TSGS#28(05)0299

Source: SA5 (Telecom Management)

Title: CR 32662-3 Kernel CM IRP Information Service / CORBA SS

Document for: Approval

Agenda Item: 7.5.3

Doc-1st-Level	Spec__#_	CR__#	R	Phase	Subject	Cat	Ver-Cur	Doc-2nd-Level	Workitem
SP-050299	32.662	0007	-	Rel-6	Apply Generic System Context	F	6.3.0	S5-056366	OAM-NIM
SP-050299	32.663	0013	-	Rel-6	Correct CORBA SS mapping of notification filterable/non-filterable IS parameters	F	6.4.0	S5-056367	OAM-NIM

CR-Form-v7

CHANGE REQUEST

⌘ 32.662 CR 0007 ⌘rev - ⌘ Current version: 6.3.0 ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network X Core Network X

Title:	⌘	Apply Generic System Context
Source:	⌘	**SA5** (yaojing@huawei.com, Huawei Technologies Co., Ltd.)
Work item code:	⌘	OAM-NIM
Date:	⌘	10/5/2005
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
- ***F*** *(correction)*
- ***A*** *(corresponds to a correction in an earlier release)*
- ***B*** *(addition of feature),*
- ***C*** *(functional modification of feature)*
- ***D*** *(editorial modification)*

Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
Rel-4	*(Release 4)*
Rel-5	*(Release 5)*
Rel-6	*(Release 6)*

Reason for change:	⌘	Today we have redundant, time-consuming and error prone duplication of the same text for the System Context in all Interface IRPs.
Summary of change:	⌘	Align the title of subclause 4.1 with other Interface IRPs and modify the text of 4.1 with a generic text, referring to the new common definition in 32.150 for the System Context for all Interface IRPs, but keep the diagrams for readability.
Consequences if not approved:	⌘	Redundant, time-consuming and error prone duplication of the same text for the System Context in all Interface IRPs.

Clauses affected:	⌘	2, 4, 5.

		Y	N		
Other specs affected:	⌘		X	Other core specifications	⌘
			X	Test specifications	
			X	O&M Specifications	

Other comments:	⌘	

Change in Clause 2

2 References

The following documents contain provisions, which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

...

[18] 3GPP TS 32.642: "Telecommunication management; Configuration Management (CM); UTRAN network resources Integration Reference Point (IRP): Network Resource Model (NRM)".

[19] 3GPP TS 32.150: "Telecommunication management; Integration Reference Point (IRP) Concept and definitions".

End of Change in Clause 2

Change in Clause 4

4 System Ooverview

4.1 System Ccontext

The general definition of the System Context for the present IRP is found in 3GPP TS 32.150 [19] subclause 4.7.

In addition, the set of related IRP(s) relevant to the present IRP is shown in the two diagrams below.~~Figures 4.1 and 4.2 identify system contexts of the IRP defined by the present specification in terms of its implementation called IRPAgent and the user of the IRPAgent, called IRPManager. For a definition of IRPManager and IRPAgent, see TS 32.102 [2].~~

~~The IRPAgent implements and supports this IRP. The IRPAgent can reside in an Element Manager (EM) or a Network Element (NE) (see also [2] clause 8). In the former case, the interfaces (represented by a thick dotted line) between the EM and the NEs are not the subject of this IRP.~~

~~An NE can be managed via System Context A or B. The criterion for choosing System Context A or B, to manage a particular NE, is implementation dependent. An IRPAgent shall support one of the two System Contexts. By observing the interaction across the Itf-N, an IRPManager cannot deduce if the EM and NE are integrated in a single system or if they run in separate systems.~~

