测量周期中可用的 PDCH 的最大数目 (包括正在使用的), 即一个测量时间内测量到的可用 PDCH 数目的最大值		GSMMeasurementType1
可用的 PDCH 的最小数目 (minNbrAvailablePDCH)	该测量值是一个测量周期中可用的 PDCH 的最小数目 (包括正在使用的), 即一个测量时间内测量到的可用 PDCH 数目的最小值		GSMMeasurementType1
占用的 PDCH 的平均数目 (meanNbrOfOccPDCH)	指一个测量周期中占用的 PDCH 的算术平均值 (包括正在使用的), 即该测量值是占用的 PDCH 的算术平均值	通过预定义 (系统设计) 的时间间隔, 对承载分组业务的 PDCH 数进行取样, 然后进行算术平均 (GSM 04.60)	GSMMeasurementType1

表 F.26 (续)

属性名	说明	触发点	类型
占用的 PDCH 的最大数目 (maxNbrOfOccPDCH)	指被占用的 PDCH 最高记录值, 即抽样值的最大值	通过预定义 (系统设计) 的时间间隔, 对承载分组业务的 PDCH 数进行取样, 取最大值 (GSM 04.60)	GSMMeasurementType1
占用的 PDCH 的最小数目 (minNbrOfOccPDCH)	指被占用的 PDCH 最低记录值, 即抽样值的最小值	通过预定义 (系统设计) 的时间间隔, 对承载分组业务的 PDCH 数进行取样, 取最小值 (GSM 04.60)	GSMMeasurementType1
所有可用 PDCH 均被分配的时间 (availablePDCHAllocatedTime)	指在一个测量周期内所有可用 PDCH 均被分配的时间	该时间段从最后一个 PDCH 被分配的時刻到第一个 PDCH 被释放的時刻间的时间间隔来进行计算	GSMMeasurementType1
CS 业务抢占 PS 业务的次数 (succPdchToTch)	指一个在性能采集周期内, PDCH 成功转换成 TCH 的次数	在一个测量周期中的, 当一个 PDCH 成功转换成 TCH 时, 该测量项加一	GSMMeasurementType1

F.2.13.6 GPRS 分组业务测量

BTS GPRS 分组业务测量的测量属性见表 F.27。

表 F.27 测量类型 btsGPRSPacketService 的测量属性

属性名	说明	触发点	类型
PCCCH 发送的分组寻呼消息数 (nbrPacketPagingMessagesPCHOnPCCCH)	指在 PCCCH 上发送的分组寻呼消息的数目 (尝试次数)	发送“分组寻呼请求” (“PACKET PAGING REQUEST”) 消息 (GSM 04.60)	GSMMeasurementType1
PCCCH 上的平均 PPCH-PAGCH 队列长度 (meanPPCHPAGCHQueueLengthOnPCCCH)	指所有在 PCCCH 的 PPCH-PAGCH 子信道上等待传送的消息数的算术平均值	该测量值是通过按照一个预先定义的时间间隔 (系统定义) 对 PPCH-PAGCH 队列长度取样, 然后计算其算术平均 (GSM 04.60)	GSMMeasurementType2
PCCCH 上的 PPCH 队列中丢弃的分组寻呼消息数 (nbrOfPSPagesDiscardedFromPPCHQueueOnPCCCH)	指在 PCCCH 信道上发送前就从 PPCH 队列上被删除的分组寻呼消息数目	把寻呼消息从队列中删除有很多原因, 包括队列溢出、优先插入导致的队列溢出和队列计数器计数终止等 (GSM 04.60)	GSMMeasurementType1
在各种原因下的分组信道指配请求的尝试次数 (attPCReqAssPerCause)	该测量值是每个原因的分组信道指配请求次数	在接收到 MS 从 PRACH 信道发送的“分组信道请求” (“PACKET CHANNEL REQUEST”) 消息或者是从 RACH 信道上发送来的“信道请求” (“CHANNEL REQUEST”) 消息 注: 建立的原因包括信道请求 (GSM 04.08) 和分组信道请求 (GSM 04.60)	GSMMeasurementType3

