Technical Specification Group Services and System Aspects
Meeting #26, Athens, GREECE, 13 - 16 December 2004

TSGS#26(04)0797

Source: **SA5 (Telecom Management)**

Title: **Rel-6 CR 32.323 Test management IRP CORBA SS**

Document for: **Approval**

Agenda Item: **7.5.3**

Doc1stevel	Specifica	CR	R	Phase	Subject	Ca	VersCu	Doc2ndLev	WorkitemsID
SP-040797	32.323	002	--	Rel-6	Align the IDL style innthe CORBA SS with the IDL Style Guide in 32.150	F	6.0.0	S5-047121	OAM-NIM

3GPP TSG-SA5 (Telecom Management) **S5-047121**
Meeting #40, Sanya, CHINA, 15 - 19 November 2004

CR-Form-v7

CHANGE REQUEST

⌘ **32.323 CR 002** ⌘ **rev** **-** ⌘ Current version: **6.0.0** ⌘

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 ☒ Core Network ☒

Title: ⌘ Align the IDL style innthe CORBA SS with the IDL Style Guide in 32.150

Source: ⌘ **SA5** (Yangli, afi@huawei.com)

Work item code: ⌘ OAM-NIM ***Date:*** ⌘ 19/11/2004

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: ⌘ Correct the mapping of IS-defined non-filterable parameters to SS-defined non-filterable fields (instead of filterable fields). Align IDL style with IDL Style Guide.

Summary of change: ⌘ Place IS-defined non-filterable parameters into remaining_body of CORBA structured event. Align IDL style with IDL Style Guide.

Consequences if not approved: ⌘ IRPAgent wastes CPU cycles on non-filterable parameters before emission of notification. This non-standard behaviour can confuse IRPManager.

Clauses affected: ⌘ Table 8, A.1, A.2, A.3

Other specs affected: ⌘

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

Other comments: ⌘

Change in Table 5

Table 5: Mapping for `notifyTestResult`

IS Parameters	OMG CORBA Structured Event attribute	Qualifier	Comment
There is no corresponding IS attribute.	domain_name	M	It carries the IRP document version number string. See subclause 3.3. It indicates the syntax and semantics of the Structured Event as defined by the present document.
notificationType	Type_name	M	This is the NOTIFY_TM_TEST_RESULT of module of TestManagementIRPConstDefs.
There is no corresponding IS attribute	event_name	M	It carries no information.
There is no corresponding IS attribute.	Variable Header		
objectClass, objectInstance	One NV pair of filterable_ body_fields	M	NV stands for name-value pair. Order arrangement of NV pairs is not significant. The name of NV-pair is always encoded in string. Name of this NV pair is the MANAGED_OBJECT_INSTANCE of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. See corresponding table in Notification IRP: CORBA SS [5].
notificationId	One NV pair of remaining_body	M	Name of NV pair is the NOTIFICATION_ID of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a long. See corresponding table in Notification IRP: CORBA SS [5].
eventTime	One NV pair of filterable_ body_fields	M	Name of NV pair is the EVENT_TIME of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is IRPTime. See corresponding table in Notification IRP: CORBA SS [5].
systemDN	One NV pair of filterable_ body_fields	M	Name of NV pair is the SYSTEM_DN of interface AttributeNameValue of module NotificationIRPConstDefs. Value of NV pair is a string. See corresponding table in Notification IRP: CORBA SS [5].
testInvocationInitiator	One NV pair of ~~filterable_ body_fields~~remaining_body	M	Name of NV pair is the TEST_INVOCATION_INITIATOR of module TestManagementIRPConstDefs. Value of NV pair is a string.
testInvocationId	One NV pair of ~~filterable_ body_fields~~remaining_body	M	Name of NV pair is the TEST_INVOCATION_ID of module TestManagementIRPConstDefs. Value of NV pair is a string.
testActualStartTime	One NV pair of ~~filterable_ body_fields~~remaining_body	O	Name of NV pair is the TEST_ACTUAL_START_TIME of module TestManagementIRPConstDefs. Value of NV pair is IRPTime. See corresponding table in Notification IRP: CORBA SS [5].
testActualStopTime	One NV pair of ~~filterable_ body_fields~~remaining_body	O	Name of NV pair is the TEST_ACTUAL_STOP_TIME of module TestManagementIRPConstDefs. Value of NV pair is IRPTime. See corresponding table in Notification IRP: CORBA SS [5].
testOutcome	One NV pair of ~~filterable_ body_fields~~remaining_body	O	Name of NV pair is the TEST_OUTCOME of module TestManagementIRPConstDefs. Value of NV pair is the enum TestOutcomeType of

