YD

中华人民共和国通信行业标准

YD/T 1584.4-2007

2GHz 数字蜂窝移动通信网 网络管理通用技术要求 第 4 部分 基于 CORBA 技术的接口设计

Technical Specification for 2GHz Digital Cellular Mobile Communications Network Management General Part 4 CORBA based Interface Design

2007-05-16 发布 2007-05-16 实施

中华人民共和国信息产业部 发布

目　次

前　言

本标准是《2GHz 数字蜂窝移动通信网网络管理技术要求》系列标准之一。该系列标准的预计结构为：系列标准由四项标准组成，各项标准又分为多个部分标准。该系列标准和部分标准的名称预计如下：

1. 2GHz数字蜂窝移动通信网网络管理通用技术要求　第1部分　基本原则
2. 2GHz数字蜂窝移动通信网网络管理通用技术要求　第2部分　接口功能
3. 2GHz数字蜂窝移动通信网网络管理通用技术要求　第3部分　接口分析
4. 2GHz数字蜂窝移动通信网网络管理通用技术要求　第4部分　基于CORBA技术的管理接口设计
5. 2GHz WCDMA数字蜂窝移动通信网网络管理技术要求（第一阶段）　第1部分　配置网络资源模型
6. 2GHz WCDMA数字蜂窝移动通信网网络管理技术要求（第一阶段）　第2部分　性能网络资源模型
7. 2GHz WCDMA数字蜂窝移动通信网网络管理技术要求（第一阶段）　第3部分　基于CORBA技术的网络资源模型设计
8. 2GHz cdma2000数字蜂窝移动通信网网络管理技术要求（第一阶段）　第1部分　配置网络资源模型
9. 2GHz cdma2000数字蜂窝移动通信网网络管理技术要求（第一阶段）　第2部分　性能网络资源模型
10. 2GHz cdma2000数字蜂窝移动通信网网络管理技术要求（第一阶段）　第3部分　基于CORBA技术的网络资源模型设计
11. 2GHz TD-SCDMA数字蜂窝移动通信网网络管理技术要求（第二阶段）　第1部分　配置网络资源模型
12. 2GHz TD-SCDMA数字蜂窝移动通信网网络管理技术要求（第二阶段）　第2部分　性能网络资源模型
13. 2GHz TD-SCDMA数字蜂窝移动通信网网络管理技术要求（第二阶段）　第3部分　基于CORBA技术的网络资源模型设计

本标准对应于第三代移动通信伙伴项目（3GPP）的以下 TS 32 系列标准和第三代移动通信伙伴项目第二组（3GPP2）的以下标准：

1. 3GPP TS 32.413 Telecommunication management；Performance Management IRP；CORBA Solution Set（电信管理；性能集成参考点；基于CORBA的接口设计）

2. 3GPP TS 32.303 Telecommunication management；Configuration Management；Notification Integration Reference Point（IRP）：CORBA Solution Set（电信管理；配置管理；通知集成参考点；基于CORBA的接口设计）

3. 3GPP TS 32.663 Telecommunication management；Kernel CM Integration Reference Point（IRP）：CORBA Solution Set（电信管理；公共配置集成参考点；基于CORBA的接口设计）

4. 3GPP TS 32.111-3 Telecommunication management；Part 2：Alarm Integration Reference Point（IRP）；CORBA Solution Set（电信管理；告警集成参考点；基于CORBA的接口设计）

5. 3GPP TS 32.603 Telecommunication management；Basic CM Integration Reference Point（IRP）：CORBA Solution Set（电信管理；基本配置集成参考点；基于CORBA的接口设计）

6. 3GPP TS 32.363 Telecommunication management；Entry Point Integration Reference Point（IRP）；CORBA Solution Set（电信管理；入口点集成参考点；基于CORBA的接口设计）

7. 3GPP TS 32.343 Telecommunication management；File Transfer IRP；CORBA Solution Set（电信管理；文件传输集成参考点；基于CORBA的接口设计）

8. 3GPP TS 32.353 Telecommunication management；Communication Surveillance Integration Reference Point (IRP); CORBA Solution Set（电信管理；链路监视集成参考点；基于CORBA的接口设计）

9. 3GPP2 S.S0028-A（Version 1.0）OAM&P for cdma2000（3GPP R4 Delta Specification）（cdma2000网络的操作、管理、维护和指配）

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；

— 取消了过滤条件filter只能取值为“TRUE”的限定；

— 对上述国际标准中的相关IDL文件编译错误情况进行更新，如StateManagement的IDL合并为一个文件；

— 增加OMG UtcT时间格式的说明和使用示例。

— 本标准与上述3GPP和3GPP2相关标准的一致性程度为非等效。

本标准的附录A为规范性附录，附录B为资料性附录。

本标准由中国标准化协会提出并归口。

本标准起草单位：北京邮电大学

本标准主要起草人： 芮兰兰、姚羿志、王 峰、李文璟、李云喜、范小磊

2GHz数字蜂窝移动通信网网络管理通用技术要求
第4部分　基于CORBA技术的接口设计

1　范围

本标准规定了2GHz数字蜂窝移动通信网（以下简称3G）网络管理接口中与网络技术无关的通用管理部分的基于CORBA技术的管理接口设计。

本标准适用于于2GHz数字蜂窝移动通信网中WCDMA，cdma2000和TD-SCDMA技术的网络管理。

2　规范性引用文件

下列文件中的条款通过本标准的引用而成为本标准的条款。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标准。然而，鼓励根据本标准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标准。

YD/T 1584.1-2007　2GHz数字蜂窝移动通信网网络管理通用技术要求　第1部分　基本原则

YD/T 1584.2-2007　2GHz数字蜂窝移动通信网网络管理通用技术要求　第3部分　接口分析

OMG TC Document telecom/98-12-09　CORBA服务：通用对象服务规范OMG通知服务

3　缩略语

下列缩略语适用于本标准。

IRP　Integrated Reference Point　集成参考点

4　公共管理接口设计

下面给出了公共管理接口分析到基于CORBA/IDL技术接口设计的映射表格。

4.1　访问入口点接口设计

4.1.1　操作和通知映射

在《2GHz数字蜂窝移动通信网网络管理通用技术要求 第3部分 接口分析》（以下简称《接口分析》）中定义了访问入口点接口操作和通知的语义，表1给出了这些操作和通知到基于CORBA/IDL技术接口设计的映射。

表1　操作映射表

分析阶段操作/通知	设计阶段映射
getIRPOutline	get_IRP_outline
getIRPReference	get_IRP_reference
releaseIRPReference	release_IRP_reference
notifyEPInfoChanges	push_structured_events
getIRPVersion[注]	get_EP_IRP_versions
getOperationProfile[注]	get_EP_IRP_operations_profile
getNotificationProfile[注]	get_EP_IRP_notification_profile
注：这3个操作是从ManagedGenericIRP中继承下来的操作	

