

YD

中华人民共和国通信行业标准

YD/T 1685.1-2007

800MHz CDMA 数字蜂窝移动通信网 网络管理接口技术要求 第 1 部分：智能网网管接口

800MHz CDMA Network Management Interface Technical Specification
Part 1: Intelligent Network Management Interface

2007-09-29 发布

2008-01-01 实施

中华人民共和国信息产业部 发布

目 次

前 言	II
1 范围	1
2 规范性引用文件	1
3 术语、定义与缩略语	1
4 网络管理体系结构	2
4.1 网络参考模型	2
4.2 接口位置	3
5 接口管理功能	3
6 接口分析	3
7 网络资源分析	3
7.1 配置网络资源模型	3
7.2 智能网设备性能数据分析	11
8 基于 CORBA 技术的网络资源模型设计	16
8.1 配置网络资源模型设计	16
8.2 性能网络资源模型设计	41
8.3 性能管理接口功能相关的文件	44

前 言

《800MHz CDMA 数字蜂窝移动通信网网络管理接口技术要求》分为两个部分：

1. YD/T 1685.1-2007 800MHz CDMA 数字蜂窝移动通信网网络管理接口技术要求 第1部分：智能网网管接口；

2. YD/T 1685.2-2007 800MHz CDMA 数字蜂窝移动通信网网络管理接口技术要求 第2部分：短信设备网管接口。

本部分是第1部分。

本部分由中国通信标准化协会提出并归口。

本部分起草单位：中兴通讯股份有限公司、中国联合通信有限公司、北京邮电大学

本部分主要起草人：郭雪峰、苏晨阳、徐 峰、王宝明、高柏峰、王 勇、腾佳欣、李文璟、芮兰兰、谢智刚

广东省网络空间安全协会受控资料

800MHz CDMA 数字蜂窝移动通信网网络管理接口技术要求

第 1 部分：智能网网管接口

1 范围

本部分规定了 800MHz CDMA 数字蜂窝移动通信网(以下简称 CDMA 通信网)中智能网系统向网络管理系统提供的网管接口。

本部分适用于 800MHz CDMA 数字蜂窝移动通信网的智能网设备网络管理接口。

2 规范性引用文件

下列文件中的条款通过本规范的引用而成为本部分的条款。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部分。然而，鼓励根据本部分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部分。

2GHz数字蜂窝移动通信网网络管理通用技术要求 第1部分 基本原则

2GHz数字蜂窝移动通信网网络管理通用技术要求 第2部分 接口功能

2GHz数字蜂窝移动通信网网络管理通用技术要求 第3部分 接口分析

2GHz数字蜂窝移动通信网网络管理通用技术要求 第4部分 基于CORBA技术的管理接口设计

2GHz cdma2000数字蜂窝移动通信网网络管理技术要求 第1部分 配置网络资源模型

2GHz cdma2000数字蜂窝移动通信网网络管理技术要求 第2部分 性能网络资源模型

2GHz cdma2000数字蜂窝移动通信网网络管理技术要求 第3部分 基于CORBA技术的网络资源模型

设计

3 缩略语

下列缩略语适用于本部分。

AC	Authentication Center	鉴权中心
CC	Cumulative Counter	累计数
CORBA	Common Object Request Broker Architecture	公共对象请求代理体系
DER	Discrete Event Registration	离散事件注册
GMSC	Gateway mobile-services switching center	移动关口局
HLR	Home Location Register	归属位置寄存器
IDL	Interface Definition Language	接口定义语言
IP	Intelligent Peripheral	智能外设
MAP	Mobile Application Part	移动应用部分
MSC	Mobile Switching Center	移动交换中心
NMC	Network Management Center	网管中心
PPC	Prepaid Charging	预付费

SCEP	Service Creation Environment Point	业务生成环境点
SCF	Service Control Function	业务控制功能
SCP	Service Control Point	业务控制点
SDF	Service Data Function	业务数据功能
SI	Status Inspection	状态观察
SMAF	Service Management Access Function	业务管理接入功能
SMAP	Service Management Access Point	业务管理接入点
SMF	Service Management Function	业务管理功能
SMP	Service Management Point	业务管理点
SRF	Special Resource Function	专用资源功能
SSF	Service Switching Point	业务交换功能
SSP	Service Switching Point	业务交换点
VC	Voucher Center	充值中心
VLR	Visit Location Register	拜访位置寄存器
VPN	Virtual Private Network	虚拟专网

4 网络管理体系结构

4.1 网络参考模型

4.1.1 CDMA 智能网网络参考模型

CDMA智能网网络参考模型如图1所示。

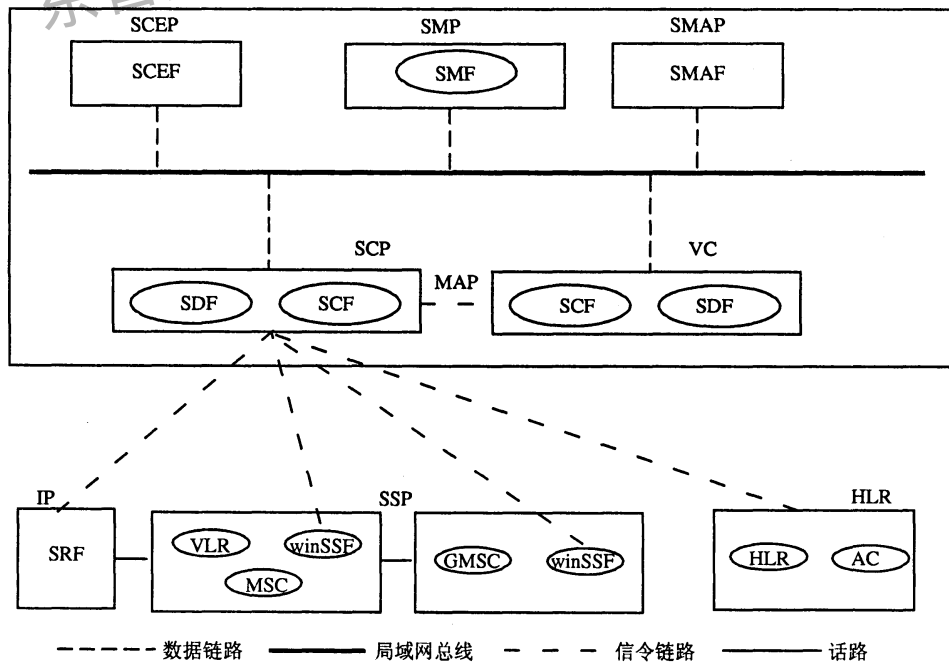


图 1 CDMA 智能网网络参考模型示意图

图1中，智能网设备包括SMP、SCP、VC、SCEP等设备，各设备又可以由多个逻辑模块或物理实体模块组成，例如SCP可以由SCF、SDF等模块组成，这些模块可以是逻辑上的，也可以是独立的物理实体。

各网络单元之间的通信使用TCP/IP协议组成局域网。

智能网系统对外和CDMA通信网基础设施HLR、MSC/SSP、GMSC/SSP以及IP之间通过信令网连接，采用MAP协议进行通信。

4.2 接口位置

本规范定义的接口为智能网系统与网管系统之间的接口，如图2所示。

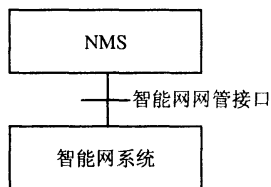


图2 DMA通信网智能网网管接口位置

智能网网管接口为智能网系统向网络管理系统提供的管理接口，通过该接口智能网系统向网络管理系统提供关于资源配置信息、告警信息及性能测量信息的上报，并接受网络管理系统的管理功能。

补充说明：智能网系统提供接口的方式，有可能是直接设备客户端接口，或者通过归属OMC提供接口。

5 接口管理功能

参见《第三代移动通信网网络管理技术要求 通用部分 接口功能》。

6 接口分析

参见《第三代移动通信网网络管理技术要求 通用部分 接口分析》。

7 网络资源分析

7.1 配置网络资源模型

7.1.1 通用配置网络资源对象类分析

在配置管理中，涉及到以下的网络资源管理对象类：

- (1) Top
- (2) SubNetwork
- (3) ManagedElement
- (4) ManagedFunction
- (5) SignallingPoint
- (6) SignallingLinkSetTP
- (7) SignallingLinkTP
- (8) ManagementNode
- (9) InventoryUnit
- (10) VsDataContainer

这些管理对象类的定义参见《第三代移动通信网网络管理技术要求 cdma2000部分 配置网络资源模型》。

7.1.2 智能网设备配置网络资源分析

7.1.2.1 概述

这部分管理对象类是智能网设备特有的配置网络资源对象类。

7.1.2.2 智能网设备网络资源对象类图

智能网设备配置网络资源对象类包含关系如图 3 所示。

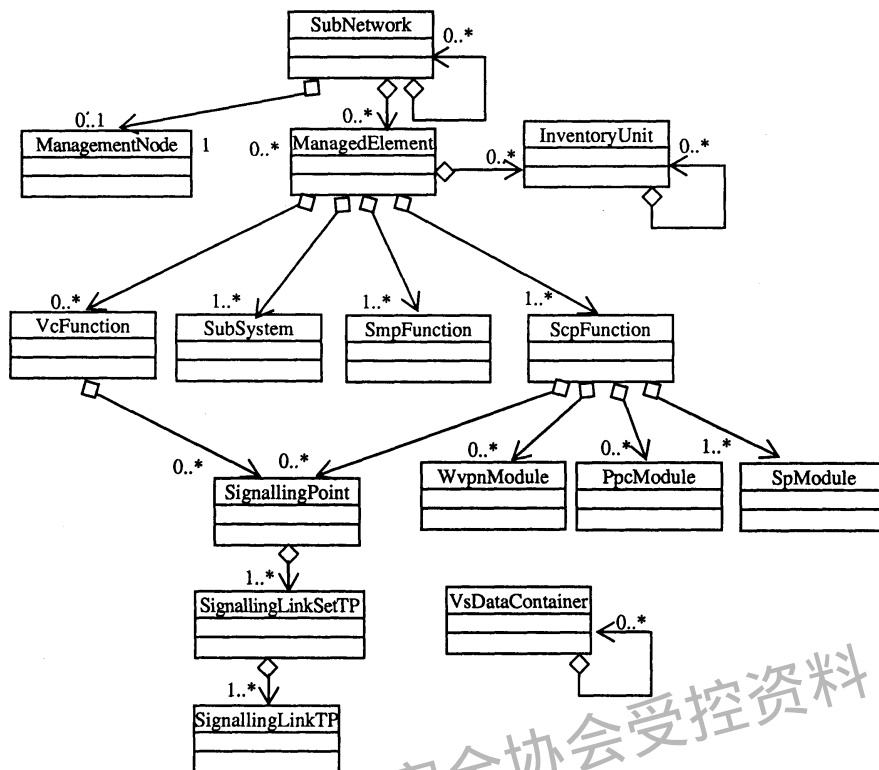


图 3 智能网设备配置网络资源对象包含关系图

说明：

- (1) 所有从 SPMODULE 中继承的***Module 可以挂在 SCPFunction 下。
- (2) 在智能网部分，可不实例化 InventoryUnit 对象，仅实例化子系统对象。