IS Parameters	OMG CORBA Structured Event attribute	Qualifier	Comment
			TestManagementIRPConstDefs.
mORT	One NV pair of ~~filterable_body_fields~~remaining_body	O	Name of NV pair is the MORT of module TestManagementIRPConstDefs. Value of NV pair is a string.
proposedRepairActions	One NV pair of remaining_body~~filterable_body_fields~~	O	Name of NV pair is the PROPOSED_REPAIR_ACTIONS of module TestManagementIRPConstDefs. Value of NV pair is a string.
additionalInformation	Two NV pair of ~~filterable_body_fields~~remaining_body	O	Name of one NV pair is the ~~TEST_~~ADDITIONAL_INFORMATION of module TestManagementIRPConstDefs. Value of NV pair is a string.
fileReference	One NV pair of ~~filterable_body_fields~~remaining_body	O	Name of NV pair is FILE_REFERENCE of module TestManagementIRPConstDefs. Value of NV pair is a string.
fileExpiryDate	One NV pair of ~~filterable_body_fields~~remaining_body	O	Name of NV pair is the FILE_EXPIRY_DATE of module TestManagementIRPConstDefs. Value of NV pair is IRPTime. See corresponding table in Notification IRP: CORBA SS [5].
~~There is no corresponding IS attribute.~~	~~remaining_body~~		

End of change in Table 5

Change in A.1,A.2,A.3

Annex A (normative): IDL specifications

A.1 IDL specification (file name "TestManagementIRPConstDefs.idl")

```
// File: TestManagementIRPConstDefs.idl

#ifndef _TESTMANAGEMENTIRPCONSTDEFS_IDL_~~TestManagementIRPConstDefs_idl~~
#define _TESTMANAGEMENTIRPCONSTDEFS_IDL_~~TestManagementIRPConstDefs_idl~~

#include "CosNotification.idl"
#include "ManagedGenericIRPConstDefs.idl"

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

/* ## Module: TestManagementIRPConstDefs
This module contains commonly used definitions for Test Management IRP
======================================================================
*/
module TestManagementIRPConstDefs
{
   /*
   This defines the notification type of this Test Management
      IRP.
   */
   const string NOTIFY_TM_TEST_RESULT = "x1";
```

```
/*
This enum defines the test state
*/
enum TestStateType {
   NotInitialized,
   Idle,
   Initializing,
   Testing,
   Terminating,
   Disabled
};

/*
This enum defines the test outcome
*/
enum TestOutcomeType {
   Inconclusive,
   Pass,
   Fail,
   TimeOut,
   PrematureTermination
};


/*
This block defines notification attributes of this IRP.
These attribute values should not clash with those used
    in Notification header (see IDL of Notification IRP).
*/
interface AttributeNameValue
{

 const string TEST_INVOCATION_INITIATOR = "f";
 const string TEST_INVOCATION_ID = "g";
 const string TEST_ACTUAL_START_TIME = "h";
 const string TEST_ACTUAL_STOP_TIME = "i";
 const string TEST_OUTCOME = "j";
 const string MORT = "k";
 const string PROPOSED_REPAIR_ACTIONS = "l";
 const string ADDITIONAL_INFORMATION = "m";
 const string FILE_REFERENCE = "n";
 const string FILE_EXPIRY_DATE = "o";
 };

typedef string TestInvocationInitiator;
typedef string ToBeMonitoredTO;

typedef CosNotification::PropertySeq NVPairs;

/*
Define a seq of to-be-initiated-test
*/
struct ToBeInitiatedTest
{
    unsigned long max_testing_state_duration;//seconds;0->no limit
    string  toBeTestedMORT; //MORT DN
    string  tOClass;        //Tester object class
    string  tODN;           //Tester object DN
    NVPairs tONVPair;       //Tester object attributes in NV pairs
};
typedef sequence <ToBeInitiatedTest> ToBeInitiatedTestSeq;

/*
Define the structure returned by initiate_tests
*/
struct InitiateTestsResponseElement
{
   // If failureReason is NULL, the test is initiated successfully and
   //     testInvocationId contains the invocation id. In case the tester object name is not
   //     provided in the request, it shall be carried by testerObjectDN. In case the tester
   //     object name is provided in the request tODN shall be NULL.
   // Else, the test initiation fails and failureReason contains
   //     the failure reason and testInvocationId contains garbage.
   string failureReason;
   string testInvocationId;
   string tODN;
};
```

```
   typedef sequence <InitiateTestsResponseElement> InitiateTestsResponse;



   /*
   Define a seq of to-be-terminated-test
   */
   typedef string TestInvocationId;
   typedef sequence <TestInvocationId> ToBeTerminatedTestSeq;

   /*
   Define the structure returned by terminate_tests
   */
   struct TerminateTestsResponseElement
   {
      // If failureReason is NULL, the test has terminated successfully and
      //     testInvocationId identifies the terminated invocation.
      // Else, the test termination fails and failureReason contains
      //     the failure reason and testInvocationId contains garbage.
      string failureReason;
      string testInvocationId;
   };
   typedef sequence <TerminateTestsResponseElement> TerminateTestsResponse;

   /*
   Define the structure of a TOAttributes.
   */
   struct TOAttributes
   {
       TestStateType   testState;
       TestOutcomeType testOutcome;
       NVPairs         attributesInNVPairs;
   };

};
#endif _TESTMANAGEMENTIRPCONSTDEFS_IDL_
```

