ICS 33.040
M 15

YD

中华人民共和国通信行业标准

YD/T 1863.3-2009

2GHz WCDMA 数字蜂窝移动通信网 高速下行分组接入（HSDPA） 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

2GHz WCDMA Digital Cell Mobile Communications Network HSDPA Management Technical Specification
Part 3：CORBA Based Network Resource Model Design

2009-06-15 发布　　2009-09-01 实施

中华人民共和国工业和信息化部　发 布

目　次

前　言

《2GHz WCDMA 数字蜂窝移动通信网高速下行分组接入（HSDPA）网络管理技术要求》分为 3 个部分：

— 第 1 部分：配置网络资源模型

— 第 2 部分：性能网络资源模型

— 第 3 部分：基于 CORBA 技术的网络资源模型设计

本部分为第 3 部分。

本部分参考第三代移动通信伙伴项目（3GPP）关于 WCDMA Release 5 HSDPA 接入技术的以下 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) Solution Set (SS)（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) Solution Set (SS)（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) Solution Set (SS)（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) Solution Set (SS)（GERAN 网络资源集成参考点：基于 CORBA 接口设计）

本标准与上述国际标准之间的主要差异为：

— 增强 IDL 文件的注释说明；

— 增强配置和性能网络资源模型参数数据类型的定义，并对其名称取值进行规范；

— 配置资源模型的 IDL 定义增加一类文件 xxxNRMProfile.idl，包括 GenericNRMProfile.idl、IMDataProfile.idl 和 UtranNRMProfile.idl。这类文件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；

— 完善配置和性能网络资源模型 Schema 文件定义，并针对行业标准的特殊要求，规范了其目标域名空间的域名取值。

本标准与上述 3GPP 相关标准的一致性程度为非等效。

附录 A 为规范性附录，附录 B 为资料性附录。

本部分由中国通信标准化协会提出并归口。

本部分起草单位：北京邮电大学、中国移动通信集团公司、北京市天元网络技术股份有限公司

本部分主要起草人：刘会永、陈兴渝、芮兰兰、朱　凯、刘思平、李冶文、梁　亮

2GHz cdma2000 数字蜂窝移动通信网 高速分组数据（HSDPA）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

1 范围

本部分规定了2GHz WCDMA数字蜂窝移动通信网高速下行分组接入（HSDPA）的网络管理接口的CORBA/IDL定义。

本部分适用于2GHz WCDMA数字蜂窝移动通信网高速下行分组接入（HSDPA）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 1863.1–2009	2GHz WCDMA数字蜂窝移动通信网高速下行分组接入（HSDPA）网络管理技术要求 第1部分：配置网络资源模型
YD/T 1863.2–2009	2GHz WCDMA数字蜂窝移动通信网高速下行分组接入（HSDPA）网络管理技术要求 第2部分：性能网络资源模型

3 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

配置网络资源模型设计中有3类idl文件，这3类文档及其用途如下：

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

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

3）xxxNRMProfile.idl包括GenericNRMProfile.idl、IMDataProfile.idl和UtranNRMProfile.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

4.1.1 GenericNRMDefs

```
//File "GenericNRMDefs.idl"
//The IRP document version number is "Generic NRM V1.0"
```

```
#ifndef GenericNRMDefs_idl
#define GenericNRMDefs_idl

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

module GenericNRMDefs
{
    //Definitions for abstract MO class Top

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

    //Definitions for MO class IRPAgent

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

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

    //Definitions for abstract MO class GenericIRP

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

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

    //Definitions for MO class SubNetwork

    interface SubNetwork: Top
    {
        const string CLASS = "SubNetwork";
    // Attribute Names
    //
    const string subNetworkId = "subNetworkId";
    const string dnPrefix = "dnPrefix";
    const string userLabel = "userLabel";
    const string setOfMcc = "setOfMcc";
    const string userDefinedNetworkType = "userDefinedNetworkType ";
};

//Definitions for MO class MeContext

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

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

//Definitions for MO class ManagementNode

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

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

//Definitions for MO class ManagedElement

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

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

//Definitions for abstract MO class ManagedFunction

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

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

//Definitions for MO class VsDataContainer

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

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

//Definitions for MO class SignallingPoint

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

    // Attribute Names
    //
```

```
        const string signallingPointId = "signallingPointId";
        const string signallingInfo = "signallingInfo";
        const string signallingPointType = "signallingPointType";
        const string userLabel = "userLabel";
    };

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

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

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

        // Attribute Names
        //
        const string signallingLinkTPId = "signallingLinkTPId";
        const string userLabel = "userLabel";
        const string slc = "slc";
        const string slsNormalList = "slsNormalList";
        const string slsCurrentList = "slsCurrentList";
        const string linkStatus = "linkStatus";
        const string bandwidth = "bandwidth";
    };

};
#endif
```

4.1.2 GenericNRMProfile

```
//File "GenericNRMProfile.idl"
//The IRP document version number is "Generic NRM V1.0"
#ifndef GenericNRMProfile_idl
#define GenericNRMProfile_idl

#include "GenericNRMSystem.idl"
```

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

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

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

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


   interface SubNetwork: Top
   {
      readonly attribute GenericNRMSystem::ObjectIdType subNetworkId;
      readonly attribute GenericNRMSystem::DNPrefixType dnPrefix;
               attribute wstring userLabel;
      readonly attribute GenericNRMSystem::MobileCountryCodeSetType setOfMcc;
      readonly attribute GenericNRMSystem::NetworkTypeType userDefinedNetworkType;
    // The following notifications may be sent from this MO,
      // notifyObjectCreation
      // notifyObjectDeletion
      // notifyAttributeValueChange
      // notifyAckStateChanged
      // notifyChangedAlarm
      // notifyClearedAlarm
      // notifyNewAlarm
      // notifyComments
      // notifyAlarmListRebuilt
      // notifyPotentialFaultyAlarmList
};

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

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


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

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

```
        // notifyPotentialFaultyAlarmList
  };

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

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

  interface ManagedFunction : Top
  {
      attribute wstring userLabel;
  };

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

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

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

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

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

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

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

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

4.1.3 GenericNRMSystem

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

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

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

typedef sequence<DN> DNListType;

typedef string ObjectIdType;

typedef DN DNPrefixType;

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

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

typedef unsigned long UserDefinedStateType;

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

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

enum SignallingPointLengthType
{
    bits_24,
    bits_14
};

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

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

enum SignallingLinkTypeType
{
    N_SS7,
    W_SS7
};

typedef unsigned long linkDirectType;
const linkDirectType toHstp=1;
const linkDirectType toLstp=2;
const linkDirectType toCdmaGmsc=3;
const linkDirectType toMsc=4;
const linkDirectType toHlr=5;
const linkDirectType toMc=6;
const linkDirectType toScp=7;
const linkDirectType toBsc=8;
const linkDirectType toCncPstn=9;
const linkDirectType toCtPstn=10;
const linkDirectType toCmcc=11;
const linkDirectType toCtt=12;
const linkDirectType toVoiceMailBox=13;
const linkDirectType toColorRing=14;
const linkDirectType toGsm=15;
const linkDirectType toOthers=16;

typedef unsigned short SlcType;

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

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

```
        const LinkStatusType remoteInhibited_UDS = 6;

        typedef unsigned long BandwidthType;
};

#endif
```

4.1.4 IMDataDefs

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

#include "GenericNRMDefs.idl"

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

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

```
#endif
```

4.1.5 IMDataProfile

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

#include "GenericNRMSystem.idl"
#include "GenericNRMProfile.idl"

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

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

#endif
```

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 for HSDPA V1.0"
#ifndef UtranNRMDefs_idl
#define UtranNRMDefs_idl

#include "GenericNRMDefs.idl"

/**
  * 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 mcc = "mcc";
                     const string mnc = "mnc";
                     const string maxCallCapability = "maxCallCapability";
                     const string maxThroughput = "maxThroughput";
        };

        //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";
        // HSDPA specific attribute(s)
        const string hsFlag = "hsFlag";
};