表 F.27 (续)

属性名	说明	触发点	类型
在各种原因下的分组信道指配请求的成功次数 (succPDTCHAssProcsPerCause)	该测量值是每个原因下成功的分组信道指配 注：一个分组信道指配成功是指“PACKET UPLINK ASSIGNMENT”消息或“IMMEDIATE ASSIGNMENT COMMAND”消息被发送	在 AGCH 上发送“立即分配请求” (“IMMEDIATE ASSIGN COMMAND”) 消息，或在 PAGCH 上发送“分组上行分配” (“PACKET UPLINK ASSIGNMENT”) 消息 注：当消息中包含两个移动台的指配信息时，如果两个原因相同，则相应原因对应的计数器加 2；如果两个原因不同，则各自原因对应的计数器分别加 1	GSMMeasurementType3
PDTCH 的平均队列长度 (meanPacketQueueLength)	指所有在队列中等待从 PDTCH 上发送的消息个数的算术平均值	该值是按照预先定义的时间间隔（系统定义）对 PDTCH 队列长度进行抽样，然后计算算术平均	GSMMeasurementType2
服务升级\降级的次数，即 CS1\CS2 转换次数 (nbrOfServiceChanges)	随着服务的升级和降级，使用的编码方式（CS1\CS2）会转换，该测量值计算每个小区升级\降级的次数	相应对象进行升级或降级服务	GSMMeasurementType9

F.3 测量属性的 IDL 定义

本节给出了 PMHandler_v2 接口中的 startMeasurementJob_v2 操作所包含的测量属性的 IDL 定义。

```
// File: NmclfIMMeasurement.idl
// This file is based on interface information model.

#ifndef _NMC_IF_IM_MEASUREMENT_IDL_
#define _NMC_IF_IM_MEASUREMENT_IDL_
#include <NmclfIMCommType.idl>

module NmclfIMMeasurement
{
    enum mAttrIdType
    {
        // common measurement attributes
        // The following attributes belong to category: xxxProcessorLoad
        // processor load measurement
        meanUsage,
        peakUsage,
        peakTime,
        peakDuration,

        // signallingLinkSetTP measurement
        // The following attributes belong to category: signallingLinkSetTPTraffic
```

```

blockedSignallingLinks,
signallingLinkBlockedTimes,
signallingLinkSetTPTrafficVolume,

// for BSC measurement
// The following attributes belong to category: bscTrafficStatistics
// request for service
unsuccReqsForService,
// paging message in BSC
attTransOfPagingMessagesInBSC,
unsuccTransOfPagingMessagesInBSC,
// immediate assignment procedures in BSC
attImmediateAssignProcsInBSC,
succImmediateAssignProcsInBSC,
// total traffic
bscTotalTrafficVolume,
// for BSC-PCU traffic measurement
meanPSInterArrivalTime, //conditional
flushRequestReceived, //conditional

// The following attributes belong to category: bscHDOStatistics
// internal handovers intra cell in BSC
succInternalHDOsIntraCellInBSC,
unsuccInternalHDOsIntraCellInBSC,
// internal handovers inter cell in BSC
succInternalHDOsInBSC,
// internal handovers per cause
succInternalHDOsPerCause,
// internal handover failures in BSC
unsuccInternalHDOsWithReconnectionInBSC,
unsuccInternalHDOsWithLossOfConnectionInBSC,

// category : bscProcissorLoad, see common part definition

// The following attributes belong to category: bscGRPSSubLayerTraffic
// for GPRS RLC SubLayer measurement in BSC
nbrUmULBlock, //conditional
nbrUmDLBlock, //conditional
// for GPRS NS SubLayer measurement in BSC
nbrNsOctetSentBsc, //conditional
nbrNsOctetRvdBsc, //conditional
nbrNsPduSentBsc, //conditional
nbrNsPduRvdBsc, //conditional
nbrNsPduDiscardedBsc, //conditional

