YD

中华人民共和国通信行业标准

YD/T 1585.3-2007

2GHz TD-SCDMA 数字蜂窝移动通信网 网络管理技术要求（第二阶段） 第3部分 基于CORBA技术的 网络资源模型设计

2GHz TD-SCDMA Digital Cell Mobile Communication Network
Management Technical Specification(Phase Ⅱ)
Part 3 CORBA-based Network Resource Model Design

2007-05-16 发布　　2007-05-16 实施

中华人民共和国信息产业部　发布

目　次

前　言

本标准是 2GHz 数字蜂窝移动通信网网络管理技术要求系列标准之一。该系列标准的预计结构为：系列标准由 4 项标准组成，各项标准又分为多个部分标准。该系列标准和部分标准的名称预计如下：

1. 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

2. 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

3. 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

4. 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

5. 2GHz WCDMA 数字蜂窝移动通信网网络管理技术要求（第一阶段）　第 1 部分　配置网络资源模型

6. 2GHz WCDMA 数字蜂窝移动通信网网络管理技术要求（第一阶段）　第 2 部分　性能网络资源模型

7. 2GHz WCDMA 数字蜂窝移动通信网网络管理技术要求（第一阶段）　第 3 部分　基于 CORBA 技术的网络资源模型设计

8. 2GHz cdma2000 数字蜂窝移动通信网网络管理技术要求（第一阶段）　第 1 部分　配置网络资源模型

9. 2GHz cdma2000 数字蜂窝移动通信网网络管理技术要求（第一阶段）　第 2 部分　性能网络资源模型

10. 2GHz cdma2000 数字蜂窝移动通信网网络管理技术要求（第一阶段）　第 3 部分　基于 CORBA 技术的网络资源模型设计

11. 2GHz TD-SCDMA 数字蜂窝移动通信网网络管理技术要求（第二阶段）　第 1 部分　配置网络资源模型

12. 2GHz TD-SCDMA 数字蜂窝移动通信网网络管理技术要求（第二阶段）　第 2 部分　性能网络资源模型

13. 2GHz TD-SCDMA 数字蜂窝移动通信网网络管理技术要求(第二阶段）第 3 部分　基于 CORBA 技术的网络资源模型设计

本标准参考 3G 移动通信伙伴项目（3GPP）的以下 TS 32 系列标准：

1. 3GPP TS 32.623 Telecommunication management;Configuration Management (CM);Generic network resources Integration Reference Point (IRP); Common Object Request Broker Architecture (CORBA)(通用网络资源集成参考点：基于 CORBA 接口设计）

2. 3GPP TS 32.633 Telecommunication management;Configuration Management (CM);Core Network Resources Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA)（核心网网络资源集成参考点：基于 CORBA 接口设计）

3. 3GPP TS 32.643 Telecommunication management;Configuration Management (CM);UTRAN network

resources Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA)（UTRAN 网络资源集成参考点：基于 CORBA 接口设计）

4. 3GPP TS 32.653 Telecommunication management;Configuration Management (CM);GERAN network resources Integration Reference Point (IRP): Common Object Request Broker Architecture (CORBA)（GERAN 网络资源集成参考点：基于 CORBA 接口设计）

与上述 3GPP 相关标准的一致性程度为非等效。

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：大唐电信科技产业集团

本标准主要起草人：吴 恒、罗云中

2GHz TD-SCDMA数字蜂窝移动通信网网络管理技术要求（第二阶段）第3部分　基于CORBA技术的网络资源模型设计

1　范围

本部分规定了2GHz数字蜂窝移动通信网（以下简称3G）网络管理接口中采用TD-SCDMA技术的网络资源模型的IDL定义。

本部分适用于2GHz TD-SCDMA数字蜂窝移动通信网（第二阶段）的网络管理。

2　规范性引用文件

下列文件中的条款通过本部分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。

YD/T 1584.3-2006　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

YD/T 1585.1-2006　2GHz TD-SCDMA数字蜂窝移动通信网网络管理技术要求（第二阶段）　第1部分　配置网络资源模型

YD/T 1585.2-2006　2GHz TD-SCDMA数字蜂窝移动通信网网络管理技术要求（第二阶段）　第2部分　性能网络资源模型

YD/T 1586.3-2006　2GHz WCDMA数字蜂窝移动通信网网络管理技术要求（第一阶段）　第3部分　基于CORBA技术的网络资源模型设计

3　术语、定义和缩略语

3.1　缩略语

下列缩略语适用于本部分。

IDL	Interface Definition Language	接口定义语言
CORBA	Common Object Request Broker Architecture	公共对象请求代理体系

4　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、UtranNRMDefs.idl和CoreNRMDefs.idl，用来定义配置网络资源对象及其属性名称；

2）xxxNRMSystem.idl,包括GenericNRMSystem.idl、UtranNRMSystem.idl 和 CoreNRMSystem.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、UtranNRMProfile.idl 和 CoreNRMProfile.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。

4.1　通用配置资源模型的 IDL 定义

参见《2GHz WCDMA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》中的通用配置资源模型的IDL定义。

4.2 无线接入网网络资源模型的 IDL 定义

4.2.1 UtranNRMDefs

```
//File "UtranNRMDefs.idl"
//The IRP document version number is "UTRAN NRM V1.0"
#ifndef UtranNRMDefs_idl
#define UtranNRMDefs_idl

#include "GenericNRMDefs.idl"

#pragma prefix "3gppsa5.org"

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

        // including all Attribute Names from
        // MO Class GenericNRMDefs::ManagedFunction
        // additional Attribute Names is as follows.
        //
        const string rncFunctionId = "rncFunctionId";
        const string rncId = "rncId";
        const string mnc = "mnc";
        const string mcc = "mcc";
    };

    //Definitions for MO class NodeBFunction
    interface NodeBFunction : GenericNRMDefs::ManagedFunction
    {
        const string CLASS = "NodeBFunction";
    // including all Attribute Names from
    // MO Class GenericNRMDefs::ManagedFunction
    // additional Attribute Names is as follows.
    //
    const string nodeBFunctionId = "nodeBFunctionId";
    const string relatedIubLink = "relatedIubLink";
};

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

    // Attribute Names
    //
    const string iubLinkId = "iubLinkId";
    const string relatedNodeB = "relatedNodeB";
    const string relatedUtranCells = "relatedUtranCells";
};

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

    // Attribute Names
    //
    const string utranCellId = "utranCellId";
    const string cId= "cId";
    const string localCellId= "localCellId";
    const string cellMode ="cellMode";
    const string maximumTransmissionPower = "maximumTransmissionPower";
    const string uarfcn = "uarfcn";
    const string cellParameterId = "cellParameterId";
    const string primaryCcpchPower = "primaryCcpchPower";
    const string dwPchPower = "dwPchPower";
    const string timeSlotList = "timeSlotList";
    const string lac = "lac";
    const string rac = "rac";
    const string sac = "sac";
    const string uraList = "uraList";
    const string relatedIubLink = "relatedIubLink";
};

//Definitions for MO class UtranRelation
interface UtranRelation : GenericNRMDefs::Top
{
    const string CLASS = "UtranRelation";

    // Attribute Names
    //
    const string utranRelationId = "utranRelationId";
    const string adjacentCell = "adjacentCell";
    const string cellMode ="cellMode";
    const string uarfcn = "uarfcn";
    const string cellParameterId = "cellParameterId";
    const string primaryCcpchPower = "primaryCcpchPower";
    const string lac = "lac";
    const string userLabel = "userLabel" ;
};

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

    // Attribute Names
    //
    const string externalUtranCellId = "externalUtranCellId";
    const string cId = "cId";
    const string mnc = "mnc";
    const string mcc = "mcc";
    const string rncId = "rncId";
    const string cellMode ="cellMode";
    const string uarfcn = "uarfcn";
    const string cellParameterId = "cellParameterId";
    const string primaryCcpchPower = "primaryCcpchPower";
    const string lac = "lac";
    const string rac = "rac";
};

//Defination for MO class GsmRelation
interface GsmRelation: GenericNRMDefs::Top
{
    const string CLASS = "GsmRelation";

    //Attribute Names
    //
    const string gsmRelationId = "gsmRelationId";
    const string adjacentCell = "adjacentCell";
    const string bcchFrequency = "bcchFrequency";
    const string ncc = "ncc";
    const string bcc = "bcc";
    const string lac = "lac";
    const string userLabel = "userLabel";
};

//Defination for MO ExternalGSMCell

interface ExternalGSMCell: GenericNRMDefs::ManagedFunction
{
    const string CLASS = "ExternalGSMCell";

    //Attribute Names
    //
    const string externalGsmCellId = "externalGsmCellId";
    const string cellIdentity = "cellIdentity";
    const string bcchFrequency = "bcchFrequency";
    const string ncc = "ncc";
    const string bcc = "bcc";
```

```
            const string lac = "lac";
            const string mcc = "mcc";
            const string mnc = "mnc";
            const string rac = "rac";
            const string racc = "racc";
        };
    };
    #endif
```

4.2.2 UtranNRMSystem

```
//File "UtranNRMSystem.idl"
//The IRP document version number is "UTRAN NRM V1.0"
#ifndef UtranNRMSystem_idl
#define UtranNRMSystem_idl

#include "GenericNRMSystem.idl"

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

    union AdjacentCellType switch(boolean)
    {
        case TRUE: GenericNRMSystem::DN    utranCell;
        case FALSE: string cellGloableId;
    };
    enum CellModeEnumType
    {
        FDD_mode,
        TDD_mode_1_28Mcps,
        TDD_mode_3_84Mcps
    };
    enum TimeSlotDirectionType
```

```
        {
            UL,
            DL
        };
        enum TimeSlotStatusType
        {
            Active,
            Not_Active
        };
        struct TimeSlotConfigStructType
        {
            unsigned short timeSlotId;
            TimeSlotDirectionType timeSlotDirection;
            TimeSlotStatusType timeSlotStatus;
        };
        typedef sequence<TimeSlotConfigStructType> TimeSlotListConfigStructType;
        typedef sequence< unsigned long> UraListType;
    };
    #endif
```

4.2.3 UtranNRMprofile

```
//File "UtranNRMProfile.idl"
//The IRP document version number is "UTRAN NRM V1.0"
#ifndef UtranNRMProfile_idl
#define UtranNRMProfile_idl

#include "GenericNRMSystem.idl"
#include "GenericNRMDefs.idl"
#include "UtranNRMSystem.idl"

//#pragma prefix "3gppsa5.org"

/**
 * This module defines the attribute names and
 * correspondig attribute types for all defined
 * MO class in Utran network. This module is
 * used for reference.
 */
```

```
module UtranNRMProfile
{
    interface RncFunction : GenericNRMDefs::ManagedFunction
    {
        readonly attribute GenericNRMSystem::ObjectIdType rncFunctionId;
                     attribute unsigned long rncId;
        readonly attribute unsigned long mnc;
        readonly attribute unsigned long mcc;

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

    interface NodeBFunction : GenericNRMDefs::ManagedFunction
    {
        readonly attribute GenericNRMSystem::ObjectIdType nodeBFunctionId;
        readonly attribute GenericNRMSystem::DN relatedIubLink;

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

interface IubLink : GenericNRMDefs::ManagedFunction
{
    readonly attribute GenericNRMSystem::ObjectIdType iubLinkId;
             attribute GenericNRMSystem::DN relatedNodeB;
             attribute GenericNRMSystem::DNListType relatedUtranCells;

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

interface UtranCell : GenericNRMDefs::ManagedFunction
{
    readonly attribute GenericNRMSystem::ObjectIdType utranCellId;
             attribute unsigned long cId;
             attribute unsigned long localCellId;
    readonly attribute UtranNRMSystem::CellModeEnumType cellMode;
    readonly attribute unsigned short maximumTransmissionPower; //0..50Dbm
    readonly attribute unsigned long uarfcn;
             attribute unsigned short cellParameterId;
    readonly attribute short primaryCcpchPower;
    readonly attribute short dwPchPower;
             attribute UtranNRMSystem::TimeSlotListConfigStructType timeSlotList;
    readonly attribute unsigned long lac;
    readonly attribute unsigned long rac;
    readonly attribute unsigned long sac;
```

```
        readonly attribute UtranNRMSystem::UraListType uraList;
        readonly attribute GenericNRMSystem::DN relatedIubLink;

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

    interface UtranRelation : GenericNRMDefs::Top
    {
        readonly attribute GenericNRMSystem::ObjectIdType utranRelationId;
                 attribute UtranNRMSystem::AdjacentCellType adjacentCell;
        readonly attribute UtranNRMSystem::CellModeEnumType cellMode;
        readonly attribute unsigned long uarfcn;      //conditional
        readonly attribute unsigned long cellParameterId;      //conditional
        readonly attribute unsigned short primaryCcpchPower;      //conditional
        readonly attribute unsigned long lac;      //conditional
                 attribute wstring userLabel;

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

    interface ExternalUtranCell : GenericNRMDefs::ManagedFunction
    {
        readonly attribute GenericNRMSystem::ObjectIdType externalUtranCellId;
                 attribute unsigned long cId;
```

```
                attribute unsigned long mcc;
                attribute unsigned long mnc;
                attribute unsigned long rncId;
      readonly attribute UtranNRMSystem::CellModeEnumType cellMode;
                attribute unsigned long uarfcn;
                attribute unsigned long cellParameterId;
                attribute unsigned short primaryCcpchPower;
                attribute unsigned long lac;
                attribute unsigned long rac;

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

   interface GsmRelation: GenericNRMDefs::Top
   {
      readonly attribute GenericNRMSystem::ObjectIdType gsmRelationId;
                attribute GenericNRMSystem::DN adjacentCell;
      readonly attribute UtranNRMSystem::AdjacentCellType bcchFrequency;
      readonly attribute unsigned long ncc;
      readonly attribute unsigned long bcc;
      readonly attribute unsigned long lac;
                attribute wstring userLabel;

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

   interface ExternalGSMCell: GenericNRMDefs::ManagedFunction
   {
      readonly attribute GenericNRMSystem::ObjectIdType externalGsmCellId;
                attribute unsigned long cellIdentity;
                attribute short bcchFrequency;
```

```
                    attribute unsigned long ncc;
                    attribute unsigned long bcc;
                    attribute unsigned long lac;
                    attribute unsigned long mcc;
                    attribute unsigned long mnc;
                    attribute unsigned long rac;
                    attribute unsigned long racc;

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

4.3 核心网网络资源模型的 IDL 定义

参见《2GHz WCDMA 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》中的核心网网络资源模型的 IDL 定义。

5 性能网络资源模型设计

5.1 性能管理资源模型的 IDL 定义

注：下面的 IDL 文件为每个 family 定义了一个独立的 module。

— "family.measurementName.subcounter" 可用于获取一个 MeasurementName 的某个 subcounter 值；

— "family.measurementName" 可用于获取一个 MeasurementName 的值。如果该 MeasurementName 有 subcounters，那么所有 subcounters 的值都应该返回。

— "family" 可用于获取该 family 下的所有 MeasurementName 的值。

其中，family 为《2GHz TD-SCDMA 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》中的英文表名；subcounter 设置时选相应的整数值（取值见下节数据类型的 IDL 定义 "TDSCDMANRMMeasurementSystem.idl" 中的定义，对按每位置区或每路由区统计的性能测量项，其 subcounter 的取值为网络中的 LAI 或 RAI 值，本标准未进行定义；对按用户归属的 HLR 的 HLRNumber 或漫游到的 VLR 的 VLRNumber 进行统计的性能测量项，其 subcounter 的取值为网络中 HLRNumber 或 VLRNumber 的值，本标准未进行定义；"0" 表示总和）。

例如：

（1）"mscBasicMeasurement.failImsiAttachsPerCauseGsm.5" 可用于获取某个 cause (imeiNotAccepted) 的 "failImsiAttachsPerCauseGsm" 的值；

（2）"mscBasicMeasurement.failImsiAttachsPerCauseGsm" 可用于获取 "failImsiAttachsPerCauseGsm" 的所有 subcounters 的值；

（3）"mscBasicMeasurement.attGetRoutingInfo" 可用于获取 "attGetRoutingInfo" 的值；

“mscBasicMeasurement”可用于获取该 family 下的所有 MeasurementType 的值。

```
TDSCDMANRMMeasurementDefs.idl
   //File "TDSCDMANRMMeasurementDefs.idl"
   #ifndef TDSCDMANRMMeasurementDefs_idl
   #define TDSCDMANRMMeasurementDefs_idl

   // #pragma prefix "3gppsa5.org"

    /**
    * This module defines measurementType names constants
    */
   module TDSCDMANRMMeasurementDefs
   {
      // msc measurement
        module mscBasicMeasurement
        {
            //get routing information from HLR
            const string attGetRoutingInfo = "attGetRoutingInfo";
            const string succGetRoutingInfo = "succGetRoutingInfo";
            //imsi attach and detach
            const string attImsiAttachs = "attImsiAttachs";
            const string succImsiAttachs = "succImsiAttachs";
            const string failImsiAttachsPerCause = "failImsiAttachsPerCause";
            const string nbrImsiDetachs= "nbrImsiDetachs";
            //location update
            const string attLocationUpdatesIntraMsc= "attLocationUpdatesIntraMsc";
            const string succLocationUpdatesIntraMsc= "succLocationUpdatesIntraMsc";
            const string failLocationUpdatesIntraMscPerCause= "failLocationUpdatesIntraMscPerCause";
            const string attLocationUpdatesInterMsc= "attLocationUpdatesInterMsc";
            const string succLocationUpdatesInterMsc= "succLocationUpdatesInterMsc";
            const string failLocationUpdatesInterMscPerCause= "failLocationUpdatesInterMscPerCause";
            //originating and terminating sms via MSC
            const string attOrigSmsCs = "attOrigSmsCs";
            const string succOrigSmsCs = "succOrigSmsCs";
            const string failOrigSmsCsPerCause = "failOrigSmsCsPerCause";
            const string attTermSsmsCs = "attTermSsmsCs";
    const string succTermSmsCs = "succTermSmsCs";
    const string failTermSmsCsPerCause = "failTermSmsCsPerCause";
    //incoming handover inter MSC
    const string attIncHosInterMsc = "attIncHosInterMsc";
    const string succIncHosInterMsc = "succIncHosInterMsc";
    //outgoing handover inter MSC
    const string attOutHosInterMsc = "attOutHosInterMsc";
    const string succOutHosInterMsc = "succOutHosInterMsc";
    //subseqent handover to MSCa
    const string attSubsequentHosToMsca = "attSubsequentHosToMsca";
    const string succSubsequentHosToMsca = "succSubsequentHosToMsca";
    //subseqent handover to MSCc
    const string attSubsequentHosToMscc = "attSubsequentHosToMscc";
    const string succSubsequentHosToMscc = "succSubsequentHosToMscc";
    //external handover
    const string attExternalHos = "attExternalHos";
    const string attExternalHosPerCause = "attExternalHosPerCause";
    const string failExternalHosWithReconn = "failExternalHosWithReconn";
    const string failExternalHosWithLossOfConn = "failExternalHosWithLossOfConn";
    //paging
    const string attPageReqsPerLa = "attPageReqsPerLa";
    const string succPageReqsPerLa = "succPageReqsPerLa";
    const string attRepageReqsPerLa = "attRepageReqsPerLa";
};

module mscQos
{
    const string meanDurOfCallSetup = "meanDurOfCallSetup";
    const string meanDurOfCallAssignGsm = "meanDurOfCallAssignGsm";
    const string meanDurOfCallRabAssignUmts = "meanDurOfCallRabAssignUmts";
    const string meanDurOfLuService = "meanDurOfLuService";
    const string meanCallDur = "meanCallDur";
    const string meanDurOfTrunkSeizure = "meanDurOfTrunkSeizure";
};