4.1.2 操作参数映射

在《接口分析》中定义了访问入口点接口操作中各个参数的语义，表2~7中逐个给出了操作中各个参数到基于CORBA/IDL技术接口设计的映射。

表2 getIRPOutline 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
iRPVersion	ManagedGenericIRPConstDefs::VersionNumber iRPVersion
supportedIRPList	返回参数类型SupportedIRPListType supportedIRPList
result	EPIRPSystem::ResultType 异常：GetIRPOutline，InvalidIRPVersion

表3 getIRPReference 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
managerIdentifier	EPIRPSystem::ManagerIdentifierType managerIdentifier
systemDN	EPIRPSystem::SystemDNType systemDn
iRPId	EPIRPSystem::IRPIdType iRPId
iRPReference	返回参数类型string iRPReference，其值为字符串形式的IOR
result	EPIRPSystem::ResultType， 异常：GetIRPReference, InvalidRequestedParameters

表4 releaseIRPReference 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
managerIdentifier	EPIRPSystem::ManagerIdentifierType managerReference
iRPReference	string iRPReference，其值为字符串形式的IOR
result	EPIRPSystem::ResultType， 异常：ReleaseIRPReference,UnknownIRPReference

表5 getIRPVersion 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
versionNumberSet	返回参数类型 ManagedGenericIRPConstDefs::VersionNumberSet
result	异常： GetEPIRPVersions

表6 getOperationProfile 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
iRPVersion	ManagedGenericIRPConstDefs::VersionNumber iRPVersion
operationNameProfile, operationParameterProfile	返回参数类型 ManagedGenericIRPConstDefs::MethodList
result	异常： GetEPIRPOperationsProfile， ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

表7 getNotificationProfile 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
iRPVersion	ManagedGenericIRPConstDefs::VersionNumber iRPVersion
notificationNameProfile, notificationParameterProfile	返回参数类型 ManagedGenericIRPConstDefs::MethodList
result	异常： GetEPIRPNotificationProfile, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

4.1.3 通知参数映射

在《接口分析》中定义了访问入口点接口通知中各个参数的语义，表8中逐个给出了通知中各个参数到基于CORBA/IDL技术接口设计的映射。

表8 NotifyIRPInfoChanges 通知参数映射表

分析阶段通知参数	OMG CORBA 结构事件参数映射	说明
分析阶段没有相应参数	域名（domain_name）	字符串，为相应IRP信息模型的版本号
notificationType	类型名（type_name）	上报的通知的类型，即入口点中IRP信息改变通知（notifyIRPInfoChanges）
--	事件名（event_name）	应置成空串
分析阶段没有相应参数	可变事件头（variable Header）	
objectClass, objectInstance	可过滤体域（filterable_body_fields）的一个名值对	发出通知的管理对象类和对象实例，即EPIRP对象类的对象实例
notificationId	可过滤体域（filterable_body_fields）的一个名值对	通知标识符，用来唯一标识通知，可进行通知的关联。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: NOTIFICATION_ID。 名值对的值为长整型long
eventTime	可过滤体域（filterable_body_fields）的一个名值对	指明事件发生时间。使用 OMG定义的UtcT [参见附录B]类型。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue::EVENT_TIME
systemDN	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串 NotificationIRPConstDefs::AttributeNameValue:: SYSTEM_DN。 名值对的值为字符串string，表示产生通知的被管系统（即IRPAgent）标识
iRPDN	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串EPIRPSystem:: AttributeNameValue::IRP_DN。 名值对的值为EPIRPSystem:: IRPDnType
changeMode	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串EPIRPSystem:: AttributeNameValue::CHANGE_MODE。 名值对的值为EPIRPSystem:: ChangeModeType
additionalText	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： EPIRPSystem::AttributeNameValue:: ADDITIONAL_TEXT 名值对的值为字符串
分析阶段没有相应参数	保留体（remaining_ body）	

4.2 通知管理接口设计

4.2.1 通知定购管理接口设计

4.2.1.1 操作映射

在《接口分析》中定义了通知定购管理接口操作的语义，表9给出了这些操作到基于CORBA/IDL技术接口设计的映射。

表9 操作映射表

分析阶段操作	设计阶段操作映射
subscribe	attach_push，attach_push_b，attach_pull
unsubscribe	detach
suspendSubscription	如果采用attach_push方式建立定购，此操作无映射，即IRPManager不能挂起定购。 如果采用attach_push_b方式建立定购，此操作映射为suspend_connection，其IDL定义不在本文档中，参见OMG通知服务，且如果IRPAgent支持可选attach_push_b方式，则必须支持此操作。 如果采用attach_pull方式建立定购，此操作无映射
resumeSubscription	如果采用attach_push方式建立定购，此操作无映射，即IRPManager不能挂起定购。 如果采用attach_push_b方式建立定购，此操作映射为resume_connection，其IDL定义不在本文档中，参见OMG通知服务，且如果IRPAgent支持可选attach_push_b方式，则必须支持此操作。 如果采用attach_pull方式建立定购，此操作无映射
changeSubscriptionFilter	如果采用attach_push方式建立定购，此操作映射为change_subscription_filter，其IDL定义参见附录A，且此操作为可选。 如果采用attach_push_b方式建立定购，此操作映射为modify_constraints，其IDL定义不在本文档中，参见OMG通知服务，且如果IRPAgent支持可选attach_push_b方式，则必须支持此操作。 如果采用attach_pull方式建立定购，此操作映射为modify_constraints，其IDL定义不在本文档中，参见OMG通知服务，且如果IRPAgent支持可选attach_pull方式，则必须支持此操作
getSubscriptionIds	get_subscription_ids
getSubscriptionStatus	get_subscription_status
getNotificationCategories	get_notification_categories
getIRPVersion [注]	get_notification_IRP_versions
getOperationProfile [注]	get_notification_IRP_operations_profile
getNotificationProfile [注]	get_notification_IRP_notification_profile
注：这3个操作是从ManagedGenericIRP中继承下来的操作	

4.2.1.2 操作参数映射

在《接口分析》中定义了通知定购管理接口操作中各个参数的语义，表10~22中逐个给出了操作中各个参数到基于CORBA/IDL技术接口设计的映射。

表10 subscribe 操作到 attach_push 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
managerReference	string manager_reference [注1]，其值为字符串形式的IOR
timeTick	unsigned long time_tick
notificationCategorySet	NotificationIRPConstDefs::NotificationCategorySet notification_categories
filter	ManagedGenericIRPConstDefs::StringTypeOpt filter [注2]