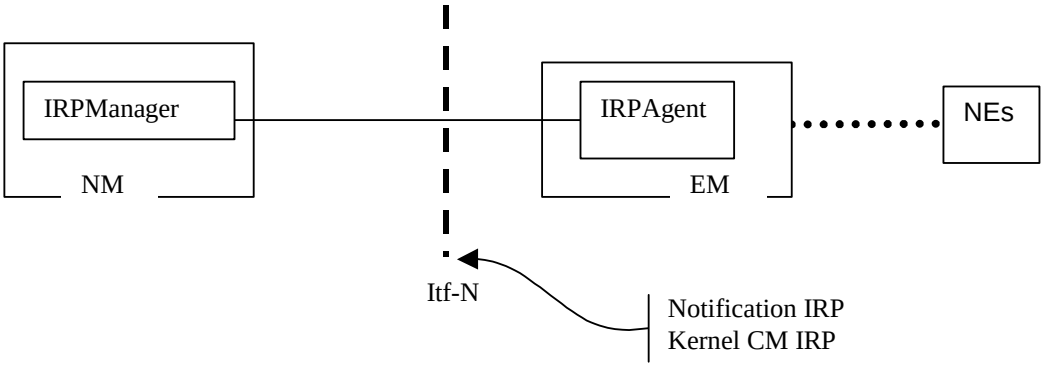

Figure 4.1: System Context A

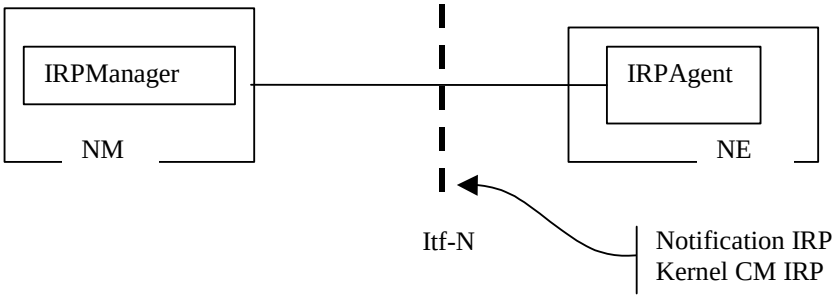

Figure 4.2: System Context B

4.2 Compliance rules

For general definitions of compliance rules related to qualifiers (Mandatory/Optional/Conditional) for *operations*, *notifications and parameters* (of operations and notifications) please refer to TS 32.150~~0~~2 [19~~2~~].

An IRPAgent that incorporates vendor-specific extensions shall support normal communication with a 3GPP SA5-compliant IRPManager with respect to all Mandatory and Optional managed object classes, attributes, associations, operations, parameters and notifications without requiring the IRPManager to have any knowledge of the extensions.

Given that

- rules for vendor-specific extensions remain to be fully specified, and
- many scenarios under which IRPManager and IRPAgent interwork may exist,

it is recognised that ~~in Release 4/5~~ the IRPManager, even though it is not required to have knowledge of vendor-specific extensions, may be required to be implemented with an awareness that extensions can exist and behave accordingly.

End of Change in Clause 4

Change in Clause 5

5 Modelling approach

See 3GPP TS 32.150 [19].~~This clause identifies the modelling approach adopted and used in this IRP.~~

~~As described in TS 32.101 [1], an IRP comprises the following components:~~

~~(1)an **IRP Information Model** that specifies the interface in a protocol neutral manner, defined as an Information Service and/or one or more Network Resource Models,~~

~~(2)a number of **IRP Solution Sets** that provide the actual realization of the operations and notifications defined in the IRP Information Model for each protocol environment.~~

~~The present document defines one such Information Service – the Kernel CM IRP: IS.~~

~~The IRP Information Service is a specification of the *operations* and *notifications* that are visible over the IRP. These operations/notifications are generic in the sense that they do not specify the Managed Objects that are retrieved/manipulated/informed about over the interface, and thus this IS is independent of the NRM being managed.~~

~~5.1 IRP Information Service modelling approach~~

~~The IRP Information Service of the subject IRP specifies a number of protocol independent operations and notifications that are needed by an IRPManager to retrieve CM information from an IRPAgent.~~