//exchange measurement
module mobileTrafficFlow
```

```
{
    //GSM originating call
    const string attOrigCallsGsm = "attOrigCallsGsm";
    const string succOrigCallsGsm = "succOrigCallsGsm";
    const string ansOrigCallsGsm = "ansOrigCallsGsm";
    const string failOrigCallsGsmPerCause = "failOrigCallsGsmPerCause";
    const string attOrigCallTrafficGsm = "attOrigCallTrafficGsm";
    const string succOrigCallTrafficGsm = "succOrigCallTrafficGsm";
    const string ansOrigCallTrafficGsm = "ansOrigCallTrafficGsm";
    //WCDMA originating call
    const string attOrigCallsUmts = "attOrigCallsUmts";
    const string succOrigCallsUmts = "succOrigCallsUmts";
    const string ansOrigCallsUmts = "ansOrigCallsUmts";
    const string failOrigCallsUmtsPerCause = "failOrigCallsUmtsPerCause";
    const string attOrigCallTrafficUmts = "attOrigCallTrafficUmts";
    const string succOrigCallTrafficUmts = "succOrigCallTrafficUmts";
    const string ansOrigCallTrafficUmts = "ansOrigCallTrafficUmts";
    //internal call
    const string attInternalCalls = "attInternalCalls";
    const string succInternalCalls = "succInternalCalls";
    const string ansInternalCalls = "ansInternalCalls";
    const string failInternalCallsPerCause = "failInternalCallsPerCause";
    const string attInternalCallTraffic = "attInternalCallTraffic";
    const string succInternalCallTraffic = "succInternalCallTraffic";
    const string ansInternalCallTraffic = "ansInternalCallTraffic";
    //GSM terminating call
    const string attTermCallsGsm = "attTermCallsGsm";
    const string succTermCallsGsm = "succTermCallsGsm";
    const string ansTermCallsGsm = "ansTermCallsGsm";
    const string failTermCallsGsmPerCause = "failTermCallsGsmPerCause";
    const string attTermCallTrafficGsm = "attTermCallTrafficGsm";
    const string succTermCallTrafficGsm = "succTermCallTrafficGsm";
    const string ansTermCallTrafficGsm = "ansTermCallTrafficGsm";
    //WCDMA terminating call
    const string attTermCallsUmts = "attTermCallsUmts";
    const string succTermCallsUmts = "succTermCallsUmts";
    const string ansTermCallsUmts = "ansTermCallsUmts";
const string failTermCallsUmtsPerCause = "failTermCallsUmtsPerCause";
const string attTermCallTrafficUmts = "attTermCallTrafficUmts";
const string succTermCallTrafficUmts = "succTermCallTrafficUmts";
const string ansTermCallTrafficUmts = "ansTermCallTrafficUmts";
//incoming call
const string attIncCalls = "attIncCalls";
const string succIncCalls = "succIncCalls";
const string ansIncCalls= "ansIncCalls";
const string failIncCallsPerCause = "failIncCallsPerCause";
const string attIncCallTraffic = "attIncCallTraffic";
const string succIncCallTraffic = "succIncCallTraffic";
const string ansIncCallTraffic = "ansIncCallTraffic";
//outgoing call
const string attOutCalls = "attOutCalls";
const string succOutCalls= "succOutCalls";
const string ansOutCalls= "ansOutCalls";
const string failOutCallsPerCause = "failOutCallsPerCause";
const string attOutCallTraffic = "attOutCallTraffic";
const string succOutCallTraffic = "succOutCallTraffic";
const string ansOutCallTraffic = "ansOutCallTraffic";
//transit call
const string attTransCalls = "attTransCalls";
const string succTransCalls= "succTransCalls";
const string ansTransCalls= "ansTransCalls";
const string failTransCallsPerCause = "failTransCallsPerCause";
const string attTransCallTraffic = "attTransCallTraffic";
const string succTransCallTraffic = "succTransCallTraffic";
const string ansTransCallTraffic = "ansTransCallTraffic";
//originating outgoing call
const string attOrigOutCalls = "attOrigOutCalls";
const string succOrigOutCalls= "succOrigOutCalls";
const string ansOrigOutCalls= "ansOrigOutCalls";
const string failOrigOutCallsPerCause = "failOrigOutCallsPerCause";
const string attOrigOutCallTraffic = "attOrigOutCallTraffic";
const string succOrigOutCallTraffic = "succOrigOutCallTraffic";
const string ansOrigOutCallTraffic = "ansOrigOutCallTraffic";
//terminating incoming call
```

```
        const string attTermIncCalls = "attTermIncCalls";
        const string succTermIncCalls= "succTermIncCalls";
        const string ansTermIncCalls= "ansTermIncCalls";
        const string failTermIncCallsPerCause = "failTermIncCallsPerCause";
        const string attTermIncCallTraffic = "attTermIncCallTraffic";
        const string succTermIncCallTraffic = "succTermIncCallTraffic";
        const string ansTermIncCallTraffic = "ansTermIncCallTraffic";
        //abnormal call
        const string nbrCallsBlockedByLoadShedding = "nbrCallsBlockedByLoadShedding";
        const string nbrCallsBlockedByInternalCongestion = "nbrCallsBlockedByInternalCongestion";
        const string nbrCallsBlockedByOutCircuitBusy= "nbrCallsBlockedByOutCircuitBusy";
};

//circuit end point subgroup measurement
module circuitEndpointSubGroup
{
        //outgoing circuit end subgroup
        const string outBids= "outBids";
        const string succOutSeizures= "succOutSeizures";
        const string succOutCalls = "succOutCalls";
        const string ansOutCalls = "ansOutCalls";
        const string failOutCallsByOverflow = "failOutCallsByOverflow";
        const string failOutCallsByUserBusy = "failOutCallsByUserBusy";
        const string failOutCallsByNoAns = "failOutCallsByNoAns";
        const string failOutCallsByUnallNum = "failOutCallsByUnallNum";
        const string failOutCallsByCongestion = "failOutCallsByCongestion";
        const string succOutSeizureTraffic = "succOutSeizureTraffic";
        const string ansOutCallTraffic = "ansOutCallTraffic";
        //incoming circuit end subgroup
        const string succIncSeizures = "succIncSeizures";
        const string succIncCalls = "succIncCalls";
        const string ansIncSeizures = "ansIncSeizures";
        const string failIncCallsByUserBusy = "failIncCallsByUserBusy";
        const string failIncCallsByNoAns = "failIncCallsByNoAns";
        const string failIncCallsByUnallNum = "failIncCallsByUnallNum";
        const string failIncCallsByCongestion = "failIncCallsByCongestion";
        const string succIncSeizureTraffic = "succIncSeizureTraffic";
        const string ansIncSeizureTraffic = "ansIncSeizureTraffic";
        const string nbrAvailTrunks = "nbrAvailTrunks";
    };

    //No.7 signalling measurement
    module mtp3SignallingLinkTP
    {
        const string durSigLinkOutOfService= "durSigLinkOutOfService";
        const string nbrSigLinkOutOfService= "nbrSigLinkOutOfService";
        const string nbrSentMsus = "nbrSentMsus";
        const string nbrSentSifsAndSios = "nbrSentSifsAndSios";
        const string nbrRecvedMsus = "nbrRecvedMsus";
        const string nbrRecvedSifsAndSios = "nbrRecvedSifsAndSios";
    };

    module mtp3SignallingLinkSetTP
    {
        const string nbrSigLinkSetOutOfService= "nbrSigLinkSetOutOfService";
        const string durSigLinkSetOutOfService= "durSigLinkSetOutOfService";
        const string nbrAvailSiglinks= "nbrAvailSiglinks";
    };

    module mtp3bSignallingLinkTP
    {
        const string durSigLinkOutOfService= "durSigLinkOutOfService";
        const string nbrSigLinkOutOfService= "nbrSigLinkOutOfService";
        const string nbrSentMsus = "nbrSentMsus";
        const string nbrSentSifsAndSios = "nbrSentSifsAndSios";
        const string nbrRecvedMsus = "nbrRecvedMsus";
        const string nbrRecvedSifsAndSios = "nbrRecvedSifsAndSios";
    };

    module mtp3bSignallingLinkSetTP
    {
        const string nbrSigLinkSetOutOfService= "nbrSigLinkSetOutOfService";
        const string durSigLinkSetOutOfService= "durSigLinkSetOutOfService";
        const string nbrAvailSiglinks= "nbrAvailSiglinks";
        };

        // observed destination measurement
        module observedDestination
        {
            const string bids= "bids";
            const string nbrNoAvailCircuits= "nbrNoAvailCircuits";
            const string succCalls = "succCalls";
            const string ansCalls = "ansCalls";
            const string succCallTraffic = "succCallTraffic";
            const string ansCallTraffic = "ansCallTraffic";
        };

        //VLR measurement
        module vlrBasicMeasurement
        {
            //identification request to PVLR
            const string attIdentReqsToPVlr= "attIdentReqsToPVlr";
            const string succIdentReqsToPVlr= "succIdentReqsToPVlr";
            //location update
            const string attLusIntraVlr= "attLusIntraVlr";
            const string succLusIntraVlr= "succLusIntraVlr";
            const string attLusInterVlr= "attLusInterVlr";
            const string succLusInterVlr= "succLusInterVlr";
            //request for authentication set to HLR
            const string attReqsForAuthSetsSentToHlr= "attReqsForAuthSetsSentToHlr";
            const string succRecvedAuthSetsFromHlr= "succRecvedAuthSetsFromHlr";
            const string succReqAuthSetWithQuintupletsFromHlr =
"succReqAuthSetWithQuintupletsFromHlr";
            const string succReqAuthSetWithTripletsFromHlr = "succReqAuthSetWithTripletsFromHlr";
            //subscriber data administration
            const string succInsertSubsData= "succInsertSubsData";
            const string succDelSubsData= "succDelSubsData";
            //provide roaming number to HLR
            const string attProvideRoamingNumber= "attProvideRoamingNumber";
            const string succProvideRoamingNumber= "succProvideRoamingNumber";
        };
```

```
module vlrSubscriberData
{
      const string nbrCurrentSubsInVlrPerHlr= "nbrCurrentSubsInVlrPerHlr";
      const string nbrCurrentSubsWithPowerOnInVlr = "nbrCurrentSubsWithPowerOnInVlr";
      const string nbrRoamingSubs= "nbrRoamingSubs";
      const string nbrRoamingSubsInternational= "nbrRoamingSubsInternational";
};

//HLR measurement
module hlrBasicMeasurement
{
      const string attGetRoutingInfo= "attGetRoutingInfo";
      const string succGetRoutingInfo = "succGetRoutingInfo";
      const string attProvideRoamingNumber = "attProvideRoamingNumber";
      const string succProvideRoamingNumber = "succProvideRoamingNumber";
      const string attLocationUpdates = "attLocationUpdates";
      const string succLocationUpdates = "succLocationUpdates";
      const string attCancelLocation = "attCancelLocation";
      const string succCancelLocation = "succCancelLocation";
      const string attInsertSubsData = "attInsertSubsData";
      const string succInsertSubsData = "succInsertSubsData";
      const string attDeleteSubsData = "attDeleteSubsData";
      const string succDeleteSubsData = "succDeleteSubsData";
      const string attSendAuthInfo = "attSendAuthInfo";
      const string succSendAuthInfo = "succSendAuthInfo";
      const string nbrReset = "nbrReset";
      const string attRestoreData = "attRestoreData";
      const string succRestoreData = "succRestoreData";
};

module hlrSubscriberData
{
      const string nbrCurrentSubsWithPowerOnInHlrPerVlr =
"nbrCurrentSubsWithPowerOnInHlrPerVlr";
      const string nbrCurrentSubsInHlr = "nbrCurrentSubsInHlr";
      const string nbrCurrentMsisdnInHlr = "nbrCurrentMsisdnInHlr";
```

```
};

module hlrSmServiceMeasurement
{
    const string attSendRoutingInfoForSm = "attSendRoutingInfoForSm";
    const string succSendRoutingInfoForSm = "succSendRoutingInfoForSm";
    const string nbrAlertServiceCentre= "nbrAlertServiceCentre";
    const string nbrInformServiceCenter= "nbrInformServiceCenter";
    const string nbrReadyForSm= "nbrReadyForSm";
};

module hlrSupplementServiceMeasurement
{
    const string attRegisterSs = "attRegisterSs";
    const string succRegisterSs = "succRegisterSs";
    const string attEraseSs = "attEraseSs";
    const string succEraseSs = "succEraseSs";
    const string attActSs = "attActSs";
    const string succActSs = "succActSs";
    const string attDeactSs = "attDeactSs";
    const string succDeactSs = "succDeactSs";
};

module hlrInServiceMeasurement
{
    const string attAnyTimeInterrogation = "attAnyTimeInterrogation";
    const string succAnyTimeInterrogation = "succAnyTimeInterrogation";
    const string attAnyTimeSubsInterrogation = "attAnyTimeSubsInterrogation";
    const string succAnyTimeSubsInterrogation = "succAnyTimeSubsInterrogation";
    const string attAnyTimeModification = "attAnyTimeModification";
    const string succAnyTimeModification = "succAnyTimeModification";
    const string nbrNoteSubsDataModified = "nbrNoteSubsDataModified";
};

module hlrPacketServiceMeasurement
{
    const string attSendRoutingInfoForGprs = "attSendRoutingInfoForGprs";
```

```
        const string succSendRoutingInfoForGprs = "succSendRoutingInfoForGprs";
        const string nbrFailReport= "nbrFailReport";
        const string nbrNoteMsPresentForGprs= "nbrNoteMsPresentForGprs";
        const string attUpdateGprsLocation = "attUpdateGprsLocation";
        const string succUpdateGprsLocation = "succUpdateGprsLocation";
};

module hlrLocationServiceMeasurement
{
        const string attSendRoutingInfoForLcs = "attSendRoutingInfoForLcs";
        const string succSendRoutingInfoForLcs = "succSendRoutingInfoForLcs";
};

//EIR measurement
module eirBasicMeasurement
{
        const string nbrCurrentWhiteSubsInEir= "nbrCurrentWhiteSubsInEir";
        const string nbrCurrentBlackSubsInEir= "nbrCurrentBlackSubsInEir";
        const string nbrCurrentGreySubsInEir= "nbrCurrentGreySubsInEir";
};

//SGSN measurement
module sessionManagementMeasurement
{
        //active PDP context by MS
        const string attActPdpContextMs= "attActPdpContextMs";
        const string succActPdpContextMs= "succActPdpContextMs";
        const string failActPdpContextMsPerCause = "failActPdpContextMsPerCause";

        //active PDP context by network
        const string attActPdpContextNetwork= "attActPdpContextNetwork";
        const string succActPdpContextNetwork= "succActPdpContextNetwork";
        const string failActPdpContextNetworkPerCause= "failActPdpContextNetworkPerCause";

        //active dynamic PDP context by MS
        const string attActPdpContextDynMs= "attActPdpContextDynMs";
        const string succActPdpContextDynMs= "succActPdpContextDynMs";
```

```
//number of subscriber with actived PDP context
const string meanSubsWithActPdpContext= "meanSubsWithActPdpContext";
const string maxSubsWithActPdpContext= "maxSubsWithActPdpContext";

//mean number of acti ed PDP context
const string meanActPDPContext= "meanActPDPContext";
const string maxActPdpContext= "maxActPdpContext";

//deactive PDP context by SGSN
const string attDeactPdpContextSgsn= "attDeactPdpContextSgsn";
const string succDeactPdpContextSgsn= "succDeactPdpContextSgsn";

//deactive PDP context by MS
const string attDeactPdpContextMs= "attDeactPdpContextMs";
const string succDeactPdpContextMs= "succDeactPdpContextMs";

//deactive PDP context by GGSN
const string attDeactPdpContextGgsn= "attDeactPdpContextGgsn";
const string succDeactPdpContextGgsn= "succDeactPdpContextGgsn";

//active secondary PDP context
const string attActSecondPdpContext= "attActSecondPdpContext";
const string succActSecondPdpContext= "succActSecondPdpContext";

//modify PDP context by MS
const string attModPdpContextMs= "attModPdpContextMs";
const string succModPdpContextMs= "succModPdpContextMs";

//modify PDP context by SGSN
const string attModPdpContextSgsn= "attModPdpContextSgsn";
const string succModPdpContextSgsn= "succModPdpContextSgsn";

//update PDP context by GGSN
const string attUpdPdpContextGgsn= "attUpdPdpContextGgsn";
const string succUpdPdpContextGgsn= "succUpdPdpContextGgsn";
```

```
        //update PDP context by SGSN
        const string attUpdPdpContextSgsn= "attUpdPdpContextSgsn";
        const string succUpdPdpContextSgsn= "succUpdPdpContextSgsn";
    };

    module subscriberManagementMeasurement
    {
        //subscriber state
        const string meanStandbySubs= "meanStandbySubs";
        const string maxStandbySubs= "maxStandbySubs";
        const string meanReadySubs= "meanReadySubs";
        const string maxReadySubs= "maxReadySubs";
        const string meanPmmIdleSubs= "meanPmmIdleSubs";
        const string maxPmmIdleSubs= "maxPmmIdleSubs";
        const string meanPmmConnectedSubs= "meanPmmConnectedSubs";
        const string maxPmmConnectedSubs= "maxPmmConnectedSubs";

        //number of attached subscribers
        const string meanAttachedSubsPerRa = "meanAttachedSubsPerRa";
        const string maxAttachedSubsPerRa= "maxAttachedSubsPerRa";
    };

    module   mobileManagementMeasurement
    {
        //GPRS attach
        const string attGprsAttachGsm= "attGprsAttach";
        const string succGprsAttach= "succGprsAttach";
        const string failGprsAttachPerCause= "failGprsAttachPerCause";

        //combined GPRS/IMSI attach
        const string attCombiAttach= "attCombiAttach";
        const string succCombiAttach= "succCombiAttach";
        const string failCombiAttachPerCause= "failCombiAttachPerCause";

        //GPRS attach with IMSI already attached
        const string attGprsAttachWithImsiAttached= "attGprsAttachWithImsiAttached";
        const string succGprsAttachWithImsiAttached= "succGprsAttachWithImsiAttached";
```

```
            const string failGprsAttachWithImsiAttachedPerCause=
"failGprsAttachWithImsiAttachedPerCause";