//Definitions for MO class IubLink

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

            // including all Attribute Names from
            // MO Class GenericNRMDefs::ManagedFunction
            // additional Attribute Names is as follows.
            //
            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 relatedIubLink = "relatedIubLink";
            const string cId = "cId";
            const string localCellId = "localCellId";
            const string uarfcnUl = "uarfcnUl";
            const string uarfcnDl = "uarfcnDl";
            const string primaryScramblingCode = "primaryScramblingCode";
            const string primaryCpichPower = "primaryCpichPower";
        const string maximumTransmissionPower = "maximumTransmissionPower";
        const string primarySchPower = "primarySchPower";
        const string secondarySchPower = "secondarySchPower";
        const string bchPower = "bchPower";
        const string lac = "lac";
        const string rac = "rac";
        const string sac = "sac";
        const string uraList = "uraList";
```

```
    const string cellMode = "cellMode";
    // HSDPA specific attribute(s)
    const string hsFlag = "hsFlag";
    const string hsStat = "hsStat";
    const string nbrCodHspdsch = "nbrCodHspdsch";
    const string nbrCodHsscch = "nbrCodeHsscch";
};

//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 uarfcnUl = "uarfcnUl";
        const string uarfcnDl = "uarfcnDl";
        const string primaryScramblingCode = "primaryScramblingCode";
        const string primaryCpichPower = "primaryCpichPower";
        const string lac = "lac";
        const string cellMode = "cellMode";
        const string userLabel = "userLabel";
    // HSDPA specific attribute(s)
        const string hsFlag = "hsFlag";
        const string hsStat = "hsStat";
     };

     //Definitions for MO class ExternalUtranCell

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

    // Attribute Names
    //
        const string externalUtranCellId = "externalUtranCellId";
        const string mcc = "mcc";
        const string mnc = "mnc";
        const string cId = "cId";
        const string rncId = "rncId";
        const string uarfcnUl = "uarfcnUl";
        const string uarfcnDl = "uarfcnDl";
```

```
            const string primaryScramblingCode = "primaryScramblingCode";
            const string primaryCpichPower = "primaryCpichPower";
            const string lac = "lac";
            const string rac = "rac";
            const string cellMode = "cellMode";
        // HSDPA specific attribute(s)
            const string hsFlag = "hsFlag";
            const string hsStat = "hsStat";
            const string nbrCodHspdsch = "nbrCodHspdsch";
            const string nbrCodHsscch = "nbrCodHsscch";
        };

        //Definitions 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";
    };

    //Definitions for MO class 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 UtranNRMProfile

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

#include "GenericNRMProfile.idl"
#include "GenericNRMDefs.idl"
#include "UtranNRMSystem.idl"

/**
  * 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 : GenericNRMProfile::ManagedFunction
            {
            readonly attribute GenericNRMSystem::ObjectIdType rncFunctionId;
                  attribute unsigned long rncId;
            readonly attribute unsigned long mcc;
            readonly attribute unsigned long mnc;
            readonly attribute unsigned long maxCallCapability;
            readonly attribute unsigned long maxThroughput;

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

        interface NodeBFunction : GenericNRMProfile::ManagedFunction
        {
            readonly attribute GenericNRMSystem::ObjectIdType nodeBFunctionId;
            readonly attribute GenericNRMSystem::DN relatedIubLink;
        // HSDPA specific attribute(s)
            readonly attribute unsigned long hsFlag;

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

    interface IubLink : GenericNRMProfile::ManagedFunction
        {
        readonly attribute GenericNRMSystem::ObjectIdType iubLinkId;
            readonly 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 : GenericNRMProfile::ManagedFunction
```

```
{
    readonly attribute GenericNRMSystem::ObjectIdType utranCellId;
    readonly attribute GenericNRMSystem::DN relatedIubLink;
              attribute unsigned long cId;
              attribute unsigned long localCellId;
    readonly attribute unsigned long uarfcnUl;
    readonly attribute unsigned long uarfcnDl;
    readonly attribute unsigned long primaryScramblingCode;
    readonly attribute float primaryCpichPower;
    readonly attribute float maximumTransmissionPower;
    readonly attribute float primarySchPower;
    readonly attribute float secondarySchPower;
    readonly attribute float bchPower;
readonly attribute unsigned long lac;
    readonly attribute unsigned long rac;
    readonly attribute unsigned long sac;
    readonly attribute UtranNRMSystem::UraListType uraList;
    readonly attribute UtranNRMSystem::CellModeEnumType cellMode;
// HSDPA specific attribute(s)
    readonly attribute unsigned long hsFlag;
          attribute unsigned long hsStat;
    readonly attribute unsigned long nbrCodHspdsch;
    readonly attribute unsigned long nbrCodHsscch;

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

interface UtranRelation : GenericNRMProfile::Top
{
    readonly attribute GenericNRMSystem::ObjectIdType utranRelationId;
              attribute UtranNRMSystem::AdjacentCellType adjacentCell;
    readonly attribute unsigned long uarfcnUl;
    readonly attribute unsigned long uarfcnDl;
    readonly attribute unsigned long primaryScramblingCode;
    readonly attribute float primaryCpichPower;
```

```
        readonly attribute unsigned long lac;
        readonly attribute UtranNRMSystem::CellModeEnumType cellMode;
                attribute string userLabel;
    // HSDPA specific attribute(s)
        readonly attribute unsigned long hsFlag;
            attribute unsigned long hsStat;

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

    interface ExternalUtranCell : GenericNRMProfile::ManagedFunction
    {
        readonly attribute GenericNRMSystem::ObjectIdType externalUtranCellId;
                    attribute unsigned long mcc;
            attribute unsigned long mnc;
            attribute unsigned long cId;
            attribute unsigned long rncId;
            attribute unsigned long uarfcnUl;
            attribute unsigned long uarfcnDl;
            attribute unsigned long primaryScramblingCode;
            attribute float primaryCpichPower;
            attribute unsigned long lac;
            attribute unsigned long rac;
        readonly attribute UtranNRMSystem::CellModeEnumType cellMode;
    // HSDPA specific attribute(s)
        readonly attribute unsigned long hsFlag;
            attribute unsigned long hsStat;
        readonly attribute unsigned long nbrCodHspdsch;
        readonly attribute unsigned long nbrCodHsscch;

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

    interface GsmRelation : GenericNRMProfile::Top
{
        readonly attribute GenericNRMSystem::ObjectIdType gsmRelationId;
         attribute GenericNRMSystem::DN adjacentCell;
    readonly attribute unsigned long bcchFrequency;
    readonly attribute unsigned long ncc;
```

```
            readonly attribute unsigned long bcc;
            readonly attribute unsigned long lac;
                    attribute string userLabel;

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

          interface ExternalGSMCell : GenericNRMProfile::ManagedFunction
          {
                  readonly attribute GenericNRMSystem::ObjectIdType externalGsmCellId;
                    attribute unsigned long cellIdentity;
                    attribute unsigned long 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
```

4.2.3 UtranNRMSystem

```
//File "UtranNRMSystem.idl"
//The IRP document version number is "UTRAN NRM for HSDPA 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 cellGlobalId;
      };

      typedef GenericNRMSystem::ULongSet UraListType;

      enum CellModeEnumType
      {
               FDDMode,
               TDDMode_3_84Mcps,
               TDDMode_1_28Mcps
      };
};
#endif
```

5 性能网络资源模型设计

5.1 性能参数的 IDL 定义

5.1.1 HsdpaMeasCollecDefs

```
//File HsdpaMeasCollecDefs.idl
#ifndef HsdpaMeasCollecDefs_idl
#define HsdpaMeasCollecDefs_idl

/**
* This module defines measurementType names constants
*/

module HsdpaMeasCollecDefs
{
       //RNC measurement
       module rabAssignmentMeasurement
       {
              //establish 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";
        //modify RAB
        const string attRabAssignModCsPerType= "attRabAssignModCsPerType";
        const string succRabAssignModCsPerType= "succRabAssignModCsPerType";
        const string failRabAssignModCsPerCause= "failRabAssignModCsPerCause";
        const string attRabAssignModPsPerType= "attRabAssignModPsPerType";
        const string succRabAssignModPsPerType= "succRabAssignModPsPerType";
        const string failRabAssignModPsPerCause= "failRabAssignModPsPerCause";
        //release RAB
        const string attRabAssignRelCsPerType= "attRabAssignRelCsPerType";
        const string succRabAssignRelCsPerType= "succRabAssignRelCsPerType";
        const string failRabAssignRelCsPerCause= "failRabAssignRelCsPerCause";
        const string attRabAssignRelPsPerType= "attRabAssignRelPsPerType";
        const string succRabAssignRelPsPerType= "succRabAssignRelPsPerType";
        const string failRabAssignRelPsPerCause= "failRabAssignRelPsPerCause";
};