```

```

nbrNsResetMegSentBsc, //conditional
nbrNsResetMegRvdBsc, //conditional
nbrNsResetAckMegSentBsc, //conditional
nbrNsResetAckMegRvdBsc, //conditional
nbrNsBlockMegSentBsc, //conditional
nbrNsBlockMegRvdBsc, //conditional
nbrNsBlockAckMegSentBsc, //conditional
nbrNsBlockAckMegRvdBsc, //conditional
nbrNsUnblockMegSentBsc, //conditional
nbrNsUnblockMegRvdBsc, //conditional
nbrNsUnblockAckMegSentBsc, //conditional
nbrNsUnblockAckMegRvdBsc, //conditional

// for BTS measurement
// The following attributes belong to category: btsTrafficData
totalNumberOfSDCCH,
availableSDCCH,
attSDCCHSeizures,
blockedSDCCH,
droppedSDCCH,
blockedTCH,
attTCHSeizures,
succTCHSeizures,
droppedTCH,
totalTrafficVolumeOfSDCCH,
droppedSDCCHOnRF,
averageHoldTimeOfSDCCH,
cellBusyTimeOfSDCCH,
averageTimeAssignOfSDCCH,
availableTCH,
totalTrafficVolumeOfTCH,
numOfSuccAssignUnSeizure,
droppedTCHOnRF,
averageHoldTimeOfTCH,
cellBusyTimeOfTCH,
averageTimeAssignOfTCH,
maxNumOfSeizuredTCH,
availablePCH,
attTransOfPagingMessagesThePCH,
unsuccTransOfPagingMessagesThePCH,
attImmediateAssignProcsInBTS,
succImmediateAssignProcsInBTS,
numOfAvailRadioCarriers,
numOfAvailTrx,

```

meanNbrOfIdleTCHsPerInterferenceBand,
 nbrOfPagesDiscardedFromPCHQueue,
 meanDurationOfSuccPagingProcs,
 meanPCHAGCHQueueLength,

// The following attribute belong to category: powerLevelOfBS
 powerOfMobileStation,
 powerOfBTS,
 strengthOfUplink,
 strengthOfDownlink,
 qualityOfUplink,
 qualityOfDownlink,
 meanCallDistance,
 attReqsForPowerIncreaseFromMS,
 attReqsForPowerReduceFromMS,
 attInstructionsForPowerIncreaseFromBSC,
 attInstructionsForPowerReduceFromBSC,

// The following attribute belong to category: btsHDOStatistics
 succInternalHDOsIntraCellInBTS,
 unsuccInternalHDOsIntraCellInBTS,
 unsuccInternalHDOsForNoCHsIntraCellInBTS,
 unsuccInternalHDOsForSeizingNewCHIntraCellInBTS,
 unsuccInternalHDOsWithReconnectionIntraCellInBTS,
 succIncomingInternalHDOsIntraBSCInBTS,
 unsuccIncomingInternalHDOsIntraBSCInBTS,
 unsuccIncomingInternalHDOsForNoCHsIntraBSCInBTS,
 succOutgoingInternalHDOsIntraBscInBTS,
 unsuccOutgoingInternalHDOsIntraBSCInBTS,
 unsuccOutgoingInternalHDOsWithReconnectionIntraBSCInBTS,
 succIncomingExternalHDOsIntraMSCInBTS,
 unsuccIncomingExternalHDOsIntraMSCInBTS,
 unsuccIncomingExternalHDOsForNoCHsIntraMSCInBTS,
 unsuccIncomingHDOsForNotSeizingNewCHIntraMSCInBTS,
 succOutgoingExternalHDOsIntraMSCInBTS,
 unsuccOutgoingExternalHDOsIntraMSCInBTS,
 unsuccOutgoingExternalHDOsWithReconnectionIntraMSCInBTS,
 attHDOReqsForLowUplinkQuality,
 attHDOReqsForLowDownlinkQuality,
 attHDOReqsForLowUplinkPowerLevel,
 attHDOReqsForLowDownlinkPowerLevel,
 attHDOReqsForDistanceTooLongFromBTS,
 attHDOReqsForBetterCellFromMSC,
 attHDOReqsForUplinkDisturbance,

attHDOReqsForDownlinkDisturbance,
attHDOReqsForBetterPowerSelectedCell,
attHDOReqsForUmbrellaCell,
attIncomingExternalHDOsInterMSCInBTS,
succIncomingExternalHDOsInterMSCInBTS,
attOutgoingExternalHDOsInterMSCInBTS,
succOutgoingExternalHDOsInterMSCInBTS,
attDoubleFrequencyHDOs,
succDoubleFrequencyHDOs,