            //GPRS detach by MS
            const string attGprsDetachMs= "attGprsDetachMs";
            //combined GPRS/IMSI detach by MS
            const string attCombiDetachMs= "attCombiDetachMs";
            //IMSI detach by MS
            const string attImsiDetachMs= "attImsiDetachMs";
            //GPRS detach by SGSN
            const string attGprsDetachSgsn= "attGprsDetachSgsn";
            const string succGprsDetachSgsn= "succGprsDetachSgsn";
            //GPRS detach by HLR
            const string attGprsDetachHlr= "attGprsDetachHlr";
            //intra-SGSN routing area update
            const string attIntraSgsnRaUpdate= "attIntraSgsnRaUpdate";
            const string succIntraSgsnRaUpdate= "succIntraSgsnRaUpdate";
            const string failIntraSgsnRaUpdatePerCause= "failIntraSgsnRaUpdatePerCause";

            //combined RA/LA intra SGSN routing area update
            const string attCombiIntraSgsnRaUpdate= "attCombiIntraSgsnRaUpdate";
            const string succCombiIntraSgsnRaUpdate= "succCombiIntraSgsnRaUpdate";
            const string failCombiIntraSgsnRaUpdatePerCause= "failCombiIntraSgsnRaUpdatePerCause";

            //inter SGSN routing area update
            const string attInterSgsnRaUpdate= "attInterSgsnRaUpdate";
            const string succInterSgsnRaUpdate= "succInterSgsnRaUpdate";
            const string failInterSgsnRaUpdatePerCause= "failInterSgsnRaUpdatePerCause";

            //combined RA/LA inter SGSN routing area update
            const string attCombiInterSgsnRaUpdate= "attCombiInterSgsnRaUpdate";
            const string succCombiInterSgsnRaUpdate= "succCombiInterSgsnRaUpdate";
            const string failCombiInterSgsnRaUpdatePerCause= "failCombiInterSgsnRaUpdatePerCause";

            //PS paging
            const string nbrPsPagingGsm= "nbrPsPagingGsm";
            const string nbrPsPagingNoRspGsm= "nbrPsPagingNoRspGsm";
```

```
    const string nbrPsPagingUmts= "nbrPsPagingUmts";
    const string nbrPsPagingNoRspUmts= "nbrPsPagingNoRspUmts";
};

module sgsnRelocationMeasurement
{
    //inter SGSN relocation
    const string attInterSgsnReloc= "attInterSgsnReloc";
    const string succInterSgsnReloc= "succInterSgsnReloc";
    //inter SGSN combined relocation and hard handover
    const string attInterSgsnCombiReloc= "attInterSgsnCombiReloc";
    const string succInterSgsnCombiReloc= "succInterSgsnCombiReloc";
    //intra SGSN relocation
    const string attIntraSgsnReloc= "attIntraSgsnReloc";
    const string succIntraSgsnReloc= "succIntraSgsnReloc";
    //intra SGSN combined relocation and hard handover
    const string attIntraSgsnCombiReloc= "attIntraSgsnCombiReloc";
    const string succIntraSgsnCombiReloc= "succIntraSgsnCombiReloc";
};

module sgsnInterSystemHandoverMeasurement
{
    //intra SGSN handover from GSM to UMTS
    const string attIntraSgsnHoGsmToUmts= "attIntraSgsnHoGsmToUmts";
    const string succIntraSgsnHoGsmToUmts= "succIntraSgsnHoGsmToUmts";
    //intra SGSN handover from UMTS to GSM
    const string attIntraSgsnHoUmtsToGsm= "attIntraSgsnHoUmtsToGsm";
    const string succIntraSgsnHoUmtsToGsm= "succIntraSgsnHoUmtsToGsm";
    //inter SGSN handover from GSM to UMTS
    const string attInterSgsnHoGsmToUmts= "attInterSgsnHoGsmToUmts";
    const string succInterSgsnHoGsmToUmts= "succInterSgsnHoGsmToUmts";
    //inter SGSN handover from UMTS to GSM
    const string attInterSgsnHoUmtsToGsm= "attInterSgsnHoUmtsToGsm";
    const string succInterSgsnHoUmtsToGsm= "succInterSgsnHoUmtsToGsm";
};

module mapServiceMeasurement
```

```
{
    //request authentication set
    const string attReqAuthSetHlr= "attReqAuthSetHlr";
    const string succReqAuthSetWithQuintupletsHlr= "succReqAuthSetWithQuintupletsHlr";
    const string succReqAuthSetWithTripletsHlr= "succReqAuthSetWithTripletsHlr";
    const string succReqAuthSetWithEmptyRspHlr= "succReqAuthSetWithEmptyRspHlr";
    //GPRS update location
    const string attUpdateGprsLocationHlr= "attUpdateGprsLocationHlr";
    const string succUpdateGprsLocationHlr= "succUpdateGprsLocationHlr";
    //insert and delete subscriber data
    const string attInsertSubsDataHlr= "attInsertSubsDataHlr";
    const string attDeleteSubsDataHlr= "attDeleteSubsDataHlr";
};

module securityManagementMeasurement
{
    //P-TMSI reallocate
    const string attPtmsiRealloc= "attPtmsiRealloc";
    const string succPtmsiRealloc= "succPtmsiRealloc";

    //authentication request
    const string attAuthReq= "attAuthReq";
    const string succAuthReq= "succAuthReq";

    //identity request
    const string attIdentReq= "attIdentReq";
    const string succIdentReq= "succIdentReq";

    //security mode setup
    const string attSecMode= "attSecMode";
    const string succSecMode= "succSecMode";
};

module gtpInGnGpMeasurement
{
    //number of incoming and outgoing signalling packets
    const string nbrIncGtpCSigPkts= "nbrIncGtpCSigPkts";
```

```
        const string nbrOutGtpCSigPkts= "nbrOutGtpCSigPkts";

        //number of octets of incoming and outgoing signalling packets
        const string nbrIncGtpCSigOcts= "nbrIncGtpCSigOcts";
        const string nbrOutGtpCSigOcts= "nbrOutGtpCSigOcts";

        //number of abnormal signalling packets
        const string nbrOutGtpCSigPktsOverflow= "nbrOutGtpCSigPktsOverflow";
        const string nbrIncGtpCSigPktsError= "nbrIncGtpCSigPktsError";

        //number of incoming and outgoing data packets
        const string nbrIncGtpUDataPkts= "nbrIncGtpUDataPkts";
        const string nbrOutGtpUDataPkts= "nbrOutGtpUDataPkts";

        //number of octets of incoming and outgoing data packets
        const string nbrIncGtpUDataOcts= "nbrIncGtpUDataOcts";
        const string nbrOutGtpUDataOcts= "nbrOutGtpUDataOcts";

        //number of abnormal data packets
        const string nbrOutGtpUDataPktsOverflow= "nbrOutGtpUDataPktsOverflow";
        const string nbrIncGtpUDataPktsError= "nbrIncGtpUDataPktsError";
    };

    module shortMessageServiceMeasurement
    {
        //originating short messages
        const string attOrigSmsPs= "attOrigSmsPs";
        const string succOrigSmsPs= "succOrigSmsPs";

        //terminating short messages
        const string attTermSmsPs= "attTermSmsPs";
        const string succTermSmsPs= "succTermSmsPs";

        //MS present for short message
        const string attSmsMsPresentPs= "attSmsMsPresentPs";
        const string succSmsMsPresentPs= "succSmsMsPresentPs";
    //memory available for short message
    const string attSmsMemoryAvaiPs= "attSmsMemoryAvaiPs";
    const string succSmsMemoryAvailPs= "succSmsMemoryAvailPs";
};

//GGSN measurement
module ggsnThroughputMeasurement
{
    //number of incoming and outgoing data packets in Gn/Gp interface
    const string nbrIncDataPktsGnGp= "nbrIncDataPktsGnGp";
    const string nbrOutDataPktsGnGp= "nbrOutDataPktsGnGp";
    const string nbrIncDataOctsGnGp = "nbrIncDataOctsGnGp";
    const string nbrOutDataOctsGnGp = "nbrOutDataOctsGnGp";
    //number of incoming and outgoing signalling packets in Gn/Gp interface
    const string nbrIncSigPktsGnGp= "nbrIncSigPktsGnGp";
    const string nbrOutSigPktsGnGp= "nbrOutSigPktsGnGp";
    const string nbrIncSigOctsGnGp = "nbrIncSigOctsGnGp";
    const string nbrOutSigOctsGnGp = "nbrOutSigOctsGnGp";
    //number of incoming and outgoing data packets in Gi interface
    const string nbrOutDataPktsGi= "nbrOutDataPktsGi";
    const string nbrIncDataPktsGi= "nbrIncDataPktsGi";
    const string nbrOutDataOctsGi = "nbrOutDataOctsGi";
    const string nbrIncDataOctsGi = "nbrIncDataOctsGi";
};

//GGSN APN measurement
module apnSessionManagementMeasurement
{
    //active PDP context
    const string attActPdpContext= "attActPdpContext";
    const string succActPdpContext= "succActPdpContext";
    const string failActPdpContextUmtsPerCause = "failActPdpContextUmtsPerCause";
    //active dynamic PDP context
    const string attDynActPdpContext= "attDynActPdpContext";
    const string succDynActPdpContext= "succDynActPdpContext";
    //active PDP context with QoS parameter
    const string succActPdpContextQos= "succActPdpContextQos";
```

```
        //fail to active PDP context because of no resource available
        const string failActPdpContextNoResource= "failActPdpContextNoResource";
        //deactive PDP context by MS
        const string attDeactPdpContextMs= "attDeactPdpContextMs";
        const string succDeactPdpContextMs= "succDeactPdpContextMs";
        //Deactive PDP context by GGSN
        const string attDeactPdpContextGgsn= "attDeactPdpContextGgsn";

        const string succDeactPdpContextGgsn= "succDeactPdpContextGgsn";
        //number of actived PDP context by GGSN
        const string nbrActPdpContexts= "nbrActPdpContexts";
        const string meanActPdpContexts= "meanActPdpContexts";
        const string maxActPdpContexts= "maxActPdpContexts";
    };

    module apnThroughputMeasurement
    {
        const string nbrIncDataPktsGnGp = "nbrIncDataPktsGnGp";
        const string nbrOutDataPktsGnGp = "nbrOutDataPktsGnGp";
        const string nbrIncDataOctsGnGp = "nbrIncDataOctsGnGp";
        const string nbrOutDataOctsGnGp = "nbrOutDataOctsGnGp";
        const string nbrIncSigPktsGnGp = "nbrIncSigPktsGnGp";
        const string nbrOutSigPktsGnGp = "nbrOutSigPktsGnGp";
        const string nbrIncSigOctsGnGp = "nbrIncSigOctsGnGp";
        const string nbrOutSigOctsGnGp = "nbrOutSigOctsGnGp";
        const string nbrOutDataPktsGi = "nbrOutDataPktsGi";
        const string nbrIncDataPktsGi = "nbrIncDataPktsGi";
        const string nbrOutDataOctsGi = "nbrOutDataOctsGi";
        const string nbrIncDataOctsGi = "nbrIncDataOctsGi";
    };

    //for RNC measurement
    module rabAssignmentMeasurement
    {
        //assignment RAB
        const string attRabAssignEstabCsPerType= "attRabAssignEstabCsPerType";
        const string succRabAssignEstabCsPerType= "succRabAssignEstabCsPerType";
    const string failRabAssignEstabCsPerCause= "failRabAssignEstabCsPerCause";
    const string attRabAssignEstabPsPerType= "attRabAssignEstabPsPerType";
    const string succRabAssignEstabPsPerType= "succRabAssignEstabPsPerType";
    const string failRabAssignEstabPsPerCause= "failRabAssignEstabPsPerCause";
};
module rabReleaseRequestMeasurement
{
    const string nbrRncRelCsRabPerCause= "nbrRncRelCsRabPerCause";
    const string nbrRncRelPsRabPerCause= "nbrRncRelPsRabPerCause";
};
module iuConnectionMeasurement
{
    //establish Iu connection
    const string attRncEstabCsIuConn= "attRncEstabCsIuConn";
    const string attRncEstabPsIuConn= "attRncEstabPsIuConn";
    //request to release Iu connection
    const string nbrRncRelCsIuConnPerCause= "nbrRncRelCsIuConnPerCause";
    const string nbrRncRelPsIuConnPerCause= "nbrRncRelPsIuConnPerCause";
    //release Iu connection
    const string attRelCsIuConnPerCause = "attRelCsIuConnPerCause";
    const string attRelPsIuConnPerCause= "attRelPsIuConnPerCause";
};
module iuInterfaceMeasurement
{
    //Iu interface reset
    const string nbrResetCsByRncPerCause= "nbrResetCsByRncPerCause";
    const string nbrResetPsByRncPerCause= "nbrResetPsByRncPerCause";
    const string nbrResetCsByCnPerCause= "nbrResetCsByCnPerCause";
    const string nbrResetPsByCnPerCause= "nbrResetPsByCnPerCause";
    //Iu interface reset resource
    const string nbrResetResCsByRncPerCause= "nbrResetResCsByRncPerCause";
    const string nbrResetResPsByRncPerCause= "nbrResetResPsByRncPerCause";
    const string nbrResetResCsByCnPerCause= "nbrResetResCsByCnPerCause";
    const string nbrResetResPsByCnPerCause= "nbrResetResPsByCnPerCause";
    //Iu interface overload control
    const string nbrOverloadControlCsByRnc= "nbrOverloadControlCsByRnc";
    const string nbrOverloadControlPsByRnc= "nbrOverloadControlPsByRnc";
```

```
        const string nbrOverloadControlCsByCn= "nbrOverloadControlCsByCn";
        const string nbrOverloadControlPsByCn= "nbrOverloadControlPsByCn";
        //Iu interface error indication
        const string nbrErrorIndCsByRncPerCause= "nbrErrorIndCsByRncPerCause";
        const string nbrErrorIndPsByRncPerCause= "nbrErrorIndPsByRncPerCause";
        const string nbrErrorIndCsByCnPerCause= "nbrErrorIndCsByCnPerCause";
        const string nbrErrorIndPsByCnPerCause= "nbrErrorIndPsByCnPerCause";
};
module rncHardHandoverMeasurement
{
        const string attHho= "attHho";
        const string failHhoPerCause= "failHhoPerCause";
};
module rncRelocationMeasurement
{
        const string relocAttPrep= "relocAttPrep";
        const string relocSuccPrep= "relocSuccPrep";
        const string relocFailPrepCause= "relocFailPrepCause";
        const string relocSucc= "relocSucc";
};
module rncInterSystemHandoverMeasurement
{
        //relocation in RAT
        const string attRelocPrepOutRATHOCSPerCause= "attRelocPrepOutRATHOCSPerCause";
        const string failRelocPrepOutRATHOCSPerCause= "failRelocPrepOutRATHOCSPerCause";
        const string succRelocPrepOutRATHOCS = "succRelocPrepOutRATHOCS";
        //CS inter system handover from 3G to 2G
        const string iRATHOAttOutCS= "iRATHOAttOutCS";
        const string iRATHOFailOutCSCause= "iRATHOFailOutCSCause";
        const string iRATHOSuccOutCS= "iRATHOSuccOutCS";
        //CS inter system handover from 2G to 3G
        const string iRATHOAttIncCS= "iRATHOAttIncCS";
        const string iRATHOFailIncCSCause= "iRATHOFailIncCSCause";
        const string iRATHOSuccIncCS= "iRATHOSuccIncCS";
        //PS inter system handover from 3G to 2G
        const string iRATHOAttOutPSUTRAN= "iRATHOAttOutPSUTRAN";
        const string iRATHOFailOutPSUTRANCause= "iRATHOFailOutPSUTRANCause";
```

```
    const string iRATHOSuccOutPSUTRAN= "iRATHOSuccOutPSUTRAN";
    const string iRATHOSuccOutPSUE= "iRATHOSuccOutPSUE";
    //PS inter system handover from 2G to 3G
    const string iRATHOAttIncPS= "iRATHOAttIncPS";
    const string iRATHOSuccIncPS= "iRATHOSuccIncPS";
};
 //No.7 signalling measurement
module signallingPointTP
{
    const string nbrUsrUnavailRx= "nbrUsrUnavailRx";
    const string nbrUsrUnavailTx= "nbrUsrUnavailTx";
    const string nbrTraTx= "nbrTraTx";
    const string nbrTraRx= "nbrTraRx";
    const string nbrMsuDropRteErr= "nbrMsuDropRteErr";
};
module signallingLinkTP
{
    const string nbrChangeOverTx= "nbrChangeOverTx";
    const string nbrChangeOverRx= "nbrChangeOverRx";
    const string nbrChangeOverAckTx= "nbrChangeOverAckTx";
    const string nbrChangeOverAckRx= "nbrChangeOverAckRx";
    const string nbrChangeBackTx= "nbrChangeBackTx";
    const string nbrChangeBackRx= "nbrChangeBackRx";
    const string nbrChangeBackAckTx= "nbrChangeBackAckTx";
    const string nbrChangeBackAckRx= "nbrChangeBackAckRx";
    const string nbrLnkInhDenTx= "nbrLnkInhDenTx";
    const string nbrLnkInhDenRx= "nbrLnkInhDenRx";
    const string nbrLnkForceUninhTx= "nbrLnkForceUninhTx";
    const string nbrLnkForceUninhRx= "nbrLnkForceUninhRx";
    const string nbrLnkLocInhTstTx= "nbrLnkLocInhTstTx";
    const string nbrLnkLocInhTstRx= "nbrLnkLocInhTstRx";
    const string nbrLnkRmtInhTstTx= "nbrLnkRmtInhTstTx";
    const string nbrLnkRmtInhTstRx= "nbrLnkRmtInhTstRx";
    const string nbrLnkConOrdTx= "nbrLnkConOrdTx";
    const string nbrLnkConOrdRx = "nbrLnkConOrdRx";
    const string nbrLnkConAckTx= "nbrLnkConAckTx";
    const string nbrLnkConAckRx= "nbrLnkConAckRx";
```

```
        const string nbrLnkTstRx= "nbrLnkTstRx";
        const string nbrLnkTstTx= "nbrLnkTstTx";
        const string nbrLnkTstAckRx= "nbrLnkTstAckRx";
        const string nbrLnkTstAckTx= "nbrLnkTstAckTx";
        const string nbrTxDrop= "nbrTxDrop";
        const string nbrTxCongDrop= "nbrTxCongDrop";
        const string nbrSifOctTx= "nbrSifOctTx";
        const string nbrSifOctRx= "nbrSifOctRx";
        const string nbrSioOctTx= "nbrSioOctTx";
        const string nbrSioOctRx= "nbrSioOctRx";
        const string nbrMsuTx= "nbrMsuTx";
        const string nbrMsuRx= "nbrMsuRx";
        const string nbrCong1= "nbrCong1";
        const string nbrCong2= "nbrCong2";
        const string nbrCong3= "nbrCong3";
        const string durSigLinkOutOfService= "durSigLinkOutOfService";
        const string durLnkCong= "durLnkCong";
        const string nbrLnkErrPduRx= "nbrLnkErrPduRx";
    };
    module signallingLinkSetTP
    {
        const string durLnkSetUnav= "durLnkSetUnav";
    };