~~The operations and notifications of the IRP Information Service are mainly based on the principles of the Common Management Information Service (CMIS) defined in ITU-T Recommendation X.710 [7] and ITU-T Recommendation X.721 [8] (M-GET etc.). Note however, that the Information Service of the subject IRP is focused on the essential operations and notifications needed for CM purposes and thus only covers a subset of the operations/notifications defined in ITU-T Recommendation X.710 [7]/ITU-T Recommendation X.721 [8].~~

~~It is expected that most Solution Sets will implement the operations and notifications by mapping them to standard operations (and possibly standard notifications) that are applicable in the corresponding protocol environment. A CMIP Solution Set should for instance map the operations to the more generic operations defined in CMIS, an SNMP Solution Set should map the operations to applicable SNMP operations, and a CORBA Solution Set should map the operations to applicable OMG/CORBA services.~~

End of Change in Clause 5
End of Document

Annex B (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Mar 2002	S_15	SP-020034	--	--	Submitted to TSG SA #15 for Information	1.0.0	
Sep 2002	S_17	SP-020465	--	--	Submitted to TSG SA #17 for Approval	2.0.0	5.0.0
Mar 2003	S_19	SP-030145	001	--	Add description of notifyCMSynchronizationRecommended notification for KernelCM IRP.	5.0.0	6.0.0
Dec 2003	S_22	SP-030630	003	--	Correction of System Context	6.0.0	6.1.0
Mar 2004	S_23	SP-040119	005	--	Correction of System Context	6.1.0	6.2.0
Jun 2004	S_24	SP-040260	006	--	Add State Management Support to Kernel CM IRP IS 32.622	6.2.0	6.3.0

CR-Form-v7.1

CHANGE REQUEST

⌘ **32.663** **CR** **0013** ⌘**rev** **-** ⌘ Current version: **6.4.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: UICC apps⌘ ☐ ME ☐ Radio Access Network **X** Core Network **X**

Title:	⌘	Correct CORBA SS mapping of notification filterable/non-filterable IS parameters
Source:	⌘	**SA5** (Nortel – Suzèle Lariven – lariven@nortel.com)
Work item code:	⌘	OAM-NIM
Date:	⌘	13/05/2005
Category:	⌘	**F**
Release:	⌘	Rel-6

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:
Ph2 *(GSM Phase 2)*
R96 *(Release 1996)*
R97 *(Release 1997)*
R98 *(Release 1998)*
R99 *(Release 1999)*
Rel-4 *(Release 4)*
Rel-5 *(Release 5)*
Rel-6 *(Release 6)*
Rel-7 *(Release 7)*

Reason for change:	⌘	Incoherent filterable/non-filterable notification parameter mappings: • non-filterable IS parameter notificationId to NV pair of filterable_body_fields • filterable IS parameter eventTime not to NV pair of filterable_body_fields Incomplete and incorrect notification parameter mapping for non-filterable IS parameters attributeList and attributeValueChangeDefinition
Summary of change:	⌘	• Mapped IS parameter notificationId to NV pair of remainder_of_body • Mapped IS parameter eventTime to NV pair of filterable_body_fields • Corrected mapping definition of IS parameter attributeList • Corrected mapping definition of IS parameter attributeValueChangeDefinition • Alignments with TS 32.150 Style Guide for CORBA SS IDL • Editorial corrections
Consequences if not approved:	⌘	The notification filterable/non-filterable IS parameter mapping in the CORBA SS would be incoherent and incomplete.