表10（续）

分析阶段操作参数	设计阶段操作参数映射
subscriptionId	返回参数类型NotificationIRPConstDefs::SubscriptionId
result	异常： Attach, ManagedGenericIRPSystem::ParameterNotSupported, ManagedGenericIRPSystem::InvalidParameter, AlreadySubscribed, AtLeastOneNotificationCategoryNotSupported
注1：IRPManager创建一个CosNotifyComm::SequencePushConsumer对象，调用CORBA::ORB::object_to_string操作将其转换为IOR字符串，然后将其作为attach_push_b的输入参数。IRPAgent接收到此参数后存储供后续操作使用。 此IOR字符串也可以作为IRPManager调用detach操作时的比较依据。 注2：过滤条件的语法是扩展的TCL，参见OMG通知服务[3]。此处的设计部分以及告警AlarmIRP的设计部分都只用此语法	

表11 subscribe 操作到 attach_push_b 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
managerReference	string manager_reference [注1]
timeTick	unsigned long time_tick
notification Categories	NotificationIRPConstDefs::NotificationCategorySet notification_categories
filter	ManagedGenericIRPConstDefs::StringTypeOpt filter [注2]
subscriptionId	返回参数类型NotificationIRPConstDefs::SubscriptionId
无	CosNotifyChannelAdmin::SequenceProxyPushSupplier system_reference [注3]
result	异常： Attach, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::ParameterNotSupported, ManagedGenericIRPSystem::InvalidParameter, AlreadySubscribed, AtLeastOneNotificationCategoryNotSupported
注1：IRPManager创建一个CosNotifyComm::SequencePushConsumer对象，调用CORBA::ORB::object_to_string操作将其转换为IOR字符串，然后将其作为attach_push_b的输入参数。IRPAgent接收到此参数后存储供后续操作使用。 此IOR字符串也可以作为IRPManager调用detach操作时的比较依据。 注2：过滤条件的语法是扩展的TCL，参见OMG通知服务[3]。此处的设计部分以及告警AlarmIRP的设计部分都只用此语法。 注3：IRPAgent提供此引用供IRPManager对定购进行管理，相关的操作不在此IRP中定义，而是在OMG通知服务中定义。参见CosNotifyChannelAdmin::SequenceProxyPushSupplier以及CosNotifyComm::SequencePushConsumer接口定义。IRPManager用此引用调用这些接口的connect_sequence_push_consumer操作与其cosNotifyComm::SequencePushConsummer连接，连接成功后，IRPAgent将系列通知push到IRPManager	

表12 subscribe 操作到 attach_pull 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
managerReference	string manager_reference [注1] ，其值为字符串形式的IOR
timeTick	unsigned long time_tick
notification Categories	NotificationIRPConstDefs::NotificationCategorySet notification_categories
filter	ManagedGenericIRPConstDefs::StringTypeOpt filter [注2]
subscriptionId	返回参数类型NotificationIRPConstDefs::SubscriptionId
无	CosNotifyChannelAdmin::SequenceProxyPullSupplier system_reference [注3]
result	异常： Attach, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::ParameterNotSupported, ManagedGenericIRPSystem::InvalidParameter, AlreadySubscribed, AtLeastOneNotificationCategoryNotSupported

表12（续）

分析阶段操作参数	设计阶段操作参数映射
注1：IRPManager创建一个CosNotifyComm::SequencePushConsumer对象，调用CORBA::ORB::object_to_string操作将其转换为IOR字符串，然后将其作为attach_push_b的输入参数。IRPAgent接收到此参数后存储供后续操作使用。 此IOR字符串也可以作为IRPManager调用detach操作时的比较依据。 注2：过滤条件的语法是扩展的TCL，参见OMG通知服务[3]。此处的设计部分以及告警AlarmIRP的设计部分都只用此语法。 注3：IRPAgent提供此引用供IRPManager对定购进行管理，相关的操作不在此IRP中定义，而是在OMG通知服务中定义。参见CosNotifyChannelAdmin::SequenceProxyPushSupplier以及CosNotifyComm::SequencePushConsumer接口定义。IRPManager用此引用调用这些接口的connect_sequence_push_consumer操作与其cosNotifyComm::SequencePushConsummer连接，连接成功后，IRPAgent将系列通知push到IRPManager	

表13 unsubscribe 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
managerReference	string manager_reference，其值为字符串形式的IOR
subscriptionId	NotificationIRPConstDefs::SubscriptionId subscription_id
result	异常： DetachException, ManagedGenericIRPSystem::ParameterNotSupported, ManagedGenericIRPSystem::InvalidParameter

表14 getIRPVersion 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
versionNumberSet	返回参数类型 ManagedGenericIRPConstDefs::VersionNumberSet
result	异常： GetNotificationIRPVersions

表15 suspendSubscription 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
subscriptionId	如果采用attach_push方式建立定购，此操作无映射，因此无参数映射。 如果采用attach_push_b方式建立定购，此操作映射为suspend_connection，其IDL定义参见OMG通知服务，且不需要参数，因此无参数映射。 如果采用attach_pull方式建立定购，此操作无映射，因此无参数映射
result	如果采用attach_push方式建立定购，此操作无映射，因此无参数映射。 如果采用attach_push_b方式建立定购，此操作映射为suspend_connection，其IDL定义参见OMG通知服务，且返回为空，因此无参数映射。suspend_connection操作可以返回OMG通知服务定义的异常ConnectionAlreadyInactive。 如果采用attach_pull方式建立定购，此操作无映射，因此无参数映射

表16 resumeSubscription 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
subscriptionId	如果采用attach_push方式建立定购，此操作无映射，因此无参数映射。 如果采用attach_push_b方式建立定购，此操作映射为resume_connection，其IDL定义参见OMG通知服务，且不需要参数，因此无参数映射。 如果采用attach_pull方式建立定购，此操作无映射，因此无参数映射
result	如果采用attach_push方式建立定购，此操作无映射，因此无参数映射。 如果采用attach_push_b方式建立定购，此操作映射为resume_connection，其IDL定义参见OMG通知服务，且返回为空，因此无参数映射。resume_connection操作可以返回OMG通知服务定义的异常ConnectionAlreadyActive。 如果采用attach_pull方式建立定购，此操作无映射，因此无参数映射

表17 changeSubscriptionFilter 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
subscriptionId	NotificationIRPConstDefs::SubscriptionId subscription_id
filter	string filter
result	异常： ChangeSubscriptionFilter, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

表18 getSubscriptionIds 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
managerReference	string manager_reference
subscriptionIdSet	返回参数类型 NotificationIRPConstDefs::SubscriptionIdSet
result	异常： GetSubscriptionIds, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

表19 getSubscriptionStatus 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
subscriptionId	NotificationIRPConstDefs::SubscriptionId subscription_id
notificationCategorySet	返回参数类型 NotificationIRPConstDefs::NotificationCategorySet
filterInEffect	ManagedGenericIRPConstDefs::StringTypeOpt filter_in_effect
subscriptionState	NotificationIRPConstDef::SubscriptionState subscription_state
timeTick	unsigned long time_tick
result	异常： GetSubscriptionStatus, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