A.2 IDL specification (file name "TestManagementIRPSystem.idl")

```
// File: TestManagementIRPSystem.idl

#ifndef ~~TestManagementIRPSystem_idl~~_TESTMANAGEMENTIRPSYSTEM_IDL_
#define ~~TestManagementIRPSystem_idl~~_TESTMANAGEMENTIRPSYSTEM_IDL_

#include "TestManagementIRPConstDefs.idl"
#include "ManagedGenericIRPSystem.idl"

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

/* ## Module: TestManagementIRPSystem
This module contains the specification of all methods of TestManagement IRP Agent.
================================================================
*/
module TestManagementIRPSystem
{
   /*
   System may fail to complete an operation.  System can provide reason
   to qualify the failed reason.  The semantics carried in reason
   is outside the scope of this IRP.
   */
   exception GetTestManagementIRPVersions { string reason; };
   exception GetTestManagementIRPOperationsProfile { string reason; };
   exception GetTestManagementIRPNotificationProfile { string reason; };
   exception InitiateTests { string reason; };
   exception TerminateTests { string reason; };
   exception MonitorTest { string reason; };


   interface TestManagementIRP
   {
      /*
      Return the list of all supported TestManagement IRP versions.
      */
      ManagedGenericIRPConstDefs::VersionNumberSet
      get_Test_Management_IRP_versions (
      )
      raises (GetTestManagementIRPVersions);


      /*
      Return the list of all supported operations and their supported
      parameters for a specific TestManagement IRP version.
      */
      ManagedGenericIRPConstDefs::MethodList
      get_Test_Management_IRP_operations_profile (
         in ManagedGenericIRPConstDefs::VersionNumber
             test_management_irp_version
      )
      raises (GetTestManagementIRPOperationsProfile,
              ManagedGenericIRPSystem::OperationNotSupported,
              ManagedGenericIRPSystem::InvalidParameter);

      /*
      Return the list of all supported notifications and their supported
      parameters for a specific TestManagement IRP version.
      */
      ManagedGenericIRPConstDefs::MethodList
      get_Test_Management_IRP_notification_profile (
         in ManagedGenericIRPConstDefs::VersionNumber
            test_management_irp_version
      )
      raises (GetTestManagementIRPNotificationProfile,
              ManagedGenericIRPSystem::OperationNotSupported,
              ManagedGenericIRPSystem::InvalidParameter);


      /*
      Request to initiate tests.
      */
      TestManagementIRPConstDefs::InitiateTestsResponse
      initiate_tests (
         in TestManagementIRPConstDefs::TestInvocationInitiator
            test_invocation_initiator,
         in TestManagementIRPConstDefs::ToBeInitiatedTestSeq
            to_be_initiated_test_seq
      )
      raises (InitiateTests,
              ManagedGenericIRPSystem::InvalidParameter);


      /*
      Request to terminate tests.
      */
      TestManagementIRPConstDefs::TerminateTestsResponse
      terminate_tests (
         in TestManagementIRPConstDefs::ToBeTerminatedTestSeq
             to_be_terminated_test_seq
      )
      raises (TerminateTests,
              ManagedGenericIRPSystem::InvalidParameter);


      /*
      Request test info (to monitor a test).
      */
      ManagedGenericIRPConstDefs::Signal monitor_test (
         in TestManagementIRPConstDefs::ToBeMonitoredTO
             to_be_monitored_TO,
         out TestManagementIRPConstDefs::TOAttributes tO_attributes
      )
      raises (MonitorTest,
              ManagedGenericIRPSystem::InvalidParameter);




   };
};
```

```
#endif _TESTMANAGEMENTIRPSYSTEM_IDL_
```