Clauses affected:	⌘	6.4, 7, annex A, annex B, annex C

Other specs affected: ⌘	Y	N		⌘
		X	Other core specifications	
		X	Test specifications	
		X	O&M Specifications	

Other comments: ⌘

Change in Clause 6.4

6.4 Notification attribute mapping

[…]

Table 6.4.3: Mapping from IS `notifyObjectCreation` attributes to SS equivalent OBJECT_CREATION

IS Attribute of `notifyObjectCreation` in 3GPP TS 32.662 [4]	SS Attribute	Qualifier
[…]		
attributeList	KernelC~~M~~mNotifications::MOCreation::~~MOAttributeSet~~InitialAttributeValues (contained in remainder_of_body)	O

Table 6.4.4: Mapping from IS `notifyObjectDeletion` attributes to SS equivalent OBJECT_DELETION

IS Attribute of `notifyObjectDeletion` in 3GPP TS 32.662 [4]	SS Attribute	Qualifier
[…]		
attributeList	KernelC~~M~~mNotifications::MODeletion::~~MOAttributeSet~~AttributeValues (contained in remainder_of_body)	O

Table 6.4.5: Mapping from IS `notifyAttributeValueChange` attributes to SS equivalent ATTRIBUTE_VALUE_CHANGE

IS Attribute of `notifyAttributeValue Change` in 3GPP TS 32.662 [4]	SS Attribute	Qualifier
[…]		
attributeValueChangeDefinition	KernelC~~M~~mNotifications:: AttributeValueChange::~~MOAttributeSet~~ModifiedAttributeSet (contained in remainder_of_body)	M

[…]

End of Change in Clause 6.4

Change in Clause 7

7 Use of OMG Structured Event

[…]

Table 7.1: Use of OMG Structured Event

SS Attribute	OMG CORBA Structured Event attribute	Comment
[…]		
~~N~~notificationId	One NV pair of ~~filterable_body_fields~~ remainder_of_body	It is an attribute of notificationHeader. Name of NV pair is a string, KernelCmNotifications::<interface>::NOTIFICATION_ID where <interface> is either MOCreation, MODeletion, AttributeValueChange, StateChange or CMSynchronizationRecommended. Value of NV pair is a long. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [9]).
~~E~~eventTime	One NV pair of ~~remainder_of_body~~ filterable_body_fields	It is an attribute of notificationHeader. Name of NV pair is a string, KernelCmNotifications::<interface>::EVENT_TIME where <interface> is either MOCreation, MODeletion, AttributeValueChange, StateChange or CMSynchronizationRecommended. Value of NV pair is a ManagedGenericIRPConstDefs::IRPTime defined in 3GPP TS 32.303 [9]. See corresponding table in Notification IRP: CORBA SS (3GPP TS 32.303 [9]).
[…]		
~~There is no corresponding SS attribute~~attributeList ~~attributeValueChangeDefinition~~	remainder_of_non_filterable_body (see 3GPP TS 32.303 [9])	Is used to transport attribute information. For Object Creation notification, this is defined by KernelCmNotifications::MOCreation::InitialAttributeValues. For Object Deletion notification, this is defined by KernelCmNotifications::MODeletion::AttributeValues. ~~For Attribute Value Change notification, this is defined by KernelCmNotifications::AttributeValueChange::ModifiedAttributeSet.~~ The name component of InitialAttributeValues~~,~~ and AttributeValues ~~and ModifiedAttributeSet~~ will be set to attribute names defined in KernelCmNRMDefs.
attributeValueChangeDefinition	remainder_of_non_filterable_body (see 3GPP TS 32.303 [9])	For Attribute Value Change notification, this is defined by KernelCmNotifications::AttributeValueChange::ModifiedAttributeSet. The name component of ModifiedAttributeSet will be set to attribute name defined in KernelCmNRMDefs.
[…]		

[…]

