



3RD GENERATION
PARTNERSHIP
PROJECT 2
"3GPP2"

TSG CORRESPONDENCE

Mr. Norikazu Yamasaki
Chair, 3GPP2 TSG-S
KDDI Corporation
Garden Air Tower
3-10-10 Iidabashi
Chiyoda-ku, Tokyo 102-8460, Japan
tsgs_chair@3gpp2.org

27 October 2005

Mr. Christian Toche
Chair, 3GPP TSG-SA WG5
NORTEL
Parc d'Activités de Magny-Châteaufort
Les Jeunes Bois – MS CTF 12B1
Châteaufort
78928 Yvelines Cedex 9, France
toche@nortel.com

RE: Comments & questions regarding 3GPP SA5 R6 IRP definitions

Dear Mr. Toche,

As you are aware, 3GPP2 TSG-S WG5 is progressing on developing 3GPP2 S.S0028 Revision C, incorporating via reference and delta definitions various 3GPP R6 IRP specifications. During this process, 3GPP2 TSG-S WG5 came across a number of aspects of concerns, which are listed below.

1. NRM Comments on IOC InventoryUnit re VsDataContainer

Hereby acknowledging the response provided in your liaison from July 01 agreeing with 3GPP2's suggestion and looking forward to see the creation of appropriate CRs to address this in R6.

3GPP2:

1. NRM Question comment on VsDataContainer contained by InventoryUnit

In 3GPP SA5 R6 TS 32.692 specification, InventoryUnit IOCs may not contain VsDataContainer IOCs. However, it seems reasonable that inventory data should be able to contain vendor-specific data (such as, Operating System, replacement information, etc.).

3GPP2 TSG-S WG5 strongly recommends that 3GPP SA5 update the 3GPP R6 TS 32.692 specification so that VsDataContainer IOCs may be contained underneath InventoryUnit IOCs.

SA5 reply:

SA5 agrees with the suggestion and will look to creating appropriate CRs to address this in R6.

2. *NRM Comments on IOC InventoryUnit attributes type and version*

TS 32.962 Table 2 defines vendorUnitTypeNumber carrying unit type information as well as version information.

vendorUnitTypeNumber	A vendor/manufacture defined and assigned number which uniquely identifies the unit type and version (used for replacing HW units, spares).	String
----------------------	---	--------

Since a) version information can change without changes in unit type information and b) there is no standard way to indicate the separation of the two information, we suggest:

- Add an attribute versionNumber attribute in InventoryUnit and
- Remove “version number” from the definition of InventoryUnit.vendorUnitTypeNumber (so that vendorUnitTypeNumber holds only unit type) such as.

vendorUnitTypeNumber	A vendor/manufacture defined and assigned number which uniquely identifies the unit type (used for replacing HW units, spares).	String
versionNumber	The version information related to vendorUnitTypeNumber.	String

3. *NRM Question on meaning of attribute “relatedObjects” in MtpSignPoint*

Hereby acknowledging the response provided in your liaison from July 01 agreeing with 3GPP2’s comments and looking forward to see the creation of appropriate CRs to address this in R6.

3GPP2:

3. NRM Question on meaning of attribute “relatedObjects” in MtpSignPoint

3GPP2 TSG-S WG5 would like 3GPP SA5 to further explain the definition of the 3GPP R6 TS 32.742 MtpSignPoint IOC relatedObjects attribute. The current definition is as follows: “This attribute indicates list of related xxxFunction object”. As an example, related in what ways? Which specific types of IOCs may be contained in this attribute?

3GPP2 TSG-S WG5 recommends that an Information Relationship Definition be defined.

SA5 reply:

SA5 can appreciate the current version of TS 32.742 specification may not provide by itself a clear understanding and that information may be missing. SA5 will look to clarifying with appropriate CRs. (In advance and for some background, examples of “xxxFunction” are SsgnFunction, RncFunction, etc, i.e. those that may need be associated with SS7 signalling).