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 rncSoftHandoverMeasurement
    {
        const string attRlAddInSho= "attRlAddInSho";
        const string failRlAddInShoPerCause= "failRlAddInShoPerCause";
        const string attRlDelInSho= "attRlDelInSho";
        const string failRlDelInShoPerCause= "failRlDelInShoPerCause";
    };

    module rncHardHandoverMeasurement
    {
        const string attHho= "attHho";
        const string failHhoPerCause= "failHhoPerCause";
    };

    module rncRelocationMeasurement
    {
        //CS relocation out with UE not involved
        const string attRelocOutPrepWithUeNotInvCsPerCause= "attRelocOutPrepWithUeNotInvCsPerCause";
        const string failRelocOutPrepWithUeNotInvCsPerCause=
"failRelocOutPrepWithUeNotInvCsPerCause";
        const string attRelocOutWithUeNotInvCs= "attRelocOutWithUeNotInvCs";
        const string failRelocOutWithUeNotInvCsPerCause= "failRelocOutWithUeNotInvCsPerCause";
        //CS relocation out with UE involved
        const string attRelocOutPrepWithUeInvCsPerCause= "attRelocOutPrepWithUeInvCsPerCause";
        const string failRelocOutPrepWithUeInvCsPerCause= "failRelocOutPrepWithUeInvCsPerCause";
        const string attRelocOutWithUeInvCs= "attRelocOutWithUeInvCs";
        const string failRelocOutWithUeInvCsPerCause= "failRelocOutWithUeInvCsPerCause";
        //PS relocation out with UE not involved
        const string attRelocOutPrepWithUeNotInvPsPerCause= "attRelocOutPrepWithUeNotInvPsPerCause";
        const string failRelocOutPrepWithUeNotInvPsPerCause= "failRelocOutPrepWithUeNotInvPsPerCause";
        const string attRelocOutWithUeNotInvPs= "attRelocOutWithUeNotInvPs";
        const string failRelocOutWithUeNotInvPsPerCause= "failRelocOutWithUeNotInvPsPerCause";
        //PS relocation out with UE involved
        const string attRelocOutPrepWithUeInvPsPerCause= "attRelocOutPrepWithUeInvPsPerCause";
        const string failRelocOutPrepWithUeInvPsPerCause= "failRelocOutPrepWithUeInvPsPerCause";
        const string attRelocOutWithUeInvPs= "attRelocOutWithUeInvPs";
        const string failRelocOutWithUeInvPsPerCause= "failRelocOutWithUeInvPsPerCause";
        //CS relocation in with UE not involved
        const string attRelocInWithUeNotInvCsPerCause= "attRelocInWithUeNotInvCsPerCause";
        const string failRelocInWithUeNotInvCsPerCause= "failRelocInWithUeNotInvCsPerCause";
        //CS relocation in with UE involved
        const string attRelocInWithUeInvCsPerCause= "attRelocInWithUeInvCsPerCause";
        const string failRelocInWithUeInvCsPerCause= "failRelocInWithUeInvCsPerCause";
        //PS relocation in with UE not involved
        const string attRelocInWithUeNotInvPsPerCause= "attRelocInWithUeNotInvPsPerCause";
        const string failRelocInWithUeNotInvPsPerCause= "failRelocInWithUeNotInvPsPerCause";
        //PS relocation in with UE involved
        const string attRelocInWithUeInvPsPerCause= "attRelocInWithUeInvPsPerCause";
        const string failRelocInWithUeInvPsPerCause= "failRelocInWithUeInvPsPerCause";
};

module rncInterSystemHandoverMeasurement
{
        //CS inter system handover from WCDMA to GSM
        const string attRelocOutPrepInterSysCsPerCause= "attRelocOutPrepInterSysCsPerCause";
        const string failRelocOutPrepInterSysCsPerCause= "failRelocOutPrepInterSysCsPerCause";
        const string attRelocOutInterSysCs= "attRelocOutInterSysCs";
        const string failRelocOutInterSysCsPerCause= "failRelocOutInterSysCsPerCause";
        //CS inter system handover from GSM to WCDMA
        const string attRelocInInterSysCsPerCause= "attRelocInInterSysCsPerCause";
        const string failRelocInInterSysCsPerCause= "failRelocInInterSysCsPerCause";
        //PS inter system handover  rom WCDMA to GPRS
        //const string attRelocOutPrepInterSysPsPerCause= "attRelocOutPrepInterSysPsPerCause";
        //const string failRelocOutPrepInterSysPsPerCause= "failRelocOutPrepInterSysPsPerCause";
        const string attRelocOutInterSysPs= "attRelocOutInterSysPs";
        const string failRelocOutInterSysPsPerCause= "failRelocOutInterSysPsPerCause";
        //PS inter system handover from GPRS to WCDMA
        const string attRelocInInterSysPs= "attRelocInInterSysPs";
        const string succRelocInInterSysPs= "succRelocInInterSysPs";
};

module iuInterfaceThroughputMeasurement
{
        const string iuUlSigThroughputCs = "iuUlSigThroughputCs";
        const string iuDlSigThroughputCs = "iuDlSigThroughputCs";
        const string iuUlDataThroughputCsPerType = "iuUlDataThroughputCsPerType";
        const string iuDlDataThroughputCsPerType = "iuDlDataThroughputCsPerType";
        const string iuUlSigThroughputPs = "iuUlSigThroughputPs";
        const string iuDlSigThroughputPs = "iuDlSigThroughputPs";
        const string iuUlDataThroughputPsPerType = "iuUlDataThroughputPsPerType";
        const string iuDlDataThroughputPsPerType = "iuDlDataThroughputPsPerType";
};

module iurInterfaceThroughputMeasurement
{
        const string iurUlSigThroughput = "iurUlSigThroughput";
        const string iurDlSigThroughput = "iurDlSigThroughput";
        const string iurUlDataThroughput = "iurUlDataThroughput";
        const string iurDlDataThroughput = "iurDlDataThroughput";
};

module rlcConnectionMeasurement
{
        const string nbrRlcBlockSentPerMode = "nbrRlcBlockSentPerMode";
        const string nbrRlcBlockRecvPerMode = "nbrRlcBlockRecvPerMode";
        const string nbrDiscardedRlcBlocksByRnc = "nbrDiscardedRlcBlocksByRnc";
        const string nbrRetransmittedRlcBlocksToUe = "nbrRetransmittedRlcBlocksToUe";
};

//Cell measurement
module cellRrcConnectionMeasurement
{
        const string attRrcConnSetupPerCause = "attRrcConnSetupPerCause";
        const string succRrcConnSetupPerCause = "succRrcConnSetupPerCause";
        const string failRrcConnSetupPerCause= "failRrcConnSetupPerCause";
        const string attRrcConnReestab= "attRrcConnReestab";
        const string failRrcConnReestabPerCause= "failRrcConnReestabPerCause";
};

module cellSoftHandoverMeasurement
{
        //soft handover
        const string attRlAddInSho= "attRlAddInSho";
        const string failRlAddInShoPerCause= "failRlAddInShoPerCause";
        const string attRlDelInSho= "attRlDelInSho";
        const string failRlDelInShoPerCause= "failRlDelInShoPerCause";
};

module hardHandoverIntraCellMeasurement
```

```
{
    const string attHhoOutIntraCell= "attHhoOutIntraCell";
    const string failHhoOutIntraCellPerCause= "failHhoOutIntraCellPerCause";
};

module iubRlManagementMeasurement
{
    const string attRlSetupIub= "attRlSetupIub";
    const string failRlSetupIubPerCause= "failRlSetupIubPerCause";
    const string attRlAddIub= "attRlAddIub";
    const string failRlAddIubPerCause= "failRlAddIubPerCause";
    const string attRlDelIub= "attRlDelIub";
    const string succRlDelIub= "succRlDelIub";
};

module iurRlManagementMeasurement
{
    const string attRlSetupIur= "attRlSetupIur";
    const string failRlSetupIurPerCause= "failRlSetupIurPerCause";
    const string attRlAddIur= "attRlAddIur";
    const string failRlAddIurPerCause= "failRlAddIurPerCause";
    const string attRlDelIur= "attRlDelIur";
    const string succRlDelIur= "succRlDelIur";
};

module cellTrafficMeasurement
{
    const string cellCcchTraffic = "cellCcchTraffic";
    const string cellCtchTraffic = "cellCtchTraffic";
    const string cellDcchTraffic = "cellDcchTraffic";
    const string cellDtchTraffic = "cellDtchTraffic";
};

    module cellPagingMeasurement
{
    const string attPagingType1FromUtran = "attPagingType1FromUtran";
    const string succPagingType1FromUtran = "succPagingType1FromUtran";
    const string attPagingType2FromUtran = "attPagingType2FromUtran";
};