    //for UtranCell measurement
    module cellRrcConnectionMeasurement
    {
        const string rrcAttConnEstabCause= "rrcAttConnEstabCause";
        const string rrcFailConnEstabCause= "rrcFailConnEstabCause";
        const string rrcSuccConnEstabCause= "rrcSuccConnEstabCause";
        const string rrcAttConnReEstab= "rrcAttConnReEstab";
        const string rrcFailConnReEstabCause= "rrcFailConnReEstabCause";
        const string rrcSuccConnReEstab= "rrcSuccConnReEstab";
        const string rrcAttConnRelDCCHCause= "rrcAttConnRelDCCHCause";
        const string rrcAttConnRelCCCHCause= "rrcAttConnRelCCCHCause";
    };
    //for UtranRelation measurement
    module hardHandoverInterCellIntraNodeBMeasurement
    {
        const string hHOAttOutIntraNodeB= "hHOAttOutIntraNodeB";
        const string hHOFailOutIntraNodeBCause= "hHOFailOutIntraNodeBCause";
        const string hHOSuccOutIntraNodeB= "hHOSuccOutIntraNodeB";
    };
    module hardHandoverInterNodeBIntraRncMeasurement
    {
        const string hHOAttOutInterNodeBIntraRNC= "hHOAttOutInterNodeBIntraRNC";
        const string hHOFailOutInterNodeBIntraRNCCause=
"hHOFailOutInterNodeBIntraRNCCause";
        const string hHOSuccOutInterNodeBIntraRNC= "hHOSuccOutInterNodeBIntraRNC";
    };
    module hardHandoverInterRncViaIurMeasurement
    {
        const string hHOAttOutInterRNCIur= "hHOAttOutInterRNCIur";
        const string hHOSuccOutInterRNCIur= "hHOSuccOutInterRNCIur";
        const string hHOFailOutInterRNCIurCause= "hHOFailOutInterRNCIurCause";
    };
    module hardHandoverInterRncMeasurement
    {
        const string attHhoOutInterRncCn= "attHhoOutInterRncCn";
        const string failHhoOutInterRncCnPerCause= "failHhoOutInterRncCnPerCause";
    };

    module hardHandoverInterSystemMeasurement
    {
        //relocation in RAT
        const string attRelocPrepOutRATHOCSPerCause= "attRelocPrepOutRATHOCSPerCause";
        const string failRelocPrepOutRATHOCSPerCause= "failRelocPrepOutRATHOCSPerCause";
        const string succRelocPrepOutRATHOCS = "succRelocPrepOutRATHOCS";
        //CS inter system handover from 3G to 2G
        const string iRATHOAttOutCS= "iRATHOAttOutCS";
        const string iRATHOFailOutCSCause= "iRATHOFailOutCSCause";
        const string iRATHOSuccOutCS= "iRATHOSuccOutCS";
        //CS inter system handover from 2G to 3G
        const string iRATHOAttIncCS= "iRATHOAttIncCS";
```

```
            const string iRATHOFailIncCSCause= "iRATHOFailIncCSCause";
            const string iRATHOSuccIncCS= "iRATHOSuccIncCS";
            //PS inter system handover from 3G to 2G
            const string iRATHOAttOutPSUTRAN= "iRATHOAttOutPSUTRAN";
            const string iRATHOFailOutPSUTRANCause= "iRATHOFailOutPSUTRANCause";
            const string iRATHOSuccOutPSUTRAN= "iRATHOSuccOutPSUTRAN";
            const string iRATHOSuccOutPSUE= "iRATHOSuccOutPSUE";
            //PS inter system handover from 2G to 3G
            const string iRATHOAttIncPS= "iRATHOAttIncPS";
            const string iRATHOSuccIncPS= "iRATHOSuccIncPS";
        };
    };
    #endif
```

5.2 数据类型的 IDL 定义

TDSCDMANRMMeasurementSystem.idl

```
    //File "TDSCDMANRMMeasurementSystem.idl"
    #ifndef TDSCDMANRMMeasurementSystem_idl
    #define TDSCDMANRMMeasurementSystem_idl

    // #pragma prefix "3gppsa5.org"

     /**
     * This module defines type definitions for performance mesurements
     */
    module TDSCDMANRMMeasurementSystem
    {

        typedef   unsigned long   TDSCDMAMeasurementType1;
        typedef   float   TDSCDMAMeasurementType2;

        // The following RANAP causes are defined in the section 9.2.1.4 of 3GPP TS 25.413 v5.5.0.
        typedef   unsigned short   RANAPCause;

        //Radio Network Layer Cause. Value range is 1 - 64.
        const RANAPCause rabPreempted = 1;
        const RANAPCause trelocoverallExpiry = 2;
```

```
const RANAPCause trelocprepExpiry = 3;
const RANAPCause treloccompleteExpiry = 4;
const RANAPCause tqueingExpiry = 5;
const RANAPCause relocationTriggered = 6;
const RANAPCause trelocallocExpiry   = 7;
const RANAPCause unableToEstablishDuringRelocation = 8;
const RANAPCause unknownTargetRnc = 9;
const RANAPCause relocationCancelled = 10;
const RANAPCause successfulRelocation = 11;
const RANAPCause requestedCipheringAndOrIntegrityProtectionAlgorithmsNotSupported = 12;
const RANAPCause conflictWithAlreadyExistingIntegrityProtectionAndOrCipheringInformation = 13;
const RANAPCause failureInTheRadioInterfaceProcedure = 14;
const RANAPCause releaseDueToUtranGeneratedReason = 15;
const RANAPCause userInactivity_RANAP = 16;
const RANAPCause timeCriticalRelocation = 17;
const RANAPCause requestedTrafficClassNotAvailable = 18;
const RANAPCause invalidRABParametersValue = 19;
const RANAPCause requestedMaximumBitRateNotAvailable = 20;
const RANAPCause requestedGuaranteedBitRateNotAvailable = 21;
const RANAPCause requestedTransferDelayNotAchievable = 22;
const RANAPCause invalidRabParametersCombination = 23;
const RANAPCause conditionViolationForSduParameters = 24;
const RANAPCause conditionViolationForTrafficHandlingPriority = 25;
const RANAPCause conditionViolationForGuaranteedBitRate = 26;
const RANAPCause userPlaneVersionsNotSupported = 27;
const RANAPCause iuUpFailure = 28;
const RANAPCause relocationFailureInTargetCnRncOrTargetSystem = 29;
const RANAPCause invalidRabId = 30;
const RANAPCause noRemainingRab = 31;
const RANAPCause interactionWithOtherProcedure = 32;
const RANAPCause requestedMaximumBitRateForDlNotAvailable = 33;
const RANAPCause requestedMaximumBitRateForUlNotAvailable = 34;
const RANAPCause requestedGuaranteedBitRateForDlNotAvailable = 35;
const RANAPCause requestedGuaranteedBitRateForUlNotAvailable = 36;
const RANAPCause repeatedIntegrityCheckingFailure = 37;
const RANAPCause requestedRequestTypeNotSupported = 38;
const RANAPCause requestSuperseded = 39;
const RANAPCause releaseDueToUeGeneratedSignallingConnectionRelease = 40;
const RANAPCause resourceOptimisationRelocation = 41;
const RANAPCause requestedInformationNotAvailable = 42;
const RANAPCause relocationDesirableForRadioReasons = 43;
const RANAPCause relocationNotSupportedInTargetRncOrTargetSystem = 44;
const RANAPCause directedRetry = 45;
const RANAPCause radioConnectionWithUeLost = 46;
const RANAPCause rncUnableToEstablishAllRfcs = 47;
const RANAPCause decipheringKeysNotAvailable = 48;
const RANAPCause dedicatedAssistanceDataNotAvailable = 49;
const RANAPCause relocationTargetNotAllowed = 50;
const RANAPCause locationReportingCongestion = 51;
const RANAPCause reduceLoadInServingCell = 52;
const RANAPCause noRadioResourcesAvailableInTargetCell = 53;
const RANAPCause geranIuModeFailure = 54;
const RANAPCause accessRestrictedDueToSharedNetworks = 55;
const RANAPCause incomingRelocationNotSupportedDueToPuesbineFeature = 56;
//Transport Layer Cause. Value range is 65 - 80.
const RANAPCause signallingTransportResourceFailure = 65;
const RANAPCause iuTransportConnectionFailedToEstablish = 66;

//NAS Cause. Value range is 81 - 96.
const RANAPCause userRestrictionStartIndication = 81;
const RANAPCause userRestrictionEndIndication = 82;
const RANAPCause normalRelease = 83;

//Protocol Cause. Value range is 97 - 112.
const RANAPCause transferSyntaxError_RANAP = 97;
const RANAPCause semanticError_RANAP = 98;
const RANAPCause messageNotCompatibleWithReceiverState_RANAP = 99;
const RANAPCause abstractSyntaxErrorReject_RANAP = 100;
const RANAPCause abstractSyntaxErrorIgnoreAndNotify_RANAP = 101;
const RANAPCause abstractSyntaxErrorFalselyConstructedMessage_RANAP = 102;

//Miscellaneous Cause. Value range is 113 - 128.
const RANAPCause operationAndMaintenanceIntervention_RANAP = 113;
const RANAPCause noResourceAvailable = 114;
const RANAPCause unspecifiedFailure = 115;
const RANAPCause networkOptimisation = 116;

//Non-standard Cause. Value range is 129 - 256. Cause value 256 shall not be used.


// The following RNSAP causes are defined in the section 9.2.1.5 of 3GPP TS 25.423 v5.6.0.
typedef   unsigned short   RNSAPCause;

//Radio Network Layer Cause.
const RNSAPCause unknownCid_RNSAP = 1;
const RNSAPCause cellNotAvailable_RNSAP = 2;
const RNSAPCause powerLevelNotSupported_RNSAP = 3;
const RNSAPCause ulScramblingCodeAlreadyInUse = 4;
const RNSAPCause dlRadioResourcesNotAvailable_RNSAP = 5;
const RNSAPCause ulRadioResourcesNotAvailable_RNSAP = 6;
const RNSAPCause measurementNotSupportedForTheObject_RNSAP = 7;
const RNSAPCause combiningResourcesNotAvailable_RNSAP = 8;
const RNSAPCause combiningNotSupported_RNSAP = 9;
const RNSAPCause reconfigurationNotAllowed = 10;
const RNSAPCause requestedConfigurationNotSupported_RNSAP = 11;
const RNSAPCause synchronisationFailure = 12;
const RNSAPCause requestedTxDiversityModeNotSupported_RNSAP = 13;
const RNSAPCause measurementTemporarilyNotAvailable_RNSAP = 14;
const RNSAPCause unspecified_RNL_RNSAP = 15;
const RNSAPCause invalidCmSettings = 16;
const RNSAPCause reconfigurationCfnNotElapsed_RNSAP = 17;
const RNSAPCause numberOfDLCodesNotSupported_RNSAP = 18;
const RNSAPCause dedicatedTransportChannelTypeNotSupported_RNSAP = 19;
const RNSAPCause dlSharedChannelTypeNotSupported = 20;
const RNSAPCause ulSharedChannelTypeNotSupported = 21;
const RNSAPCause commonTransportChannelTypeNotSupported_RNSAP = 22;
const RNSAPCause ulSpreadingFactorNotSupported = 23;
const RNSAPCause dlSpreadingFactorNotSupported = 24;
const RNSAPCause cmNotSupported_RNSAP = 25;
const RNSAPCause transactionNotSupportedByDestinationNodeB = 26;
const RNSAPCause rlAlreadyActivatedAllocated_RNSAP = 27;
const RNSAPCause numberOfUICodesNotSupported_RNSAP = 28;
const RNSAPCause cellReservedForOperatorUse = 29;
const RNSAPCause dpcModeChangeNotSupported_RNSAP = 30;
const RNSAPCause informationTemporarilyNotAvailable_RNSAP = 31;
const RNSAPCause informationProvisionNotSupportedForTheObject_RNSAP = 32;
const RNSAPCause powerBalancingStatusNotCompatible_RNSAP = 33;
const RNSAPCause delayedActivationNotSupported_RNSAP = 34;
const RNSAPCause rlTimingAdjustmentNotSupported_RNSAP = 35;
const RNSAPCause unknownRnti = 36;

//Transport Layer Cause.
const RNSAPCause transportResourceUnavailable_RNSAP = 37;
const RNSAPCause unspecified_TL_RNSAP = 38;

//Protocol Cause.
const RNSAPCause transferSyntaxError_RNSAP = 39;
const RNSAPCause abstractSyntaxErrorReject_RNSAP = 40;
const RNSAPCause abstractSyntaxErrorIgnoreAndNotify_RNSAP = 41;
const RNSAPCause messageNotCompatibleWithReceiverState_RNSAP = 42;
const RNSAPCause semanticError_RNSAP = 43;
const RNSAPCause unspecified_Protocol_RNSAP = 44;
const RNSAPCause abstractSyntaxErrorFalselyConstructedMessage_RNSAP = 45;

//Miscellaneous Cause.
const RNSAPCause controlProcessingOverload_RNSAP = 46;
const RNSAPCause hardwareFailure_RNSAP = 47;
const RNSAPCause operationAndMaintenanceIntervention_RNSAP = 48;
const RNSAPCause notEnoughUserPlaneProcessingResources_RNSAP = 49;
const RNSAPCause Unspecified_Misc_RNSAP = 50;

// The following NBAP causes are defined in the section 9.2.1.6 of 3GPP TS 25.433 v5.5.0.
typedef   unsigned short   NBAPCause;

//Radio Network Layer Cause.
const NBAPCause unknownCid_NBAP = 1;
const NBAPCause cellNotAvailable_NBAP = 2;
const NBAPCause powerLevelNotSupported_NBAP = 3;
const NBAPCause dlRadioResourcesNotAvailable_NBAP = 4;
const NBAPCause ulRadioResourcesNotAvailable_NBAP = 5;
const NBAPCause rlAlreadyActivatedAllocated_NBAP = 6;
const NBAPCause nodeBResourcesUnavailable = 7;
const NBAPCause measurementNotSupportedForTheObject_NBAP = 8;
const NBAPCause combiningResourcesNotAvailable_NBAP = 9;
const NBAPCause requestedConfigurationNotSupported_NBAP = 10;
const NBAPCause synchronizationFailure = 11;
const NBAPCause priorityTransportChannelEstablished = 12;
const NBAPCause sibOriginationInNodeBNotSupported = 13;
const NBAPCause requestedTxDiversityModeNotSupported_NBAP = 14;
const NBAPCause unspecified_RNL_NBAP = 15;
const NBAPCause bcchSchedulingError = 16;
const NBAPCause measurementTemporarilyNotAvailable_NBAP = 17;
const NBAPCause invalidCmSetting = 18;
const NBAPCause reconfigurationCfnNotElapsed_NBAP = 19;
const NBAPCause numberOfDlCodesNotSupported_NBAP = 20;
const NBAPCause scpichNotSupported = 21;
const NBAPCause combiningNotSupported_NBAP = 22;
const NBAPCause ulSfNotSupported = 23;
const NBAPCause dlSfNotSupported = 24;
const NBAPCause commonTransportChannelTypeNotSupported_NBAP = 25;
const NBAPCause dedicatedTransportChannelTypeNotSupported_NBAP = 26;
const NBAPCause downlinkSharedChannelTypeNotSupported = 27;
const NBAPCause uplinkSharedChannelTypeNotSupported = 28;
const NBAPCause cmNotSupported_NBAP = 29;
const NBAPCause txDiversityNoLongerSupported = 30;
const NBAPCause unknownLocalCellId = 31;
const NBAPCause numberOfUlCodesNotSupported_NBAP = 32;
const NBAPCause informationTemporarilyNotAvailable_NBAP = 33;
const NBAPCause informationProvisionNotSupportedForTheObject_NBAP = 34;
const NBAPCause cellSynchronisationNotSupported = 35;
const NBAPCause cellSynchronisationAdjustmentNotSupported = 36;
const NBAPCause dpcModeChangeNotSupported_NBAP = 37;
const NBAPCause ipdlAlreadyActivated = 38;
const NBAPCause ipdlNotSupported = 39;
const NBAPCause ipdlParametersNotAvailable = 40;
const NBAPCause frequencyAcquisitionNotSupported = 41;
const NBAPCause powerBalancingStatusNotCompatible_NBAP = 42;
const NBAPCause requestedTypeOfBearerRearrangementNotSupported = 43;
const NBAPCause signallingBearerRearrangementNotSupported = 44;
const NBAPCause bearerRearrangementNeeded = 45;
const NBAPCause delayedActivationNotSupported_NBAP = 46;
const NBAPCause rlTimingAdjustmentNotSupported_NBAP = 47;

//Transport Layer Cause.
const NBAPCause transportResourceUnavailable_NBAP = 48;
const NBAPCause unspecified_TL_NBAP = 49;

//Protocol Cause.
const NBAPCause transferSyntaxError_NBAP = 50;
const NBAPCause abstractSyntaxErrorReject_NBAP = 51;
const NBAPCause abstractSyntaxErrorIgnoreAndNotify_NBAP = 52;
const NBAPCause messageNotCompatibleWithReceiverState_NBAP = 53;
const NBAPCause semanticError_NBAP = 54;
const NBAPCause Unspecified_Protocol_NBAP = 55;
const NBAPCause abstractSyntaxErrorFalselyConstructedMessage_NBAP = 56;

//Miscellaneous Cause.
const NBAPCause controlProcessingOverload_NBAP = 57;
const NBAPCause hardwareFailure_NBAP = 58;
const NBAPCause operationAndMaintenanceIntervention_NBAP = 59;
const NBAPCause notEnoughUserPlaneProcessingResources_NBAP = 60;
const NBAPCause unspecified_Misc_NBAP = 61;

// The following cell update causes are defined in the section 10.3.3.3 of 3GPP TS 25.331 v5.5.0.
typedef    unsigned short    CellUpdateCause;
const CellUpdateCause cellReselection = 1;
const CellUpdateCause periodicalCellUpdate = 2;
const CellUpdateCause uplinkDataTransmission = 3;
const CellUpdateCause pagingResponse = 4;
const CellUpdateCause reenteredServiceArea = 5;
const CellUpdateCause radioLinkFailure = 6;
const CellUpdateCause rlcUnrecoverableError = 7;
```

```
// The following establishment causes are defined in the section 10.3.3.11 of 3GPP TS 25.331 v5.5.0.
typedef   unsigned short   EstablishmentCause;
const EstablishmentCause originatingConversationalCall = 1;
const EstablishmentCause originatingStreamingCall = 2;
const EstablishmentCause originatingInteractiveCall = 3;
const EstablishmentCause originatingBackgroundCall = 4;
const EstablishmentCause originatingSubscribedTrafficCall = 5;
const EstablishmentCause terminatingConversationalCall = 6;
const EstablishmentCause terminatingStreamingCall = 7;
const EstablishmentCause terminatingInteractiveCall = 8;
const EstablishmentCause terminatingBackgroundCall = 9;
const EstablishmentCause emergencyCall = 10;
const EstablishmentCause interRatCellReselection = 11;
const EstablishmentCause interRatCellChangeOrder = 12;
const EstablishmentCause registration = 13;
const EstablishmentCause detach = 14;
const EstablishmentCause originatingHighPrioritySignalling = 15;
const EstablishmentCause originatingLowPrioritySignalling = 16;
const EstablishmentCause callReestablishment = 17;
const EstablishmentCause terminatingHighPrioritySignalling = 18;
const EstablishmentCause terminatingLowPrioritySignalling = 19;
const EstablishmentCause terminatingCauseUnknown = 20;