表20 getNotificationCategories 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
notificationCategorySet	返回参数类型 NotificationIRPConstDefs::NotificationCategorySet
无	NotificationIRPConstDefs::NotificationTypesSet notification_type_list（可选）
result	异常： GetNotificationCategories, ManagedGenericIRPSystem:: OperationNotSupported

表21 getOperationProfile 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
iRPVersion	ManagedGenericIRPConstDefs::VersionNumber notification_irp_version
operationNameProfile, operationParameterProfile	返回参数类型 ManagedGenericIRPConstDefs::MethodList
result	异常： GetNotificationIRPOperationsProfile, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

表22 getNotificationProfile 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
iRPVersion	ManagedGenericIRPConstDefs::VersionNumber notification_irp_version
notificationNameProfile, notificationParameterProfile	返回参数类型 ManagedGenericIRPConstDefs::MethodList
result	异常： GetNotificationIRPNotificationProfile, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

4.2.2 通知上报管理接口设计

4.2.2.1 通知头的参数映射

表23 通知头参数映射表

分析阶段通知头参数	设计阶段通知头参数映射
objectClass	在各个具体通知中映射
objectInstance	在各个具体通知中映射
notificationId	在各个具体通知中映射
eventTime	在各个具体通知中映射
systemDN	在各个具体通知中映射
notificationType	在各个具体通知中映射

4.3 链路监视功能接口设计

4.3.1 操作和通知映射

在《接口分析》中定义了链路监视功能接口操作和通知的语义，表24给出了这些操作和通知到基于CORBA/IDL技术接口设计的映射。

表24 操作/通知映射表

分析阶段操作/通知	设计阶段映射
getHeartbeatPeriod	get_heartbeat_period
setHeartbeatPeriod	set_heartbeat_period
triggerHeartbeat	trigger_heartbeat
getIRPVersion[注]	get_CS_IRP_versions
getOperationProfile[注]	get_CS_IRP_operations_profile
getNotificationProfile[注]	get_CS_IRP_notification_profile
notifyHeartbeat	push_structured_events
注：这三个操作是从ManagedGenericIRP中继承下来的操作	

4.3.2 操作参数映射

在《接口分析》中定义了链路监视功能接口操作中各个参数的语义，表25~30逐个给出了操作中各个参数到基于CORBA/IDL技术接口设计的映射。

表25 getHeartbeatPeriod 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
heartbeatPeriod	返回参数值 CSIRPConstDefs::HeartbeatPeriodType heartbeatPeriod
result	CSIRPConstDefs::ResultType 异常：GetHeartbeatPeriod

表26 setHeartbeatPeriod 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
heartbeatPeriod	CSIRPConstDefs::HeartbeatPeriodType heartbeatPeriod
result	CSIRPConstDefs::ResultType 异常：SetHeartbeatPeriod， InvalidHeartbeatPeriod，ConflictingHeartbeatPeriod

表27 triggerHeartbeat 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
managerIdentifier	CSIRPConstDefs:: managerIdentifier Type managerIdentifier
result	CSIRPConstDefs::ResultType 异常： TriggerHeartbeat，InvalidManagerIdentifier

表28 getIRPVersion 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
versionNumberSet	返回参数类型 ManagedGenericIRPConstDefs::VersionNumberSet
result	异常： GetCSIRPVersions

表29 getOperationProfile 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
iRPVersion	ManagedGenericIRPConstDefs::VersionNumber iRPVersion
operationNameProfile, operationParameterProfile	返回参数类型 ManagedGenericIRPConstDefs::MethodList
result	异常： GetCSIRPOperationsProfile, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

表30 getNotificationProfile 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
iRPVersion	ManagedGenericIRPConstDefs::VersionNumber iRPVersion
notificationNameProfile, notificationParameterProfile	返回参数类型 ManagedGenericIRPConstDefs::MethodList
result	异常： GetCSIRPNotificationProfile, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

4.3.3 通知参数映射

在《接口分析》中定义了链路监视功能接口通知中各个参数的语义，表31给出了通知中各个参数到基于CORBA/IDL技术接口设计的映射。

表31 NotifyHeartbeat通知参数映射表

分析阶段通知参数	OMG CORBA结构事件参数映射	说 明
分析阶段没有相应参数	域名（domain_name）	字符串，为相应IRP信息模型的版本号
notificationType	类型名（type_name）	上报的通知的类型，即心跳通知（notifyHeartbeat）
--	事件名（event_name）	应置成空串
分析阶段没有相应参数	可变事件头（variable Header）	
objectClass，objectInstance	可过滤体域（filterable_body_fields）的一个名值对	发出通知的管理对象类和对象实例，即CSIRP对象类的对象实例
notificationId	可过滤体域（filterable_body_fields）的一个名值对	通知标识符，用来唯一标识通知，可进行通知的关联。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: NOTIFICATION_ID。 名值对的值为长整型long
eventTime	可过滤体域（filterable_body_fields）的一个名值对	指明事件发生时间。使用 OMG定义的UtcT类型。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue::EVENT_TIME
systemDN	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串 NotificationIRPConstDefs:: AttributeNameValue::SYSTEM_DN。 名值对的值为字符串string，表示产生通知的被管系统（即IRPAgent）标识
heartbeatPeriod	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串CSIRPNotifDefs:: notifyHeartbeat::HEARTBEAT_PERIOD。 名值对的值为CSIRPConstDefs:: HeartbeatPeriodType，表示心跳时间间隔（由于IRPManager可能修改EMS心跳的间隔，该属性用来指明当前定义的时间间隔）
triggerFlag	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串 CSIRPNotifDefs::notifyHeartbeat::TRIGGER_FLAG。 名值对的值为CSIRPConstDefs:: TriggerFlagType，指明心跳通知的触发方
locator	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串CSIRPNotifDefs:: notifyHeartbeat::CHANNEL_ID 名值对的值为CSIRPConstDefs::ChannelIdType， 该参数映射为OMG通知服务中channel的标识。其取值可以等同于OMG中的Channel ID的取值，也可以不同
managerIdentifier	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串 CSIRPNotifDefs::notifyHeartbeat::MANAGER_IDENTIFIER 名值对的值为CSIRPSystem::ManagerIdentifierType，指明触发此通知的NMC
分析阶段没有相应参数	保留体（remaining_ body）	

4.4 通知日志管理接口设计

待补充。

4.5 文件传输接口设计

4.5.1 操作和通知映射

在《接口分析》中定义了文件传输接口功能的操作和通知的语义，表32给出了这些操作和通知到基于CORBA/IDL技术接口设计的映射。

表32 操作/通知映射表

分析阶段操作/通知	设计阶段映射
listAvailableFiles	list_available_files
notifyFileReady	push_structured_events
notifyFilePreparationError	push_structured_events
getIRPVersion[注]	get_FileTransfer_IRP_versions
getOperationProfile[注]	get_FileTransfer_IRP_operations_profile
getNotificationProfile [注]	get_FileTransfer_IRP_notifications_profile
注：这3个操作是从ManagedGenericIRP中继承下来的操作	