End of Change in Clause 7

Change in Annex A

Annex A (normative):
IDL specification (file name "KernelCmConstDefs.idl")

```
//~File: KernelCmConstDefs.idl
```

```
#ifndef _KERNEL_CM_CONST_DEFS_IDL_
#define _KERNEL_CM_CONST_DEFS_IDL_

// This statement must appear after all include statements
#pragma prefix "3gppsa5.org"

module KernelCmConstDefs
{
      /**
       * Information about one attribute
       * - name defines the name of the attribute
       * - value defines the value of the attribute
       *
       */
      struct MOAttribute
      {
         string name;
         any value;
      };


      /**
       * A set of attribute names and values
       */
      typedef sequence<MOAttribute> MOAttributeSet;


    /**
     * ScopeType defines the kind of scope to use in a CM synchronization
     * request together with ScopePara.level, in the SCOPE field.
     *
     * ScopePara.level is always >= 0. If a level is bigger than the
     * depth of the tree there will be no exceptions thrown.
     * BASE_ONLY: level ignored, just return the base object.
     * BASE_NTH_LEVEL: return all subordinate objects that are on "level"
     * distance from the base object, where 0 is the base object.
     * BASE_SUBTREE: return the base object and all of its subordinates
     * down to and including the nth level.
     * BASE_ALL: level ignored, return the base object and all of it's
     * subordinates.
     */
    enum ScopeType
    {
       BASE_ONLY,
       BASE_NTH_LEVEL,
       BASE_SUBTREE,
       BASE_ALL
    };

    struct ScopePara
    {
       ScopeType type;
       unsigned long level;
    };

    /* The format of Distinguished Name (DN) is specified in 3GPP TS 32.300
    "Name Conventions for Managed Objects".
    */
    typedef string DN;

   typedef sequence <long> NotifIdSetType;

   /*
   This holds identifiers of notifications that are correlated.
   */
   struct CorrelatedNotification
   {
      DN source;  // Contains DN of MO that emitted the set of notifications
                  // DN string format in compliance with Name Convention for
                  // Managed Object.
                  // This may be a zero-length string. –In this case, the MO
                  // is identified by the value of the MOI attribute
                  // of the Structured Event, i.e., the notification.
      NotifIdSetType notif_id_set;  // Set of related notification ids
   };
```

```
    /*
    Correlated Notification sets are sets of Correlated Notification
    structures.
    */
    typedef sequence <CorrelatedNotification> CorrelatedNotificationSetType;

    /*
    This block identifies attributes which are included as part of the Kernel
    CM IRP. These attribute values should not clash with those defined for the
    attributes of notification header (see IDL of Notification IRP).
    */
    interface AttributeNameValue
    {
       const string SOURCE_INDICATOR = "SOURCE";
       const string ADDITIONAL_TEXT = "ADD_TEXT";
       const string CORRELATED_NOTIFICATIONS = "CORREL_NOTIFS";
       const string BASE_MO_CLASS = "BASE_MOC";
       const string BASE_MO_INSTANCE = "BASE_MOI";
       const string SCOPE = "SCOPE";
    };

};

#endif // _KERNELCMNOTIFDEFS_IDL_ _KERNEL_CM_CONST_DEFS_IDL_
```