// The following attributes belong to category: btsLocationUpdate
repagingReqsForMS,
attLocationUpdateInBTS,
unsuccLocationUpdate,
periodicLocationUpdate,
iMSIAttachments,

// The following attributes belong to category: btsGPRSRadioResource
succPDTCHSeizures,
meanNbrAvailablePDCH,
maxNbrAvailablePDCH,
minNbrAvailablePDCH,
meanNbrOfOccPDCH,
maxNbrOfOccPDCH,
minNbrOfOccPDCH,
availablePDCHAllocatedTime,
succPdchToTch,

// The following attributes belong to category: btsGPRSPacketService
nbrPacketPagingMessagesPCHOnPCCCH,
meanPPCHPAGCHQueueLengthOnPCCCH,
nbrOfPSPagesDiscardedFromPPCHQueueOnPCCCH,
attPCReqAssPerCause,
succPDTCHAssProcsPerCause,
meanPacketQueueLength,
nbrOfServiceChanges,

// circuit end point subgroup measurement
// The following attributes belong to category: circuitEndpointSubGroup
numberOfAvailCircuits,
// incoming circuit end subgroup measurement
incomingBids,
incomingSeizures,
answeredIncomingSeizures,


```
incomingBidsOverflow,  
incomingTrafficVolume,  
// outgoing circuit end subgroup measurement  
outgoingBids,  
outgoingSeizures,  
answeredOutgoingSeizures,  
// overflow, has been defined above  
outgoingBidsOverflow,  
outgoingTrafficVolume,  
  
// for exchange measurement  
// The following attributes belong to category: trafficFlow  
// for traffic flow measurement  
// mobile originating call measurement  
attMobileOriginatingCalls,  
succMobileOriginatingCalls,  
ansMobileOriginatingCalls,  
failReasonForMobileOriginatingCalls,  
seizureTrafficVolumeForMobileOriginatingCalls,  
succTrafficVolumeForMobileOriginatingCalls,  
ansTrafficVolumeForMobileOriginatingCalls,  
// mobile terminating call measurement  
attMobileTerminatingCalls,  
succMobileTerminatingCalls,  
ansMobileTerminatingCalls,  
failReasonForMobileTerminatingCalls,  
seizureTrafficVolumeForMobileTerminatingCalls,  
succTrafficVolumeForMobileTerminatingCalls,  
ansTrafficVolumeForMobileTerminatingCalls,  
// incoming Call measurement  
attIncomingCalls,  
succIncomingCalls,  
ansIncomingCalls,  
failReasonForIncomingCalls,  
seizureTrafficVolumeForIncomingCalls,  
succTrafficVolumeForIncomingCalls,  
ansTrafficVolumeForIncomingCalls,  
// outgoing Call measurement  
attOutgoingCalls,  
succOutgoingCalls,  
ansOutgoingCalls,  
failReasonForOutgoingCalls,  
seizureTrafficVolumeForOutgoingCalls,  
succTrafficVolumeForOutgoingCalls,
```

```

ansTrafficVolumeForOutgoingCalls,
// local call measurement
attLocalCalls,
succLocalCalls,
ansLocalCalls,
failReasonForLocalCalls,
seizureTrafficVolumeForLocalCalls,
succTrafficVolumeForLocalCalls,
ansTrafficVolumeForLocalCalls,
// transit Call measurement
attTransitCalls,
succTransitCalls,
ansTransitCalls,
failReasonForTransitCalls,
seizureTrafficVolumeForTransitCalls,
succTrafficVolumeForTransitCalls,
ansTrafficVolumeForTransitCalls,
// incoming Terminating Call measurement
attIncomingTerminatingCalls,
succIncomingTerminatingCalls,
ansIncomingTerminatingCalls,
failReasonForIncomingTerminatingCalls,
seizureTrafficVolumeForIncomingTerminatingCalls,
succTrafficVolumeForIncomingTerminatingCalls,
ansTrafficVolumeForIncomingTerminatingCalls,
// originating outgoing call measurement
attOriginatingOutgoingCalls,
succOriginatingOutgoingCalls,
ansOriginatingOutgoingCalls,
failReasonForOriginatingOutgoingCalls,
seizureTrafficVolumeForOriginatingOutgoingCalls,
succTrafficVolumeForOriginatingOutgoingCalls,
ansTrafficVolumeForOriginatingOutgoingCalls,
// abnormal call measurement
callsBlockedByLoadShedding,
internalCongestionTrafficVolume,
blockedTrafficVolumeBecauseOfTrunkBusy,

// for MSC measurement
// The following attributes belong to category: mscPerformance
// incoming inter MSC handovers measurement
attIncomingInterMSCHDOs,
succIncomingInterMSCHDOs,
// outgoing inter MSC handovers measurement