   //UtranCell measurement for HSDPA
module cellHsSetupMeas
{
    const string attHsRabAssignEstabPerType = "attHsRabAssignEstabPerType";
    const string succHsRabAssignEstabperType = "succHsRabAssignEstabperType";
```

```
        const string nbrhsRncRelbyUserInact = "nbrhsRncRelbyUserInact";
        const string nbrHsCnRel = "nbrHsCnRel";
        const string attMacdFlowSetup = "attMacdFlowSetup";
        const string succMacdFlowSetup = "succMacdFlowSetup";
        const string attHsRbSetup = "attHsRbSetup";
        const string succHsRbSetup = "succHsRbSetup";
    };

    module cellHsChSwMeas
    {
        const string attFachToHsDsch = "attFachToHsDsch";
        const string attDchToHsDsch = "attDchToHsDsch";
        const string succFachToHsDsch = "succFachToHsDsch";
        const string succDchToHsDsch = "succDchToHsDsch";
        const string attHsDschToFach = "attHsDschToFach";
        const string attHsDschToDch = "attHsDschToDch";
        const string succHsDschToFach = "succHsDschToFach";
        const string succHsDschToDch = "succHsDschToDch";
    };

    module cellHsRepointMeas
    {
        const string attServCellUpd = "attServCellUpd";
        const string succServCellUpd = "succServCellUpd";
    };

    module cellHsTraffMeas
    {
        const string nbrMacHsPduTx = "nbrMacHsPduTx";
        const string nbrMacHsAckedPduTx = "nbrMacHsAckedPduTx";
        const string nbrOctMacHsAckedPduTx = "nbrOctMacHsAckedPduTx";
        const string nbrNonEptBuffTti = "nbrNonEptBuffTti";
        const string nbrSubsNonEptBuffPerTti = "nbrSubsNonEptBuffPerTti";
        const string nbrMeanSubs = "nbrMeanSubs";
    };

    module cellHsResMeas
    {
        const string nbrMeanUsedHspdschCode = "nbrMeanUsedHspdschCode";
        const string nbrMeanUsedHsscchCode = "nbrMeanUsedHsscchCode";
        const string cellNonHsMeanTxPower = "cellNonHsMeanTxPower";
        const string cellNonHsMaxTxPower = "cellNonHsMaxTxPower";
        const string cellMeanTxPower = "cellMeanTxPower";
        const string cellMaxTxPower = "cellMaxTxPower";
    };
    //UtranRelation measurement
module hardHandoverInterCellIntraNodeBMeasurement
{
    const string attHhoOutInterCellIntraNodeB= "attHhoOutInterCellIntraNodeB";
    const string failHhoOutInterCellIntraNodeBPerCause= "failHhoOutInterCellIntraNodeBPerCause";
};

module hardHandoverInterNodeBIntraRncMeasurement
{
    const string attHhoOutInterNodeBIntraRnc= "attHhoOutInterNodeBIntraRnc";
    const string failHhoOutInterNodeBIntraRncPerCause= "failHhoOutInterNodeBIntraRncPerCause";
};

module hardHandoverInterRncViaIurMeasurement
{
    const string attHhoOutInterRncViaIur= "attHhoOutInterRncViaIur";
    const string failHhoOutInterRncViaIurPerCause= "failHhoOutInterRncViaIurPerCause";
};

module hardHandoverInterRncMeasurement
{
    const string attHhoOutInterRncCn = "attHhoOutInterRncCn";
    const string failHhoOutInterRncCnPerCause = "failHhoOutInterRncCnPerCause";
};

//GsmRelation measurement
module hardHandoverInterSystemMeasurement
{
    //CS inter system handover from WCDMA to GSM
    const string attRelocOutPrepInterSysCsPerCause= "attRelocOutPrepInterSysCsPerCause";
    const string failRelocOutPrepInterSysCsPerCause= "failRelocOutPrepInterSysCsPerCause";
    const string attRelocOutInterSysCs= "attRelocOutInterSysCs";
    const string failRelocOutInterSysCsPerCause= "failRelocOutInterSysCsPerCause";
    //CS inter system handover from GSM to WCDMA
    const string attRelocInInterSysCsPerCause= "attRelocInInterSysCsPerCause";
    const string failRelocInInterSysCsPerCause= "failRelocInInterSysCsPerCause";
    //PS inter system handover from WCDMA to GPRS
    //const string attRelocOutPrepInterSysPsPerCause= "attRelocOutPrepInterSysPsPerCause";
    //const string failRelocOutPrepInterSysPsPerCause= "failRelocOutPrepInterSysPsPerCause";
    const string attRelocOutInterSysPs= "attRelocOutInterSysPs";
    const string failRelocOutInterSysPsPerCause= "failRelocOutInterSysPsPerCause";
    //PS inter system handover from GPRS to WCDMA
    const string attRelocInInterSysPs= "attRelocInInterSysPs";
    const string succRelocInInterSysPs= "succRelocInInterSysPs";
```

```
        };

};
#endif
```

5.2 数据类型的 IDL 定义

5.2.1 HsdpaMeasCollecSystem

```
//File "HspdaMeasCollecSystem.idl"
#ifndef HspdaMeasCollecSystem_idl
#define HspdaMeasCollecSystem_idl

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

    // typedef    unsigned long    CountType;
    typedef    unsigned long    WCDMAMeasurementType1;
    typedef    float    WCDMAMeasurementType2;

    typedef unsigned short CauseType;
    const CauseType sum = 0;
    const CauseType other = 65535;
    const CauseType noResponse = 65534;

    // The following RANAP causes are defined in the section 9.2.1.4 of 3GPP TS 25.413 v5.5.0.
    typedef CauseType 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; // HSDPA specified
    const RANAPCause requestedCipheringAndOrIntegrityProtectionAlgorithmsNotSupported = 12;
    const RANAPCause conflictWithAlreadyExistingIntegrityProtectionAndOrCipheringInformation = 13;
    const RANAPCause failureInTheRadioInterfaceProcedure = 14; // HSDPA specified
const RANAPCause releaseDueToUtranGeneratedReason = 15;
const RANAPCause userInactivity_RANAP = 16; // HSDPA specified
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; // HSDPA specified
const RANAPCause resourceOptimisationRelocation = 41; // HSDPA specified
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; // HSDPA specified