智能网设备配置网络资源对象类继承关系如图 4 所示。

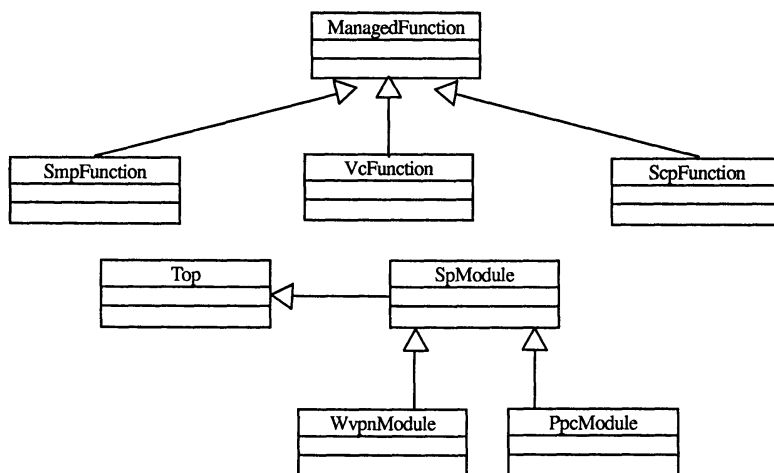


图 4 智能网设备配置网络资源对象继承关系图

说明：SPMOUDULE为其他业务父类，从TOP继承。

7.1.2.3 智能网设备网络资源对象

7.1.2.3.1 SCP 功能-ScpFunction

7.1.2.3.1.1 被管对象类描述

ScpFunction完成业务控制点SCP的所有逻辑功能。该对象从ManagedFunction对象继承而来。

7.1.2.3.1.2 属性描述

ScpFunction属性描述见表1。

表1 ScpFunction的信息

条 目	中文名称	说 明	类型和 取值说明	限 定
scpFunctionId	SCP 标识符	Scpfunction 对象的命名属性	字符串	M,R
workingMode	工作模式	网元的工作方式, 可取值为: 0: 单机工作方式; 1: 主从工作方式; 2: 集群工作方式	整型	M,R
workingState	工作状态	只在当工作方式为主从工作时有效, 0: 主用; 1: 备用	整型	C,R
maxCaps	最大 CAPS	SCP 系统处理能力——设计的最大 CAPS 数	整型	M,R
coverArea	覆盖地区	SCP 覆盖的省份	字符串	M,R/W
relatedSmp	归属 SMP	该 SCP 所属的 SMP 的网元标识(DN)	字符串	M,R/W
usageState	使用状态	表明网元是否在用 0: 不用; 1: 在用	整型	M,R
funcList	SCP 上驻留的功能实体	SCP 上驻留的功能实体序列, 如 SMF、SCF、SSF、SRF 等	集合, 字符串的列表	M,R
relatedOmc	归属 OMC	管理此 SCP 的 OMC 的标识符(DN), 若没有 OMC, 则该属性值为空。	字符串	O, R
totalMachine	集群方式下包含的机器总数	该属性只在 SCP 工作方式为集群方式下存在	整数	O, R
maxNumSignLink2M	设计的 2Mbit/s 链路数	设计的 SCP 最大可支持的 2Mbit/s 信令链路数	整数	M, R
maxNumSignLink	设计的 64kbit/s 链路数	设计的 SCP 最大可支持的 64kbit/s 信令链路数	整数	M, R
numSignLink	SCP 当前已配置的 64kbit/s 信令链路数	SCP 当前已配置的 64kbit/s 信令链路数	整数	M, R
numSignLink2M	SCP 当前已配置的 2Mbit/s 信令链路数	SCP 当前已配置的 2Mbit/s 信令链路数	整数	M, R
serviceList	支持的智能网业务列表	SCP 所支持的智能网业务的列表	字符串的序列	M, R
maxBhca	设计的忙时 BHCA 值	SCP 的处理能力——设计的最大 BHCA 值	整数	M, R
putIntoUseTime	投入使用时间	设备投入使用的时间	时间	M, R

7.1.2.3.1.3 可发送的通知描述

ScpFunction可发送的通知描述见表2。

表2 ScpFunction可发送通知

中文名称	英文名称	限 定
对象创建通知	notifyObjectCreation	O
对象删除通知	notifyObjectDeletion	O
对象属性值改变通知	notifyAttributeValueChange	O
告警状态改变通知	notifyAckStateChanged	O
变化的告警通知	notifyChangedAlarm	C
清除的告警通知	notifyClearedAlarm	M
新的告警通知	notifyNewAlarm	M

7.1.2.3.2 SCP 信令点功能

7.1.2.3.2.1 被管对象类描述

SCP信令点信息由三个对象类来表示：信令点（SignallingPoint）、信令链路集终端点（SignallingLinkSetTP）、信令链路终端点（SignallingLinkTP）。这些类具有通用性，具体定义参见《第三代移动通信网网络管理技术要求 cdma2000部分 配置网络资源模型》。

7.1.2.3.3 SCP 业务模块功能-SpModule

7.1.2.3.3.1 被管对象类描述

SpModule对象完成CDMA移动智能网节点中业务处理模块的所有逻辑功能，该对象不实例化，被其他具体业务模块对象继承。该对象从Top对象继承而来。

7.1.2.3.3.2 属性描述

SpModule属性描述见表3。

表3 SpModule信息

条 目	中文名称	说 明	类型和取值说明	限 定
serviceName	业务名称	该模块所支持的业务名称。	字符串	M,R
serviceDesc	业务描述	对该模块所支持的业务详细描述	字符串	M,R
maxNumUsers	最大支持用户数	可支持的该业务的最大用户数	整型	M,R
userTraffic	每用户话务量	设计的该业务每用户的话务量	浮点型	M,R
maxCaps	设计每秒试呼数	设计的该业务的最大CAPS值	整型	M,R

7.1.2.3.3.3 可发送的通知描述

SpModule可发送的通知描述见表4。

表4 SpModule可发送通知

中文名称	英文名称	限定
对象创建通知	NotifyObjectCreation	O
对象删除通知	NotifyObjectDeletion	O
对象属性值改变通知	notifyAttributeValueChange	O
告警状态改变通知	NotifyAckStateChanged	O
变化的告警通知	NotifyChangedAlarm	C
清除的告警通知	notifyClearedAlarm	M
新的告警通知	NotifyNewAlarm	M

7.1.2.3.4 PPC 业务模块功能-PpcModule

7.1.2.3.4.1 被管对象类描述

PpcModule为PPC业务处理模块，完成PPC业务处理功能，该对象从SpModule对象继承而来。

7.1.2.3.4.2 属性描述

PpcModule属性描述见表5。

表5 PpcModule的信息

条 目	中文名称	说 明	类型和取值说明	限 定
ppcModuleId	命名属性	该对象的命名属性，移动智能业务标识，与其业务键的取值相同	字符串	M,R

7.1.2.3.4.3 可发送的通知描述

PpcModule可发送的通知描述见表6。

表6 PpcModule可发送通知

中文名称	英文名称	限定
对象创建通知	notifyObjectCreation	O
对象删除通知	notifyObjectDeletion	O
对象属性值改变通知	notifyAttributeValueChange	O
告警状态改变通知	notifyAckStateChanged	O
变化的告警通知	notifyChangedAlarm	C
清除的告警通知	notifyClearedAlarm	M
新的告警通知	notifyNewAlarm	M

7.1.2.3.5 WVPN 业务模块功能-WvpnModule

7.1.2.3.5.1 被管对象类描述

WvpnModule为WVPN业务处理模块，完成wvpn业务处理功能，该对象从SpModule对象继承而来。

7.1.2.3.5.2 属性描述

WvpnModule属性描述见表7。

表7 WvpnModule的信息

条目	中文名称	说明	类型和取值说明	限定
wvpnModuleId	命名属性	该对象的命名属性，移动智能业务标识，与其业务键的取值相同	字符串	M,R

7.1.2.3.5.3 可发送的通知描述

WvpnModule可发送的通知描述见表8。

表8 WvpnModule可发送通知

中文名称	英文名称	限定
对象创建通知	notifyObjectCreation	O
对象删除通知	notifyObjectDeletion	O
对象属性值改变通知	notifyAttributeValueChange	O
告警状态改变通知	notifyAckStateChanged	O
变化的告警通知	notifyChangedAlarm	C
清除的告警通知	notifyClearedAlarm	M
新的告警通知	notifyNewAlarm	M

7.1.2.3.6 VC 功能——VcFunction

7.1.2.3.6.1 被管对象类描述

VcFunction完成充值中心的所有逻辑功能。该对象从ManagedFunction对象继承而来。

7.1.2.3.6.2 网元属性描述

VcFunction网元属性描述见表9。

表9 VcFunction信息

条 目	中文名称	说 明	类型和取值说明	限 定
vcFunctionId	VcFunction 对象标识符	VcFunction 对象的命名属性	字符串	M,R
maxNumCard	最大卡容量	网元设计支持的最大卡容量	整型	M,R
workingMode	工作模式	网元的工作方式, 可取值为: 0: 单机工作方式; 1: 主从工作方式; 2: 集群工作方式	整型	M,R
workingState	工作状态	只有当工作方式为主从方式时有效, 0: 主用; 1: 备用	整型	M,R
coverArea	覆盖地区	VC/SDP 覆盖的省份	字符串	M,R/W
relatedSmp	归属 SMP	该 VC 所属的 SMP 的网元标识(DN)	字符串	M,R/W
usageState	使用状态	表明网元是否在用, 0 表示不用, 1 表示在用)	整型	M,R
relatedOmc	归属 OMC	管理此 SCP 的 OMC 的标识符(DN), 若没有 OMC, 则该属性值为空	字符串	O, R
totalMachine	集群方式下包含的机器总数	该属性只在 SCP 工作方式为集群方式下存在	整数	O, R
numSignLink	SCP 当前已配置的 64kbit/s 信令链路数	SCP 当前已配置的 64kbit/s 信令链路数	整数	M, R
numSignLink2M	SCP 当前已配置的 2Mbit/s 信令链路数	SCP 当前已配置的 2Mbit/s 信令链路数	整数	M, R
maxNumSignLink2M	设计的 2Mbit/s 链路数	设计的 SCP 最大可支持的 2Mbit/s 信令链路数	整数	M, R
maxNumSignLink	设计的 64kbit/s 链路数	设计的 SCP 最大可支持的 64kbit/s 信令链路数	整数	M, R
backupVc	备份的 VC	该属性只在 VC 做主用时存在。表示为该 VC 做备份的 VC 的标识符(DN)	字符串	MC, R
putIntoUseTime	投入使用时间	设备投入使用的时间	时间	M, R