4.5.2 操作参数映射

在《接口分析》中定义了文件传输接口操作中各个参数的语义，表33~36逐个给出了操作中各个参数到基于CORBA/IDL技术接口设计的映射。

表33 listAvailableFiles 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
managementDataType	FileTransferIRPConstDefs::ManagementDataTypeType
beginTime	FileTransferIRPConstDefs::BeginTimeType
endTime	FileTransferIRPConstDefs::EndTimeType
fileInfoList	FileTransferIRPConstDefs::FileInfoListType
result	FileTransferIRPConstDefs::ResultType 异常：ListAvailableFiles, InvalidTimes

表34 getIRPVersion 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
versionNumberSet	返回参数类型 ManagedGenericIRPConstDefs::VersionNumberSet
result	异常： GetFileTransferIRPVersions

表35 getOperationProfile 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
iRPVersion	ManagedGenericIRPConstDefs::VersionNumber iRPVersion
operationNameProfile, operationParameterProfile	返回参数类型 ManagedGenericIRPConstDefs::MethodList
result	异常： GetFileTransferIRPOperationsProfile, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

表36 getNotificationProfile 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
iRPVersion	ManagedGenericIRPConstDefs::VersionNumber iRPVersion
notificationNameProfile, notificationParameterProfile	返回参数类型 ManagedGenericIRPConstDefs::MethodList
result	异常： GetFileTransferIRPNotificationsProfile, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

4.5.3 通知参数映射

在《接口分析》中定义了文件传输接口通知中各个参数的语义，表37和38中逐个给出了通知中各个参数到基于CORBA/IDL技术接口设计的映射。

表37 NotifyFileReady 通知参数映射表

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
分析阶段没有相应参数	域名（domain_name）	字符串，为相应IRP信息模型的版本号
notificationType	类型名（type_name）	上报的通知的类型，即文件准备好通知（notifyFileReady）
--	事件名（event_name）	应置成空串
分析阶段没有相应参数	可变事件头（variable Header）	
objectClass, objectInstance	可过滤体域（filterable_body_fields）的一个名值对	发出通知的管理对象类和对象实例，即FileTransferIRP对象类的对象实例
notificationId	可过滤体域（filterable_body_fields）的一个名值对	通知标识符，用来唯一标识通知，可进行通知的关联。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: NOTIFICATION_ID。 名值对的值为长整型long
eventTime	可过滤体域（filterable_body_fields）的一个名值对	指明事件发生时间。使用 OMG定义的UtcT类型。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue::EVENT_TIME
systemDN	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串 NotificationIRPConstDefs:: AttributeNameValue::SYSTEM_DN。 名值对的值为字符串string，表示产生通知的被管系统（即IRPAgent）标识
fileInfoList	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串 FileTransferIRPNotifications:: notifyFileReady::FILE_INFO_LIST。 名值对的值为FileTransferIRPConstDefs:: FileInfoListType，指明已经准备好的数据文件的地址和名称信息列表
additionalText	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： FileTransferIRPNotifications:: notifyFileReady::ADDITIONAL_TEXT 名值对的值为字符串
分析阶段没有相应参数	保留体（remaining_ body）	

表38 NotifyFilePreparationError通知参数映射表

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
分析阶段没有相应参数	域名（domain_name）	字符串，为相应IRP信息模型的版本号
notificationType	类型名（type_name）	上报的通知的类型，即文件准备异常通知（notifyFilePreparationError）
--	事件名（event_name）	应置成空串
分析阶段没有相应参数	可变事件头（variable Header）	
objectClass，objectInstance	可过滤体域（filterable_body_fields）的一个名值对	发出通知的管理对象类和对象实例，即FileTransferIRP对象类的对象实例
notificationId	可过滤体域（filterable_body_fields）的一个名值对	通知标识符，用来唯一标识通知，可进行通知的关联。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: NOTIFICATION_ID。 名值对的值为长整型long
eventTime	可过滤体域（filterable_body_fields）的一个名值对	指明事件发生时间。使用 OMG定义的UtcT类型。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue::EVENT_TIME
systemDN	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串 NotificationIRPConstDefs:: AttributeNameValue::SYSTEM_DN。 名值对的值为字符串string，表示产生通知的被管系统（即IRPAgent）标识
fileInfoList	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串FileTransferIRPNotifications:: notifyFilePreparationError:: FILE_INFO_LIST。 名值对的值为FileTransferIRPConstDefs:: FileInfoListType，指明已经准备好的数据文件的地址和名称信息列表
reason	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串FileTransferIRPNotifications:: notifyFilePreparationError::REASON。 名值对的值为string
additionalText	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： FileTransferIRPNotifications::notifyFilePreparationError:: ADDITIONAL_TEXT 名值对的值为字符串
分析阶段没有相应参数	保留体（remaining_ body）	

5 配置管理接口设计

下面给出了配置管理接口分析到基于CORBA/IDL技术接口设计的映射表格。

5.1 公共配置管理接口设计

5.1.1 操作和通知映射

在《接口分析》中定义了公共配置功能管理对象类KernelCMIRP及其操作和通知的语义，表39给出了这些操作和通知到基于CORBA/IDL技术接口设计的映射。

表39 操作映射表

分析阶段操作/通知	设计阶段映射
getNRMIRPVersion	get_NRM_IRP_version
getIRPVersion [注]	get_kernel_CM_IRP_versions
getOperationProfile [注]	get_kernel_CM_IRP_operation_profile
getNotificationProfile [注]	get_kernel_CM_IRP_notification_profile
notifyObjectCreation	push_structured_events
notifyObjectDeletion	push_structured_events
notifyAttributeValueChange	push_structured_events
notifyStateChange	push_structured_events
notifyCMSynchronizationRecommended	push_structured_events
注：这3个操作是从ManagedGenericIRP中继承下来的操作	

5.1.2 操作参数映射

在《接口分析》中定义了操作中各个参数的语义，表40~43逐个给出了操作中各个参数到基于CORBA/IDL技术接口设计的映射。

表40 getNRMIRPVersion 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
versionNumberList	返回值类型 ManagedGenericIRPConstDefs::VersionNumberSet
vSEVersionNumberList	返回值类型 ManagedGenericIRPConstDefs::VersionNumberSet
result	异常： GetNRMIRPVersion

表41 getIRPVersion 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
versionNumberList	返回值类型 ManagedGenericIRPConstDefs::VersionNumberSet
result	异常：GetKernelCMIRPVersionsException