//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 CauseType 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 numberOfUlCodesNotSupported_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 CauseType 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 CauseType 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 CauseType 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 CauseType 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 CauseType 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 CauseType 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 CauseType 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 CauseType 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 CauseType 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 Location Update failure causes are defined in the section 10.5.3.6 of 3GPP TS 24.008 v6.1.0.
    typedef CauseType LocationUpdateFailureCause;
    const LocationUpdateFailureCause imsiUnknownInHLR_Imsi = 2;
    const LocationUpdateFailureCause illegalMS_Imsi = 3;
    const LocationUpdateFailureCause imsiUnknownInVLR = 4;
    const LocationUpdateFailureCause imeiNotAccepted = 5;
    const LocationUpdateFailureCause illegalME_Imsi = 6;
    const LocationUpdateFailureCause plmnNotAllowed_Imsi = 11;
    const LocationUpdateFailureCause locationAreaNotAllowed_Imsi =12;
    const LocationUpdateFailureCause roamingNotAllowedInThisLocationArea_Imsi = 13;
    const LocationUpdateFailureCause noSuitableCellsInLocationArea_Imsi = 15;
    const LocationUpdateFailureCause networkFailure_Imsi = 17;
    const LocationUpdateFailureCause macFailure_Imsi = 20;
    const LocationUpdateFailureCause synchFailure_Imsi = 21;
    const LocationUpdateFailureCause congestion_Imsi = 22;
    const LocationUpdateFailureCause gsmAuthenticationUnacceptable_Imsi = 23;
    const LocationUpdateFailureCause serviceOptionNotSupported_Imsi = 32;
    const LocationUpdateFailureCause requestedServiceOptionNotSubscribed_Imsi = 33;
    const LocationUpdateFailureCause serviceOptionTemporarilyOutOfOrder_Imsi = 34;
    const LocationUpdateFailureCause callCannotBeIdentified = 38;
    const LocationUpdateFailureCause failRetryUponEntryIntoANewCell_Imsi = 48;
    //value range 48 - 63 is used to retry upon entry into a new cell;
    const LocationUpdateFailureCause semanticallyIncorrectMessage_Imsi = 95;
    const LocationUpdateFailureCause invalidMandatoryInformation_Imsi = 96;
    const LocationUpdateFailureCause messageTypeNon_existentOrNotImplemented_Imsi = 97;
    const LocationUpdateFailureCause messageTypeNotCompatibleWithTheProtocolState_Imsi = 98;
    const LocationUpdateFailureCause informationElementNon_existentOrNotImplemented_Imsi = 99;
    const LocationUpdateFailureCause conditionalIeError_Imsi = 100;
    const LocationUpdateFailureCause messageNotCompatibleWithTheProtocolState_Imsi = 101;
    const LocationUpdateFailureCause 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 CauseType 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 CauseType 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 attach failure causes are defined in the section 10.5.5.14 of 3GPP TS 24.008 v6.1.0.
    typedef CauseType 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 CauseType 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 TrafficClassType;
    const TrafficClassType conversational = 1;
    const TrafficClassType streaming = 2;
    const TrafficClassType interactive = 3;
    const TrafficClassType background = 4;

    typedef unsigned short RlcMode;
    const RlcMode transparentMode = 1;
    const RlcMode unacknowledgedMode   = 2;
    const RlcMode acknowledgedMode = 3;
};
#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<HspdaMeasCollec.xsd>

下面的Schema文件中用到的字段的说明参见附录A，示例参见附录B。

版本号：1.0

```
<?xml version="1.0" encoding="UTF-8"?>
<!--REFERENCE: 3GPP TS 32.435 V7.2.0 measCollec.xsd-->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:mc="std://yd-t/hsdpa/itf-n/mc/2007"
targetNamespace="std://yd-t/hsdpa/itf-n/mc/2007" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.0">
    <xsd:element name="measCollecFile">
        <xsd:complexType>
```

```
            <xsd:sequence>
                <xsd:element name="fileHeader">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="fileSender">
                                <xsd:complexType>
                                    <xsd:attribute name="localDn" type="xsd:string" use="optional"/>
                                    <xsd:attribute name="elementType" type="xsd:string"
use="optional"/>
                                    <xsd:attribute name="inDomain" type="mc:domainType"
use="optional"/>
                                    <!--add an optional attribute "inDomain" indicating in which domain
the ME resides-->
                                </xsd:complexType>
                            </xsd:element>
                            <xsd:element name="measCollec">
                                <xsd:complexType>
                                    <xsd:attribute name="beginTime" type="xsd:dateTime"
use="required"/>
                                </xsd:complexType>
                            </xsd:element>
                        </xsd:sequence>
                        <xsd:attribute name="fileFormatVersion" type="xsd:string" use="required"/>
                        <xsd:attribute name="vendorName" type="xsd:string" use="optional"/>
                        <xsd:attribute name="dnPrefix" type="xsd:string" use="optional"/>
                    </xsd:complexType>
                </xsd:element>
                <xsd:element name="measData" minOccurs="0" maxOccurs="unbounded">
                    <xsd:complexType>
                        <xsd:sequence>
                            <xsd:element name="managedElement">
                                <xsd:complexType>
                                    <xsd:attribute name="localDn" type="xsd:string" use="optional"/>
                                    <xsd:attribute name="userLabel" type="xsd:string"
use="optional"/>
                                    <xsd:attribute name="swVersion" type="xsd:string"
use="optional"/>
                                    <xsd:attribute name="inDomain" type="mc:domainType"
use="optional"/>
                                    <!--add an optional attribute "inDomain" indicating in which domain
the ME resides-->
                                </xsd:complexType>
                            </xsd:element>
                            <xsd:element name="measInfo" minOccurs="0" maxOccurs="unbounded">
                                <xsd:complexType>
```