```

```

attOutgoingInterMSCHDOs,
succOutgoingInterMSCHDOs,
// subsequent inter MSC handovers to MACa measurement
attSubsequentInterMSCHDOsMSCa,
succSubsequentInterMSCHDOsMSCa,
// subsequent Inter MSC Handovers To MACc measurement
attSubsequentInterMSCHDOsMSCc,
succSubsequentInterMSCHDOsMSCc,
// external handovers measurement
externalHDOs,
// external handovers per cause measurement
externalHDOsPerCause,
// external Handover Failure Per MSC measurement
unsuccExternHDOsWithReconnectionPerMSC,
unsuccExternHDOsWithLossOfConnectionPerMSC,
// MSC paging measurement
attSystemPaging,
// MSC fault measurement
numOfMSCFault,
falutTimeOfMSC,
// MSC end code measurement
numOfEndCode,
// MSC IMSI attach detach measurement
imsiDetachProcs,
imsiAttachProcs,
// interrogating HLR measurement
attInterrogationOfHLRsForRouting, //conditional
succInterrogationOfHLRsMSRNObtained, //conditional
succInterrogationOfHLRsCallForwarding, //conditional

// The following attributes belong to category: mscQos
meanTimeToCallSetupService,
meanTimeToLocationUpdateService,
meanCallDuration,
meanTrunkSeizureDuration,

// category : mscProcessorLoad, see common part definition.

// The following attributes belong to category : ssfTraffic
attIncomingCamelCallsToSSF, //conditional
attOutgoingCamelCallsFromSSF, //conditional
attCamelToCamelCalls, //conditional
answeredCamelToCamelCalls, //conditional
unsuccCamelCallsForSystemReason, //conditional

```

```

unsuccCamelCallsForSubscriberReason, //conditional
camelOriginatingTrafficVolume, //conditional
camelTerminatingTrafficVolume, //conditional
meanDurationOfCamelOriginatingCalls, //conditional
meanDurationOfCamelTerminatingCalls, //conditional
attCamelOriginatingCalls, //conditional
seizuredCamelOriginatingCalls, //conditional
connectedCamelOriginatingCalls, //conditional
answeredCamelOrigiantingCalls, //conditional
attCamelTerminatingCalls, //conditional
seizuredCamelTerminatingCalls, //conditional
connectedCamelTerminatingCalls, //conditional
answeredCamelTerminatingCalls, //conditional
seizuredCamelCalls, //conditional
connectedCamelCalls, //conditional
answeredCamelCalls, //conditional
answeredCamelCallsTrafficVolume, //conditional
connectedCamelCallsTrafficVolume, //conditional
seizuredCamelCallsTrafficVolume, //conditional
callingPartyEarlyReleaseCamelCalls, //conditional
callingPartyRinginReleaseCamelCalls, //conditional
calledPartyBusyCamelCalls, //conditional
calledPartyNoAnswerCamelCalls, //conditional
dp2, //conditional
dp4, //conditional
dp5, //conditional
dp6, //conditional
dp7, //conditional
dp9, //conditional
dp10, //conditional
dp12, //conditional
dp13, //conditional
dp14, //conditional
dp15, //conditional
dp17, //conditional
dp18, //conditional

// observed destination measurement
// The following attributes belong to category: observedDestination
bids,
// outgoingSeizures //This two attributes have been defined above;
// answeredOutgoingSeizures
noCircuitsAvailable,
seizureTrafficVolume,