// The following failure causes are defined in the section 10.3.3.13 of 3GPP TS 25.331 v5.5.0.
typedef   unsigned short   FailureCause;
const FailureCause configurationUnsupported = 1;
const FailureCause physicalChannelFailure_Failure = 2;
const FailureCause incompatibleSimultaneousReconfiguration = 3;
const FailureCause protocolError_Failure = 4;
const FailureCause compressedModeRuntimeError = 5;
const FailureCause cellUpdateOccurred = 6;
const FailureCause invalidConfiguration = 7;
const FailureCause configurationIncomplete = 8;
const FailureCause unsupportedMeasurement = 9;

// The following rejection causes are defined in the section 10.3.3.31 of 3GPP TS 25.331 v5.5.0.
```

```
typedef    unsigned short    RejectionCause;
const RejectionCause congestion_Reject = 1;
const RejectionCause unspecified_Reject = 2;

// The following release causes are defined in the section 10.3.3.32 of 3GPP TS 25.331 v5.5.0.
typedef    unsigned short    ReleaseCause;
const ReleaseCause normalEvent = 1;
const ReleaseCause preemptiveRelease = 2;
const ReleaseCause congestion_Release = 3;
const ReleaseCause reestablishmentReject = 4;
const ReleaseCause userInactivity_Release = 5;
const ReleaseCause directedSignallingConnectionReestablishment = 6;
const ReleaseCause unspecified_Release = 7;


// The following inter-RAT change failure causes are defined in the section 10.3.8.5 of 3GPP TS
25.331 v5.5.0.
typedef    unsigned short    InterRatChangeFailureCause;
const InterRatChangeFailureCause configurationUnacceptable_IRATChange = 1;
const InterRatChangeFailureCause physicalChannelFailure_IRATChange = 2;
const InterRatChangeFailureCause protocolError_IRATChange = 3;
const InterRatChangeFailureCause unspecified_IRATChange = 4;

// The following inter-RAT handover failure causes are defined in the section 10.3.8.6 of 3GPP TS
25.331 v5.5.0.
typedef    unsigned short    InterRatHandoverFailureCause;
const InterRatHandoverFailureCause configurationUnacceptable_IRATHo = 1;
const InterRatHandoverFailureCause physicalChannelFailure_IRATHo = 2;
const InterRatHandoverFailureCause protocolError_IRATHo = 3;
const InterRatHandoverFailureCause interRatProtocolError = 4;
const InterRatHandoverFailureCause unspecified_IRATHo = 5;

//The following call failure causes are used in the category "mobileTrafficFlow".
typedef    unsigned short    CallFailureCause;
const CallFailureCause callingPartAuthFail = 1;
const CallFailureCause callingPartCipherModeFail = 2;
const CallFailureCause interfaceABusy = 3;
const CallFailureCause callingPartAssignFail = 4;
const CallFailureCause exchangeCongestion = 5;
const CallFailureCause userEarlyRelease = 6;
const CallFailureCause calledPartAssignFail = 7;
const CallFailureCause calledPartDetermineBusy = 8;
const CallFailureCause userUnreachable = 9;
const CallFailureCause alertingEarlyRelease = 10;
const CallFailureCause outCircuitOverflow = 11;
const CallFailureCause calledPartBusy = 12;
const CallFailureCause noAnswer = 13;

//The following Imsi attatch failure causes are defined in the section 10.5.3.6 of 3GPP TS 24.008 v6.1.0.
typedef unsigned short ImsiAttatchFailureCause;
const ImsiAttatchFailureCause imsiUnknownInHLR_Imsi = 2;
const ImsiAttatchFailureCause illegalMS_Imsi = 3;
const ImsiAttatchFailureCause imsiUnknownInVLR = 4;
const ImsiAttatchFailureCause imeiNotAccepted = 5;
const ImsiAttatchFailureCause illegalME_Imsi = 6;
const ImsiAttatchFailureCause plmnNotAllowed_Imsi = 11;
const ImsiAttatchFailureCause locationAreaNotAllowed_Imsi =12;
const ImsiAttatchFailureCause roamingNotAllowedInThisLocationArea_Imsi = 13;
const ImsiAttatchFailureCause noSuitableCellsInLocationArea_Imsi = 15;
const ImsiAttatchFailureCause networkFailure_Imsi = 17;
const ImsiAttatchFailureCause macFailure_Imsi = 20;
const ImsiAttatchFailureCause synchFailure_Imsi = 21;
const ImsiAttatchFailureCause congestion_Imsi = 22;
const ImsiAttatchFailureCause gsmAuthenticationUnacceptable_Imsi = 23;
const ImsiAttatchFailureCause serviceOptionNotSupported_Imsi = 32;
const ImsiAttatchFailureCause requestedServiceOptionNotSubscribed_Imsi = 33;
const ImsiAttatchFailureCause serviceOptionTemporarilyOutOfOrder_Imsi = 34;
const ImsiAttatchFailureCause callCannotBeIdentified = 38;
const ImsiAttatchFailureCause failRetryUponEntryIntoANewCell_Imsi = 48;
//value range 48 - 63 is used to retry upon entry into a new cell;
const ImsiAttatchFailureCause semanticallyIncorrectMessage_Imsi = 95;
const ImsiAttatchFailureCause invalidMandatoryInformation_Imsi = 96;
const ImsiAttatchFailureCause messageTypeNon_existentOrNotImplemented_Imsi = 97;
const ImsiAttatchFailureCause messageTypeNotCompatibleWithTheProtocolState_Imsi = 98;
const ImsiAttatchFailureCause informationElementNon_existentOrNotImplemented_Imsi = 99;
const ImsiAttatchFailureCause conditionalIeError_Imsi = 100;
const ImsiAttatchFailureCause messageNotCompatibleWithTheProtocolState_Imsi = 101;
const ImsiAttatchFailureCause protocolError_Imsi = 111; // unspecified
//The following activate PDP context MS failure causes are defined in the section 10.5.6.6 of 3GPP TS 24.008 v6.1.0.
typedef unsigned short ActPdpContextMsFailureCause;
const ActPdpContextMsFailureCause operatorDeterminedBarring_Ms = 8;
const ActPdpContextMsFailureCause llcOrSndcpFailure= 25;
const ActPdpContextMsFailureCause insufficientResources = 26;
const ActPdpContextMsFailureCause unknownOrMissingAccessPointName = 27;
const ActPdpContextMsFailureCause unknownPdpAddressOrPdpType_Ms = 28;
const ActPdpContextMsFailureCause userAuthenticationFailed_Ms = 29;
const ActPdpContextMsFailureCause activationRejectedByGgsn = 30;
const ActPdpContextMsFailureCause activationRejected = 31; //unspecified
const ActPdpContextMsFailureCause serviceOptionNotSupported_Ms = 32;
const ActPdpContextMsFailureCause requestedServiceOptionNotSubscribed_Ms = 33;//redefined
const ActPdpContextMsFailureCause serviceOptionTemporarilyOutOfOrder_Ms = 34;//redefined
const ActPdpContextMsFailureCause nsapiAlreadyUsed = 35;
const ActPdpContextMsFailureCause regularPdpContextDeactivation = 36;
const ActPdpContextMsFailureCause qosNotAccepted = 37;
const ActPdpContextMsFailureCause networkFailure_Ms = 38;
const ActPdpContextMsFailureCause reactivationRequested = 39;
const ActPdpContextMsFailureCause featureNotSupported = 40;
const ActPdpContextMsFailureCause semanticErrorInTheTftOperation_Ms = 41;
const ActPdpContextMsFailureCause syntacticalErrorInTheTftOperation = 42;
const ActPdpContextMsFailureCause unknownPdpContext = 43;
const ActPdpContextMsFailureCause semanticErrorsInPacketFilters_Ms= 44;
const ActPdpContextMsFailureCause syntacticalErrorInPacketFilters= 45;
const ActPdpContextMsFailureCause PdpContextWithoutTftAlreadyActivated_Ms = 46;
const ActPdpContextMsFailureCause InvalidTransactionIdentifierValue = 81;
const ActPdpContextMsFailureCause semanticallyIncorrectMessage_Ms = 95;
//const ActPdpContextMsFailureCause invalidMandatoryInformation_Ms = 96;//redefined
const ActPdpContextMsFailureCause messageTypeNon_existentOrNotImplemented_Ms = 97;
const ActPdpContextMsFailureCause messageTypeNotCompatibleWithTheProtocolState_Ms = 98;
const ActPdpContextMsFailureCause informationElementNon_existentOrNotImplemented_Ms = 99;
```

```
    const ActPdpContextMsFailureCause conditionalIeError_Ms = 100;
    const ActPdpContextMsFailureCause messageNotCompatibleWithTheProtocolState_Ms = 101;
    const ActPdpContextMsFailureCause protocolError_Ms = 111; // unspecified

    //The following activate PDP context UMTS failure causes are defined in the section 7.7.1 of 3GPP
TS 29.060 v6.1.0. and 3GPP TS 32.215 v5.4.0.
    typedef unsigned short ActPdpContextUtmsFailureCause;
    const ActPdpContextUtmsFailureCause non_existent = 192;
    const ActPdpContextUtmsFailureCause invalidMessageFormat = 193;
    const ActPdpContextUtmsFailureCause imsiNotKnown = 194;
    const ActPdpContextUtmsFailureCause msIsGprsDetached = 195;
    const ActPdpContextUtmsFailureCause msIsNotGprsResponding = 196;
    const ActPdpContextUtmsFailureCause msRefuses = 197;
    const ActPdpContextUtmsFailureCause versionNotSupported = 198;
    const ActPdpContextUtmsFailureCause noResourcesAvailable = 199;
    const ActPdpContextUtmsFailureCause serviceNotSupported = 200;
    const ActPdpContextUtmsFailureCause mandatoryIeIncorrect = 201;
    const ActPdpContextUtmsFailureCause mandatoryIeMissing = 202;
    const ActPdpContextUtmsFailureCause optionalIeIncorrect = 203;
    const ActPdpContextUtmsFailureCause systemFailure = 204;
    const ActPdpContextUtmsFailureCause roamingRestriction = 205;
    const ActPdpContextUtmsFailureCause p_tmsiSignatureMismatch = 206;
    const ActPdpContextUtmsFailureCause gprsConnectionSuspended = 207;
    const ActPdpContextUtmsFailureCause authenticationFailure = 208;
    const ActPdpContextUtmsFailureCause userAuthenticationFailed_Utms = 209;
    const ActPdpContextUtmsFailureCause contextNotFound = 210;
    const ActPdpContextUtmsFailureCause allDynamicPdpAddressesAreOccupied = 211;
    const ActPdpContextUtmsFailureCause noMemoryIsAvailable = 212;
    const ActPdpContextUtmsFailureCause relocationFailure = 213;
    const ActPdpContextUtmsFailureCause unknownMandatoryExtensionHeader = 214;
    const ActPdpContextUtmsFailureCause semanticErrorInTheTftOperation_Utms = 215;
    const ActPdpContextUtmsFailureCause syntacticErrorInTheTftOperation = 216;
    const ActPdpContextUtmsFailureCause semanticErrorsInPacketFilters_Utms= 217;
    const ActPdpContextUtmsFailureCause syntacticErrorsInPacketFilters= 218 ;
    const ActPdpContextUtmsFailureCause missingOrUnknownApn = 219;
    const ActPdpContextUtmsFailureCause unknownPdpAddressOrPdpType_Utms = 220;
    const ActPdpContextUtmsFailureCause pdpContextWithoutTftAlreadyActivated_Utms = 221;
    const ActPdpContextUtmsFailureCause apnAccessDenied_noSubscription = 222;
```

```
//value range 223-240 is for future use;
//value range 241-255 is reserved for GPRS charging protocol use;
const ActPdpContextUtmsFailureCause requestRelatedToPossiblyDuplicatedPacketsAlreadyFulfilled = 252;
const ActPdpContextUtmsFailureCause requestAlreadyFulfilled = 253;
const ActPdpContextUtmsFailureCause sequenceNumbersOfReleasedOrCancelledPacketsIeIncorrect = 254;
const ActPdpContextUtmsFailureCause requestNotFulfilled = 255;

//The following GPRS attatch failure causes are defined in the section 10.5.5.14 of 3GPP TS 24.008 v6.1.0.
typedef unsigned short gprsAttathFailureCause;
const gprsAttathFailureCause imsiUnknownInHLR_Gprs = 2;
const gprsAttathFailureCause illegalMS_Gprs = 3;
const gprsAttathFailureCause illegalME_Gprs = 6;
const gprsAttathFailureCause gprsServicesNotAllowed = 7;
const gprsAttathFailureCause gprsServicesAndNon_GprsServicesNotAllowed = 8;
const gprsAttathFailureCause msIdentityCannotBeDerivedByTheNetwork = 9;
const gprsAttathFailureCause implicitlyDetached = 10;
const gprsAttathFailureCause plmnNotAllowed_Gprs = 11;
const gprsAttathFailureCause locationAreaNotAllowed_Gprs =12;
const gprsAttathFailureCause roamingNotAllowedInThisLocationArea_Gprs = 13;
const gprsAttathFailureCause noSuitableCellsInLocationArea_Gprs = 15;
const gprsAttathFailureCause networkFailure_Gprs = 17;
const gprsAttathFailureCause macFailure_Gprs = 20;
const gprsAttathFailureCause synchFailure_Gprs = 21;
const gprsAttathFailureCause congestion_Gprs = 22;
const gprsAttathFailureCause gsmAuthenticationUnacceptable_Gprs = 23;
const gprsAttathFailureCause NoPdpContextActivated = 40;
const gprsAttathFailureCause failRetryUponEntryIntoANewCell_Gprs = 48;
//value range 48 - 63 is used to retry upon entry into a new cell;
const gprsAttathFailureCause semanticallyIncorrectMessage_Gprs = 95;
const gprsAttathFailureCause invalidMandatoryInformation_Gprs = 96;
const gprsAttathFailureCause messageTypeNon_existentOrNotImplemented_Gprs = 97;
const gprsAttathFailureCause messageTypeNotCompatibleWithTheProtocolState_Gprs = 98;
const gprsAttathFailureCause informationElementNon_existentOrNotImplemented_Gprs = 99
const gprsAttathFailureCause conditionalIeError_Gprs = 100;
const gprsAttathFailureCause messageNotCompatibleWithTheProtocolState_Gprs = 101;
const gprsAttathFailureCause protocolError_Gprs = 111; // unspecified
```

```
// The following originating and terminating SMS failure causes are defined in the section 8.2.5.4 of 3GPP TS 24.011 v5.2.0.
typedef unsigned short smsFailureCause;
// Cause values in a mobile originating SM_transfer attempt failure
const smsFailureCause unassignedOrUnallocatedNumber = 1;
const smsFailureCause operatorDeterminedBarring_Sms = 8;
const smsFailureCause callBarred = 10;
const smsFailureCause reserved = 11;
const smsFailureCause shortMessageTransferRejected = 21;
const smsFailureCause destinationOutOfOrder = 27;
const smsFailureCause unidentifiedSubscriber = 28;
const smsFailureCause facilityRejected = 29;
const smsFailureCause unknownSubscriber = 30;
const smsFailureCause networkOutOfOrder = 38;
const smsFailureCause temporaryFailure = 41;
const smsFailureCause congestion_Sms = 42;
const smsFailureCause resourcesUnavailable = 47; //unspecified
const smsFailureCause requestedFacilityNotSubscribed = 50;
const smsFailureCause requestedFacilityNotImplemented = 69;
const smsFailureCause invalidShortMessageTransferReferenceValue = 81;
const smsFailureCause semanticallyIncorrectMessage_Sms = 95;
const smsFailureCause invalidMandatoryInformation_Sms = 96;
const smsFailureCause messageTypeNon_existentOrNotImplemented_Sms = 97;
const smsFailureCause messageNotCompatibleWithShortMessageProtocolState = 98;
const smsFailureCause informationElementNon_existentOrNotImplemented_Sms = 99;
const smsFailureCause protocolError_Sms = 111; //unspecified
const smsFailureCause interworking = 127; //unspecified
// Cause values in a mobile terminating SM_transfer attempt failure
const smsFailureCause memoryCapacityExceeded = 22;
//const smsFailureCause invalidShortMessageTransferReferenceValue = 81; //redefined
//const smsFailureCause semanticallyIncorrectMessage_Sms = 95; //redefined
//const smsFailureCause invalidMandatoryInformation_Sms = 96; //redefined
//const smsFailureCause messageTypeNon_existentOrNotImplemented_Sms = 97; //redefined
//const smsFailureCause messageNotCompatibleWithShortMessageProtocolState = 98; //redefined
//const smsFailureCause informationElementNon_existentOrNotImplemented_Sms = 99; //redefined
//const smsFailureCause protocolError_Sms = 111; //unspecified    //redefined

typedef unsigned short CauseType;
const CauseType sum = 0;
const CauseType other = 65535;
```

```
    const CauseType noResponse = 65534;
    struct   CausePairType
    {
        CauseType   cause;
        unsigned long   value;
    };
    typedef sequence<CausePairType> TDSCDMAMeasurementType3;

    typedef unsigned short TrafficClass;
    const TrafficClass conversational = 1;
    const TrafficClass streaming = 2;
    const TrafficClass interactive = 3;
    const TrafficClass background = 4;
    struct   ClassPairType
    {
        TrafficClass   class;
        unsigned long   value;
    };
    typedef sequence<ClassPairType> TDSCDMAMeasurementType4;
    typedef string LocationAreaIdentificationType;
    //LocationAreaIdentificationType is composed of MCC, MNC and LAC;
    struct LocationAreaMeasurementType
    {
        LocationAreaIdentificationType LocationAreaIdentification;
        unsigned long value;
    };
    typedef sequence<LocationAreaMeasurementType> TDSCDMAMeasurementType5;

    typedef string RoutingAreaIdentificationType;
    // RoutingAreaIdentificationType is composed of LAI and RAC;
    struct RoutingAreaMeasurementType
    {
        RoutingAreaIdentificationType RoutingAreaIdentification;
        unsigned long value;
    };
    typedef sequence<RoutingAreaMeasurementType> TDSCDMAMeasurementType6;
};