7.1.2.3.6.3 可发送的通知描述

VcFunction可发送的通知描述见表10。

表10 VcFunction 可发送通知

中文名称	英文名称	限 定
对象创建通知	notifyObjectCreation	O
对象删除通知	notifyObjectDeletion	O
对象属性值改变通知	notifyAttributeValueChange	O
告警状态改变通知	notifyAckStateChanged	O
变化的告警通知	notifyChangedAlarm	C
清除的告警通知	notifyClearedAlarm	M
新的告警通知	notifyNewAlarm	M

7.1.2.3.7 VC 信令点功能

7.1.2.3.7.1 被管对象类描述

VC 信令点功能由 3 个对象类来表示：信令点（SignallingPoint）、信令链路集终端点（SignallingLinkSetTP）、信令链路终端点（SignallingLinkTP）。这些类具有通用性，具体定义参见《第三代移动通信网网络管理技术要求 cdma2000 部分 配置网络资源模型》。

7.1.2.3.8 SMP 功能-SmpFunction

7.1.2.3.8.1 被管对象类描述

SmpFunction完成业务管理点SMP的所有逻辑功能。该对象从ManagedFunction对象继承而来。

7.1.2.3.8.2 网元属性描述

SmpFunction网元属性描述见表11。

表11 SmpFunction信息

条 目	中文名称	说 明	类型和取值说明	限 定
smpFunctionId	SMP 标识符	SmpFunction 对象的命名属性	字符串	M,R
workingMode	工作模式	网元的工作方式，可取值为：0：单机工作方式；1：主从工作方式；2：集群工作方式	整型	M,R
workingState	工作状态	当工作方式为主从方式时有效，取值只能为 0 或 1； 0：主用；1：备用	整型	M,R
maxNumSmap	最大支持 SMAP 终端数目	SMP 最多可连接的 SMAP 终端数	整型	M,R
usageState	使用状态	表明网元是否在用	整型（0 表示不用，1 表示在用）	M,R
coverArea	覆盖范围	SMP 覆盖的省份	字符串	M,R/W
relatedSepList	所管理的 SCP 列表	所管理的 SCP 标识符的列表（DN 的列表）	DN 的列表	M, R
relatedOmc	归属 OMC	管理此 SMP 的 OMC 的标识符(DN)，若没有 OMC，则该属性值为空	字符串（255）	O, R
totalMachine	机器个数	该属性只在 SMP 工作方式为集群方式下存在	整数	MC, R
putIntoUseTime	投入使用时间	设备投入使用的时间	时间	M, R

7.1.2.3.8.3 可发送的通知描述

SmpFunction可发送的通知描述见表12。

表12 SmpFunction可发送通知

中文名称	英文名称	限定
对象创建通知	notifyObjectCreation	O
对象删除通知	notifyObjectDeletion	O
对象属性值改变通知	notifyAttributeValueChange	O
告警状态改变通知	notifyAckStateChanged	O
变化的告警通知	notifyChangedAlarm	C
清除的告警通知	notifyClearedAlarm	M
新的告警通知	notifyNewAlarm	M

7.1.2.3.9 子系统功能-SubSystem

7.1.2.3.9.1 被管对象类描述

SubSystem对象表示CDMA移动智能网中各节点（如SMP，SCP等）所包括的一些独立设施，如SMP和SCP单机工作方式下或集群工作方式下的一台计算机等[注]。该对象从Top对象继承而来。

注：子系统对象表示了智能网实体的物理配置，但由于所要求的配置属性在通用对象InventoryUnit中没有相应的表示，因此单独定义了子系统对象。在智能网部分，可不实例化InventoryUnit对象，仅实例化子系统对象。

7.1.2.3.9.2 网元属性描述

SubSystem网元属性描述见表13。

表13 SubSystem属性

英文名称	中文名称	说明	类型	限定
subSystemId	命名属性	SubSystem对象的命名属性	字符串	M, R
subsysModel	子系统型号	子系统型号	字符串 (31)	M, R
subsysState	子系统工作状态	0: 正常工作 1: 异常 2: 未投入使用	整数	M, R
subsysDesc	子系统描述	其他必要的描述信息	字符串 (255)	M, R
cpuInfoList	子系统 CPU 信息	CPU 信息包括: CPU 标识和 CPU 主频	结构的列表, 列表中的每个项为结构, 定义为: { cpu_id CPU 标识 (整数), cpu_frequence CPU 主频 (整数 MHz) }	M, R
memInfoList	子系统的内存信息	内存信息包括: 内存标识和内存容量	结构的列表, 列表中的每个项为结构, 定义为: { memory_id 内存标识 (整数), memory_size 内存容量 (整数 MB) }	M, R
diskInfoList	子系统的硬盘信息	硬盘信息包括: 硬盘标识和硬盘容量	结构的列表, 列表中的每个项为结构, 定义为: { disk_id 硬盘标识 (整数), disk_size 硬盘容量 (整数 MB) }	M, R
ioPortInfoList	子系统的 I/O 端口信息	I/O 端口信息包括: I/O 端口标识、名称、 类型、速率和状态	结构的列表, 列表中的每个项为结构, 定义为: { ioport_id I/O 端口标识 (整数), ioport_name 端口名称 (字符串), ioport_type 端口类型 (整数) [注 1], ioport_speed 端口速率 (整数 Kbps), ioport_state 端口状态 (整数) [注 2]}	M, R
osInfoList	子系统的操作系统信息	操作系统信息包括: 操作系统标识、名称和 版本	结构的列表, 列表中的每个项为结构, 定义为: { os_id 操作系统标识 (整数), os_name 操作系统名称 (字符串 31), os_version 操作系统版本 (字符串 31) }	M, R

表13 (续)

英文名称	中文名称	说 明	类 型	限 定
appInfoList	子系统的应用软件/平台信息	应用软件/平台信息包括： 应用软件/平台标识、名称、版本和开发商	结构的列表，列表中的每个项为结构，定义为： { app_id 应用软件/平台标识（整数）， app_name 应用软件/平台名称（字符串 31）， app_version 应用软件/平台版本（字符串 31）， app_vendor 应用软件/平台开发商（字符串 31）}	M, R
dbInfoList	子系统的数据库信息	数据库信息包括： 数据库标识、名称和版本	结构的列表，列表中的每个项为结构，定义为： { db_id 数据库标识（整数）， db_name 数据库名称（字符串 31）， db_version 数据库版本（字符串 31）， db_size 数据库容量（整数 MB）}	O, R
locationName	位置名称	子系统设备所在地名称 (细化到机房)	字符串	M, R/W

7.2 智能网设备性能数据分析

7.2.1 SCP 性能测量数据

7.2.1.1 信令链路性能测量数据

参见《第三代移动通信网网络管理技术要求 cdma2000部分 性能网络资源模型》。

7.2.1.2 SCP 整体话务统计

相关的配置对象类：ScpFunction。

SCP整体话务属性见表14。

表14 SCP整体话务属性 (scpMeas)

性能测量项	描 述	触发点	类 型	采集方式
智能试呼次数 numCallsAtt	SCF 在性能采集周期内收到的智能试呼次数	SCP 接收到 SSP 的 TC_BEGIN 消息	整型	CC
由于过负荷被丢弃的智能呼叫次数 numCallsDiscardOverload	由于过负荷控制而丢弃、没有被处理的智能呼叫次数	SCP 在处理过程中，过负荷	整型	CC
由于协议错误被丢弃的智能呼叫次数 numCallsDiscardErr	由于协议错误而引起智能呼叫处理失败的次数	SCP 在处理过程中，发生协议错误	整型	CC
主叫话务量 callingTraffic	由于智能业务用户作主叫而引起的智能呼叫话务量	所有因智能网用户作为主叫所产生的话务量	整型	DER
被叫话务量 calledTraffic	由于智能业务用户作被叫而发出的智能呼叫话务量	所有因智能网用户作为被叫所产生的话务量	整型	DER
主叫应答次数 numCallingResp	主叫成功应答的次数	主叫流程中收到 O_Answer 应答消息	整型	CC
被叫应答次数 numCalledResp	被叫成功接续的次数	被叫流程中收到 T_Answer 应答消息	整型	CC

表14 (续)

性能测量项	描述	触发点	类型	采集方式
主叫通话完成次数 numCallingFinish	主叫成功完成通话的次数	主叫流程中收到 O_Disconnect 挂机消息	整型	CC
被叫通话完成次数 numCalledFinish	被叫成功完成通话的次数	被叫流程中收到 T_Disconnect 挂机消息	整型	CC
实际每秒试呼数 scpCaps	实际的 CAPS 值	SCP 收到的所有试呼次数	整型	CC
生成的话单数 numRecord	性能统计周期内, SCP 生成的话 单总数目	SCP 成功生成话单	整型	CC
创建失败的话单数 numRecordFailed	性能统计周期内, SCP 创建失败 的话单总数目	SCP 未成功生成话单	整型	CC
零时长的话单数 numRecord0	性能统计周期内, SCP 生成的零 时长的话单总数目	SCP 成功生成话单, 话单中通 话时长为零	整型	CC
超时消息的个数 numMsgOvertime	性能统计周期内, SCP 超时未处 理的操作个数	SCP 在处理过程中, 等待 SSP 或其他网元响应超时	整型	CC
解码错误的个数 numErrDecode	性能统计周期内, SCP 解码错误 的操作个数	SCP 在处理过程汇总, 发生解 码错误	整型	CC
主机处理的总消息个数 numMsgTotal	SCP 处理的总信令消息个数	SCP 在处理过程中, 收到来自 其他网元的总消息数目	整型	CC
SCP 实际 BHCA 值 scpBhca	SCP 实际产生的 BHCA 值	SCP 收到的所有试呼次数	整型	DER

7.2.1.3 SCP 智能业务性能测量数据

本部分定义了 SCP 智能业务的统计信息, 包括预付费业务 (PPC) 和虚拟专网业务 (WVPN)。

7.2.1.3.1 预付费业务统计

相关的配置对象类: PpcModule。

预付费业务性能属性见表15。

表15 预付费业务性能属性 (ppcMeas)

性能测量项	描述	触发点	类型	采集方式
当前 PPC 业务用户数 numUsers	统计周期结束时, PPC 用户总个数	无	整型	GAUGE
新增用户数 numUsersAdd	统计周期内, 新开户并激活的用户个数	无	整型	GAUGE
使用期用户数 numUsersUse	统计周期结束时, 处于有效期状态用户的个数	无	整型	GAUGE
充值期用户数 numUsersHold	统计周期结束时, 处于充值期状态用户的个数	无	整型	GAUGE
锁定期用户数 numUsersLocked	统计周期结束时, 处于锁定期状态用户的个数	无	整型	GAUGE
冷冻期用户数 numUsersFrosen	统计周期结束时, 处于冷冻期状态用户的个数	无	整型	GAUGE
黑名单用户数 numUsersBlack	统计周期结束时, 处于黑名单状态用户的个数	无	整型	GAUGE

表15 (续)