```

```

answerTrafficVolume,

// The following attributes belong to category: handoverObservation
//incoming external inter MSC handovers per cell measurement
attIncomingInterMSCHDOsPerOriginatingCell,
succIncomingInterMSCHDOsPerOriginatingCell,
//outgoing external inter MSC handovers per cell measurement
attOutgoingInterMSCHDOsPerTargetCell,
succOutgoingInterMSCHDOsPerTargetCell,
// incoming external intra MSC handovers per cell measurement
attIncomingExternalIntraMSCHDOsPerOriginatingCell, //conditional
succIncomingExternalIntraMSCHDOsPerOriginatingCell, //conditional
// outgoing external intra MSC handovers per cell measurement
attOutgoingExternalIntraMSCHDOsPerTargetCell, //conditional
succOutgoingExternalIntraMSCHDOsPerTargetCell, //conditional

// for EIR measurement
// The following attributes belong to eirPerformanceStatistics
// received IMEI check request
nbrOfReceivedIMEICheckReqs,
// white answers in EIR
nbrOfWhiteAnsInEIR,
// grey answers in EIR
nbrOfGreyAnsInEIR,
// black answers in EIR
nbrOfBlackAnsInEIR,
// unknown IMEI answers in EIR
nbrOfUnknownIMEIAnsInEIR,

// category : eirProcessorLoad, see common definition

// for HLR measurement
// The following attributes belong to category: hlrPerformanceStatistics
// send alerts
attNbrOfSendAlerts,
succNbrOfSendAlerts,
// request for MSRN measurement
attReqForMSRN,
succReqForMSRN,
// authentication sets HLR to VLR measurement
attReqForAuthSetsReceivedByHLRFromVLRs,
succReturnedAuthSetsFromHLRToVLRs,
emptyResponsesForAuthSetsFromHLRToVLRs,
// insert subscriber data service measurement

```

```

attInsertSubDataService,
succInsertSubDataService,
// MS roaming outside HPLMN measurement
nbrOfCurrentMSsRoamingOutsideHLR,
// location update measurement
attLocationUpdate,
succLocationUpdate,
// SS related operations in HLR measurement
attSSRelatedOperationsInHLR, //conditional
succSSRelatedOperationsInHLR, //conditional
// GPRS users in HLR measurement
numOfGSMGPRSUsers, //conditional
numOfGPRSOnlyUsers, //conditional
numOfRoamingGPRSUsers, //conditional
// GPRS routing requests in HLR measurement
attGPRSRoutingReqsInHLR, //conditional
succGPRSRoutingReqsInHLR, //conditional

// category: hlrProcessorLoad, see common part definitions

// for SMS measurement
// The following attributes belong to category: smsPerformanceStatistics
// mobile originating short message forwarding measurement
attMobileOriginatingSMForwardings,
succMobileOriginatingSMForwardings,
// mobile terminating short message forwarding measurement
attMobileTerminatingSMForwardings,
succMobileTerminatingSMForwardings,

// category: smsProcessorLoad, see common part definition.

// for VLR measurement
// The following attribute belong to category: vlrPerformanceStatistics
// page request measurement
attPageReqs,
succPageReqs,
// authentication in VLR measurement
attAuthProcsInVLR,
succAuthProcsInVLR,
// intraVLR location update measurement
attIntraVLRLocationUpdates,
succIntraVLRLocationUpdates,
// interVLR location update measurement
attInterVLRLocationUpdates,