#endif
```

6 性能管理接口功能相关的文件

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

下面Schema文件中用到的字段说明参见附录A，示例参见附录B。

版本号：PM FILE V1.0

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Measurement collection data file XML schema measCollec.xsd -->
<schema targetNamespace="http://latest/nmc-omc/cmNrm.doc#measCollec" elementFormDefault="qualified"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:mc="http://latest/nmc-omc/cmNrm.doc#measCollec">
    <!-- Measurement 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="optional"/>
                                </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>
        <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>
                    </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="(mscBasicMeasurement.failImsiAttachsPerCause.|mscBasicMeasurement.failLocationUpdatesIntraMscP
erCause.|mscBasicMeasurement.failLocationUpdatesInterMscPerCause.|mscBasicMeasurement.failOrigSmsCsP
erCause.|mscBasicMeasurement.failTermSmsCsPerCause.|mscBasicMeasurement.attExternalHosPerCause.|msc
BasicMeasurement.attPageReqsPerLa.|mscBasicMeasurement.succPageReqsPerLa.|mscBasicMeasurement.attR
epageReqsPerLa.|mobileTrafficFlow.failOrigCallsGsmPerCause.|mobileTrafficFlow.failOrigCallsUmtsPerCaus
e.|mobileTrafficFlow.failInternalCallsPerCause.|mobileTrafficFlow.failTermCallsGsmPerCause.|mobileTrafficF
low.failTermCallsUmtsPerCause.|mobileTrafficFlow.failIncCallsPerCause.|mobileTrafficFlow.failOutCallsPerC
ause.|mobileTrafficFlow.failTransCallsPerCause.|mobileTrafficFlow.failOrigOutCallsPerCause.|mobileTrafficFl
ow.failTermIncCallsPerCause.|vlrSubscriberData.nbrCurrentSubsInVlrPerHlr.|hlrSubscriberData.nbrCurrentSub
sWithPowerOnInHlrPerVlr.|sessionManagementMeasurement.failActPdpContextMsPerCause.|sessionManagem
entMeasurement.failActPdpContextNetworkPerCause.|subscriberManagementMeasurement.meanAttachedSubs
PerRa.|subscriberManagementMeasurement.maxAttachedSubsPerRa.|mobileManagementMeasurement.failGprs
AttachPerCause.|mobileManagementMeasurement.failCombiAttachPerCause|mobileManagementMeasurement.f
ailGprsAttachWithImsiAttachedPerCause.|mobileManagementMeasurement.failIntraSgsnRaUpdatePerCause.|m
obileManagementMeasurement.failCombiIntraSgsnRaUpdatePerCause.|mobileManagementMeasurement.failInt
erSgsnRaUpdatePerCause.|mobileManagementMeasurement.failCombiInterSgsnRaUpdatePerCause.|apnSession
ManagementMeasurement.failActPdpContextUmtsPerCause.|
rabAssignmentMeasurement.attRabAssignEstabCsPerType.|rabAssignmentMeasurement.succRabAssignEstabC
sPerType.|rabAssignmentMeasurement.failRabAssignEstabCsPerCause.|rabAssignmentMeasurement.attRabAssi
gnEstabPsPerType.|rabAssignmentMeasurement.succRabAssignEstabPsPerType.|rabAssignmentMeasurement.f
ailRabAssignEstabPsPerCause.|rabAssignmentMeasurement.attRabAssignModCsPerType.|rabAssignmentMeas
```