4. *Proposed changes to CORBA SS Style Guide*

In 3GPP2 TSG-S WG5, for the CORBA Solution Sets, we define both the attribute constants (as is done in 3GPP SA5 R6), but also define the attribute types for each of the attributes. We feel that this provides less ambiguity, type checks of the mapping tables and provides a place to discover all of the attribute types. As an example of the format, the GenericNetworkResourcesNRMDefs.idl CORBA IDL file defined in 3GPP TS 32.623 V6.50 would look like the following when adding attribute type typedefs (with updates in yellow):

```

//File: GenericNetworkResourcesNRMDefs.idl
#ifndef _GENERIC_NETWORK_RESOURCES_NRM_DEFS_IDL_
#define _GENERIC_NETWORK_RESOURCES_NRM_DEFS_IDL_

#include <GenericNetworkResourcesIRPSystem.idl>

// This statement must appear after all include statements
#pragma prefix "3gppsa5.org"
/**
 * This module defines constants for each MO class name and
 * the attribute names for each defined MO class.
 */
module GenericNetworkResourcesNRMDefs
{
    /**
     * Definitions for MO class Top
     */
    interface Top
    {
        // Attribute Names
        //
        const string CLASS = "Top";
    };

    typedef string CLASSType;

    /**
     * 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 userDefinedNetworkType = "userDefinedNetworkType";
        const string setOfMcc = "setOfMcc";
    };

    typedef string SubNetworkIdType;
    typedef string DnPrefixType;
    typedef string UserLabelType;
    typedef string UserDefinedNetworkTypeType;
    typedef GenericNetworkResourcesIRPSystem::AttributeTypes::StringSet
        SetOfMccType;
    // CLASS

    /**
     * Definitions for MO class ManagedElement
     */
    interface ManagedElement : Top
    {
        const string CLASS = "ManagedElement";
        // Attribute Names
        //
        const string managedElementId = "managedElementId";
        const string dnPrefix = "dnPrefix";
        const string managedElementType = "managedElementType";
        const string userLabel = "userLabel";
        const string vendorName = "vendorName";
        const string userDefinedState = "userDefinedState";
        const string locationName = "locationName";
        const string managedBy = "managedBy";
        const string swVersion = "swVersion";
    };

    typedef string ManagedElementIdType;
    // dnPrefix
    // userLabel

```

```

typedef string LocationNameType;
typedef string VendorNameType;
typedef string UserDefinedStateType;
typedef GenericNetworkResourcesIRPSystem::AttributeTypes::StringSet
    ManagedElementType; // An error in 32.623 Mapping Tables
typedef GenericNetworkResourcesIRPSystem::AttributeTypes::MOReferenceSet
    ManagedByType; // An error in 32.623 Mapping Tables
typedef string SwVersionType;
    // CLASS

/**
 * Definitions for MO class MeContext
 */
interface MeContext : Top
{
    const string CLASS = "MeContext";
    // Attribute Names
    //
    const string meContextId = "meContextId";
    const string dnPrefix = "dnPrefix";
};

typedef string MeContextIdType;
    // dnPrefix
    // CLASS

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

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

typedef string ManagementNodeIdType;
    // userLabel
    // locationName
    // vendorName
    // userDefinedState
typedef GenericNetworkResourcesIRPSystem::AttributeTypes::MOReferenceSet
    ManagedElementsType; // An error in 32.623 Mapping Tables
    // swVersion
    // CLASS

/**
 * Definitions for abstract MO class ManagedFunction
 */
interface ManagedFunction : Top
{
    const string CLASS = "ManagedFunction";
    // Attribute Names
    //
    const string userLabel = "userLabel";
};

    // CLASS
    // userLabel

/**
 * Definitions for MO class IRPAgent

```

```

*/
interface IRPAgent : Top
{
    const string CLASS = "IRPAgent";
    // Attribute Names
    //
    const string irpAgentId = "irpAgentId";
    const string systemDN = "systemDN";
};

```

```

typedef string IrpAgentIdType;
typedef string SystemDNType;
// CLASS

```

```

/**
 * Definitions for abstract MO class Link
 * This inherits from ManagedFunction
 * The attributes aEnd and zEnd are populated with the DNs
 * of the entities associated via the link class.
 * The aEnd takes the DN of the 1st entity in alphabetical order,
 * the zEnd takes the 2nd entity in alphabetical order of the class
 * names.
 */

```

```

interface Link : ManagedFunction
{
    const string CLASS = "Link";
    // Attribute Names
    //
    const string linkId = "linkId";
    const string aEnd = "aEnd";
    const string zEnd = "zEnd";
    const string linkType = "linkType";
    const string protocolName = "protocolName";
    const string protocolVersion = "protocolVersion";
};

```

```

enum LinkType
{
    SIGNALLING,
    BEARER,
    OAM_AND_P,
    OTHER
};

```

```

typedef string LinkIdType;
typedef GenericNetworkResourcesIRPSystem::AttributeTypes::MOReference AEndType;
typedef GenericNetworkResourcesIRPSystem::AttributeTypes::MOReference ZEndType;
typedef sequence <LinkType> LinkTypeType;
typedef string ProtocolNameType;
typedef string ProtocolVersionType;
// CLASS
// 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";
};