性能测量项	描述	触发点	类型	采集方式
挂失用户数 numUsersLoss	统计周期结束时, 处于挂失状态用户个数	无	整型	GAUGE
尚未头次使用用户数 numUsersUnused	统计周期结束时, 处于尚未使用状态的用户个数	无	整型	GAUGE
已回收用户数 numUsersRecy	统计周期内, 回收的用户个数	无		GAUGE
总呼叫次数 numCalls	统计周期内, PPC 用户总呼叫次数	PPC 业务收到的所有试呼次数	整型	CC
总呼叫占用时长 durCalls	统计周期内, PPC 用户总呼叫占用时长	无	整型	CC
主叫应答次数 numOAnswer	PPC 用户作为主叫的总应答次数	PPC 业务收到 OANSWER 消息, 操作码为 82, 触发器取值为 40	整型	CC
被叫应答次数 numTAnswer	PPC 用户作为被叫的总应答次数	PPC 业务收到 TANSWER 消息, 操作码为 85, 触发器取值为 69	整型	CC
点到点短信条数 numSmsP2P	PPC 用户使用点到点短信业务的总条数	PPC 业务收到 PDSCP 点到点短信鉴权消息	整型	CC
联通在信短信条数 numSmsOnline	PPC 用户使用联通在信业务的总条数	PPC 业务收到 PDSCP 联通在信鉴权消息	整型	CC
实时扣费短信费用 feeSmsReal	PPC 用户使用短信业务产生的总费用 (实时扣费)	成功从用户账号上扣除短信业务费用 (实时扣费)	整型	CC
实时扣费短信费用 feeSmsNoReal	PPC 用户使用短信业务产生的总费用 (非实时扣费)	成功从用户账号上扣除短信业务费用 (实时扣费)	整型	CC
数据业务次数 numData	PPC 用户使用数据业务的总次数	PPC 业务收到的数据业务上网次数 (非实时扣费)	整型	CC
数据业务费用 FeeDataReal	PPC 用户使用数据业务产生的总费用 (实时扣费)	成功从用户账号上扣除数据业务使用费用 (实时扣费)	整型	CC
数据业务费用 FeeDataNoReal	PPC 用户使用数据业务产生的总费用 (非实时扣费)	成功从用户账号上扣除数据业务使用费用 (非实时扣费)	整型	CC
PPC 沉淀话费 depositFee	进入冷冻期用户的沉淀话费总数	用户进入冷冻期余额大于零	整型	CC
numFailPerCause	按照原因统计的业务逻辑处理呼叫失败次数, PPC 呼叫失败原因包括: (1) 鉴权未通过, 被拒绝 (如停机、停呼入、挂失、不在有效期、余额不足、漫游未允许、呼出限制、呼入限制、来话筛选、呼出筛选、密码鉴权等因权限不够而被拒绝)。 (2) 接续不成功 (如用户忙、无应答、路由不可及等情况, 由于 C 网 SSP 不向 SCP 上报忙、无应答、路由不可及消息, 因此 C 网 PPC 业务暂不包括此类失败统计)。 (3) 管理流程——充值、挂失、解挂等错误。 (4) 其他错误	(1) PPC 响应鉴权不成功 (2) PPC 指示呼叫接续, 但未收到应答消息 (3) 管理流程处理不成功	结构型, 结构中的每项为整数	CC

7.2.1.3.2 WVPN 业务统计

相关的配置对象类：WvpnModule。

WVPN业务性能属性见表16。

表16 WVPN业务性能属性 (wvpnMeas)

性能测量项	描述	触发点	类型	采集方式
VPN 集团总数 numGroup	VPN 业务当前总集团个数	无	整型	GAUGE
新增 VPN 集团个数 numGroupAdd	统计周期内, 新增加的集团总个数	无	整型	GAUGE
删除 VPN 集团个数 numGroupDel	统计周期内, 删除的集团总个数	无	整型	GAUGE
VPN 用户总数 numUsers	统计周期结束时, VPN 用户总数目	无	整型	GAUGE
新增 VPN 用户数 numUsersAdd	统计周期内, 新增加的 VPN 用户数目	无	整型	GAUGE
最大 VPN 集团的成员数 maxMembers	当前最大 VPN 集团的成员数	无	整形	GAUGE
VPN 用户中 G 网用户总数 numUsersGsm	当前 VPN 总用户中 GSM 网络用户总数	无	整形	GAUGE
VPN 用户中 PBX 网用户总数 numUsersPbx	当前 VPN 总用户中 PBX 用户总数	无	整形	GAUGE
删除的 VPN 用户数 numUsersDel	统计时间段内删除的 VPN 用户数	无	整形	GAUGE
VPN 用户总试呼叫次数 numCalls	统计周期内, VPN 用户的总呼叫次数	VPN 业务收到或发出的所有呼叫次数[注]	整型	CC
VPN 用户总呼叫占用时长 durCalls	统计时间段内, VPN 用户总呼叫占用时长	统计时间段内, VPN 用户总呼叫占用时长[注]	整型	CC
网内呼叫次数 numCallsIntra	统计周期内, VPN 用户的集团内呼叫次数	VPN 业务收到或发出呼叫的类型为集团内呼叫[注]	整型	CC
网内呼叫时长 durCallsIntra	统计周期内, VPN 用户的集团内呼叫时长	集团内呼叫发生的通话时长[注]	整型	CC
闭合用户群呼叫次数 numCallsClosed	统计周期内, VPN 用户的闭合用户群总呼叫次数	VPN 业务收到或发出呼叫的类型为闭合用户群呼叫[注]	整型	CC
闭合用户群呼叫总时长 durCallsClosed	统计周期内, VPN 用户的闭合用户群总呼叫时长	闭合用户群内用户的通话时长[注]	整型	CC
按原因统计的业务处理呼叫失败次数 numFailPerCause	统计周期内, 按原因统计的业务处理呼叫失败次数, 或者总失败次数	WVPN 业务返回处理失败消息, 指示呼叫释放	结构型, 结构中的每项为整数	CC

[注]: 同一集团内/闭合用户群内的用户间发生的呼叫/通话只算一次呼叫/通话。

7.2.2 VC/SDP 性能测量数据

7.2.2.1 信令链路性能测量数据

参见《第三代移动通信网网络管理技术要求 cdma2000 部分 性能网络资源模型》。

7.2.2.2 VC/SDP 的充值统计

相关的配置对象类：VcFunction。

VC/SCP充值统计的性能属性见表17。

表17 VC性能属性 (vcMeas)

性能测量项	描述	触发点	类型	采集方式
当前卡容量 numCards	VC 当前系统的卡容量	无	整型	CC
智能试呼次数 numCallsAtt	VC 在性能采集周期内收到的智能试呼次数	VC 接收到 SEARCH 请求消息	整型	CC
由于过负荷被丢弃的智能呼叫次数 numCallsDiscardOverload	由于过负荷控制而丢弃、没有被处理的智能呼叫次数	VC 在处理过程中，过负荷	整型	CC
由于协议错误被丢弃的智能呼叫次数 numCallsDiscardErr	由于协议错误而引起智能呼叫处理失败的次数	VC 在处理过程中，发生协议错误	整型	CC
充值成功的次数 numRechargeSucc	SCP 到 VC/SDP 的充值成功的次数	VC 成功更改充值卡状态为已使用	整型	CC
成功充值金额数 feeRecharge	成功充值金额数	VC 成功充值的总金额	整型	CC
充值失败的次数 numRechargeFail	SCP 到 VC/SDP 的充值失败的次数	VC 返回充值失败	整型	CC
按原因统计的 VC 充值失败的次数 numRechargeFailPerCause	统计时间段内，按不同原因统计的 VC 充值请求失败的次数，原因分别为： 0: 充值卡鉴权未通过 1: 充值卡已使用 2: 充值卡已超出有效期 3: 充值卡已被系统封锁 4: 充值卡不存在	VC 返回各种充值失败原因给 SCP	结构型，结构中的每项为整数	CC

7.2.3 SMP 性能数据

7.2.3.1 SMP 整体性能统计

相关的配置对象类：SmpFunction。

SMP整体性能统计属性见表18。

表18 SMP性能属性 (smpMeas)

性能测量项	描述	触发点	类型	采集方式
接入的 SMAP 数量 numAccessSMAP	实际接入的 SMAP 数量	无	整型	GAUGE
发送给 SCP 操作数 numToScpAtt	统计时间段内发送给 SCP 的操作数	发送给 SCP 操作	整型	CC
接收的 SCP 操作响应数 numFromScpResult	统计时间段内接收的来自 SCP 的操作结果数	收到来自 SCP 的操作	整型	CC
SMP 接收的 SCP 发送的操作错误数 numFromScpErr	统计时间段内收到的 SCP 操作错误数	判定 SCP 发送操作错误	整型	CC

7.2.4 子系统性能测量数据

相关配置对象类：SubSystem

子系统性能测量数据见表19。

表19 子系统性能测量数据(subSystemMeas)

英文名称	中文名称	说明	数据类型	单位	限定
cpuAveLoad	CPU 平均负荷	CPU 在每个性能统计周期内的平均负荷（占用率）	浮点数	百分比，取百分号前的数字	M
diskSpareSize	磁盘剩余空间	磁盘剩余空间	浮点数	MB	M
diskIoPer	磁盘 I/O 占用率	读写磁盘的频度，用于描述磁盘的繁忙程度	浮点数	百分比，取百分号前的数字	M
dbSpareSize	数据库剩余空间	数据库剩余空间	浮点数	MB	M
memSpareSize	内存剩余空间	内存剩余空间	浮点数	MB	M

8 基于 CORBA 技术的网络资源模型设计

8.1 配置网络资源模型设计

注：配置网络资源模型设计中有3类idl文件,这3类文档及其用途如下：

1) xxxNRMDefs.idl,包括GenericNRMDefs.idl、IMDataDefs.idl和 INNRMDefs.idl, 用来定义配置网络对象及其属性名称；

2) xxxNRMSystem.idl,包括GenericNRMSystem.idl和INNRMSystem.idl, 用来定义配置网络资源对象的属性使用的数据类型；

3) xxxNRMProfile.idl, 包括GenericNRMProfile.idl、IMDataProfile.idl和INNRMProfile.idl, 只是用来描述配置网络资源对象的属性名称及其数据类型的对应关系，实现时并不使用此类idl文件。