表42 getOperationProfile 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
irpVersion	ManagedGenericIRPConstDefs::VersionNumber kernel_CM_IRP_version
operationNameProfile, operationParameterProfile	返回值类型ManagedGenericIRPConstDefs::MethodList
result	异常： GetKernelCMIRPOperationProfileException, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

表43 getNotificationProfile 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
IrpVersion	ManagedGenericIRPConstDefs::VersionNumber kernel_CM_IRP_version
notificationNameProfile, notificationParameterProfile	返回值类型ManagedGenericIRPConstDefs::MethodList
result	异常： GetKernelCMIRPNotificationProfileException, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

5.1.3 通知参数映射

在《接口分析》中定义了公共配置功能接口通知中各个参数的语义，表44~48中逐个给出了通知中各个参数到基于CORBA/IDL技术接口设计的映射。

表44 notifyObjectCreation 通知参数映射表

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
分析阶段没有相应参数	域名（domain_name）	字符串，为相应IRP信息模型的版本号
notificationType	类型名（type_name）	上报的通知的类型，即对象创建通知（notifyObjectCreation）
--	事件名（event_name）	应置成空串
分析阶段没有相应参数	可变事件头（variable Header）	
objectClass, objectInstance	可过滤体域（filterable_body_fields）的一个名值对	发出通知的网络资源对象以及PMIRP对象类和对象实例。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: MANAGED_OBJECT_INSTANCE。 名值对的值为字符串，包括了对象类和对象实例的值
notificationId	可过滤体域（filterable_body_fields）的一个名值对	通知标识符，用来唯一标识通知，可进行通知的关联。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: NOTIFICATION_ID。 名值对的值为长整型long
eventTime	可过滤体域（filterable_body_fields）的一个名值对	指明事件发生时间。使用 OMG定义的UtcT类型。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue::EVENT_TIME
systemDN	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： NotificationIRPConstDefs:: AttributeNameValue:: SYSTEM_DN。 名值对的值为字符串string，表示产生通知的被管系统（即IRPAgent）标识
correlatedNotifications	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： KernelCmNotifDefs::NotificationCommon:: CORRELATED_NOTIFICATIONS 名值对的值为 KernelCmNotifDefs::CorrelatedNotificationSetType

表44（续）

分析阶段通知参数	OMG CORBA 结构事件参数映射	说　明
additionalText	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： KernelCmNotifDefs:: NotificationCommon:: ADDITIONAL_TEXT 名值对的值为字符串
sourceIndicator	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： KernelCmNotifDefs:: NotificationCommon:: SOURCE_INDICATOR 名值对的值为字符串，其取值为 KernelCmNotifDefs:: NotificationCommon:: RESOURCE_OPERATION 或 KernelCmNotifDefs:: NotificationCommon:: MANAGEMENT_OPERATION 或 KernelCmNotifDefs:: NotificationCommon:: UNKNOWN_OPERATION
attributeList	保留体（remaining_ body）	KernelCMNotifDefs::MOCreation::InitialAttributeValues (在保留体中)

表45　notifyObjectDeletion 通知参数映射表

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
分析阶段没有相应参数	域名（domain_name）	字符串，为相应IRP信息模型的版本号
notificationType	类型名（type_name）	上报的通知的类型，即对象删除通知（notifyObjectDeletion）
--	事件名（event_name）	应置成空串
分析阶段没有相应参数	可变事件头（variable Header）	
objectClass， objectInstance	可过滤体域（filterable_body_fields）的一个名值对	发出通知的网络资源对象以及PMIRP对象类和对象实例。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: MANAGED_OBJECT_INSTANCE 名值对的值为字符串，包括了对象类和对象实例的值
notificationId	可过滤体域（filterable_body_fields）的一个名值对	通知标识符，用来唯一标识通知，可进行通知的关联。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: NOTIFICATION_ID。 名值对的值为长整型long
eventTime	可过滤体域（filterable_body_fields）的一个名值对	指明事件发生时间。使用OMG定义的UtcT类型。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue::EVENT_TIME
systemDN	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： NotificationIRPConstDefs:: AttributeNameValue:: SYSTEM_DN。 名值对的值为字符串string，表示产生通知的被管系统（即IRPAgent）标识

表45（续）

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
correlatedNotifications	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： KernelCmNotifDefs:: NotificationCommon:: CORRELATED_NOTIFICATIONS 名值对的值为 KernelCmNotifDefs::CorrelatedNotificationSetType
additionalText	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： KernelCmNotifDefs:: NotificationCommon:: ADDITIONAL_TEXT 名值对的值为字符串
sourceIndicator	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： KernelCmNotifDefs:: NotificationCommon:: SOURCE_INDICATOR 名值对的值为字符串，其取值为 KernelCmNotifDefs:: NotificationCommon:: RESOURCE_OPERATION 或 KernelCmNotifDefs:: NotificationCommon:: MANAGEMENT_OPERATION 或 KernelCmNotifDefs:: NotificationCommon:: UNKNOWN_OPERATION
attributeList	保留体（remaining_ body）	KernelCmNotifDefs::MODeletion::AttributeValues (在保留体中)

表46　notifyAttributeValueChange 通知参数映射表

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
分析阶段没有相应参数	域名（domain_name）	字符串，为相应IRP信息模型的版本号
notificationType	类型名（type_name）	上报的通知的类型，即对象属性值改变通知（notifyAttributeValueChange）
--	事件名（event_name）	应置成空串
分析阶段没有相应参数	可变事件头（variable Header）	
objectClass， objectInstance	可过滤体域（filterable_body_fields）的一个名值对	发出通知的网络资源对象类和对象实例。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: MANAGED_OBJECT_INSTANCE。 名值对的值为字符串，包括了对象类和对象实例的值
notificationId	可过滤体域（filterable_body_fields）的一个名值对	通知标识符，用来唯一标识通知，可进行通知的关联。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: NOTIFICATION_ID。 名值对的值为长整型long
eventTime	可过滤体域（filterable_body_fields）的一个名值对	指明事件发生时间。使用 OMG定义的UtcT类型。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue::EVENT_TIME

表46（续）

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
systemDN	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： NotificationIRPConstDefs:: AttributeNameValue:: SYSTEM_DN。 名值对的值为字符串string，表示产生通知的被管系统（即IRPAgent）标识
correlatedNotifications	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： KernelCmNotifDefs:: NotificationCommon:: CORRELATED_NOTIFICATIONS 名值对的值为 NotificationIRPConstDefs::CorrelatedNotificationSetType
additionalText	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： KernelCmNotifDefs:: NotificationCommon:: ADDITIONAL_TEXT 名值对的值为字符串
sourceIndicator	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： KernelCmNotifDefs:: NotificationCommon:: SOURCE_INDICATOR 名值对的值为字符串，其取值为 KernelCmNotifDefs:: NotificationCommon:: RESOURCE_OPERATION 或 KernelCmNotifDefs:: NotificationCommon:: MANAGEMENT_OPERATION 或 KernelCmNotifDefs:: NotificationCommon:: UNKNOWN_OPERATION
attributeValue ChangeDefinition	保留体（remaining_ body）	KernelCmNotifDefs::AttributeValueChange:: ModifiedAttributeSet (在保留体中)