```

```

typedef string VsDataContainerIdType;
typedef string VsDataTypeType;
typedef any VsDataType;
typedef string VsDataFormatVersionType;
// CLASS

```

```
};
#endif // _GENERIC_NETWORK_RESOURCES_NRM_DEFS_IDL_
```

An extra benefit of this is that you discover errors in the SS mapping tables. As an example, the CORBA Solution Set mapping table for ManagementNode needs to be:

NRM Attributes/association roles of IOC ManagementNode in 3GPP TS 32.622 [4]	SS Attributes	SS Type	Qualifier
managementNodeId	managementNodeId	String	Read-Only, M
userLabel	userLabel	String	Read-Write, M
locationName	locationName	String	Read-Only, M
vendorName	vendorName	String	Read-Only, M
userDefinedState	userDefinedState	String	Read-Write, M
managedElements	managedElements	GenericNetworkResources! RPSystem::AttributeTypes::MOReferenceSet	Read-Only, M
swVersion	swVersion	String	Read-Only, M

and the CORBA Solution Set mapping table for ManagedElement needs to be:

NRM Attributes/Association roles	SS Attributes	SS Type	Qualifier
managedElementId	managedElementId	String	Read-Only, M
dnPrefix	dnPrefix	String	Read-Only, M
userLabel	userLabel	String	Read-Write, M
locationName	locationName	String	Read-Only, M
vendorName	vendorName	String	Read-Only, M
userDefinedState	userDefinedState	String	Read-Write, M
managedElementType	managedElementType	GenericNetworkResources! RPSystem::AttributeTypes::StringSet	Read-Only, M
managedBy	managedBy	GenericNetworkResources! RPSystem::AttributeTypes::MOReferenceSet	Read-Only, M
swVersion	swVersion	String	Read-Only, M

3GPP2 TSG-S WG5 proposes that 3GPP SA5 update the following:

- 3GPP SA5 R6 TS 32.623 with the above updates.
- CORBA IDL updates for all of the 3GPP SA5 R6 <>NRMDefs.idl CORBA IDL files.
- 3GPP SA5 R6 TS 32.150, with the following updates:

D.4 NRM IRP

Use one module to define the IDL constructs for the managed object classes and the attribute types. The name of this module is XxxNRIRPConstDefs where Xxx is the name of the subject NRM IRP.

An example is UtranNRIRPConstDefs.

Within the module, define a set of IDL interfaces each of which corresponds to a managed object class specified. The interface definition respects the inheritance relation specified. An example of managed object class RncFunction, which inherits from GenericNRIRPConstDefs::ManagedFunction, is shown below.

Within the module, define a set of CORBA IDL typedefs each of which corresponds to attributes contained in the managed object class. The typedef definitions respect the inheritance relation specified. Each attribute type is generated by capitalizing the attribute name (if not already capitalized) and appending "Type" to the end of the attribute name. Each attribute type may only appear once in the file, if it has already appeared, a comment containing the attribute name will be used in its place. An example of managed object class RncFunction, which inherits from GenericNRIRPConstDefs::ManagedFunction, is shown below (assuming the CLASS attribute was already defined in the file).

```
...
#include <GenericNetworkResourcesNRMDefs.idl>
...
module UtranNRIRPConstDefs
{
...

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

    // Attribute Names
    //
    const string rncFunctionId = "rncFunctionId";

    const string mcc= "mcc";
    const string mnc= "mnc";
    const string rncId= "rncId";
};

typedef string RncFunctionIdType;
typedef long MccType;
typedef long MncType;
typedef long RncIdType;
typedef GenericNetworkResourcesNRMDefs::UserLabelType UserLabelType;
    // CLASS
...
};
```

5. Use Of Point Codes in Signalling Transport Network

In the Signalling Transport Network IRP (3GPP SA5 R6 TS 32.742), point codes are used to identify MtpSignPoint IOCs. This mimics what is used in the ITU-T Q.751.1 specification. However, using point codes as a reference to other IOCs is very inefficient. To match up an adjacent point code requires an extensive search for the correct IOC with the matching point code.

Even more inefficient is trying to match Signalling Link Codes. The SignLinkTp Signalling Link Code value matches the SignLinkTp Signalling Link Code in the SignLinkTp IOC contained under the MtpSignPoint IOC (with the matching point code of the parent SignLinkTp adjacent point code) contained

under the SignLinkSetTp IOC (with the matching adjacent point code of the parent SignLinkSetTp of the parent MtpSignPoint) contained under a ManagedElement IOC.