8.1.1 通用配置资源模型的 IDL 定义

8.1.1.1 GenericNRMDefs

```
//File "GenericNRMDefs.idl"
//The IRP document version number is "GENERIC NRM V1.0"
#ifndef GenericNRMDefs_idl
#define GenericNRMDefs_idl

// #pragma prefix "3gppsa5.org"

//This module defines constants for each MO class name and
//the attribute names for each Generic MO class.

module GenericNRMDefs
{
    //Definitions for abstract MO class Top

    interface Top
    {
        const string ObjectClass = "ObjectClass";
    }
}
```

```
    const string ObjectInstance = "ObjectInstance";
};

//Definitions for MO class IRPAgent

interface IRPAgent: Top
{
    const string CLASS = "IRPAgent";

    // Attribute Names
    //
    const string iRPAgentId = "iRPAgentId";
    const string systemDN = "systemDN";
};

//Definitions for abstract MO class GenericIRP

interface GenericIRP: Top
{
    const string CLASS = "GenericIRP";

    // Attribute Names
    //
    const string iRPId = "iRPId";
};

//Definitions for MO class SubNetwork

interface SubNetwork: Top
{
    const string CLASS = "SubNetwork";

    // Attribute Names
    //
    const string subNetworkId = "subNetworkId";
    const string dnPrefix = "dnPrefix";
    const string userLabel = "userLabel";
    const string setOfMcc = "setOfMcc";
    const string userDefinedNetworkType = "userDefinedNetworkType ";
};

//Definitions for MO class MeContext
```

```
interface MeContext: Top
{
    const string CLASS = "MeContext";

    // Attribute Names
    //
    const string meContextId = "meContextId";
    const string dnPrefix = "dnPrefix";
};

//Definitions for MO class ManagementNode

interface ManagementNode: Top
{
    const string CLASS = "ManagementNode";

    // Attribute Names
    //
    const string managementNodeId = "managementNodeId";
    const string managedElements = "managedElements";
    const string userLabel = "userLabel";
    const string userDefinedState = "userDefinedState";
    const string swVersion = "swVersion";
    const string locationName = "locationName";
    const string vendorName = "vendorName";
};

//Definitions for MO class ManagedElement

interface ManagedElement: Top
{
    const string CLASS = "ManagedElement";

    // Attribute Names
    //
    const string managedElementId = "managedElementId";
    const string dnPrefix = "dnPrefix";
    const string userLabel = "userLabel";
    const string vendorName = "vendorName";
    const string locationName = "locationName";
    const string managedElementType = "managedElementType";
    const string managedBy = "managedBy";
    const string userDefinedState = "userDefinedState";
};
```

```
    const string swVersion = "swVersion";
};

//Definitions for abstract MO class ManagedFunction

interface ManagedFunction : Top
{
    const string CLASS = "ManagedFunction";

    // Attribute Names
    //
    const string userLabel = "userLabel";
};

//Definitions for MO class VsDataContainer

interface VsDataContainer: Top
{
    const string CLASS = "VsDataContainer";

    //Attribute Names
    //
    const string vsDataContainerId = "vsDataContainerId";
    const string vsDataType = "vsDataType";
    const string vsData = "vsData";
    const string vsDataFormatVersion = "vsDataFormatVersion";
};

//Defination for MO class InventoryUnit

interface InventoryUnit: Top
{
    const string CLASS = "InventoryUnit";

    //Attribute Names
    //
    const string inventoryUnitId = "inventoryUnitId";
    const string inventoryUnitType = "inventoryUnitType";
    const string vendorUnitFamilyType = "vendorUnitFamilyType";
    const string vendorUnitTypeNumber = "vendorUnitTypeNumber";
    const string vendorName = "vendorName";
    const string serialNumber = "serialNumber";
    const string versionNumber = "versionNumber";
};
```

```
const string dateOfManufacture = "dateOfManufacture";
const string dateOfLastService = "dateOfLastService";
const string unitPosition = "unitPosition";
const string manufacturerData = "manufacturerData";
};

//Definitions for MO class SignallingPoint

interface SignallingPoint: Top
{
    const string CLASS = "SignallingPoint";

    // Attribute Names
    //
    const string signallingPointId = "signallingPointId";
    const string signallingInfo = "signallingInfo";
    const string signallingPointType = "signallingPointType";
    const string userLabel = "userLabel";
};

interface SignallingLinkSetTP: Top
{
    const string CLASS = "SignallingLinkSetTP";

    // Attribute Names
    //
    const string signallingLinkSetTPId = "signallingLinkSetTPId";
    const string adjacentSignallingInfo = "adjacentSignallingInfo";
    const string userLabel = "userLabel";
    const string signallingLinkType = "signallingLinkType";
};

interface SignallingLinkTP: Top
{
    const string CLASS = "SignallingLinkTP";

    // Attribute Names
    //
    const string signallingLinkTPId = "signallingLinkTPId";
    const string userLabel = "userLabel";
    const string slc = "slc";
    const string slsNormalList = "slsNormalList";
    const string slsCurrentList = "slsCurrentList";
```

```

        const string linkStatus = "linkStatus";
        const string bandwidth = "bandwidth";
    };

};
#endif

```

8.1.1.2 GenericNRMPProfile

```

//File "GenericNRMPProfile.idl"
//The IRP document version number is "GENERIC NRM V1.0"
#ifndef GenericNRMPProfile_idl
#define GenericNRMPProfile_idl

#include "GenericNRMSystem.idl"

#pragma prefix "3gppsa5.org"

/**
 * This module defines the attribute names and
 * correspondig attribute types for all defined
 * MO class. This module is used for reference.
 */
module GenericNRMPProfile
{
    interface Top
    {
        readonly attribute string objectClass;
        readonly attribute string objectInstance;
    };

    interface IRPAgent : Top
    {
        readonly attribute GenericNRMSystem::ObjectIdType iRPAgentId;
        readonly attribute GenericNRMSystem::DN systemDN;

        // The following notifications may be sent from this MO,
        // notifyObjectCreation
        // notifyObjectDeletion
        // notifyAttributeValueChange
        // notifyAckStateChanged
        // notifyChangedAlarm
        // notifyClearedAlarm
        // notifyNewAlarm

```



```

        // notifyComments
        // notifyAlarmListRebuilt
        // notifyPotentialFaultyAlarmList
    };

interface GenericIRP : Top
{
    readonly attribute string iRPIId;
};

interface SubNetwork: Top
{
    readonly attribute GenericNRMSystem::ObjectIdType subNetworkId;
    readonly attribute GenericNRMSystem::DNPrefixType dnPrefix;
        attribute wstring userLabel;
    readonly attribute GenericNRMSystem::MobileCountryCodeSetType setOfMcc;
    readonly attribute GenericNRMSystem::NetworkTypeType userDefinedNetworkType;

    // The following notifications may be sent from this MO,
    // notifyObjectCreation
    // notifyObjectDeletion
    // notifyAttributeValueChange
    // notifyAckStateChanged
    // notifyChangedAlarm
    // notifyClearedAlarm
    // notifyNewAlarm
    // notifyComments
    // notifyAlarmListRebuilt
    // notifyPotentialFaultyAlarmList
};

interface MeContext: Top
{
    readonly attribute GenericNRMSystem::ObjectIdType meContextId;
    readonly attribute GenericNRMSystem::DNPrefixType dnPrefix;

    // The following notifications may be sent from this MO,
    // notifyObjectCreation
    // notifyObjectDeletion
    // notifyAttributeValueChange
};

```

```

interface ManagementNode : Top
{
    readonly attribute GenericNRMSystem::ObjectIdType managementNodeId;
    readonly attribute GenericNRMSystem::DNListType managedElements;
        attribute wstring userLabel;
        attribute GenericNRMSystem::UserDefinedStateType userDefinedState;
    readonly attribute string swVersion; // software version
        attribute wstring locationName;
    readonly attribute string vendorName;

    // The following notifications may be sent from this MO,
        // notifyObjectCreation
        // notifyObjectDeletion
        // notifyAttributeValueChange
        // notifyAckStateChanged
        // notifyChangedAlarm
        // notifyClearedAlarm
        // notifyNewAlarm
        // notifyComments
        // notifyAlarmListRebuilt
        // notifyPotentialFaultyAlarmList
};

interface ManagedElement : Top
{
    readonly attribute GenericNRMSystem::ObjectIdType managedElementId;
    readonly attribute GenericNRMSystem::DNPrefixType dnPrefix;
        attribute wstring userLabel;
    readonly attribute string vendorName;
        attribute wstring locationName;
    readonly attribute GenericNRMSystem::StringSet managedElementType;
    readonly attribute GenericNRMSystem::DN managedBy;
        attribute GenericNRMSystem::UserDefinedStateType userDefinedState;
    readonly attribute string swVersion; // software version

    // The following notifications may be sent from this MO,
        // notifyObjectCreation
        // notifyObjectDeletion
        // notifyAttributeValueChange
        // notifyAckStateChanged
        // notifyChangedAlarm
        // notifyClearedAlarm

```

```

        // notifyNewAlarm
        // notifyComments
        // notifyAlarmListRebuilt
        // notifyPotentialFaultyAlarmList
};

interface ManagedFunction : Top
{
    attribute wstring userLabel;
};

interface VsDataContainer : Top
{
    readonly attribute GenericNRMSystem::ObjectIdType vsDataContainerId;
    readonly attribute string vsDataType;
        attribute any vsData;
    readonly attribute string vsDataFormatVersion;

    // The following notifications may be sent from this MO,
        // notifyObjectCreation
        // notifyObjectDeletion
        // notifyAttributeValueChange
        // notifyAckStateChanged
        // notifyChangedAlarm
        // notifyClearedAlarm
        // notifyNewAlarm
        // notifyComments
        // notifyAlarmListRebuilt
        // notifyPotentialFaultyAlarmList
};

interface InventoryUnit : Top
{
    readonly attribute GenericNRMSystem::ObjectIdType inventoryUnitId;
    readonly attribute string inventoryUnitType;
    readonly attribute string vendorUnitFamilyType;
    readonly attribute string vendorUnitTypeNumber;
    readonly attribute string vendorName;
    readonly attribute string serialNumber;
    readonly attribute string versionNumber;
    readonly attribute string dateOfManufacture;
    readonly attribute string dateOfLastService;
    readonly attribute wstring unitPosition;
};

```

```

readonly attribute string manufacturerData;

// The following notifications may be sent from this MO,
// notifyObjectCreation
// notifyObjectDeletion
// notifyAttributeValueChange
// notifyAckStateChanged
// notifyChangedAlarm
// notifyClearedAlarm
// notifyNewAlarm
// notifyComments
// notifyAlarmListRebuilt
// notifyPotentialFaultyAlarmList
};

interface SignallingPoint : Top
{
    readonly attribute GenericNRMSystem::ObjectIdType signallingPointId;
    readonly attribute GenericNRMSystem::SignallingInfoType signallingInfo;
    readonly attribute GenericNRMSystem::SignallingPointType signallingPointType;
    attribute wstring userLabel;

// The following notifications may be sent from this MO,
// notifyObjectCreation
// notifyObjectDeletion
// notifyAttributeValueChange
// notifyAckStateChanged
// notifyChangedAlarm
// notifyClearedAlarm
// notifyNewAlarm
// notifyComments
// notifyAlarmListRebuilt
// notifyPotentialFaultyAlarmList
};

interface SignallingLinkSetTP : Top
{
    readonly attribute GenericNRMSystem::ObjectIdType signallingLinkSetTPId;
    attribute GenericNRMSystem::SignallingInfoType adjacentSignallingInfo;
    attribute wstring userLabel;
    readonly attribute GenericNRMSystem::SignallingLinkTypeType signallingLinkType;

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