表47　notifyRequestCMSynchronization 通知参数映射表

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
分析阶段没有相应参数	域名（domain_name）	字符串，为相应IRP信息模型的版本号
notificationType	类型名（type_name）	上报的通知的类型，即请求配置信息同步通知（notifyCMSynchronizationRecommended）
--	事件名（event_name）	应置成空串
分析阶段没有相应参数	可变事件头（variable Header）	
objectClass， objectInstance	可过滤体域（filterable_body_fields）的一个名值对	发出通知的管理对象类和对象实例，即KernelCMIRP的对象类和对象实例。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: MANAGED_OBJECT_INSTANCE 名值对的值为字符串，包括了对象类和对象实例的值

表47（续）

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
notificationId	可过滤体域（filterable_body_fields）的一个名值对	通知标识符，用来唯一标识通知，可进行通知的关联。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: NOTIFICATION_ID。 名值对的值为长整型long
eventTime	可过滤体域（filterable_body_fields）的一个名值对	指明事件发生时间。使用 OMG定义的UtcT类型。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue::EVENT_TIME
systemDN	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： NotificationIRPConstDefs:: AttributeNameValue:: SYSTEM_DN。 名值对的值为字符串string，表示产生通知的被管系统（即IRPAgent）标识
baseMOClass	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： KernelCmNotifDefs::CMSynchronization Recommended::BASE_MO_CLASS. 名值对的值为字符串，为基类的类名
baseMOInstance	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： KernelCmNotifDefs::CMSynchronization Recommended::BASE_MO_INSTANCE 名值对的值为字符串，为基类实例的DN串
scope	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： KernelCmNotifDefs::CMSynchronization Recommended::SCOPE 名值对的值为KernelCmNotifDefs::ScopePara类型
additionalText	保留体（remaining_ body）	名值对的名称为字符串： KernelCmNotifDefs:: NotificationCommon:: ADDITIONAL_TEXT 名值对的值为字符串

表48　notifyStateChange 通知参数映射表

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
分析阶段没有相应参数	域名（domain_name）	字符串，为相应IRP信息模型的版本号
notificationType	类型名（type_name）	上报的通知的类型，即 对象属性值改变通知（notifyStateChange）
--	事件名（event_name）	应置成空串
分析阶段没有相应参数	可变事件头（variable Header）	
objectClass， objectInstance	可过滤体域（filterable_body_fields）的一个名值对	发出通知的管理对象类和对象实例。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: MANAGED_OBJECT_INSTANCE。 名值对的值为字符串，包括了对象类和对象实例的值

表48（续）

分析阶段通知参数	OMG CORBA结构事件参数映射	说明
notificationId	可过滤体域（filterable_body_fields）的一个名值对	通知标识符，用来唯一标识通知，可进行通知的关联。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: NOTIFICATION_ID。 名值对的值为长整型long
eventTime	可过滤体域（filterable_body_fields）的一个名值对	指明事件发生时间。使用 OMG定义的UtcT类型。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: EVENT_TIME 名值对的值为ManagedGenericIRPConstDefs::IRPTime
systemDN	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： NotificationIRPConstDefs:: AttributeNameValue:: SYSTEM_DN。 名值对的值为字符串string，表示产生通知的被管系统（即IRPAgent）标识
correlatedNotifications	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： KernelCmNotifDefs:: NotificationCommon:: CORRELATED_NOTIFICATIONS 名值对的值为 KernelCmNotifDefs::CorrelatedNotificationSetType
additionalText	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： KernelCmNotifDefs:: NotificationCommon:: ADDITIONAL_TEXT 名值对的值为字符串
sourceIndicator	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串： KernelCmNotifDefs:: NotificationCommon:: SOURCE_INDICATOR 名值对的值为字符串，其取值为 KernelCmNotifDefs:: NotificationCommon:: RESOURCE_OPERATION 或 KernelCmNotifDefs:: NotificationCommon:: MANAGEMENT_OPERATION 或 KernelCmNotifDefs:: NotificationCommon:: UNKNOWN_OPERATION
stateChange	保留体（remaining_body）	KernelCmNotifDefs::AttributeValueChange::ModifiedAttributeSet (在保留体中)

5.2 基本配置管理接口设计

5.2.1 操作和通知映射

在《接口分析》中定义了基本配置功能管理对象类BasicCMIRP 及其操作和通知的语义，管理对象类BasicCMIRP 由接口 BasicCmIrpOperations和BasicCmInformationIterator共同实现。

表49给出了这些操作和通知到基于CORBA/IDL技术接口设计的映射。

表49 操作映射表

分析阶段操作/通知	设计阶段映射
getMOAttributes	BasicCmIrpOperations::find_managed_objects BasicCmInformationIterator::next_basicCmInformations [注2]
getContainment	BasicCmIrpOperations::find_managed_objects BasicCmInformationIterator::next_basicCmInformations [注2]
createMO	BasicCmIrpOperations::create_managed_object
deleteMO	BasicCmIrpOperations::delete_managed_objects
setMOAttributes	BasicCmIrpOperations::modify_managed_objects
cancelOperation	BasicCmInformationIterator::destroy [注2]
getIRPVersion [注1]	get_basicCm_IRP_version
getOperationProfile [注1]	get_basicCm_IRP_operation_profile
getNotificationProfile [注1]	get_basicCm_IRP_notification_profile
注1：这3个操作是从ManagedGenericIRP中继承下来的操作。 注2：Iterator方式通常用来分段返回数据量较大的结果。每次返回1到how_many个对象信息。how_many为非0数值。其操作返回值为布尔型，TRUE表示后续还有结果返回，FALSE表示所有结果都已经返回。Iterator的destory操作意味着Agent回收Iterator资源，此后Iterator对Manager将不再可见。对于Iterator的使用方法全文相同	

5.2.2 操作参数映射

在《接口分析》中定义了操作中各个参数的语义，表50~58中逐个给出了操作中各个参数到基于CORBA/IDL技术接口设计的映射。

表50 getMOAttributes 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
invokeIdentifierIn	--
invokeIdentifierOut	返回值类型 BasicCmInformationIterator
baseObjectInstance	in DN baseObject
scope	in SearchControlType searchControl (SearchControlType.type和SearchControlType.level)
filter	in SearchControlType searchControl (SearchControlType.filter)
attributeListIn	in AttributeNameSet requestedAttributes
getListOut	返回值类型 BasicCmInformationIterator 方法 next_basicCmInformations 中的返回参数类型 ResultSet fetchedElements
result	异常: FindManagedObjects, ManagedGenericIRPSystem::ParameterNotSupported, ManagedGenericIRPSystem::InvalidParameter, ManagedGenericIRPSystem::ValueNotSupported, UndefinedMOException, IllegalDNFormatException, UndefinedScopeException, IllegalScopeTypeException, IllegalScopeLevelException, IllegalFilterFormatException, FilterComplexityLimit