It also does not meet the current methodology used by 3GPP SA5 to model object relationships. Instead, the Signalling Transport Network IRP should use what the rest of the 3GPP SA5 specifications use to model relationships between objects, the Distinguished Name.

3GPP2 TSG-S WG5 recommends that the following attributes be added to the R6 Signalling Transport Network IRP to ease the traversing of these objects:

- adjacentMtpSignPoint (SignLinkSetTp) – This corresponds to the MtpSignPoint IOC referenced by the adjacent point code.
- destinationMtpSignPoint (SignRouteSetNePart) – This corresponds to the MtpSignPoint IOC referenced by the destination point code.
- adjacentSignLinkTp (SignLinkTp) – This corresponds to the SignLinkTp IOC with the same Signalling Link Code under the corresponding adjacentMtpSignPoint in its parent SignLinkSetTp IOC.

6. VsDataContainer use in different NRMs

Currently, 3GPP SA5 R6 TS 32.622 contains the following statement regarding the VsDataContainer IOC:

The 'VsDataContainer' managed object is a container for vendor specific data. The number of instances of the 'VsDataContainer' can differ from vendor to vendor. This IOC shall only be used by the Bulk CM IRP for the UTRAN, GERAN and CN NRMs.

However, the VsDataContainer is also used in 3GPP SA5 R6 TS 32.742 and 3GPP SA5 R6 TS 32.712. These NRMs may be used outside of the UTRAN, GERAN and Core Network scope. The VsDataContainer IOC is also used in 3GPP2 TSG-S WG5 NRM specifications.

3GPP2 TSG-S WG5 recommends that 3GPP SA5 resolves these contradicting definitions.

7. InventoryUnit use of String attribute types versus Date attribute types

Currently, 3GPP SA5 R6 TS 32.692 defines the types for the dateOfManufacture and dateOfLastService attributes as string. 3GPP2 TSG-S WG5 feels this is appropriate, because with Inventory Units this information may not be known as a true date, but as a partial date. As an example, a Date Of Manufacture value of “1999” may all that is known of a particular part.

However, 3GPP SA5 R6 TS 32.695 defines the types for the dateOfManufacture and dateOfLastService attributes as date. 3GPP2 TSG-S WG5 feels that this changes the meaning of the two attributes.

3GPP2 TSG-S WG5 recommends that the type for the dateOfManufacture and dateOfLastService attributes as defined in 3GPP SA5 R6 TS 32.695 be changed from date to string.

8. Legal size for attribute issue

TS 3GPP 32.436 specifies minimum and maximum sizes for valid attributes, while the corresponding attribute definitions in TS 3GPP 32.432 and in TS 3GPP 32.435 do not have sizes defined (noting that this is only one example).

In addition, the Naming Convention for Managed Objects (TS 3GPP 32.300) specifies 400 characters as the maximum size. A legal name expressed in CORBA string may require more than 400 characters when expressed in XML document (a particular solution may require more than 400 characters due to its particular encoding).

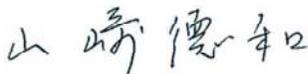
We want to investigate if “attribute legal size” should be an IS defined concept. In principle, we think not since the size of an attribute, as encoded on the wire, is related to technology chosen. However, without a common attribute size definition, each SS may define its own legal sizes that will result in difficulty when NMS needs to deal with IRPAgents of various Solution Sets.

9. *Comment on Top/CLASS definitions*

3GPP2 TSG-S WG5 would like 3GPP SA5 to clarify if a) Basic CM IRP getMO() response include information about CLASS attribute and b) Bulk CM IRP download upload can include information about CLASS attribute (noting that 3GPP TSG-S WG5 is of the impression that 3GPP TS 32.623 section 5.2.9 may not be correct, as may leading to issues such as since every object is either directly or indirectly inherited from Top, does every IOC contains an attribute called “CLASS”? - this does not seem to be followed in the XSD files).

3GPP2 TSG-S would appreciate consideration of these aspects of concern and welcome further discussions. If you have additional questions, please contact: Randy Scheer (rjscheer@lucent.com).

Regards,



Norikazu Yamasaki
Chair, 3GPP2 TSG-S

cc: Y.K. Kim
Henry Cuschieri
Stephen Hayes

Chair, 3GPP2 SC
3GPP2 Secretariat
Chair, 3GPP TSG-SA

ykkim@lgtel.co.kr
hcuschieri@tiaonline.org
stephen.hayes@ericsson.com