```

```

        // notifyObjectCreation
        // notifyObjectDeletion
        // notifyAttributeValueChange
        // notifyAckStateChanged
        // notifyChangedAlarm
        // notifyClearedAlarm
        // notifyNewAlarm
        // notifyComments
        // notifyAlarmListRebuilt
        // notifyPotentialFaultyAlarmList
};

interface SignallingLinkTP : Top
{
    readonly attribute GenericNRMSystem::ObjectIdType signallingLinkTPId;
    attribute wstring userLabel;
    readonly attribute GenericNRMSystem::SlcType slc;
    readonly attribute GenericNRMSystem::SLSListType slsNormalList;
    readonly attribute GenericNRMSystem::SLSListType slsCurrentList;
    readonly attribute GenericNRMSystem::LinkStatusType linkStatus;
    readonly attribute GenericNRMSystem::BandwidthType bandwidth;

    // The following notifications may be sent from this MO,
    // notifyObjectCreation
    // notifyObjectDeletion
    // notifyAttributeValueChange
    // notifyAckStateChanged
    // notifyChangedAlarm
    // notifyClearedAlarm
    // notifyNewAlarm
    // notifyComments
    // notifyAlarmListRebuilt
    // notifyPotentialFaultyAlarmList
};
};
#endif

```

8.1.1.3 GenericNRMSystem

```

//File "GenericNRMSystem.idl"
//The IRP document version number is "Generic NRM V1.0"
#ifndef GenericNRMSystem_idl
#define GenericNRMSystem_idl

```

```

module GenericNRMSystem
{
    /**
     * This module adds datatype definitions for types
     * used in the Generic NRM which are not basic datatypes defined
     * already in CORBA.
     */

    /**
     * The format of Distinguished Name (DN) is specified in "Name Conventions
     * for Managed Objects revision B".
     */
    typedef string DN;

    typedef sequence<DN> DNListType;

    typedef string ObjectIdType;

    typedef DN DNPrefixType;

    typedef string MobileCountryCodeType;
    typedef string ISDNAddrStringType;
    typedef sequence<MobileCountryCodeType> MobileCountryCodeSetType;

    typedef string NetworkTypeType;
    const NetworkTypeType AN = "Access Netowrk";
    const NetworkTypeType CN = "Core Netowrk";
    const NetworkTypeType AN_CN = "AN and CN";

    typedef unsigned long UserDefinedStateType;

    /**
     * A set of strings.
     */
    typedef sequence<string> StringSet;
    typedef sequence <unsigned long> ULongSet;

    enum NetworkIndicatorType
    {
        international,
        spare,
        national,
        nationalSpare
    }
}

```

```
};

enum SignallingPointLengthType
{
    bits_24,
    bits_14
};

struct SignallingInfoType
{
    SignallingPointLengthType signallingPointLength;
    unsigned long signallingPointCode;
    NetworkIndicatorType networkIndicator;
};

typedef unsigned long SignallingPointType;
const SignallingPointType SP=0;
const SignallingPointType HSTP=1;
const SignallingPointType LSTP=2;
const SignallingPointType HLSTP=3;

enum SignallingLinkTypeType
{
    N_SS7,
    W_SS7
};

typedef unsigned long linkDirectType;
const linkDirectType toHstp=1;
const linkDirectType toLstp=2;
const linkDirectType toCdmaGmsc=3;
const linkDirectType toMsc=4;
const linkDirectType toHlr=5;
const linkDirectType toMc=6;
const linkDirectType toScp=7;
const linkDirectType toBsc=8;
const linkDirectType toCncPstn=9;
const linkDirectType toCtPstn=10;
const linkDirectType toCmcc=11;
const linkDirectType toCtt=12;
const linkDirectType toVoiceMailBox=13;
const linkDirectType toColorRing=14;
const linkDirectType toGsm=15;
```

```

const linkDirectType toOthers=16;

typedef unsigned short SlcType;

typedef unsigned short SLSType;
typedef sequence<SLSType> SLSListType;

typedef unsigned short LinkStatusType;
const LinkStatusType normal_UDS = 0;
const LinkStatusType deactivated_UDS = 1;
const LinkStatusType failed_UDS = 2;
const LinkStatusType localBlocked_UDS = 3;
const LinkStatusType remoteBlocked_UDS = 4;
const LinkStatusType localInhibited_UDS = 5;
const LinkStatusType remoteInhibited_UDS = 6;

typedef unsigned long BandwidthType;
};

#endif

```

8.1.1.4 IMDataDefs

```

//File "IMDataDefs.idl"
//The IRP document version number is "Inventory NRM V1.0"
#ifndef IMDataDefs_idl
#define IMDataDefs_idl

#include "GenericNRMDefs.idl"

/**
 * This module defines constants for each MO class name and
 * the attribute names for each defined MO class.
 */
module IMDataDefs
{
    /**
     * Definitions for MO class InventoryUnit
     */
    interface InventoryUnit:GenericNRMDefs::Top
    {
        const string CLASS = "InventoryUnit";

        //Attribute Names

```



```

//
const string inventoryUnitId = "inventoryUnitId";
const string inventoryUnitType = "inventoryUnitType";
const string vendorUnitFamilyType = "vendorUnitFamilyType";
const string vendorUnitTypeNumber = "vendorUnitTypeNumber";
const string vendorName = "vendorName";
const string serialNumber = "serialNumber";
const string versionNumber = "versionNumber";
const string dateOfManufacture = "dateOfManufacture";
const string dateOfLastService="dateOfLastService";
const string unitPosition = "unitPosition";
const string manufacturerData= "manufacturerData";

};

};

#endif

```

8.1.1.5 IMDataProfile

```

//File "IMDataProfile.idl"
//The IRP document version number is "Inventory NRM V1.0"
#ifndef IMDataProfile_idl
#define IMDataProfile_idl

#include "GenericNRMSystem.idl"
#include "GenericNRMPProfile.idl"

module IMDataProfile
{
    interface InventoryUnit:GenericNRMPProfile::Top
    {
        readonly attribute GenericNRMSystem::ObjectIdType inventoryUnitId;
        readonly attribute string inventoryUnitType ;
        readonly attribute string vendorUnitFamilyType;
        readonly attribute string vendorUnitTypeNumber;
        readonly attribute string vendorName;
        readonly attribute string serialNumber;
        readonly attribute string versionNumber;
        readonly attribute string dateOfManufacture;
        readonly attribute string dateOfLastService;
        readonly attribute wstring unitPosition;
        readonly attribute string manufacturerData;

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

```

```

        // notifyObjectCreation
        // notifyObjectDeletion
        // notifyAttributeValueChange
        // notifyAckStateChanged
        // notifyChangedAlarm
        // notifyClearedAlarm
        // notifyNewAlarm
        // notifyComments
        // notifyAlarmListRebuilt
        // notifyPotentialFaultyAlarmList
    };
};

#endif

```

8.1.2 移动智能网网络资源模型的 IDL 定义

8.1.2.1 INNRMDefs

```

//File "INNRMDefs.idl"
//The IRP document version number is "IN NRM V1.0"
#ifndef INNRMDefs_idl
#define INNRMDefs_idl

#include <GenericNRMDefs.idl>

/**
 * This module defines constants for each MO class name and
 * the attribute names for each defined MO class.
 */
module INNRMDefs
{
    /**
     * Definitions for MO class ScpFunction
     */
    interface ScpFunction : GenericNRMDefs::ManagedFunction
    {
        const string CLASS = "ScpFunction";

        // Attribute Names
        //
        const string scpFunctionId = "scpFunctionId";
        const string maxCaps = "maxCaps";
        const string maxBhca = "maxBhca";
        const string funcList = "funcList";
    };
};

```

```
const string serviceList = "serviceList";
const string maxNumSignLink = "maxNumSignLink";
const string maxNumSignLink2M = "maxNumSignLink2M";
const string numSignLink = "numSignLink";
const string numSignLink2M = "numSignLink2M";
const string putIntoUseTime = "putIntoUseTime";
const string workingMode = "workingMode";
const string workingState = "workingState";
const string usageState = "usageState";
const string totalMachine = "totalMachine";
const string relatedSmp = "relatedSmp";
const string relatedOmc = "relatedOmc";
const string coverArea = "coverArea";
};

/**
 * Definitions for MO class SmpFunction
 */
interface SmpFunction : GenericNRMDefs::ManagedFunction
{
    const string CLASS = "SmpFunction";

    // Attribute Names
    const string smpFunctionId = "smpFunctionId";
    const string maxNumSmap = "maxNumSmap";
    const string putIntoUseTime = "putIntoUseTime";
    const string workingMode = "workingMode";
    const string workingState = "workingState";
    const string usageState = "usageState";
    const string totalMachine = "totalMachine";
    const string relatedScpList = "relatedScpList";
    const string relatedOmc = "relatedOmc";
    const string coverArea = "coverArea";
};

/**
 * Definitions for MO class VcFunction
 */
interface VcFunction : GenericNRMDefs::ManagedFunction
{
    const string CLASS = "VcFunction";
```

```

// Attribute Names
//
const string vcFunctionId = "vcFunctionId";
const string maxNumCard = "maxNumCard";
const string maxNumSignLink = "maxNumSignLink";
const string maxNumSignLink2M = "maxNumSignLink2M";
const string numSignLink = "numSignLink";
const string numSignLink2M = "numSignLink2M";
const string putIntoUseTime = "putIntoUseTime";
const string workingMode = "workingMode";
const string workingState = "workingState";
const string backupVc = "backupVc";
const string usageState = "usageState";
const string totalMachine = "totalMachine";
const string relatedSmp = "relatedSmp";
const string relatedOmc = "relatedOmc";
const string coverArea = "coverArea";
};

```

```

/**
 * Definitions for MO class SpModule
 */
interface SpModule : GenericNRMDefs::Top
{

```

```

    const string CLASS = "SpModule";

    // Attribute Names
    //
    const string serviceName = "serviceName";
    const string serviceDesc = "serviceDesc";
    const string maxNumUsers = "maxNumUsers";
    const string userTraffic = "userTraffic";
    const string maxCaps = "maxCaps";
};

```

```

/**
 * Definitions for MO class WvpnModule
 */
interface WvpnModule : INNRMDefs::SpModule
{

```

```
const string CLASS = "WvpnModule";

// Attribute Names
//
const string wvpnModuleId = "wvpnModuleId";
};

/**
 * Definitions for MO class PpcModule
 */
interface PpcModule : INNRMDefs::SpModule
{
    const string CLASS = "PpcModule";

    // Attribute Names
    //
    const string PpcModuleId = "PpcModuleId";
};

/**
 * Definitions for MO class SubSystem
 */
interface SubSystem : GenericNRMDDefs::Top
{
    const string CLASS = "SubSystem";

    // Attribute Names
    //
    const string subSystemId = "subSystemId";
    const string locationName = "locationName";
    const string subsysModel = "subsysModel";
    const string subsysState = "subsysState";
    const string subsysDesc = "subsysDesc";
    const string cpuInfoList = "cpuInfoList";
    const string memInfoList = "memInfoList";
    const string diskInfoList = "diskInfoList";
    const string ioPortInfoList = "ioPortInfoList";
    const string osInfoList = "osInfoList";
    const string appInfoList = "appInfoList";
    const string dbInfoList = "dbInfoList";
};
```

```
};

#endif
```