End of Change in Annex A

Change in Annex B

Annex B (normative): IDL specification (file name "KernelCmIRPSystem.idl")

```
//File: KernelCmIRPSystem.idl

#ifndef _KERNEL_CM_IRP_SYSTEM_IDL_
#define _KERNEL_CM_IRP_SYSTEM_IDL_

#include "<ManagedGenericIRPConstDefs.idl">
#include "<ManagedGenericIRPSystem.idl">

// This statement must appear after all include statements
#pragma prefix "3gppsa5.org"

module KernelCmIRPSystem
{

    exception GetKernelCMIRPNotificationProfileException { string reason; };
    exception GetKernelCMIRPOperationProfileException { string reason; };
    exception GetNRMIRPVersion { string reason; };
    exception GetKernelCMIRPVersionsException { string reason; };

    /**
     * The KernelCmIrpOperations interface.
     * Supports a number of Resource Model versions.
     */
    interface KernelCmIrpOperations
    {
      /*
      Return the list of all supported Kernel CM IRP versions.
      */
      ManagedGenericIRPConstDefs::VersionNumberSet get_kernel_CM_IRP_versions (
      )
      raises (GetKernelCMIRPVersionsException);


     /**
      * Get the version(s) of the interface
      *
      * @raises GetNRMIRPVersion when the system for some reason
      *   can not return the supported versions.
```

```
        * @returns all supported versions.
        */
        void get_NRM_IRP_version
        (
           out ManagedGenericIRPConstDefs::VersionNumberSet versionNumberList,
           out ManagedGenericIRPConstDefs::VersionNumberSet vSEVersionNumberList
        )
        raises (GetNRMIRPVersion);

        /*
        Return the list of all supported operations and their supported
        parameters for a specific KernelCM IRP version.
        */
        ManagedGenericIRPConstDefs::MethodList get_kernel_CM_IRP_operation_profile (
           in ManagedGenericIRPConstDefs::VersionNumber kernel_CM_IRP_version
        )
        raises (GetKernelCMIRPOperationProfileException,
                ManagedGenericIRPSystem::OperationNotSupported,
                ManagedGenericIRPSystem::InvalidParameter);

        /*
        Return the list of all supported notifications and their supported
        parameters for a specific KernelCM IRP version.
        */
        ManagedGenericIRPConstDefs::MethodList
           get_kernel_CM_IRP_notification_profile
        (
           in ManagedGenericIRPConstDefs::VersionNumber kernel_CM_IRP_version
        )
        raises (GetKernelCMIRPNotificationProfileException,
                ManagedGenericIRPSystem::OperationNotSupported,
                ManagedGenericIRPSystem::InvalidParameter);

    };
};
#endif // _KERNEL_CM_IRP_SYSTEM_IDL_
```