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

```
                        <xsd:sequence>
                          <xsd:choice>
                            <xsd:element name="measResults">
                              <xsd:simpleType>
                                <xsd:list
itemType="mc:measResultType"/>
                              </xsd:simpleType>
                            </xsd:element>
                            <xsd:element name="r" minOccurs="0"
maxOccurs="unbounded">
                              <xsd:complexType>
                                <xsd:simpleContent>
                                  <xsd:extension
base="mc:measResultType">
                                    <xsd:attribute
name="p" type="xsd:positiveInteger" use="required"/>
                                  </xsd:extension>
                                </xsd:simpleContent>
                              </xsd:complexType>
                            </xsd:element>
                          </xsd:choice>
                          <xsd:element name="suspect"
type="xsd:boolean" minOccurs="0"/>
                        </xsd:sequence>
                        <xsd:attribute name="measObjLdn"
type="xsd:string" use="required"/>
                      </xsd:complexType>
                    </xsd:element>
                  </xsd:sequence>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
        </xsd:element>
        <xsd:element name="fileFooter">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="measCollec">
                <xsd:complexType>
                  <xsd:attribute name="endTime" type="xsd:dateTime"
use="required"/>
                </xsd:complexType>
              </xsd:element>
            </xsd:sequence>
          </xsd:complexType>
```

```
                    </xsd:element>
                </xsd:sequence>
            </xsd:complexType>
        </xsd:element>
        <!--New attribute type-->
        <xsd:simpleType name="domainType">
            <xsd:restriction base="xsd:string">
                <xsd:enumeration value="CN_CS"/>
                <xsd:enumeration value="CN_PS"/>
                <xsd:enumeration value="UTRAN"/>
                <xsd:enumeration value="IMS"/>
                <!--Legacy compatibility -->
                <xsd:enumeration value="EMS"/>
                <xsd:enumeration value="CAMEL"/>
                <xsd:enumeration value="OTHER"/>
            </xsd:restriction>
        </xsd:simpleType>
        <!--PM name extensions-->
        <xsd:simpleType name="measName">
            <xsd:union memberTypes="mc:measNameWithSubCounter mc:measNameWithoutSubCounter"/>
        </xsd:simpleType>
        <xsd:simpleType name="measResultType">
            <xsd:union memberTypes="xsd:decimal">
                <xsd:simpleType>
                    <xsd:restriction base="xsd:string">
                        <xsd:enumeration value="NIL"/>
                    </xsd:restriction>
                </xsd:simpleType>
            </xsd:union>
        </xsd:simpleType>
        <xsd:simpleType name="measNameWithSubCounter">
            <xsd:restriction base="xsd:string">
                <xsd:pattern
value="(rabAssignmentMeasurement.attRabAssignEstabCsPerType.|rabAssignmentMeasurement.succRabAssignE
stabCsPerType.|rabAssignmentMeasurement.failRabAssignEstabCsPerCause.|rabAssignmentMeasurement.attRab
AssignEstabPsPerType.|rabAssignmentMeasurement.succRabAssignEstabPsPerType.|rabAssignmentMeasurement.
failRabAssignEstabPsPerCause.|rabAssignmentMeasurement.attRabAssignModCsPerType.|rabAssignmentMeasure
ment.succRabAssignModCsPerType.|rabAssignmentMeasurement.failRabAssignModCsPerCause.|rabAssignment
Measurement.attRabAssignModPsPerType.|rabAssignmentMeasurement.succRabAssignModPsPerType.|rabAssign
mentMeasurement.failRabAssignModPsPerCause.|rabAssignmentMeasurement.attRabAssignRelCsPerType.|rabAs
signmentMeasurement.succRabAssignRelCsPerType.|rabAssignmentMeasurement.failRabAssignRelCsPerCause.|r
abAssignmentMeasurement.attRabAssignRelPsPerType.|rabAssignmentMeasurement.succRabAssignRelPsPerTyp
e.|rabAssignmentMeasurement.failRabAssignRelPsPerCause.|rabReleaseRequestMeasurement.nbrRncRelCsRabPe
rCause.|rabReleaseRequestMeasurement.nbrRncRelPsRabPerCause.|iuConnectionMeasurement.nbrRncRelCsIuCo
nnPerCause.|iuConnectionMeasurement.nbrRncRelPsIuConnPerCause.|iuConnectionMeasurement.attRelCsIuConn
PerCause.|iuConnectionMeasurement.attRelPsIuConnPerCause.|iuInterfaceMeasurement.nbrResetCsByRncPerCau
se.|iuInterfaceMeasurement.nbrResetPsByRncPerCause.|iuInterfaceMeasurement.nbrResetCsByCnPerCause.|iuInte
rfaceMeasurement.nbrResetPsByCnPerCause.|iuInterfaceMeasurement.nbrResetResCsByRncPerCause.|iuInterface
Measurement.nbrResetResPsByRncPerCause.|iuInterfaceMeasurement.nbrResetResCsByCnPerCause.|iuInterface
Measurement.nbrResetResPsByCnPerCause.|iuInterfaceMeasurement.nbrErrorIndCsByRncPerCause.|iuInterfaceM
easurement.nbrErrorIndPsByRncPerCause.|iuInterfaceMeasurement.nbrErrorIndCsByCnPerCause.|iuInterfaceMeas
urement.nbrErrorIndPsByCnPerCause.|rncSoftHandoverMeasurement.failRlAddInSofterHoPerCause.|rncSoftHand
overMeasurement.failRlDelInSofterHoPerCause.|rncSoftHandoverMeasurement.failRlAddInShoPerCause.|rncSoft
HandoverMeasurement.failRlDelInShoPerCause.|rncHardHandoverMeasurement.failHhoPerCause.|rncRelocation
Measurement.attRelocOutPrepWithUeNotInvCsPerCause.|rncRelocationMeasurement.failRelocOutPrepWithUeNo
tInvCsPerCause.|rncRelocationMeasurement.failRelocOutWithUeNotInvCsPerCause.|rncRelocationMeasurement.a
ttRelocOutPrepWithUeInvCsPerCause.|rncRelocationMeasurement.failRelocOutPrepWithUeInvCsPerCause.|rncRe
locationMeasurement.failRelocOutWithUeInvCsPerCause.|rncRelocationMeasurement.attRelocOutPrepWithUeNot
InvPsPerCause.|rncRelocationMeasurement.failRelocOutPrepWithUeNotInvPsPerCause.|rncRelocationMeasureme
nt.failRelocOutWithUeNotInvPsPerCause.|rncRelocationMeasurement.attRelocOutPrepWithUeInvPsPerCause.|rnc
RelocationMeasurement.failRelocOutPrepWithUeInvPsPerCause.|rncRelocationMeasurement.failRelocOutWithUe
InvPsPerCause.|rncRelocationMeasurement.attRelocInWithUeNotInvCsPerCause.|rncRelocationMeasurement.fail
RelocInWithUeNotInvCsPerCause.|rncRelocationMeasurement.attRelocInWithUeInvCsPerCause.|rncRelocationM
easurement.failRelocInWithUeInvCsPerCause.|rncRelocationMeasurement.attRelocInWithUeNotInvPsPerCause.|r
ncRelocationMeasurement.failRelocInWithUeNotInvPsPerCause.|rncRelocationMeasurement.attRelocInWithUeIn
vPsPerCause.|rncRelocationMeasurement.failRelocInWithUeInvPsPerCause.|rncInterSystemHandoverMeasuremen
t.attRelocOutPrepInterSysCsPerCause.|rncInterSystemHandoverMeasurement.failRelocOutPrepInterSysCsPerCaus
e.|rncInterSystemHandoverMeasurement.failRelocOutInterSysCsPerCause.|rncInterSystemHandoverMeasurement.
attRelocInInterSysCsPerCause.|rncInterSystemHandoverMeasurement.failRelocInInterSysCsPerCause.|rncInterSyst
emHandoverMeasurement.failRelocOutInterSysPsPerCause.|iuInterfaceThroughputMeasurement.iuUlDataThrough
putCsPerType.|iuInterfaceThroughputMeasurement.iuDlDataThroughputCsPerType.|iuInterfaceThroughputMeasur
ement.iuUlDataThroughputPsPerType.|iuInterfaceThroughputMeasurement.iuDlDataThroughputPsPerType.|rlcCon
nectionMeasurement.nbrRlcBlockSentPerMode.|rlcConnectionMeasurement.nbrRlcBlockRecvedPerMode.|cellRrc
ConnectionMeasurement.attRrcConnSetupPerCause.|cellRrcConnectionMeasurement.succRrcConnSetupPerCause.|
cellRrcConnectionMeasurement.failRrcConnSetupPerCause.|cellRrcConnectionMeasurement.failRrcConnReestabP
erCause.|cellSoftHandoverMeasurement.failRlAddInShoPerCause.|cellSoftHandoverMeasurement.failRlDelInShoP
erCause.|hardHandoverIntraCellMeasurement.failHhoOutIntraCellPerCause.|iubRlManagementMeasurement.failRl
SetupIubPerCause.|iubRlManagementMeasurement.failRlAddIubPerCause.|iurRlManagementMeasurement.failRlS
etupIurPerCause.|iurRlManagementMeasurement.failRlAddIurPerCause.|cellHsSetupMeas.attHsRabAssignEstabPe
rType|cellHsSetupMeas.succHsRabAssignEstabPerType|cellHsSetupMeas.nbrHsCnRel|hardHandoverInterCellIntra
NodeBMeasurement.failHhoOutInterCellIntraNodeBPerCause.|hardHandoverInterNodeBIntraRncMeasurement.fail
HhoOutInterNodeBIntraRncPerCause.|hardHandoverInterRncViaIurMeasurement.failHhoOutInterRncViaIurPerCa
use.|hardHandoverInterRncMeasurement.failHhoOutInterRncCnPerCause.|hardHandoverInterSystemMeasurement.
attRelocOutPrepInterSysCsPerCause.|hardHandoverInterSystemMeasurement.failRelocOutPrepInterSysCsPerCaus
e.|hardHandoverInterSystemMeasurement.failRelocOutInterSysCsPerCause.|hardHandoverInterSystemMeasureme
nt.attRelocInInterSysCsPerCause.|hardHandoverInterSystemMeasurement.failRelocInInterSysCsPerCause.|hardHa
ndoverInterSystemMeasurement.failRelocOutInterSysPsPerCause.)\d{1,5}"/>
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name="measNameWithoutSubCounter">
```