8.1.2.2 INNRMProfile

```
//File "INNRMProfile.idl"
//The IRP document version number is "IN NRM V1.0"
#ifndef INNRMProfile_idl
#define INNRMProfile_idl

#include <GenericNRMSystem.idl>
#include <GenericNRMProfile.idl>
#include <INNRMSystem.idl>
#include <TimeBase.idl>

/**
 * This module defines the attribute names and
 * correspondig attribute types for all defined
 * MO class in IN network. This module is
 * used for reference.
 */
module INNRMProfile
{
    interface ScpFunction : GenericNRMProfile::ManagedFunction
    {
        readonly attribute GenericNRMSystem::ObjectIdType scpFunctionId;
        readonly attribute unsigned long maxCaps;
        readonly attribute unsigned long maxBhca;
        readonly attribute INNRMSystem::funcListType funcList;
        readonly attribute INNRMSystem::serviceListType serviceList;
        readonly attribute unsigned long maxNumSignLink;
        readonly attribute unsigned long maxNumSignLink2M;
        readonly attribute unsigned long numSignLink;
        readonly attribute unsigned long numSignLink2M;
        readonly attribute INNRMSystem::putIntoUseTimeType putIntoUseTime;
        readonly attribute INNRMSystem::workingModeType workingMode;
        readonly attribute INNRMSystem::workingStateType workingState;
        readonly attribute INNRMSystem::usageStateType usageState;
        readonly attribute unsigned long totalMachine;
        readonly attribute string relatedSmp;
        readonly attribute string relatedOmc;
        readonly attribute string coverArea;
```

```

// The following notifications may be sent from this MO,
// notifyObjectCreation
// notifyObjectDeletion
// notifyAttributeValueChange
// notifyAckStateChanged
// notifyChangedAlarm
// notifyClearedAlarm
// notifyNewAlarm

};

interface SmpFunction : GenericNRMProfile::ManagedFunction
{
    readonly attribute GenericNRMSystem::ObjectIdType smpFunctionId;
    readonly attribute unsigned long maxNumSmap;
    readonly attribute INNRMSystem::putIntoUseTimeType putIntoUseTime;
    readonly attribute INNRMSystem::workingModeType workingMode;
    readonly attribute INNRMSystem::workingStateType workingState;
    readonly attribute INNRMSystem::usageStateType usageState;
    readonly attribute unsigned long totalMachine;
    readonly attribute GenericNRMSystem::DNListType relatedScpList;
    readonly attribute unsigned long relatedOmc;
    readonly attribute string coverArea;

// The following notifications may be sent from this MO,
// notifyObjectCreation
// notifyObjectDeletion
// notifyAttributeValueChange
// notifyAckStateChanged
// notifyChangedAlarm
// notifyClearedAlarm
// notifyNewAlarm
};

interface VcFunction : GenericNRMProfile::ManagedFunction
{
    readonly attribute GenericNRMSystem::ObjectIdType vcFunctionId;
    readonly attribute unsigned long maxNumCard;
    readonly attribute unsigned long maxNumSignLink;
    readonly attribute unsigned long maxNumSignLink2M;
    readonly attribute unsigned long numSignLink;

```

```

readonly attribute unsigned long numSignLink2M;
readonly attribute INNRMSystem::putIntoUseTimeType putIntoUseTime;
readonly attribute INNRMSystem::workingModeType workingMode;
readonly attribute INNRMSystem::workingStateType workingState;
readonly attribute string backupVc;
readonly attribute INNRMSystem::usageStateType usageState;
readonly attribute unsigned long totalMachine;
readonly attribute string relatedSmp;
readonly attribute string relatedOmc;
readonly attribute string coverArea;

// The following notifications may be sent from this MO,
// notifyObjectCreation
// notifyObjectDeletion
// notifyAttributeValueChange
// notifyAckStateChanged
// notifyChangedAlarm
// notifyClearedAlarm
// notifyNewAlarm
};

interface SpModule : GenericNRMPProfile::Top
{
    readonly attribute string GserviceName;
    readonly attribute string serviceDesc;
    readonly attribute unsigned long maxNumUsers;
    readonly attribute float userTraffic;
    readonly attribute unsigned long maxCaps;

// The following notifications may be sent from this MO,
// notifyObjectCreation
// notifyObjectDeletion
// notifyAttributeValueChange
// notifyAckStateChanged
// notifyChangedAlarm
// notifyClearedAlarm
// notifyNewAlarm
};

interface WvpnModule : SpModule

```



```
{
    readonly attribute GenericNRMSystem::ObjectIdType wvpnModuleId;

};

interface PpcModule : SpModule
{
    readonly attribute GenericNRMSystem::ObjectIdType PpcModuleId;

};

interface SubSystem : GenericNRMProfile::Top
{
    readonly attribute GenericNRMSystem::ObjectIdType subSystemId;
    readonly attribute string locationName;
    readonly attribute string subsysModel;
    readonly attribute unsigned short subsysState;
    readonly attribute string subsysDesc;
    readonly attribute INNRMSystem::cpuInfoListType cpuInfoList;
    readonly attribute INNRMSystem::memInfoListType memInfoList;
    readonly attribute INNRMSystem::diskInfoListType diskInfoList;
    readonly attribute INNRMSystem::ioPortInfoListType ioPortInfoList;
    readonly attribute INNRMSystem::osInfoListType osInfoList;
    readonly attribute INNRMSystem::appInfoListType appInfoList;
    readonly attribute INNRMSystem::dbInfoListType dbInfoList;

    // The following notifications may be sent from this MO,
    // notifyObjectCreation
    // notifyObjectDeletion
    // notifyAttributeValueChange
    // notifyAckStateChanged
    // notifyChangedAlarm
    // notifyClearedAlarm
    // notifyNewAlarm
};

};

#endif
```

8.1.2.3 INNRMSystem

```

//File "INNRMSystem.idl"
//The IRP document version number is "IN NRM V1.0"
#ifndef INNRMSystem_idl
#define INNRMSystem_idl

#include <TimeBase.idl>
#include <GenericNRMSystem.idl>

module INNRMSystem
{
    /**
     * This module adds datatype definitions for types
     * used in the IN NRM which are not basic datatypes defined
     * already in CORBA and datatypes defined already in
     * GenericNRMSystem.
     */

    typedef sequence<string> funcListType;
    typedef sequence<string> serviceListType;

    typedef TimeBase::UtcTimeIntoUseTimeType;
    typedef unsigned short workingModeType;
    typedef unsigned short workingStateType;
    typedef unsigned short usageStateType;

    struct cpuInfoType
    {
        unsigned long cpu_id;
        unsigned long cpu_frequence;
    };
    typedef sequence<cpuInfoType> cpuInfoListType;

    struct memInfoType
    {
        unsigned long memory_id;
        unsigned long memory_size;
    };
    typedef sequence<memInfoType> memInfoListType;

    struct diskInfoType

```

```
{
    unsigned long disk_id;
    unsigned long disk_size;
};
typedef sequence<diskInfoType> diskInfoListType;

struct ioPortInfoType
{
    unsigned long ioport_id;
    string ioport_name;
    unsigned short ioport_type;
    unsigned long ioport_speed;
    unsigned short ioport_state;
};
typedef sequence<ioPortInfoType> ioPortInfoListType;

struct osInfoType
{
    unsigned long os_id;
    string os_name;
    string os_version;
};
typedef sequence<osInfoType> osInfoListType;

struct appInfoType
{
    unsigned long app_id;
    string app_name;
    string app_version;
    string app_vendor;
};
typedef sequence<appInfoType> appInfoListType;

struct dbInfoType
{
    unsigned long db_id;
    string db_name;
    string db_version;
    unsigned long adb_size;
};
typedef sequence<dbInfoType> dbInfoListType;
```

```
};
#endif
```

8.2 性能网络资源模型设计

8.2.1 INMeasurementDefs

```
//File INMeasurementDefs.idl
#ifndef INMeasurementDefs_idl
#define INMeasurementDefs_idl

/**
 * This module defines measurementType names constants
 */
module INMeasurementDefs
{
    //Scp measurement
    module scpMeas
    {
        const string numCallsAtt = "numCallsAtt";
        const string numCallsDiscardOverload = "numCallsDiscardOverload";
        const string numCallsDiscardErr = "numCallsDiscardErr";
        const string callingTraffic = "callingTraffic";
        const string calledTraffic = "calledTraffic";
        const string numCallingResp = "numCallingResp";
        const string numCalledResp = "numCalledResp";
        const string numCallingFinish = "numCallingFinish";
        const string numCalledFinish = "numCalledFinish";
        const string numRecord = "numRecord";
        const string numRecordFailed = "numRecordFailed";
        const string numRecord0 = "numRecord0";
        const string numMsgOvertime = "numMsgOvertime";
        const string numErrDecode = "numErrDecode";
        const string numMsgTotal = "numMsgTotal";
        const string scpBhca = "scpBhca";
        const string scpCaps = "scpCaps";
    };

    //WvpnModule measurement
    module wvpnMeas
    {
        const string numGroup = "numGroup";
        const string numGroupAdd = "numGroupAdd";
        const string numGroupDel = "numGroupDel";
    };
};
```

```

const string maxMembers = "maxMembers";
const string numUsers = "numUsers";
const string numUsersAdd = "numUsersAdd";
const string numUsersDel = "numUsersDel";
const string numCalls = "numCalls";
const string numCallsIntra = "numCallsIntra";
const string numCallsClosed = "numCallsClosed";
const string durCalls = "durCalls";
const string durCallsIntra = "durCallsIntra";
const string durCallsClosed = "durCallsClosed";
const string numFailPerCause = "numFailPerCause";
const string numUsersGsm= "numUsersGsm";
const string numUsersPbx= "numUsersPbx";
};

//Ppc measurement
module ppcMeas
{
    const string numUsers = "numUsers";
    const string numUsersAdd = "numUsersAdd";
    const string numUsersUse = "numUsersUse";
    const string numUsersHold = "numUsersHold";
    const string numUsersLocked = "numUsersLocked";
    const string numUsersBlack = "numUsersBlack";
    const string numUsersLoss = "numUsersLoss";
    const string numUsersFrosen = "numUsersFrosen";
    const string numUsersUnused = "numUsersUnused";
    const string numUsersRecy = "numUsersRecy";
    const string depositFee = "depositFee";
    const string numCalls = "numCalls";
    const string durCalls = "durCalls";
    const string numOAnswer = "numOAnswer";
    const string numTAnswer = "numTAnswer";
    const string numSmsP2P = "numSmsP2P";
    const string numSmsOnline = "numSmsOnline";
    const string feeSmsReal = "feeSmsReal";
    const string feeSmsNoReal = "feeSmsNoReal";
    const string numData = "numData";
    const string feeDataReal = "feeDataReal";
    const string feeDataNoReal = "feeDataNoReal";
    const string numFailPerCause = "numFailPerCause";
};