End of Change in Annex B

Change in Annex C

Annex C (normative): IDL specification (file name "KernelCmNotifications.idl")

```
//File: KernelCmNotifications.idl

#ifndef _KERNEL_CM_NOTIFICATIONS_IDL_
#define _KERNEL_CM_NOTIFICATIONS_IDL_

#include "NotificationIRPConstDefs.idl"
#include "StateManagementIRPConstDefs.idl"
#include "<KernelCmConstDefs.idl">
#include "<NotificationIRPNotifications.idl">

// This statement must appear after all include statements
#pragma prefix "3gppsa5.org"

module KernelCmNotifications
{

        /**
         *  This interface defines fields that are common for all
         *  notification types.
         *  All constants in the scope of this interface will be
         *  visible in the interfaces that inherits this.
         *  For instance constant
         *  NotificationCommon::MANAGED_OBJECT_CLASS
         *  can be addressed by MODeletion::MANAGED_OBJECT_CLASS
         */

        interface NotificationCommon: NotificationIRPNotifications::Notify
```

```
{

   /**
    *  This constant defines the name of the
    *  source indicator property.
    */
   const string SOURCE_INDICATOR =
     KernelCmConstDefs::AttributeNameValue::SOURCE_INDICATOR;

   /**
    * Valid values for the SOURCE_INDICATOR
    * property
    */
   const string RESOURCE_OPERATION = "RESOURCE OPERATION";
   const string MANAGEMENT_OPERATION = "MANAGEMENT OPERATION";
   const string UNKNOWN_OPERATION = "UNKNOWN";

   /**
    *  This constant defines the name of the
    *  additional text property.
    *  The data type for the value of this property
    *  is a string.
    */
   const string ADDITIONAL_TEXT =
     KernelCmConstDefs::AttributeNameValue::ADDITIONAL_TEXT;


   /**
    *  This constant defines the name of the
    *  correlated notifications property.
    *  The value part of the property is
    *    KernelCmConstDefs::CorrelatedNotificationSetType
    */
   const string CORRELATED_NOTIFICATIONS =
     KernelCmConstDefs::AttributeNameValue::CORRELATED_NOTIFICATIONS;
};


/**
 *  Constant definitions for the MO deleted notification
 */
interface MODeletion : NotificationCommon
{
   const string EVENT_TYPE = "x7";

   /**
    * This information mapped into the remainder_of_body
    * in the StructuredEvent
    */
   typedef KernelCmConstDefs::MOAttributeSet AttributeValues;
};



/**
 *  Constant definitions for the MO created notification
 */
interface MOCreation : NotificationCommon
{
   const string EVENT_TYPE = "x6";

   /**
    * This information mapped into the remainder_of_body
    * in the StructuredEvent
    */
   typedef KernelCmConstDefs::MOAttributeSet InitialAttributeValues;
};


/**
 *  Constant definitions for the Attribute Value Change
 *  notification
 */
interface AttributeValueChange : NotificationCommon
{
   const string EVENT_TYPE = "x8";

   /**
```

```
         * Information about modidified attributes for
         * one MO instance.
         * - name defines the name of the attribute
         * - newValue defines the new value of the attribute
         * - oldValue defines the previous value of the attribute
         *    The value is optional, which means that it may contain
         *    an empty any (null inserted in the any).
         *
         */
        struct ModifiedAttribute
        {
           string name;
           any newValue;
           any oldValue;
        };


        /**
         * This information mapped into the remainder_of_body
         * in the StructuredEvent.
         */
        typedef sequence<ModifiedAttribute> ModifiedAttributeSet;


    };

    /**
     *  Constant definitions for the CM Synchronization Recommended notification
     */
    interface CMSynchronizationRecommended: NotificationIRPNotifications::Notify
    {
       const string EVENT_TYPE = "x9";

       /**
        *  This constant defines the name of the
        *  additional text property.
        *  The data type for the value of this property
        *  is a string.
        */
       const string ADDITIONAL_TEXT =
         KernelCmConstDefs::AttributeNameValue::ADDITIONAL_TEXT;

        /**
        *  This constant defines the name of the
        *  base MO class property.
        *  The value part of this property will carry
        *  the base MO class name as a string.
        */
       const string BASE_MO_CLASS =
         KernelCmConstDefs::AttributeNameValue::BASE_MO_CLASS;

       /**
        *  This constant defines the name of the
        *  base MO instance property.
        *  The value part of this property will carry
        *  the base MO distinguished name as a string.
        */
       const string BASE_MO_INSTANCE =
         KernelCmConstDefs::AttributeNameValue::BASE_MO_INSTANCE;


       /**
        *  This constant defines the name of the
        *  scope property.
        *  The data type for the value of this property
        *  is KernelCmConstDefs::ScopePara.
        */
       const string SCOPE =
         KernelCmConstDefs::AttributeNameValue::SCOPE;

    };

    /**
     *  Constant definitions for the State Change notification
     */
    interface StateChange : NotificationCommon
    {
       const string EVENT_TYPE = "xA";
```

```
      };

};

#endif // _KERNELCMNOTIFDEFS_IDL_ _KERNEL_CM_NOTIFICATIONS_IDL_
```

End of Change in Annex C
End of Document

Annex D (informative): Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Sep 2002	S_17	SP-020466	--	--	Submitted to TSG SA #17 for Approval	1.0.0	5.0.0
Mar 2003	S_19	SP-030143	001	--	CORBA IDL Compiler Errors	5.0.0	5.1.0
Mar 2003	S_19	SP-030145	002	--	Add IDL definition of notifyCMSynchronizationRecommended notification for KernelCM IRP	5.1.0	6.0.0
Jun 2004	S_24	SP-040261	004	--	Add Missing CorrelatedNotificationSetType definition	6.0.0	6.1.0
Sep 2004	S_25	SP-040568	007	--	Add missing DN definition	6.1.0	6.2.0
Sep 2004	S_25	SP-040568	009	--	Add missing IDL for get_kernel_CM_IRP_versions	6.1.0	6.2.0
Sep 2004	S_25	SP-040569	010	--	Add State Management Support to Kernel CM IRP CORBA SS	6.1.0	6.2.0
Dec 2004	S_26	SP-040812	011	--	Correct the mapping of IS-defined non-filterable parameters to SS-defined non-filterable fields (instead of filterable fields) - Align with IS in 32.662	6.2.0	6.3.0
Mar 2005	S_27	SP-050050	012	--	IDL incompliant to the style guide	6.3.0	6.4.0