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

```
                <xsd:enumeration value="cellTrafficMeasurement.cellDtchTraffic"/>
                <xsd:enumeration value="cellPagingMeasurement.attPagingType1FromUtran"/>
                <xsd:enumeration value="cellPagingMeasurement.succPagingType1FromUtran"/>
                <xsd:enumeration value="cellPagingMeasurement.attPagingType2FromUtran"/>
                <xsd:enumeration
value="hardHandoverInterCellIntraNodeBMeasurement.attHhoOutInterCellIntraNodeB"/>
                <xsd:enumeration
value="hardHandoverInterNodeBIntraRncMeasurement.attHhoOutInterNodeBIntraRnc"/>
                <xsd:enumeration value="hardHandoverInterRncViaIurMeasurement.attHhoOutInterRncViaIur"/>
                <xsd:enumeration value="hardHandoverInterRncMeasurement.attHhoOutInterRncCn"/>
                <xsd:enumeration value="hardHandoverInterSystemMeasurement.attRelocOutInterSysCs"/>
                <xsd:enumeration value="hardHandoverInterSystemMeasurement.attRelocOutInterSysPs"/>
                <xsd:enumeration value="hardHandoverInterSystemMeasurement.attRelocInInterSysPs"/>
                <xsd:enumeration value="hardHandoverInterSystemMeasurement.succRelocInInterSysPs"/>
                <xsd:enumeration value="cellHsSetupMeas.nbrhsRncRelbyUserInact"/>
                <xsd:enumeration value="cellHsSetupMeas.attMacdFlowSetup"/>
                <xsd:enumeration value="cellHsSetupMeas.succMacdFlowSetup"/>
                <xsd:enumeration value="cellHsSetupMeas.attHsRbSetup"/>
                <xsd:enumeration value="cellHsSetupMeas.succHsRbSetup"/>
                <xsd:enumeration value="cellHsChSwMeas.attFachToHsDsch"/>
                <xsd:enumeration value="cellHsChSwMeas.attDchToHsDsch"/>
                <xsd:enumeration value="cellHsChSwMeas.succFachToHsDsch"/>
                <xsd:enumeration value="cellHsChSwMeas.succDchToHsdsch"/>
                <xsd:enumeration value="cellHsChSwMeas.attHsDschToFach"/>
                <xsd:enumeration value="cellHsChSwMeas.attHsDschToDch"/>
                <xsd:enumeration value="cellHsChSwMeas.succHsDschToFach"/>
                <xsd:enumeration value="cellHsChSwMeas.succHsDschToDch"/>
                <xsd:enumeration value="cellHsRepointMeas.attServCellUpd"/>
                <xsd:enumeration value="cellHsRepointMeas.succServCellUpd"/>
                <xsd:enumeration value="cellHsTraffMeas.nbrMacHsPduTx"/>
                <xsd:enumeration value="cellHsTraffMeas.nbrMacHsAckedPduTx"/>
                <xsd:enumeration value="cellHsTraffMeas.nbrOctMacHsAckedPduTx"/>
                <xsd:enumeration value="cellHsTraffMeas.nbrNonEptBuffTti"/>
                <xsd:enumeration value="cellHsTraffMeas.nbrSubsNonEptBuffPerTti"/>
                <xsd:enumeration value="cellHsTraffMeas.nbrMeanSubs"/>
                <xsd:enumeration value="cellHsResMeas.nbrMeanUsedHspdschCode"/>
                <xsd:enumeration value="cellHsResMeas.nbrMeanUsedHsscchCode"/>
                <xsd:enumeration value="cellHsResMeas.cellNonHsMeanTxPower"/>
                <xsd:enumeration value="cellHsResMeas.cellNonHsMaxTxPower"/>
                <xsd:enumeration value="cellHsResMeas.cellMeanTxPower"/>
                <xsd:enumeration value="cellHsResMeas.cellMaxTxPower"/>
                <!--HSDPA mesurements on UtranCell beginning with family name "cellhsXXX"-->
            </xsd:restriction>
        </xsd:simpleType>
</xsd: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="HsdpaMeasCollec.xsl"?>
<measCollecFile xmlns="std://yd-t/hsdpa/itf-n/mc/2007">
```

附　录　A
（规范性附录）
XML Schema 文档补充说明

A.1　XML Schema文档结构标记

XML Schema文档结构标记约定如图A.1所示。

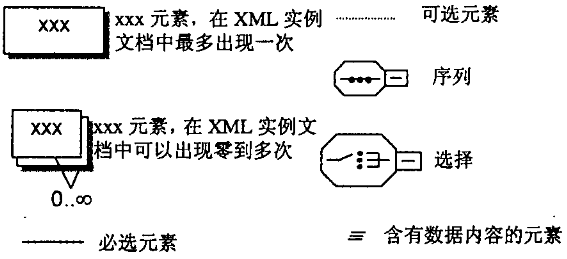
xxx
xxx 元素，在 XML 实例文档中最多出现一次
可选元素
序列
xxx
xxx 元素，在 XML 实例文档中可以出现零到多次
0..∞
选择
必选元素
含有数据内容的元素

图 A.1　XML Schema 文档结构标记

A.2　性能测量数据文件的Schema定义

XML Schema文档<HsdpaMeasCollec.xsd>结构如图A.2所示。

XML Schema文档元素/属性说明见表A.1。

表 A.1　HsdpaMeasCollec.xsd 文档元素/属性说明

元素/属性名称		元素/属性描述
元素名称	包含属性	
measCollecFile		性能数据采集文件，是该Schema的根元素，由文件头部（fileHeader）、采集数据（measData）以及文件尾部（FileFooter）3个子元素组成
fileHeader	fileFormatVersion	文件头部，由文件发送方（fileSender）、测量采集开始时间（measCollec）两个子元素组成。包含文件格式版本（fileFormatVersion）、制造商名称（vendorName）和识别名前缀（dnPrefix）3个属性
	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，由管理网元（managedElement）及其性能采集结果（measInfo））两个子元素组成
fileFooter		文件尾部，包含测量采集结束时间（measCollec）子元素
fileSender	localDn	文件发送发，包含本地识别名（localDN）、网元类型（elementType））两个属性
	elementType	
managedElement	localDn	被管网元，包括本地识别名（localDn）、用户友好名（userLabel）、软件版本（swVersion）3个属性
	userLabel	
	swVersion	
measInfo	measInfoId	测量信息，由测量任务（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）4个子元素组成，包含一个可选属性测量信息标识符（measInfoId）
job	jobId	测量任务，该元素为可选元素由其属性jobId唯一标识
granPeriod	duration	测量粒度周期，包含持续时间（duration）、结束时间（endTime）两个属性
	endTime	
repPeriod	duration	测量上报周期，该元素为可选元素，包含唯一属性持续时间（duration）

表A.1（续）

元素/属性名称		元素/属性描述
元素名称	元素名称	
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（position），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）两个子元素组成。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，分为包含子测量项（measNameWithSubCounter）和不含子测量项（measNameWithoutSubCounter）两类，从3GPP TS 32.435中扩展而来
measNameWithSubCounter		含子测量项的数据测量项名称，表示为familyname.measurename.subcounter形式，从3GPP TS 32.435中扩展而来
measNameWithoutSubCounter		不含子测量项的数据测量项名称，表示为familyname.measurename形式，从3GPP TS 32.435中扩展而来