urement.succRabAssignModCsPerType.|rabAssignmentMeasurement.failRabAssignModCsPerCause.|rabAssignmentMeasurement.attRabAssignModPsPerType.|rabAssignmentMeasurement.succRabAssignModPsPerType.|rabAssignmentMeasurement.failRabAssignModPsPerCause.|rabAssignmentMeasurement.attRabAssignRelCsPerType.|rabAssignmentMeasurement.succRabAssignRelCsPerType.|rabAssignmentMeasurement.failRabAssignRelCsPerCause.|rabAssignmentMeasurement.attRabAssignRelPsPerType.|rabAssignmentMeasurement.succRabAssignRelPsPerType.|rabAssignmentMeasurement.failRabAssignRelPsPerCause.|rabReleaseRequestMeasurement.nbrRncRelCsRabPerCause.|rabReleaseRequestMeasurement.nbrRncRelPsRabPerCause.|iuConnectionMeasurement.nbrRncRelCsIuConnPerCause.|iuConnectionMeasurement.nbrRncRelPsIuConnPerCause.|iuConnectionMeasurement.attRelCsIuConnPerCause.|iuConnectionMeasurement.attRelPsIuConnPerCause.|iuInterfaceMeasurement.nbrResetCsByRncPerCause.|iuInterfaceMeasurement.nbrResetPsByRncPerCause.|iuInterfaceMeasurement.nbrResetCsByCnPerCause.|iuInterfaceMeasurement.nbrResetPsByCnPerCause.|iuInterfaceMeasurement.nbrResetResCsByRncPerCause.|iuInterfaceMeasurement.nbrResetResPsByRncPerCause.|iuInterfaceMeasurement.nbrResetResCsByCnPerCause.|iuInterfaceMeasurement.nbrResetResPsByCnPerCause.|iuInterfaceMeasurement.nbrErrorIndCsByRncPerCause.|iuInterfaceMeasurement.nbrErrorIndPsByRncPerCause.|iuInterfaceMeasurement.nbrErrorIndCsByCnPerCause.|iuInterfaceMeasurement.nbrErrorIndPsByCnPerCause.|rncSoftHandoverMeasurement.failRlAddInSofterHoPerCause.|rncSoftHandoverMeasurement.failRlDelInSofterHoPerCause.|rncSoftHandoverMeasurement.failRlAddInShoPerCause.|rncSoftHandoverMeasurement.failRlDelInShoPerCause.|rncHardHandoverMeasurement.failHhoPerCause.|rncRelocationMeasurement.attRelocOutPrepWithUeNotInvCsPerCause.|rncRelocationMeasurement.failRelocOutPrepWithUeNotInvCsPerCause.|rncRelocationMeasurement.failRelocOutWithUeNotInvCsPerCause.|rncRelocationMeasurement.attRelocOutPrepWithUeInvCsPerCause.|rncRelocationMeasurement.failRelocOutPrepWithUeInvCsPerCause.|rncRelocationMeasurement.failRelocOutWithUeInvCsPerCause.|rncRelocationMeasurement.attRelocOutPrepWithUeNotInvPsPerCause.|rncRelocationMeasurement.failRelocOutPrepWithUeNotInvPsPerCause.|rncRelocationMeasurement.failRelocOutWithUeNotInvPsPerCause.|rncRelocationMeasurement.attRelocOutPrepWithUeInvPsPerCause.|rncRelocationMeasurement.failRelocOutPrepWithUeInvPsPerCause.|rncRelocationMeasurement.failRelocOutWithUeInvPsPerCause.|rncRelocationMeasurement.attRelocInWithUeNotInvCsPerCause.|rncRelocationMeasurement.failRelocInWithUeNotInvCsPerCause.|rncRelocationMeasurement.attRelocInWithUeInvCsPerCause.|rncRelocationMeasurement.failRelocInWithUeInvCsPerCause.|rncRelocationMeasurement.attRelocInWithUeNotInvPsPerCause.|rncRelocationMeasurement.failRelocInWithUeNotInvPsPerCause.|rncRelocationMeasurement.attRelocInWithUeInvPsPerCause.|rncRelocationMeasurement.failRelocInWithUeInvPsPerCause.|rncInterSystemHandoverMeasurement.attRelocOutPrepInterSysCsPerCause.|rncInterSystemHandoverMeasurement.failRelocOutPrepInterSysCsPerCause.|rncInterSystemHandoverMeasurement.failRelocOutInterSysCsPerCause.|rncInterSystemHandoverMeasurement.attRelocInInterSysCsPerCause.|rncInterSystemHandoverMeasurement.failRelocInInterSysCsPerCause.|rncInterSystemHandoverMeasurement.failRelocOutInterSysPsPerCause.|iuInterfaceThroughputMeasurement.iuUlDataThroughputCsPerType.|iuInterfaceThroughputMeasurement.iuDlDataThroughputCsPerType.|iuInterfaceThroughputMeasurement.iuUlDataThroughputPsPerType.|iuInterfaceThroughputMeasurement.iuDlDataThroughputPsPerType.|rlcConnectionMeasurement.nbrRlcBlockSentPerMode.|rlcConnectionMeasurement.nbrRlcBlockRecvedPerMode.|cellRrcConnection

```
Measurement.attRrcConnSetupPerCause.|cellRrcConnectionMeasurement.succRrcConnSetupPerCause.|cellRrc
ConnectionMeasurement.failRrcConnSetupPerCause.|cellRrcConnectionMeasurement.failRrcConnReestabPerC
ause.|cellSoftHandoverMeasurement.failRlAddInShoPerCause.|cellSoftHandoverMeasurement.failRlDelInShoP
erCause.|hardHandoverIntraCellMeasurement.failHhoOutIntraCellPerCause.|iubRlManagementMeasurement.fai
lRlSetupIubPerCause.|iubRlManagementMeasurement.failRlAddIubPerCause.|iurRlManagementMeasurement.f
ailRlSetupIurPerCause.|iurRlManagementMeasurement.failRlAddIurPerCause.|hardHandoverInterCellIntraNode
BMeasurement.failHhoOutInterCellIntraNodeBPerCause.|hardHandoverInterNodeBIntraRncMeasurement.failH
hoOutInterNodeBIntraRncPerCause.|hardHandoverInterRncViaIurMeasurement.failHhoOutInterRncViaIurPerC
ause.|hardHandoverInterRncMeasurement.failHhoOutInterRncCnPerCause.|hardHandoverInterSystemMeasure
ment.attRelocOutPrepInterSysCsPerCause.|hardHandoverInterSystemMeasurement.failRelocOutPrepInterSysCs
PerCause.|hardHandoverInterSystemMeasurement.failRelocOutInterSysCsPerCause.|hardHandoverInterSystem
Measurement.attRelocInInterSysCsPerCause.|hardHandoverInterSystemMeasurement.failRelocInInterSysCsPer
Cause.|hardHandoverInterSystemMeasurement.failRelocOutInterSysPsPerCause.)\d{1,5}"/>
        </restriction>
    </simpleType>
    <simpleType name="measNameWithoutSubCounter">
        <restriction base="string">
            <enumeration value="mscBasicMeasurement.attGetRoutingInfo"/>
            <enumeration value="mscBasicMeasurement.succGetRoutingInfo"/>
            <enumeration value="mscBasicMeasurement.attImsiAttachs"/>
            <enumeration value="mscBasicMeasurement.succImsiAttachs"/>
            <enumeration value="mscBasicMeasurement.nbrImsiDetachs"/>
            <enumeration value="mscBasicMeasurement.attLocationUpdatesIntraMsc"/>
            <enumeration value="mscBasicMeasurement.succLocationUpdatesIntraMsc"/>
            <enumeration value="mscBasicMeasurement.attLocationUpdatesInterMsc"/>
            <enumeration value="mscBasicMeasurement.succLocationUpdatesInterMsc"/>
            <enumeration value="mscBasicMeasurement.attOrigSmsCs"/>
            <enumeration value="mscBasicMeasurement.succOrigSmsCs"/>
            <enumeration value="mscBasicMeasurement.attTermSsmsCs"/>
            <enumeration value="mscBasicMeasurement.succTermSmsCs"/>
            <enumeration value="mscBasicMeasurement.attIncHosInterMsc"/>
            <enumeration value="mscBasicMeasurement.succIncHosInterMsc"/>
            <enumeration value="mscBasicMeasurement.attOutHosInterMsc"/>
            <enumeration value="mscBasicMeasurement.succOutHosInterMsc"/>
            <enumeration value="mscBasicMeasurement.attSubsequentHosToMsca"/>
            <enumeration value="mscBasicMeasurement.succSubsequentHosToMsca"/>
            <enumeration value="mscBasicMeasurement.attSubsequentHosToMscc"/>
```

```
<enumeration value="mscBasicMeasurement.succSubsequentHosToMscc"/>
<enumeration value="mscBasicMeasurement.attExternalHos"/>
<enumeration value="mscBasicMeasurement.failExternalHosWithReconn"/>
<enumeration value="mscBasicMeasurement.failExternalHosWithLossOfConn"/>
<enumeration value="mscQos.meanDurOfCallSetup"/>
<enumeration value="mscQos.meanDurOfCallAssignGsm"/>
<enumeration value="mscQos.meanDurOfCallRabAssignUmts"/>
<enumeration value="mscQos.meanDurOfLuService"/>
<enumeration value="mscQos.meanCallDur"/>
<enumeration value="mscQos.meanDurOfTrunkSeizure"/>
<enumeration value="mobileTrafficFlow.attOrigCallsGsm"/>
<enumeration value="mobileTrafficFlow.succOrigCallsGsm"/>
<enumeration value="mobileTrafficFlow.ansOrigCallsGsm"/>
<enumeration value="mobileTrafficFlow.attOrigCallTrafficGsm"/>
<enumeration value="mobileTrafficFlow.succOrigCallTrafficGsm"/>
<enumeration value="mobileTrafficFlow.ansOrigCallTrafficGsm"/>
<enumeration value="mobileTrafficFlow.attOrigCallsUmts"/>
<enumeration value="mobileTrafficFlow.succOrigCallsUmts"/>
<enumeration value="mobileTrafficFlow.ansOrigCallsUmts"/>
<enumeration value="mobileTrafficFlow.attOrigCallTrafficUmts"/>
<enumeration value="mobileTrafficFlow.succOrigCallTrafficUmts"/>
<enumeration value="mobileTrafficFlow.ansOrigCallTrafficUmts"/>
<enumeration value="mobileTrafficFlow.attInternalCalls"/>
<enumeration value="mobileTrafficFlow.succInternalCalls"/>
<enumeration value="mobileTrafficFlow.ansInternalCalls"/>
<enumeration value="mobileTrafficFlow.attInternalCallTraffic"/>
<enumeration value="mobileTrafficFlow.succInternalCallTraffic"/>
<enumeration value="mobileTrafficFlow.ansInternalCallTraffic"/>
<enumeration value="mobileTrafficFlow.attTermCallsGsm"/>
<enumeration value="mobileTrafficFlow.succTermCallsGsm"/>
<enumeration value="mobileTrafficFlow.ansTermCallsGsm"/>
<enumeration value="mobileTrafficFlow.attTermCallTrafficGsm"/>
<enumeration value="mobileTrafficFlow.succTermCallTrafficGsm"/>
<enumeration value="mobileTrafficFlow.ansTermCallTrafficGsm"/>
<enumeration value="mobileTrafficFlow.attTermCallsUmts"/>
<enumeration value="mobileTrafficFlow.succTermCallsUmts"/>
<enumeration value="mobileTrafficFlow.ansTermCallsUmts"/>
```

```
<enumeration value="mobileTrafficFlow.attTermCallTrafficUmts"/>
<enumeration value="mobileTrafficFlow.succTermCallTrafficUmts"/>
<enumeration value="mobileTrafficFlow.ansTermCallTrafficUmts"/>
<enumeration value="mobileTrafficFlow.attIncCalls"/>
<enumeration value="mobileTrafficFlow.succIncCalls"/>
<enumeration value="mobileTrafficFlow.ansIncCalls"/>
<enumeration value="mobileTrafficFlow.attIncCallTraffic"/>
<enumeration value="mobileTrafficFlow.succIncCallTraffic"/>
<enumeration value="mobileTrafficFlow.ansIncCallTraffic"/>
<enumeration value="mobileTrafficFlow.attOutCalls"/>
<enumeration value="mobileTrafficFlow.attOutCalls"/>
<enumeration value="mobileTrafficFlow.ansOutCalls"/>
<enumeration value="mobileTrafficFlow.attOutCallTraffic"/>
<enumeration value="mobileTrafficFlow.succOutCallTraffic"/>
<enumeration value="mobileTrafficFlow.ansOutCallTraffic"/>
<enumeration value="mobileTrafficFlow.attTransCalls"/>
<enumeration value="mobileTrafficFlow.succTransCalls"/>
<enumeration value="mobileTrafficFlow.ansTransCalls"/>
<enumeration value="mobileTrafficFlow.attTransCallTraffic"/>
<enumeration value="mobileTrafficFlow.succTransCallTraffic"/>
<enumeration value="mobileTrafficFlow.ansTransCallTraffic"/>
<enumeration value="mobileTrafficFlow.attOrigOutCalls"/>
<enumeration value="mobileTrafficFlow.succOrigOutCalls"/>
<enumeration value="mobileTrafficFlow.ansOrigOutCalls"/>
<enumeration value="mobileTrafficFlow.attOrigOutCallTraffic"/>
<enumeration value="mobileTrafficFlow.succOrigOutCallTraffic"/>
<enumeration value="mobileTrafficFlow.ansOrigOutCallTraffic"/>
<enumeration value="mobileTrafficFlow.attTermIncCalls"/>
<enumeration value="mobileTrafficFlow.succTermIncCalls"/>
<enumeration value="mobileTrafficFlow.ansTermIncCalls"/>
<enumeration value="mobileTrafficFlow.attTermIncCallTraffic"/>
<enumeration value="mobileTrafficFlow.succTermIncCallTraffic"/>
<enumeration value="mobileTrafficFlow.ansTermIncCallTraffic"/>
<enumeration value="mobileTrafficFlow.nbrCallsBlockedByLoadShedding"/>
<enumeration value="mobileTrafficFlow.nbrCallsBlockedByInternalCongestion"/>
<enumeration value="mobileTrafficFlow.nbrCallsBlockedByOutCircuitBusy"/>
<enumeration value="circuitEndpointSubGroup.outBids"/>
```

```
<enumeration value="circuitEndpointSubGroup.succOutSeizures"/>
<enumeration value="circuitEndpointSubGroup.succOutCalls"/>
<enumeration value="circuitEndpointSubGroup.ansOutCalls"/>
<enumeration value="circuitEndpointSubGroup.failOutCallsByOverflow"/>
<enumeration value="circuitEndpointSubGroup.failOutCallsByUserBusy"/>
<enumeration value="circuitEndpointSubGroup.failOutCallsByNoAns"/>
<enumeration value="circuitEndpointSubGroup.failOutCallsByUnallNum"/>
<enumeration value="circuitEndpointSubGroup.failOutCallsByCongestion"/>
<enumeration value="circuitEndpointSubGroup.succOutSeizureTraffic"/>
<enumeration value="circuitEndpointSubGroup.ansOutCallTraffic"/>
<enumeration value="circuitEndpointSubGroup.succIncSeizures"/>
<enumeration value="circuitEndpointSubGroup.succIncCalls"/>
<enumeration value="circuitEndpointSubGroup.ansIncSeizures"/>
<enumeration value="circuitEndpointSubGroup.failIncCallsByUserBusy"/>
<enumeration value="circuitEndpointSubGroup.failIncCallsByNoAns"/>
<enumeration value="circuitEndpointSubGroup.failIncCallsByUnallNum"/>
<enumeration value="circuitEndpointSubGroup.failIncCallsByCongestion"/>
<enumeration value="circuitEndpointSubGroup.succIncSeizureTraffic"/>
<enumeration value="circuitEndpointSubGroup.ansIncSeizureTraffic"/>
<enumeration value="circuitEndpointSubGroup.nbrAvailTrunks"/>
<enumeration value="mtp3SignallingLinkTP.durSigLinkOutOfService"/>
<enumeration value="mtp3SignallingLinkTP.nbrSigLinkOutOfService"/>
<enumeration value="mtp3SignallingLinkTP.nbrSentMsus"/>
<enumeration value="mtp3SignallingLinkTP.nbrSentSifsAndSios"/>
<enumeration value="mtp3SignallingLinkTP.nbrRecvedMsus"/>
<enumeration value="mtp3SignallingLinkTP.nbrRecvedSifsAndSios"/>
<enumeration value="mtp3SignallingLinkSetTP.nbrSigLinkSetOutOfService"/>
<enumeration value="mtp3SignallingLinkSetTP.durSigLinkSetOutOfService"/>
<enumeration value="mtp3SignallingLinkSetTP.nbrAvailSiglinks"/>
<enumeration value="mtp3bSignallingLinkTP.durSigLinkOutOfService"/>
<enumeration value="mtp3bSignallingLinkTP.nbrSigLinkOutOfService"/>
<enumeration value="mtp3bSignallingLinkTP.nbrSentMsus"/>
<enumeration value="mtp3bSignallingLinkTP.nbrSentSifsAndSios"/>
<enumeration value="mtp3bSignallingLinkTP.nbrRecvedMsus"/>
<enumeration value="mtp3bSignallingLinkTP.nbrRecvedSifsAndSios"/>
<enumeration value="mtp3bSignallingLinkSetTP.nbrSigLinkSetOutOfService"/>
<enumeration value="mtp3bSignallingLinkSetTP.durSigLinkSetOutOfService"/>
```

```
<enumeration value="mtp3bSignallingLinkSetTP.nbrAvailSiglinks"/>
<enumeration value="observedDestination.bids"/>
<enumeration value="observedDestination.nbrNoAvailCircuits"/>
<enumeration value="observedDestination.succCalls"/>
<enumeration value="observedDestination.ansCalls"/>
<enumeration value="observedDestination.succCallTraffic"/>
<enumeration value="observedDestination.ansCallTraffic"/>
<enumeration value="vlrBasicMeasurement.attIdentReqsToPVlr"/>
<enumeration value="vlrBasicMeasurement.succIdentReqsToPVlr"/>
<enumeration value="vlrBasicMeasurement.attLusIntraVlr"/>
<enumeration value="vlrBasicMeasurement.succLusIntraVlr"/>
<enumeration value="vlrBasicMeasurement.attLusInterVlr"/>
<enumeration value="vlrBasicMeasurement.succLusInterVlr"/>
<enumeration value="vlrBasicMeasurement.attReqsForAuthSetsSentToHlr"/>
<enumeration value="vlrBasicMeasurement.succRecvedAuthSetsFromHlr"/>
<enumeration value="vlrBasicMeasurement.succReqAuthSetWithQuintupletsFromHlr"/>
<enumeration value="vlrBasicMeasurement.succReqAuthSetWithTripletsFromHlr"/>
<enumeration value="vlrBasicMeasurement.succInsertSubsData"/>
<enumeration value="vlrBasicMeasurement.succDelSubsData"/>
<enumeration value="vlrBasicMeasurement.attProvideRoamingNumber"/>
<enumeration value="vlrBasicMeasurement.succProvideRoamingNumber"/>
<enumeration value="vlrSubscriberData.nbrCurrentSubsWithPowerOnInVlr"/>
<enumeration value="vlrSubscriberData.nbrRoamingSubs"/>
<enumeration value="vlrSubscriberData.nbrRoamingSubsInternational"/>
<enumeration value="hlrBasicMeasurement.attGetRoutingInfo"/>
<enumeration value="hlrBasicMeasurement.succGetRoutingInfo"/>
<enumeration value="hlrBasicMeasurement.attProvideRoamingNumber"/>
<enumeration value="hlrBasicMeasurement.succProvideRoamingNumber"/>
<enumeration value="hlrBasicMeasurement.attLocationUpdates"/>
<enumeration value="hlrBasicMeasurement.succLocationUpdates"/>
<enumeration value="hlrBasicMeasurement.attCancelLocation"/>
<enumeration value="hlrBasicMeasurement.succCancelLocation"/>
<enumeration value="hlrBasicMeasurement.attInsertSubsData"/>
<enumeration value="hlrBasicMeasurement.succInsertSubsData"/>
<enumeration value="hlrBasicMeasurement.attDeleteSubsData"/>
<enumeration value="hlrBasicMeasurement.succDeleteSubsData"/>
<enumeration value="hlrBasicMeasurement.attSendAuthInfo"/>
```

```
<enumeration value="hlrBasicMeasurement.succSendAuthInfo"/>
<enumeration value="hlrBasicMeasurement.nbrReset"/>
<enumeration value="hlrBasicMeasurement.attRestoreData"/>
<enumeration value="hlrBasicMeasurement.succRestoreData"/>
<enumeration value="hlrSubscriberData.nbrCurrentSubsInHlr"/>
<enumeration value="hlrSubscriberData.nbrCurrentMsisdnInHlr"/>
<enumeration value="hlrSmServiceMeasurement.attSendRoutingInfoForSm"/>
<enumeration value="hlrSmServiceMeasurement.succSendRoutingInfoForSm"/>
<enumeration value="hlrSmServiceMeasurement.nbrAlertServiceCentre"/>
<enumeration value="hlrSmServiceMeasurement.nbrInformServiceCenter"/>
<enumeration value="hlrSmServiceMeasurement.nbrReadyForSm"/>
<enumeration value="hlrSupplementServiceMeasurement.attRegisterSs"/>
<enumeration value="hlrSupplementServiceMeasurement.succRegisterSs"/>
<enumeration value="hlrSupplementServiceMeasurement.attEraseSs"/>
<enumeration value="hlrSupplementServiceMeasurement.succEraseSs"/>
<enumeration value="hlrSupplementServiceMeasurement.attActSs"/>
<enumeration value="hlrSupplementServiceMeasurement.succActSs"/>
<enumeration value="hlrSupplementServiceMeasurement.attDeactSs"/>
<enumeration value="hlrSupplementServiceMeasurement.succDeactSs"/>
<enumeration value="hlrInServiceMeasurement.attAnyTimeInterrogation"/>
<enumeration value="hlrInServiceMeasurement.succAnyTimeInterrogation"/>
<enumeration value="hlrInServiceMeasurement.attAnyTimeSubsInterrogation"/>
<enumeration value="hlrInServiceMeasurement.succAnyTimeSubsInterrogation"/>
<enumeration value="hlrInServiceMeasurement.attAnyTimeModification"/>
<enumeration value="hlrInServiceMeasurement.succAnyTimeModification"/>
<enumeration value="hlrInServiceMeasurement.nbrNoteSubsDataModified"/>
<enumeration value="hlrPacketServiceMeasurement.attSendRoutingInfoForGprs"/>
<enumeration value="hlrPacketServiceMeasurement.succSendRoutingInfoForGprs"/>
<enumeration value="hlrPacketServiceMeasurement.nbrFailReport"/>
<enumeration value="hlrPacketServiceMeasurement.nbrNoteMsPresentForGprs"/>
<enumeration value="hlrPacketServiceMeasurement.attUpdateGprsLocation"/>
<enumeration value="hlrPacketServiceMeasurement.succUpdateGprsLocation"/>
<enumeration value="hlrLocationServiceMeasurement.attSendRoutingInfoForLcs"/>
<enumeration value="hlrLocationServiceMeasurement.succSendRoutingInfoForLcs"/>
<enumeration value="eirBasicMeasurement.nbrCurrentWhiteSubsInEir"/>
<enumeration value="eirBasicMeasurement.nbrCurrentBlackSubsInEir"/>
<enumeration value="eirBasicMeasurement.nbrCurrentGreySubsInEir"/>
```

```
<enumeration value="sessionManagementMeasurement.attActPdpContextMs"/>
<enumeration value="sessionManagementMeasurement.succActPdpContextMs"/>
<enumeration value="sessionManagementMeasurement.attActPdpContextNetwork"/>
<enumeration value="sessionManagementMeasurement.succActPdpContextNetwork"/>
<enumeration value="sessionManagementMeasurement.attActPdpContextDynMs"/>
<enumeration value="sessionManagementMeasurement.succActPdpContextDynMs"/>
<enumeration value="sessionManagementMeasurement.meanSubsWithActPdpContext"/>
<enumeration value="sessionManagementMeasurement.maxSubsWithActPdpContext"/>
<enumeration value="sessionManagementMeasurement.meanActPDPContexts"/>
<enumeration value="sessionManagementMeasurement.maxActPdpContexts"/>
<enumeration value="sessionManagementMeasurement.attDeactPdpContextSgsn"/>
<enumeration value="sessionManagementMeasurement.succDeactPdpContextSgsn"/>
<enumeration value="sessionManagementMeasurement.attDeactPdpContextMs"/>
<enumeration value="sessionManagementMeasurement.succDeactPdpContextMs"/>
<enumeration value="sessionManagementMeasurement.attDeactPdpContextGgsn"/>
<enumeration value="sessionManagementMeasurement.succDeactPdpContextGgsn"/>
<enumeration value="sessionManagementMeasurement.attActSecondPdpContext"/>
<enumeration value="sessionManagementMeasurement.succActSecondPdpContext"/>
<enumeration value="sessionManagementMeasurement.attModPdpContextMs"/>
<enumeration value="sessionManagementMeasurement.succModPdpContextMs"/>
<enumeration value="sessionManagementMeasurement.attModPdpContextSgsn"/>
<enumeration value="sessionManagementMeasurement.succModPdpContextSgsn"/>
<enumeration value="sessionManagementMeasurement.attUpdPdpContextGgsn"/>
<enumeration value="sessionManagementMeasurement.succUpdPdpContextGgsn"/>
<enumeration value="sessionManagementMeasurement.attUpdPdpContextSgsn"/>
<enumeration value="sessionManagementMeasurement.succUpdPdpContextSgsn"/>
<enumeration value="subscriberManagementMeasurement.meanStandbySubs"/>
<enumeration value="subscriberManagementMeasurement.maxStandbySubs"/>
<enumeration value="subscriberManagementMeasurement.meanReadySubs"/>
<enumeration value="subscriberManagementMeasurement.maxReadySubs"/>
<enumeration value="subscriberManagementMeasurement.meanPmmIdleSubs"/>
<enumeration value="subscriberManagementMeasurement.maxPmmIdleSubs"/>
<enumeration value="subscriberManagementMeasurement.meanPmmConnectedSubs"/>
<enumeration value="subscriberManagementMeasurement.maxPmmConnectedSubs"/>
<enumeration value="mobileManagementMeasurement.attGprsAttach"/>
<enumeration value="mobileManagementMeasurement.succGprsAttach"/>
<enumeration value="mobileManagementMeasurement.attCombiAttach"/>
```

```
<enumeration value="mobileManagementMeasurement.succCombiAttach"/>
<enumeration value="mobileManagementMeasurement.attGprsAttachWithImsiAttached"/>
<enumeration value="mobileManagementMeasurement.succGprsAttachWithImsiAttached"/>
<enumeration value="mobileManagementMeasurement.attGprsDetachMs"/>
<enumeration value="mobileManagementMeasurement.attCombiDetachMs"/>
<enumeration value="mobileManagementMeasurement.attImsiDetachMs"/>
<enumeration value="mobileManagementMeasurement.attGprsDetachSgsn"/>
<enumeration value="mobileManagementMeasurement.succGprsDetachSgsn"/>
<enumeration value="mobileManagementMeasurement.attGprsDetachHlr"/>
<enumeration value="mobileManagementMeasurement.attIntraSgsnRaUpdate"/>
<enumeration value="mobileManagementMeasurement.succIntraSgsnRaUpdate"/>
<enumeration value="mobileManagementMeasurement.attCombiIntraSgsnRaUpdate"/>
<enumeration value="mobileManagementMeasurement.succCombiIntraSgsnRaUpdate"/>
<enumeration value="mobileManagementMeasurement.attInterSgsnRaUpdate"/>
<enumeration value="mobileManagementMeasurement.succInterSgsnRaUpdate"/>
<enumeration value="mobileManagementMeasurement.attCombiInterSgsnRaUpdate"/>
<enumeration value="mobileManagementMeasurement.succCombiInterSgsnRaUpdate"/>
<enumeration value="mobileManagementMeasurement.nbrPsPagingGsm"/>
<enumeration value="mobileManagementMeasurement.nbrPsPagingNoRspGsm"/>
<enumeration value="mobileManagementMeasurement.nbrPsPagingUmts"/>
<enumeration value="mobileManagementMeasurement.nbrPsPagingNoRspUmts"/>
<enumeration value="sgsnRelocationMeasurement.attInterSgsnReloc"/>
<enumeration value="sgsnRelocationMeasurement.succInterSgsnReloc"/>
<enumeration value="sgsnRelocationMeasurement.attInterSgsnCombiReloc"/>
<enumeration value="sgsnRelocationMeasurement.succInterSgsnCombiReloc"/>
<enumeration value="sgsnRelocationMeasurement.attIntraSgsnReloc"/>
<enumeration value="sgsnRelocationMeasurement.succIntraSgsnReloc"/>
<enumeration value="sgsnRelocationMeasurement.attIntraSgsnCombiReloc"/>
<enumeration value="sgsnRelocationMeasurement.succIntraSgsnCombiReloc"/>
<enumeration value="sgsnInterSystemHandoverMeasurement.attIntraSgsnHoGsmToUmts"/>
<enumeration value="sgsnInterSystemHandoverMeasurement.succIntraSgsnHoGsmToUmts"/>
<enumeration value="sgsnInterSystemHandoverMeasurement.attIntraSgsnHoUmtsToGsm"/>
<enumeration value="sgsnInterSystemHandoverMeasurement.succIntraSgsnHoUmtsToGsm"/>
<enumeration value="sgsnInterSystemHandoverMeasurement.attInterSgsnHoGsmToUmts"/>
<enumeration value="sgsnInterSystemHandoverMeasurement.succInterSgsnHoGsmToUmts"/>
<enumeration value="sgsnInterSystemHandoverMeasurement.attInterSgsnHoUmtsToGsm"/>
<enumeration value="sgsnInterSystemHandoverMeasurement.succInterSgsnHoUmtsToGsm"/>
```

```
<enumeration value="mapServiceMeasurement.attReqAuthSetHlr"/>
<enumeration value="mapServiceMeasurement.succReqAuthSetWithQuintupletsHlr"/>
<enumeration value="mapServiceMeasurement.succReqAuthSetWithTripletsHlr"/>
<enumeration value="mapServiceMeasurement.succReqAuthSetWithEmptyRspHlr"/>
<enumeration value="mapServiceMeasurement.attUpdateGprsLocationHlr"/>
<enumeration value="mapServiceMeasurement.succUpdateGprsLocationHlr"/>
<enumeration value="mapServiceMeasurement.attInsertSubsDataHlr"/>
<enumeration value="mapServiceMeasurement.attDeleteSubsDataHlr"/>
<enumeration value="securityManagementMeasurement.attPtmsiRealloc"/>
<enumeration value="securityManagementMeasurement.succPtmsiRealloc"/>
<enumeration value="securityManagementMeasurement.attAuthReq"/>
<enumeration value="securityManagementMeasurement.succAuthReq"/>
<enumeration value="securityManagementMeasurement.attIdentReq"/>
<enumeration value="securityManagementMeasurement.succIdentReq"/>
<enumeration value="securityManagementMeasurement.attSecMode"/>
<enumeration value="securityManagementMeasurement.succSecMode"/>
<enumeration value="gtpInGnGpMeasurement.nbrIncGtpCSigPkts"/>
<enumeration value="gtpInGnGpMeasurement.nbrOutGtpCSigPkts"/>
<enumeration value="gtpInGnGpMeasurement.nbrIncGtpCSigOcts"/>
<enumeration value="gtpInGnGpMeasurement.nbrOutGtpCSigOcts"/>
<enumeration value="gtpInGnGpMeasurement.nbrOutGtpCSigPktsOverflow"/>
<enumeration value="gtpInGnGpMeasurement.nbrIncGtpCSigPktsError"/>
<enumeration value="gtpInGnGpMeasurement.nbrIncGtpUDataPkts"/>
<enumeration value="gtpInGnGpMeasurement.nbrOutGtpUDataPkts"/>
<enumeration value="gtpInGnGpMeasurement.nbrIncGtpUDataOcts"/>
<enumeration value="gtpInGnGpMeasurement.nbrOutGtpUDataOcts"/>
<enumeration value="gtpInGnGpMeasurement.nbrOutGtpUDataPktsOverflow"/>
<enumeration value="gtpInGnGpMeasurement.nbrIncGtpUDataPktsError"/>
<enumeration value="shortMessageServiceMeasurement.attOrigSmsPs"/>
<enumeration value="shortMessageServiceMeasurement.succOrigSmsPs"/>
<enumeration value="shortMessageServiceMeasurement.attTermSmsPs"/>
<enumeration value="shortMessageServiceMeasurement.succTermSmsPs"/>
<enumeration value="shortMessageServiceMeasurement.attSmsMsPresentPs"/>
<enumeration value="shortMessageServiceMeasurement.succSmsMsPresentPs"/>
<enumeration value="shortMessageServiceMeasurement.attSmsMemoryAvaiPs"/>
<enumeration value="shortMessageServiceMeasurement.succSmsMemoryAvailPs"/>
<enumeration value="ggsnThroughputMeasurement.nbrIncDataPktsGnGp"/>
```

```
<enumeration value="ggsnThroughputMeasurement.nbrOutDataPktsGnGp"/>
<enumeration value="ggsnThroughputMeasurement.nbrIncDataOctsGnGp"/>
<enumeration value="ggsnThroughputMeasurement.nbrOutDataOctsGnGp"/>
<enumeration value="ggsnThroughputMeasurement.nbrIncSigPktsGnGp"/>
<enumeration value="ggsnThroughputMeasurement.nbrOutSigPktsGnGp"/>
<enumeration value="ggsnThroughputMeasurement.nbrIncSigOctsGnGp"/>
<enumeration value="ggsnThroughputMeasurement.nbrOutSigOctsGnGp"/>
<enumeration value="ggsnThroughputMeasurement.nbrOutDataPktsGi"/>
<enumeration value="ggsnThroughputMeasurement.nbrIncDataPktsGi"/>
<enumeration value="ggsnThroughputMeasurement.nbrOutDataOctsGi"/>
<enumeration value="ggsnThroughputMeasurement.nbrIncDataOctsGi"/>
<enumeration value="apnSessionManagementMeasurement.attActPdpContext"/>
<enumeration value="apnSessionManagementMeasurement.succActPdpContext"/>
<enumeration value="apnSessionManagementMeasurement.attDynActPdpContext"/>
<enumeration value="apnSessionManagementMeasurement.succDynActPdpContext"/>
<enumeration value="apnSessionManagementMeasurement.succActPdpContextQos"/>
<enumeration value="apnSessionManagementMeasurement.attDeactPdpContextMs"/>
<enumeration value="apnSessionManagementMeasurement.succDeactPdpContextMs"/>
<enumeration value="apnSessionManagementMeasurement.attDeactPdpContextGgsn"/>
<enumeration value="apnSessionManagementMeasurement.succDeactPdpContextGgsn"/>
<enumeration value="apnSessionManagementMeasurement.nbrActPdpContexts"/>
<enumeration value="apnSessionManagementMeasurement.meanActPdpContexts"/>
<enumeration value="apnSessionManagementMeasurement.maxActPdpContexts"/>
<enumeration value="apnThroughputMeasurement.nbrIncDataPktsGnGp"/>
<enumeration value="apnThroughputMeasurement.nbrOutDataPktsGnGp"/>
<enumeration value="apnThroughputMeasurement.nbrIncDataOctsGnGp"/>
<enumeration value="apnThroughputMeasurement.nbrOutDataOctsGnGp"/>
<enumeration value="apnThroughputMeasurement.nbrIncSigPktsGnGp"/>
<enumeration value="apnThroughputMeasurement.nbrOutSigPktsGnGp"/>
<enumeration value="apnThroughputMeasurement.nbrIncSigOctsGnGp"/>
<enumeration value="apnThroughputMeasurement.nbrOutSigOctsGnGp"/>
<enumeration value="apnThroughputMeasurement.nbrOutDataPktsGi"/>
<enumeration value="apnThroughputMeasurement.nbrIncDataPktsGi"/>
<enumeration value="apnThroughputMeasurement.nbrOutDataOctsGi"/>
<enumeration value="apnThroughputMeasurement.nbrIncDataOctsGi"/>
<enumeration value="iuConnectionMeasurement.attRncEstabCsIuConn"/>
<enumeration value="iuConnectionMeasurement.attRncEstabPsIuConn"/>
```

```
<enumeration value="iuInterfaceMeasurement.nbrOverloadControlCsByRnc"/>
<enumeration value="iuInterfaceMeasurement.nbrOverloadControlPsByRnc"/>
<enumeration value="iuInterfaceMeasurement.nbrOverloadControlCsByCn"/>
<enumeration value="iuInterfaceMeasurement.nbrOverloadControlPsByCn"/>
<enumeration value="rncSoftHandoverMeasurement.attRlAddInSofterHo"/>
<enumeration value="rncSoftHandoverMeasurement.attRlDelInSofterHo"/>
<enumeration value="rncSoftHandoverMeasurement.attRlAddInSho"/>
<enumeration value="rncSoftHandoverMeasurement.attRlDelInSho"/>
<enumeration value="rncHardHandoverMeasurement.attHho"/>
<enumeration value="rncRelocationMeasurement.attRelocOutWithUeNotInvCs"/>
<enumeration value="rncRelocationMeasurement.attRelocOutWithUeInvCs"/>
<enumeration value="rncRelocationMeasurement.attRelocOutWithUeNotInvPs"/>
<enumeration value="rncRelocationMeasurement.attRelocOutWithUeInvPs"/>
<enumeration value="rncInterSystemHandoverMeasurement.attRelocOutInterSysCs"/>
<enumeration value="rncInterSystemHandoverMeasurement.attRelocOutInterSysPs"/>
<enumeration value="rncInterSystemHandoverMeasurement.attRelocInInterSysPs"/>
<enumeration value="rncInterSystemHandoverMeasurement.succRelocInInterSysPs"/>
<enumeration value="iuInterfaceThroughputMeasurement.iuUlSigThroughputCs"/>
<enumeration value="iuInterfaceThroughputMeasurement.iuDlSigThroughputCs"/>
<enumeration value="iuInterfaceThroughputMeasurement.iuUlSigThroughputPs"/>
<enumeration value="iuInterfaceThroughputMeasurement.iuDlSigThroughputPs"/>
<enumeration value="iurInterfaceThroughputMeasurement.iurUlSigThroughput"/>
<enumeration value="iurInterfaceThroughputMeasurement.iurDlSigThroughput"/>
<enumeration value="iurInterfaceThroughputMeasurement.iurUlDataThroughput"/>
<enumeration value="iurInterfaceThroughputMeasurement.iurDlDataThroughput"/>
<enumeration value="rlcConnectionMeasurement.nbrDiscardedRlcBlocksByRnc"/>
<enumeration value="rlcConnectionMeasurement.nbrRetransmittedRlcBlocksToUe"/>
<enumeration value="cellRrcConnectionMeasurement.attRrcConnReestab"/>
<enumeration value="cellSoftHandoverMeasurement.attRlAddInSho"/>
<enumeration value="cellSoftHandoverMeasurement.attRlDelInSho"/>
<enumeration value="hardHandoverIntraCellMeasurement.attHhoOutIntraCell"/>
<enumeration value="iubRlManagementMeasurement.attRlSetupIub"/>
<enumeration value="iubRlManagementMeasurement.attRlAddIub"/>
<enumeration value="iubRlManagementMeasurement.attRlDelIub"/>
<enumeration value="iubRlManagementMeasurement.succRlDelIub"/>
<enumeration value="iurRlManagementMeasurement.attRlSetupIur"/>
<enumeration value="iurRlManagementMeasurement.attRlAddIur"/>
```

```
      <enumeration value="iurRlManagementMeasurement.attRlDelIur"/>
      <enumeration value="iurRlManagementMeasurement.succRlDelIur"/>
      <enumeration value="cellTrafficMeasurement.cellCcchTraffic"/>
      <enumeration value="cellTrafficMeasurement.cellCtchTraffic"/>
      <enumeration value="cellTrafficMeasurement.cellDcchTraffic"/>
      <enumeration value="cellTrafficMeasurement.cellDtchTraffic"/>
      <enumeration value="cellPagingMeasurement.attPagingType1FromUtran"/>
      <enumeration value="cellPagingMeasurement.succPagingType1FromUtran"/>
      <enumeration value="cellPagingMeasurement.attPagingType2FromUtran"/>
      <enumeration
value="hardHandoverInterCellIntraNodeBMeasurement.attHhoOutInterCellIntraNodeB"/>
      <enumeration
value="hardHandoverInterNodeBIntraRncMeasurement.attHhoOutInterNodeBIntraRnc"/>
      <enumeration value="hardHandoverInterRncViaIurMeasurement.attHhoOutInterRncViaIur"/>
      <enumeration value="hardHandoverInterRncMeasurement.attHhoOutInterRncCn"/>
      <enumeration value="hardHandoverInterSystemMeasurement.attRelocOutInterSysCs"/>
      <enumeration value="hardHandoverInterSystemMeasurement.attRelocOutInterSysPs"/>
      <enumeration value="hardHandoverInterSystemMeasurement.attRelocInInterSysPs"/>
      <enumeration value="hardHandoverInterSystemMeasurement.succRelocInInterSysPs"/>
    </restriction>
  </simpleType>
  <simpleType name="measName">
    <union memberTypes="mc:measNameWithSubCounter mc:measNameWithoutSubCounter"/>
  </simpleType>
  <simpleType name="measResultType">
    <union memberTypes="decimal">
      <simpleType>
        <restriction base="string">
          <enumeration value="NIL"/>
        </restriction>
      </simpleType>
    </union>
  </simpleType>
</schema>
```