```

```
//Smp measurement
module smpMeas
{
    const string numToScpAtt = "numToScpAtt";
    const string numFromScpResult = "numFromScpResult";
    const string numFromScpErr = "numFromScpErr";
    const string numAccessSMAP = "numAccessSMAP";
};

//VC measurement
module vcMeas
{
    const string numCards = "numCards";
    const string numCallsAtt = "numCallsAtt";
    const string numCallsDiscardOverload = "numCallsDiscardOverload";
    const string numCallsDiscardErr = "numCallsDiscardErr";
    const string numRechargeSucc = "numRechargeSucc";
    const string feeRecharge = "feeRecharge";
    const string numRechargeFail = "numRechargeFail";
    const string numRechargeFailPerCause = "numRechargeFailPerCause";
};

//SubSystem measurement
module subSystemMeas
{
    const string cpuAveLoad = "cpuAveLoad";
    const string diskSpareSize = "diskSpareSize";
    const string diskIoPer = "diskIoPer";
    const string dbSpareSize = "dbSpareSize";
    const string memSpareSize = "memSpareSize";
};

};

#endif
```

8.3 性能管理接口功能相关的文件

8.3.1 性能测量数据文件的 Schema 定义<measCollec.xsd>

下面的Schema文件中用到的字段的说明参见附录A，示例参见附录B。

版本号：PM FILE V1.0

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Meas collection data file XML schema measCollec.xsd -->
<schema targetNamespace="http://latest/nmc-omc/cmNrm.doc#measCollec"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:mc="http://latest/nmc-omc/cmNrm.doc#measCollec"
elementFormDefault="qualified">
  <!-- Meas collection data file root XML element -->
  <element name="measCollecFile">
    <complexType>
      <sequence>
        <element name="fileHeader">
          <complexType>
            <sequence>
              <element name="fileSender">
                <complexType>
                  <attribute name="localDn" type="string" use="optional"/>
                  <attribute name="elementType" type="string" use="required"/>
                </complexType>
              </element>
              <element name="measCollec">
                <complexType>
                  <attribute name="beginTime" type="dateTime" use="required"/>
                </complexType>
              </element>
            </sequence>
            <attribute name="fileFormatVersion" type="string" use="required"/>
            <attribute name="vendorName" type="string" use="optional"/>
            <attribute name="dnPrefix" type="string" use="optional"/>
          </complexType>
        </element>
        <element name="measData" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <sequence>
              <element name="managedElement">
                <complexType>
                  <attribute name="localDn" type="string" use="optional"/>
                  <attribute name="userLabel" type="string" use="optional"/>
                  <attribute name="swVersion" type="string" use="optional"/>
                </complexType>
              </element>
            </sequence>
          </complexType>
        </element>
      </sequence>
    </complexType>
  </element>
</schema>
```

```

<element name="measInfo" minOccurs="0" maxOccurs="unbounded">
  <complexType>
    <sequence>
      <element name="job" minOccurs="0">
        <complexType>
          <attribute name="jobId" type="string" use="required"/>
        </complexType>
      </element>
      <element name="granPeriod">
        <complexType>
          <attribute name="duration" type="duration" use="required"/>
          <attribute name="endTime" type="dateTime" use="required"/>
        </complexType>
      </element>
      <element name="repPeriod" minOccurs="0">
        <complexType>
          <attribute name="duration" type="duration" use="required"/>
        </complexType>
      </element>
      <choice>
        <element name="measTypes">
          <simpleType>
            <list itemType="mc:measName"/>
          </simpleType>
        </element>
        <element name="measType" minOccurs="0" maxOccurs="unbounded">
          <complexType>
            <simpleContent>
              <extension base="mc:measName">
                <attribute name="p" type="positiveInteger" use="required"/>
              </extension>
            </simpleContent>
          </complexType>
        </element>
      </choice>
      <element name="measValue" minOccurs="0" maxOccurs="unbounded">
        <complexType>
          <sequence>
            <choice>
              <element name="measResults">
                <simpleType>
                  <list itemType="mc:measResultType"/>
                </simpleType>
            </choice>
          </sequence>
        </complexType>
      </element>
    </sequence>
  </complexType>

```



```

        </element>
        <element name="r" minOccurs="0" maxOccurs="unbounded">
            <complexType>
                <simpleContent>
                    <extension base="mc:measResultType">
                        <attribute name="p" type="positiveInteger"
use="required"/>
                    </extension>
                </simpleContent>
            </complexType>
        </element>
    </choice>
    <element name="suspect" type="boolean" minOccurs="0"/>
</sequence>
    <attribute name="measObjLdn" type="string" use="required"/>
</complexType>
</element>
</sequence>
</complexType>
</element>
</sequence>
</complexType>
</element>
<element name="fileFooter">
    <complexType>
        <sequence>
            <element name="measCollec">
                <complexType>
                    <attribute name="endTime" type="dateTime" use="required"/>
                </complexType>
            </element>
        </sequence>
    </complexType>
</element>
</sequence>
</complexType>
</element>
<simpleType name="measNameWithSubCounter">
    <restriction base="string">
        <pattern
value="(wvpnMeas.numFailPerCause.lppcMeas.numFailPerCause.lvcMeas.numRechargeFailPerCause.)([a-fA-F0-9]{
1,8})"/>
    </restriction>

```

```

</simpleType>
<simpleType name="measNameWithOutSubCounter">
  <restriction base="string">
    <enumeration value="scpMeas.numCallsAtt"/>
    <enumeration value="scpMeas.numCallsDiscardOverload"/>
    <enumeration value="scpMeas.numCallsDiscardErr"/>
    <enumeration value="scpMeas.callingTraffic"/>
    <enumeration value="scpMeas.calledTraffic"/>
    <enumeration value="scpMeas.numCallingResp"/>
    <enumeration value="scpMeas.numCalledResp"/>
    <enumeration value="scpMeas.numCallingFinish"/>
    <enumeration value="scpMeas.numCalledFinish"/>
    <enumeration value="scpMeas.numRecord"/>
    <enumeration value="scpMeas.numRecordFailed"/>
    <enumeration value="scpMeas.numRecord0"/>
    <enumeration value="scpMeas.numMsgOvertime"/>
    <enumeration value="scpMeas.numErrDecode"/>
    <enumeration value="scpMeas.numMsgTotal"/>
    <enumeration value="scpMeas.scpBhca"/>
    <enumeration value="scpMeas.scpCaps"/>
    <enumeration value="wvpnMeas.numGroup"/>
    <enumeration value="wvpnMeas.numGroupAdd"/>
    <enumeration value="wvpnMeas.numGroupDel"/>
    <enumeration value="wvpnMeas.maxMembers"/>
    <enumeration value="wvpnMeas.numUsers"/>
    <enumeration value="wvpnMeas.numUsersAdd"/>
    <enumeration value="wvpnMeas.numUsersDel"/>
    <enumeration value="wvpnMeas.numCalls"/>
    <enumeration value="wvpnMeas.numCallsIntra"/>
    <enumeration value="wvpnMeas.numCallsClosed"/>
    <enumeration value="wvpnMeas.durCalls"/>
    <enumeration value="wvpnMeas.durCallsIntra"/>
    <enumeration value="wvpnMeas.durCallsClosed"/>
    <enumeration value="wvpnMeas.numUsersGsm"/>
    <enumeration value="wvpnMeas.numUsersPbx"/>
    <enumeration value="ppcMeas.numUsers"/>
    <enumeration value="ppcMeas.numUsersAdd"/>
    <enumeration value="ppcMeas.numUsersUse"/>
    <enumeration value="ppcMeas.numUsersHold"/>
    <enumeration value="ppcMeas.numUsersLocked"/>
    <enumeration value="ppcMeas.numUsersBlack"/>
    <enumeration value="ppcMeas.numUsersLoss"/>
    <enumeration value="ppcMeas.numUsersFrosen"/>
  </restriction>
</simpleType>

```

```

    <enumeration value="ppcMeas.numUsersUnused"/>
    <enumeration value="ppcMeas.numUsersRecy"/>
    <enumeration value="ppcMeas.depositFee"/>
    <enumeration value="ppcMeas.numCalls"/>
    <enumeration value="ppcMeas.durCalls"/>
    <enumeration value="ppcMeas.numOAnswer"/>
    <enumeration value="ppcMeas.numTAnswer"/>
    <enumeration value="ppcMeas.numSmsP2P"/>
    <enumeration value="ppcMeas.numSmsOnline"/>
    <enumeration value="ppcMeas.feeSmsReal"/>
    <enumeration value="ppcMeas.feeSmsNoReal"/>
    <enumeration value="ppcMeas.numData"/>
    <enumeration value="ppcMeas.feeDataReal"/>
    <enumeration value="ppcMeas.feeDataNoReal"/>
    <enumeration value="smpMeas.numToScpAtt"/>
    <enumeration value="smpMeas.numFromScpResult"/>
    <enumeration value="smpMeas.numFromScpErr"/>
    <enumeration value="smpMeas.numAccessSMAP"/>
    <enumeration value="vcMeas.numCards"/>
    <enumeration value="vcMeas.numCallsAtt"/>
    <enumeration value="vcMeas.numCallsDiscardOverload"/>
    <enumeration value="vcMeas.numCallsDiscardErr"/>
    <enumeration value="vcMeas.numRechargeSucc"/>
    <enumeration value="vcMeas.feeRecharge"/>
    <enumeration value="vcMeas.numRechargeFail"/>
    <enumeration value="subSysMeas.cpuAveLoad"/>
    <enumeration value="subSysMeas.diskSpareSize"/>
    <enumeration value="subSysMeas.diskIoPer"/>
    <enumeration value="subSysMeas.dbSpareSize"/>
    <enumeration value="subSysMeas.memSpareSize"/>
  </restriction>
</simpleType>
<simpleType name="measName">
  <union memberTypes="mc:measNameWithSubCounter mc:measNameWithOutSubCounter"/>
</simpleType>
<simpleType name="measResultType">
  <union memberTypes="integer decimal">
    <simpleType>
      <restriction base="string">
        <enumeration value="NIL"/>
      </restriction>
    </simpleType>
  </union>

```

```
</simpleType>  
</schema>
```

8.3.2 性能测量数据文件的 XML header 定义

在实际性能测量数据文件中应该使用下面的XML header定义:

```
<?xml version="1.0" encoding="UTF-8"?>  
<?xml-stylesheet type="text/xsl" href="MeasDataCollection.xsl"?>  
<measCollecFile  
  xmlns=  
  " http://latest/nmc-omc/cmNrm.doc#measCollec "  
>
```

广东省网络空间安全协会受控资料

广东省网络空间安全协会受控资料

中华人民共和国
通信行业标准

800MHz CDMA 数字蜂窝移动通信网网络管理接口技术要求
第1部分：智能网网管接口

YD/T 1685.1-2007

*

人民邮电出版社出版发行
北京市崇文区夕照寺街14号A座
邮政编码：100061

*

版权所有 不得翻印

*

本书如有印装质量问题，请与本社联系 电话：(010)67114922