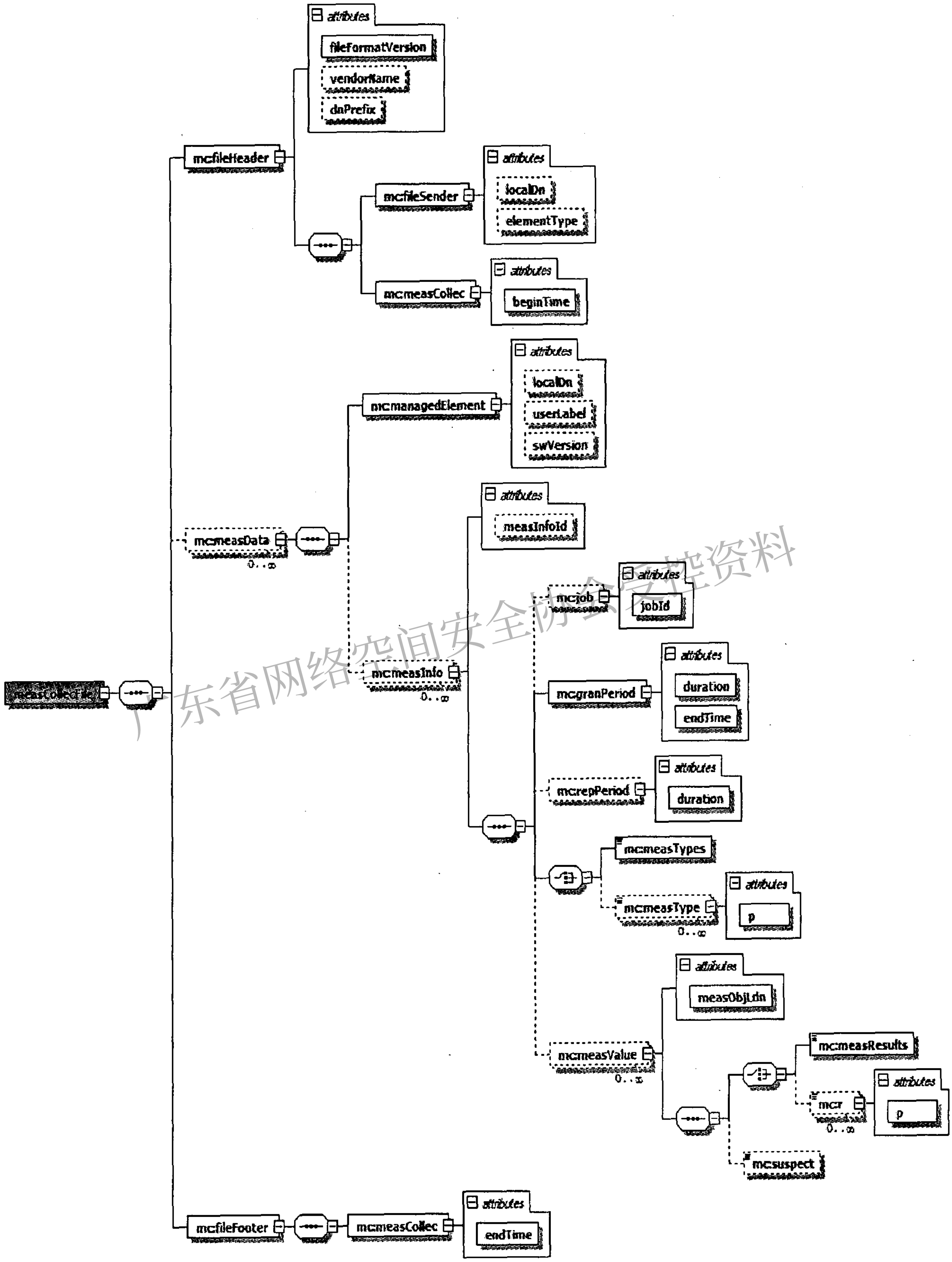

图 A.2　HsdpaMeasCollec.xsd 文档结构

附 录 B
(资料性附录)
性能管理功能相关 XML 文件示例

B.1　性能测量数据XML文件示例一：不使用可选的p属性

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="HsdpaMeasCollec.xsl"?>
<!-- XML schema based XML measurement report file without use of optional positioning attributes on
measurement types and results. All values are hypothetical but syntactically correct -->
<measCollecFile xmlns="std://yd-t/hsdpa/itf-n/mc/2007"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="std://yd-t/hsdpa/itf-n/mc/2007
HsdpaMeasCollec.xsd">
    <fileHeader fileFormatVersion="1.0" vendorName="Company NN"
dnPrefix="DC=a1.companyNN.com,SubNetwork=1,IRPAgent=1">
        <fileSender
localDn="SubNetwork=CountryNN,MeContext=MEC-Gbg-1,ManagedElement=BSC-Gbg-1"
elementType="BSC" inDomain="UTRAN"/>
        <measCollec beginTime="2007-03-01T14:00:00+02:00"/>
    </fileHeader>
    <measData>
        <managedElement
localDn="SubNetwork=CountryNN,MeContext=MEC-Gbg-1,ManagedElement=BSC-Gbg-1" userLabel="BSC
Telecomville"/>
        <measInfo>
            <job jobId="1231"/>
            <granPeriod duration="PT900S" endTime="2000-03-01T14:14:30+02:00"/>
            <repPeriod duration="PT1800S"/>
            <measTypes>  mobileTrafficFlow.failOrigCallsPerCause.0
mobileTrafficFlow.failTermIncCallsPerCause.1   mobileTrafficFlow.failTermIncCallsPerCause.2
mobileTrafficFlow.failTermIncCallsPerCause.3 </measTypes>
            <measValue measObjLdn="BscFunction=RF-1,Sector=Gbg-997">
                <measResults>234 345 567 789</measResults>
            </measValue>
            <measValue measObjLdn="BscFunction=RF-1,Sector=Gbg-998">
                <measResults>890 901 123 234</measResults>
            </measValue>
            <measValue measObjLdn="BscFunction=RF-1,Sector=Gbg-999">
                <measResults>456 567 678 789</measResults>
                <suspect>true</suspect>
            </measValue>
        </measInfo>
    </measData>
    <fileFooter>
```

```
          <measCollec endTime="2007-03-01T14:15:00+02:00"/>
     </fileFooter>
</measCollecFile>
```

B.2 性能测量数据XML文件示例二：使用可选的p属性

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="HsdpaMeasCollec.xsl"?>
<!-- XML schema based XML measurement report file with use of optional positioning attributes on measurement
types and results. All values are hypothetical but syntactically correct. -->
<measCollecFile xmlns="std://yd-t/hsdpa/itf-n/mc/2007"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="std://yd-t/hsdpa/itf-n/mc/2007
HsdpaMeasCollec.xsd">
     <fileHeader fileFormatVersion="1.0" vendorName="Company NN"
dnPrefix="DC=a1.companyNN.com,SubNetwork=1,IRPAgent=1">
          <fileSender
localDn="SubNetwork=CountryNN,MeContext=MEC-Gbg-1,ManagedElement=BSC-Gbg-1"
elementType="BSC" inDomain="UTRAN"/>
          <measCollec beginTime="2007-03-01T14:00:00+02:00"/>
     </fileHeader>
     <measData>
          <managedElement
localDn="SubNetwork=CountryNN,MeContext=MEC-Gbg-1,ManagedElement=BSC-Gbg-1" userLabel="BSC
Telecomville"/>
          <measInfo>
               <job jobId="1231"/>
               <granPeriod duration="PT900S" endTime="2000-03-01T14:14:30+02:00"/>
               <repPeriod duration="PT1800S"/>
               <measType p="1">mobileTrafficFlow.failOrigCallsPerCause.0</measType>
               <measType p="2">mobileTrafficFlow.failOrigCallsPerCause.1</measType>
               <measType p="3">mobileTrafficFlow.failOrigCallsPerCause.2</measType>
               <measType p="4">mobileTrafficFlow.failTermIncCallsPerCause.3</measType>
               <measValue measObjLdn="BscFunction=RF-1,Sector=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="BscFunction=RF-1,Sector=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="BscFunction=RF-1,Sector=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="2007-03-01T14:15:00+02:00"/>
     </fileFooter>
</measCollecFile>
```

B.3 性能测量数据XML文件示例三：使用可选的measInfoId属性

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="HsdpaMeasCollec.xsl"?>
<!-- XML schema based XML measurement report file with use of optional measInfoId attribute. All values are
hypothetical but syntactically correct. -->
<measCollecFile xmlns="std://yd-t/hsdpa/itf-n/mc/2007"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="std://yd-t/hsdpa/itf-n/mc/2007
HsdpaMeasCollec.xsd">
     <fileHeader fileFormatVersion="1.0" vendorName="Company NN" dnPrefix="SubNetwork=1">
          <fileSender localDn="OMC_PS=10" elementType="Element Manager"/>
          <measCollec beginTime="2007-03-01T14:00:00+02:00"/>
     </fileHeader>
     <measData>
          <managedElement localDn="ManagedElement=PS_Core" userLabel="SGSN" swVersion="R30.1.5"
inDomain="UTRAN"/>
          <measInfo measInfoId="Category A">
               <job jobId="01"/>
               <granPeriod duration="PT900S" endTime="2007-03-01T14:15:00+02:00"/>
               <repPeriod duration="PT1800S"/>
               <measTypes>MM.AttGprsAttach MM.SuccGprsAttach MM. AbortedGprsAttach
MM.AttIntraSgsnRaUpdate</measTypes>
               <measValue measObjLdn="SgsnFunction=1">
                    <measResults>10 20 30 40</measResults>
               </measValue>
          </measInfo>
          <measInfo measInfoId="Category B">
               <job jobId="02"/>
               <granPeriod duration="PT900S" endTime="2007-03-01T14:15:00+02:00"/>
               <repPeriod duration="PT1800S"/>
               <measTypes>MM.AttCombiAttach MM.SuccCombiAttach MM.
MM.AbortedCombiAttachMM.AttCombiDetachMs</measTypes>
```

```
            <measValue measObjLdn="SgsnFunction=2">
                <measResults>10 20 30 40</measResults>
            </measValue>
        </measInfo>
        <measInfo measInfoId="Category C">
            <job jobId="03"/>
            <granPeriod duration="PT1800S" endTime="2007-03-01T14:15:00+02:00"/>
            <repPeriod duration="PT900S"/>
            <measTypes>MM.AttPsPagingProcIu MM.SuccPsPagingProcIu</measTypes>
            <measValue measObjLdn="SgsnFunction=3">
                <measResults>25 25</measResults>
            </measValue>
        </measInfo>
    </measData>
    <fileFooter>
        <measCollec endTime="2007-03-01T14:15:00+02:00"/>
    </fileFooter>
</measCollecFile>
```

YD/T 1863.3-2009

中华人民共和国
通信行业标准
2GHz WCDMA 数字蜂窝移动通信网
高速下行分组接入（HSDPA）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
YD/T 1863.3-2009

*

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

*

*

本书如有印装质量问题，请与本社联系 电话：(010)67114922