A.3 IDL specification (file name "TestManagementIRPNotifications.idl")

```
// File: TestManagementIRPNotifications.idl

#ifndef _TESTMANAGEMENTIRPNOTIFICATIONS_IDL
#define _TESTMANAGEMENTIRPNOTIFICATIONS_IDL

#include "TestManagementIRPConstDefs.idl"
#include "NotificationIRPConstDefs.idl"
#include "NotificationIRPNotifications.idl"

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

/* ## Module: TestManagementIRPNotifications
This module contains the specification of all notifications of Test Management IRP Agent.
============================================================================
*/

module TestManagementIRPNotificationsfDefs
{
   /**
   *  Constant definitions for the notifyTestResult notification
   */
   interface notifyTestResult: NotificationIRPNotifications::Notify
   {
     const string EVENT_TYPE = "notifyTestResult";

     /**
     * This constant defines the name of the  period property,
     * which is transported in the remaining_body fields.
     * The data type for the value of this property
     * is TestManagementIRPConstDefs:: TestInvocationInitiator.
     */
     const string TEST_INVOCATION_INITIATOR =
         TestManagementIRPConstDefs::AttributeNameValue::TEST_INVOCATION_INITIATOR;

     /*
     * This constant defines the name of the
     * TestInvocationId property,
     * which is transported in the remaining body
     * fields.
     * The data type for the value of this property
     * is TestManagementIRPConstDefs:: TestInvocationId.
     */
     const string TEST_INVOCATION_ID =
         TestManagementIRPConstDefs::AttributeNameValue::TEST_INVOCATION_ID;

     /*
     * This constant defines the name of the
     * TestActualStartTime property,
     * which is transported in the remaining_body
     * fields.
     * The data type for the value of this property
     * is TestManagementIRPConstDefs:: TestActualStartTime.
     */
      const string TEST_ACTUAL_START_TIME =
          TestManagementIRPConstDefs::AttributeNameValue::TEST_ACTUAL_START_TIME;

     /*
     * This constant defines the name of the
     * TestActualStopTime property,
     * which is transported in the remaining_body
     * fields.
     * The data type for the value of this property
     * is TestManagementIRPConstDefs:: TestActualStopTime.
     */
      const string TEST_ACTUAL_STOP_TIME =
          TestManagementIRPConstDefs::AttributeNameValue::TEST_ACTUAL_STOP_TIME;
     /*
     * This constant defines the name of the
     * testOutcome property,
     * which is transported in the remaining_body
     * fields.
     * The data type for the value of this property
     * is TestManagementIRPConstDefs:: testOutcome.
     */
      const string TEST_OUTCOME = TestManagementIRPConstDefs::AttributeNameValue::TEST_OUTCOME;

     /*
     * This constant defines the name of the
     * MORT property,
     * which is transported in the remaining_body
     * fields.
     * The data type for the value of this property
     * is TestManagementIRPConstDefs::MORT.
     */
      const string MORT = TestManagementIRPConstDefs::AttributeNameValue::MORT;

     /*
     * This constant defines the name of the
     * ProposedRepairActions property,
     * which is transported in the remaining_body
     * fields.
     * The data type for the value of this property
     * is TestManagementIRPConstDefs::ProposedRepairActions.
     */
      const string PROPOSED_REPAIR_ACTIONS =
          TestManagementIRPConstDefs::AttributeNameValue::PROPOSED_REPAIR_ACTIONS;

     /*
     * This constant defines the name of the
     * AdditionalInformation property,
     * which is transported in the remaining_body
     * fields.
     * The data type for the value of this property
     * is TestManagementIRPConstDefs:: AdditionalInformation.
     */
      const string ADDITIONAL_INFORMATION =
          TestManagementIRPConstDefs::AttributeNameValue::ADDITIONAL_INFORMATION;

     /*
     * This constant defines the name of the
     * FileReference property,
     * which is transported in the remaining_body
     * fields.
     * The data type for the value of this property
     * is TestManagementIRPConstDefs:: FdditionalInformation.
     */
      const string FILE_REFERENCE = TestManagementIRPConstDefs::AttributeNameValue::FILE_REFERENCE;

     /*
     * This constant defines the name of the
     * FileExpiryDate property,
     * which is transported in the remaining_body
     * fields.
     * The data type for the value of this property
     * is TestManagementIRPConstDefs:: FileExpiryDate.
     */
      const string FILE_EXPIRY_DATE =
          TestManagementIRPConstDefs::AttributeNameValue::FILE_EXPIRY_DATE;


   };

};

#endif _TESTMANAGEMENTIRPNOTIFICATIONS_IDL_
```

Change in A.1,A.2,A.3
End of document