```

```

succInterVLRLocationUpdates,
// current subscribers in VLR measurement
nbrOfCurrentSubscriberInVLR,
// visitors from other PLMN measurement
nbrOfVisitorsInVLR,
// request and reply for MSRN measurement
attReqsForMSRNFromHLR,
unsuccRepsForMSRNToHLR,
// insert subscriber data service measurement
attInsertSubDataServiceFromHLR,
unsuccInsertSubDataServiceToHLR,
// identification request to PVLR measurement
attIdentificationReqToPVLRS,
succIdentificationReqToPVLRS,
// page requests per location area measurement
attPageReqsPerLocationArea,
succPageReqsPerLocationArea,
// authentication Sets VLR to HLR measurement
attReqForAuthSetsSentToHLR,
succReceivedAuthSetsFromHLR,
emptyResponsesForAuthFromHLR

// category: vlrProcessorLoad, see common part definition.
);

```

// The following type definitions are performance measurements:

```

typedef string GeneralizedTimeType;
typedef GeneralizedTimeType PeakTimeType;
typedef unsigned long CountType;
typedef unsigned long GSMMeasurementType1;
typedef float GSMMeasurementType2;

```

```

enum eCauseType
{

```

```

// Internal and External Handover causes (GSM 08.08)
uplinkQuality,
uplinkStrength,
downlinkQuality,
downlinkStrength,
distance,
betterCell,
operationAndMaintenanceIntervention,
directedRetry,

```

```

// valid for external handovers only:
responseToMscInvocation,

// Immediate assignment procedure causes (GSM 04.08 [2]):
emergencyCall,
callReEstablishment,
answerToPaging,
originatingCall,
locationUpdating,
otherProcedures,
reservedEstablishmentCause
};
struct CausePairType
{
    eCauseType cause;
    unsigned long value;
};
typedef sequence<CausePairType> GSMMeasurementType3;

enum eSSOperationType
{
    register,
    erase,
    activate,
    deactivate,
    registerPassword,
    interrogateSSOperation,
    processRequest
};
struct SSOperationPairType
{
    eSSOperationType ssOperaiton;
    unsigned long value;
};
typedef sequence<SSOperationPairType> GSMMeasurementType4;

struct GSMMeasurementType5
{
    unsigned long nbrOfCallLostBecauseOfTrunkUnavailable;
    unsigned long nbrOfCallLostBecauseOfExchangeCongested;
    unsigned long nbrOfAbnormalCalls;
};

```



```

struct LocationAreaMeasurementType
{
    NmcIfIMCommType::LocationAreaCodeType locationAreaCode;
    unsigned long value;
};
typedef sequence<LocationAreaMeasurementType> GSMMeasurementType6;

```

```

enum eEndCodeType
{
    longCallNoAnswer, //0
    calledPartyBusy, //1
    incompleteAddress, //2
    emptyNumber, //3
    timeoutRelease, //4
    equipmentCongestion, //5
    circuitCongestion, //6
    pagingNoResponse//7
};

```

```

struct EndCodeNumberType
{
    eEndCodeType endCode;
    unsigned long value;
};
typedef sequence<EndCodeNumberType> GSMMeasurementType7;

```

```

struct GSMMeasurementType8
{
    float channelsPerInterferenceBand1;
    float channelsPerInterferenceBand2;
    float channelsPerInterferenceBand3;
    float channelsPerInterferenceBand4;
    float channelsPerInterferenceBand5;
};

```

```

struct GSMMeasurementType9
{
    unsigned long numOfUpgrades;
    unsigned long numOfDowngrades;
};

```

```

struct NsSubLayerMeasurementType
{
    unsigned long nsvci;
};

```

YD/T 1279-2003

unsigned long value;

};

typedef sequence<NsSubLayerMeasurementType> CPRSMMeasurementType3;

};

#endif