6.2 性能测量数据文件的 XML header 定义

在实际性能测量数据文件的中应该使用下面的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 "
>
```

附　录　A

（规范性附录）

Schema 文档补充说明

A.1　XML Schema文档结构标记约定如图A.1 所示。

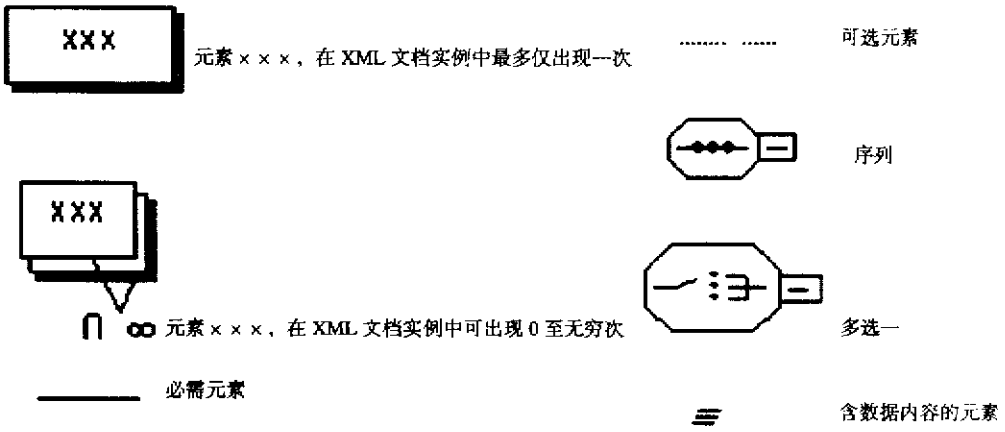
XXX
元素×××，在XML文档实例中最多仅出现一次
可选元素
序列
XXX
0 ∞ 元素×××，在XML文档实例中可出现0至无穷次
多选一
必需元素
含数据内容的元素

图 A.1　XML Schema 文档结构标记约定

A.2　性能测量数据文件的Schema定义<measCollec.xsd>

1）XML Schema 文档结构如图 A.2 所示。

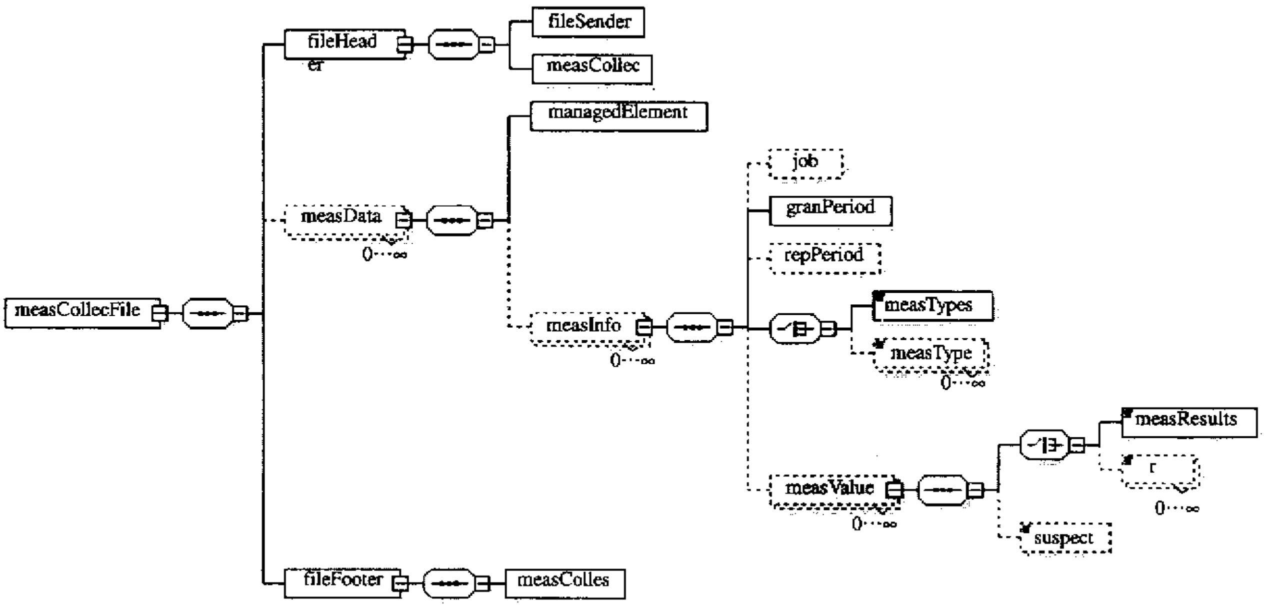

图 A.2　XML Schema 文档结构

2）XML Schema 文档元素/属性说明见表 A.1。

元素/属性　名称		元素/属性描述
measCollecFile		性能数据采集文件是该 Schema 的根元素。由三个子元素组成：文件头部（fileHeader)、采集数据(measData)以及文件尾部(FileFooter)
fileHeader	fileFormatVersion	文件头部。由两个子元素组成：文件发送方（fileSender）、测量采集开始时间（measCollec）。自身包含三个属性：文件格式版本（fileFormatVersion）、制造商名称（vendorName）和识别名前缀（dnPrefix）
	vendorName	
	dnPrefix	
measData		性能测量数据。在一份采集上报文件中可出现零（未采集到数据）至多次。由两个子元素组成：管理网元(managedElement)及其性能采集结果(measInfo)
fileFooter		文件尾部。包含子元素测量采集结束时间（measCollec）
fileSender	localDn	文件发送发。包含两个属性：本地识别名（localDN）、网元类型（elementType）
	elementType	
managedElement	localDn	被管网元。包括三个属性本地识别名（localDn）、用户友好名（userLabel）、软件版本（swVersion）
	userLabel	
	swVersion	
measInfo		测量信息。由 4 个子元素组成：测量任务（job）、测量粒度周期（granPeriod）、测量上报周期（repPeriod）、测量类型（measType/ measTypes）和测量值（measValue）
job		测量任务。该元素为可选元素。由其类型 JobID 惟一标识
granPeriod	duration	测量粒度周期。包含两个属性：持续时间（duration）、结束时间（endTime）
	endTime	
repPeriod	duration	测量上报周期。该元素为可选元素，包含惟一属性：持续时间（duration）
measTypes/measType		采集类型。均由 measName 扩展而来。在 XML 文件实例中，两个元素择一使用。不同的是 measTypes 是以列表方式呈现，且只出现一次；measType 可出现多次，由属性值为非负数的 p 加以区分
measType p		p 为属性限定。属性用于区分不同的 measType
measResults/r		采集结果。均由 measResultType 扩展而来。在 XML 文件实例中，两个元素择一使用。值为空表示该采集项的取值无法获得。不同的是 measResults 是以列表方式呈现，且只出现一次；r 可出现多次，由属性值为非负数的 p 加以区分。r 的 p 属性应与 measType 的 p 属性一一对应
r p		p 为属性限定。表示对<measType p>的一个采集结果应答。<r p>需和<measType p>一一对应
measValue	measObjLdn	采集值。由两个子元素组成：采集结果列表(measResults/r)和一个标记采集数据是否可信的标志位(suspect)。本身还包含一个属性：测量对象本地识别名(measObjLdn)
suspect		用于标记采集值是否可信。默认值为 False（即可信）
measCollec	beginTime	性能采集开始时间
	endTime	性能采集结束时间
measName		性能测量项名称。分为包含 SubCounter（measNameWithSubCounter）和不含 SubCounter（measNameWithoutSubCounter）两类。从 3GPP 规范中扩展而来
measNameWithSubCounter		含 SubCounter 的数据测量项名称。表示为 familyname.measurename.subcounter 形式。从 3GPP 规范中扩展而来
measNameWithoutSubCounter		不含 SubCounter 的数据测量项名称。表示为 familyname.measurename 形式。从 3GPP 规范中扩展而来

附录 B
（资料性附录）
性能管理功能相关 XML 文件示例

B.1 性能管理功能相关XML文件示例一

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="MeasDataCollection.xsl"?>

<!-- The following is an example of a XML schema based XML measurement report file without use of optional
positioning attributes on measurement types and results -->

<measCollecFile xmlns="http://latest/nmc-omc/cmNrm.doc#measCollec"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://latest/nmc-omc/cmNrm.doc#measCollec
D:\Downloads\GB\WCDMAM~2.XSD">

    <fileHeader fileFormatVersion="PM FILE V1.0" vendorName="Company NN"
dnPrefix="DC=a1.companyNN.com,SubNetwork=1,IRPAgent=1">
        <fileSender
localDn="SubNetwork=CountryNN,MeContext=MEC-Gbg-1,ManagedElement=RNC-Gbg-1"
elementType="RNC"/>
        <measCollec beginTime="2000-03-01T14:00:00+02:00"/>
    </fileHeader>
    <measData>
        <managedElement
localDn="SubNetwork=CountryNN,MeContext=MEC-Gbg-1,ManagedElement=RNC-Gbg-1" userLabel="RNC
Telecomville"/>
        <measInfo>
            <job jobId="1231"/>
            <granPeriod duration="PT900S" endTime="2000-03-01T14:14:30+02:00"/>
            <repPeriod duration="PT1800S"/>
            <measTypes>mscBasicMeasurement.failImsiAttachsPerCause.50000
hardHandoverInterSystemMeasurement.failRelocOutInterSysPsPerCause.0
mobileManagementMeasurement.failIntraSgsnRaUpdatePerCause.22222
mobileManagementMeasurement.failIntraSgsnRaUpdatePerCause.1</measTypes>
            <measValue measObjLdn="RncFunction=RF-1,UtranCell=Gbg-997">
                <measResults>234 345 567 789</measResults>
            </measValue>
            <measValue measObjLdn="RncFunction=RF-1,UtranCell=Gbg-998">
                <measResults>890 901 123 234</measResults>
            </measValue>
            <measValue measObjLdn="RncFunction=RF-1,UtranCell=Gbg-999">
                <measResults>456 567 678 789</measResults>
                <suspect>true</suspect>
            </measValue>
        </measInfo>
    </measData>
    <fileFooter>
        <measCollec endTime="2000-03-01T14:15:00+02:00"/>
    </fileFooter>
</measCollecFile>
```

B.2 性能管理功能相关XML文件示例二

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="MeasDataCollection.xsl"?>

<!-- The following is an example of a XML schema based XML measurement report file with use of optional
positioning attributes on measurement types and results -->

<measCollecFile xmlns="http://latest/nmc-omc/cmNrm.doc#measCollec"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://latest/nmc-omc/cmNrm.doc#measCollec
D:\Downloads\GB\WCDMAM~2.XSD">

    <fileHeader fileFormatVersion="PM FILE V1.0" vendorName="Company NN"
dnPrefix="DC=a1.companyNN.com,SubNetwork=1,IRPAgent=1">
        <fileSender
localDn="SubNetwork=CountryNN,MeContext=MEC-Gbg-1,ManagedElement=RNC-Gbg-1"
elementType="RNC"/>
        <measCollec beginTime="2000-03-01T14:00:00+02:00"/>
    </fileHeader>
    <measData>
        <managedElement
localDn="SubNetwork=CountryNN,MeContext=MEC-Gbg-1,ManagedElement=RNC-Gbg-1" userLabel="RNC
```

```
Telecomville"/>
        <measInfo>
            <job jobId="1231"/>
            <granPeriod duration="PT900S" endTime="2000-03-01T14:14:30+02:00"/>
            <repPeriod duration="PT1800S"/>
            <measType p="1">mscBasicMeasurement.attGetRoutingInfo</measType>
            <measType
p="2">hardHandoverInterSystemMeasurement.failRelocOutInterSysPsPerCause.0</measType>
            <measType
p="3">cellRrcConnectionMeasurement.failRrcConnReestabPerCause.50000</measType>
            <measType
p="5">hardHandoverInterSystemMeasurement.failRelocOutInterSysPsPerCause.5</measType>
            <measValue measObjLdn="RncFunction=RF-1,UtranCell=Gbg-997">
                <r p="1">234</r>
                <r p="2">345</r>
                <r p="3">567</r>
                <r p="4">789</r>
            </measValue>
            <measValue measObjLdn="RncFunction=RF-1,UtranCell=Gbg-998">
                <r p="1">890</r>
                <r p="2">901</r>
                <r p="3">123</r>
                <r p="4">234</r>
            </measValue>
            <measValue measObjLdn="RncFunction=RF-1,UtranCell=Gbg-999">
                <r p="1">456</r>
                <r p="2">567</r>
                <r p="3">678</r>
                <r p="4">789</r>
                <suspect>true</suspect>
            </measValue>
        </measInfo>
    </measData>
    <fileFooter>
        <measCollec endTime="2000-03-01T14:15:00+02:00"/>
    </fileFooter>
</measCollecFile>
```

参考文献

3GPP TS 32.401 Telecommunication management; Performance Management (PM); Concept and requirements

3GPP TS 32.403 Telecommunication management; Performance Management (PM); Performance measurements - UMTS and combined UMTS/GSM