表51 getContainment 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
invokeIdentifierIn	--
invokeIdentifierOut	返回值类型 BasicCmInformationIterator
baseObjectInstance	in DN baseObject
scope	in SearchControlType searchControl (SearchControlType.type 和SearchControlType.level)
分析阶段无参数	in SearchControlType searchControl (SearchControlType.filter)
containment	返回值类型 BasicCmInformationIterator 方法 next_basicCmInformations 中的返回参数类型 ResultSet fetchedElements
result	异常： FindManagedObjects, ManagedGenericIRPSystem::ParameterNotSupported, ManagedGenericIRPSystem::InvalidParameter, ManagedGenericIRPSystem::ValueNotSupported, UndefinedMOException, IllegalDNFormatException, UndefinedScopeException, IllegalScopeTypeException, IllegalScopeLevelException, IllegalFilterFormatException, FilterComplexityLimit

表52 createMO 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
managedObjectClass, managedObjectInstance	in DN objectName
referenceObjectInstance	in DN referenceObject
attributeListIn attributeListOut	inout MOAttributeSet attributes
result	out AttributeErrorSeq attributeErrors 异常： CreateManagedObject, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::ParameterNotSupported, ManagedGenericIRPSystem::InvalidParameter, UndefinedMOException, IllegalDNFormatException, DuplicateMO, CreateNotAllowed, ObjectClassMismatch, NoSuchObjectClass ParentObjectNotExists

表53 deleteMO 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
baseObjectInstance	in DN baseObject
scope	in SearchControlType searchControl (SearchControlType.type 和SearchControlType.level)
filter	in SearchControlType searchControl (SearchControlType.filter)
deletionList	返回值类型 BasicCmInformationIterator 方法 next_basicCmInformations 中的返回参数类型 ResultSet fetchedElements
result	返回值类型 DeleteResultIterator 方法 next_deleteErrors 中的返回参数类型 DeleteErrorSeq fetchedDeleteErrors 异常: DeleteManagedObjects, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter, UndefinedMoException, IllegalDNFormatException, UndefinedScopeException, IllegalScopeTypeException, IllegalScopeLevelException, IllegalFilterFormatException, FilterComplexityLimit

表54 setMOAttributes 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
baseObjectInstance	in DN baseObject
scope	in SearchControlType searchControl (SearchControlType.type 和SearchControlType.level)
filter	in SearchControlType searchControl (SearchControlType.filter)
modificationListIn	in AttributeModificationSet modifications
modificationListOut	返回值类型 ModifyResultIterator 方法 next_basicCmInformations 中的返回参数类型ResultSet fetchedElements
result	返回值类型 ModifyResultIterator 方法 ModifyResultIterator 中的返回参数类型 ModifyAttributeErrorsSeq fetchedModifyErrors 异常: ModifyManagedObjects, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter, UndefinedMoException, IllegalDNFormatException, UndefinedScopeException, IllegalScopeTypeException, IllegalScopeLevelException, IllegalFilterFormatException, FilterComplexityLimit

表55 cancelOperation 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
invokeIdentifier	--
result	异常：DestroyException

表56 getIRPVersion 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
VersionNumberSet	返回值类型 ManagedGenericIRPConstDefs::VersionNumberSet
result	异常：GetBasicCmIRPVersion

表57 getOperationProfile 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
irpVersion	ManagedGenericIRPConstDefs::VersionNumber basicCm_IRP_version
operationNameProfile, operationParameterProfile	返回值类型ManagedGenericIRPConstDefs::MethodList
result	异常： GetBasicCmIRPOperationProfile, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

表58 getNotificationProfile 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
irpVersion	ManagedGenericIRPConstDefs::VersionNumber basicCm_IRP_version
notificationNameProfile, notificationParameterProfile	返回值类型ManagedGenericIRPConstDefs::MethodList
result	异常： GetBasicCmIRPNotificationProfile, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

5.2.3 通知参数映射

无。

6 故障管理接口设计

下面给出了故障管理接口分析到基于CORBA/IDL技术接口设计的映射表格。

6.1 操作和通知映射

在《接口分析》中定义了故障管理接口操作和通知的语义，表59给出了这些操作到基于CORBA/IDL技术接口设计的映射。

表59 操作映射表

分析阶段操作/通知	设计阶段映射
getAlarmList	get_alarm_list AlarmInformationIterator::next_alarmInformations[注1]
acknowledgeAlarms	acknowledge_alarms
unacknowledgeAlarms	unacknowledge_alarms
getIRPVersion [注2]	get_alarm_IRP_versions
getOperationProfile [注2]	get_alarm_IRP_operations_profile
getNotificationProfile [注2]	get_alarm_IRP_notification_profile
getAlarmCount	get_alarm_count
setComment	comment_alarms
clearAlarms	clear_alarms
notifyPotentialFaultyAlarmList	push_structured_events
notifyAlarmListRebuilt	push_structured_events
notifyAckStateChanged	push_structured_events
notifyComments	push_structured_events
notifyNewAlarm	push_structured_events
notifyClearedAlarm	push_structured_events
notifyChangedAlarm	push_structured_events
注 1：Iterator 方式通常用来分段返回数据量较大的结果。每次返回 1 到 how_many 个对象信息。how_many 为非 0 数值。其操作返回值为布尔型，TRUE 表示后续还有结果返回，FALSE 表示所有结果都已经返回。Iterator 的 destory 操作意味着 Agent 回收 Iterator 资源，此后 Iterator 对 Manager 将不再可见。对于 Iterator 的使用方法全文相同。 注 2：这三个操作是从 ManagedGenericIRP 中继承下来的操作	

6.2 操作参数映射

在《接口分析》中定义了故障管理接口操作中各个参数的语义，表60~69中逐个给出了操作中各个参数到基于CORBA/IDL技术接口设计的映射。

表60 getAlarmList 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
alarmAckState, filter	ManagedGenericIRPConstDefs::StringTypeOpt filter
baseObjectClass, baseObjectInstance	AlarmIRPConstDefs::DNTypeOpt base_object
分析阶段没有相应参数	boolean flag，用来标识返回的告警列表是通过调用返回值返回还是通过iterator返回
alarmInformationList	返回值类型AlarmIRPConstDefs::AlarmInformationSeq，alarmInformation通过eventBatch数据结构返回，其固定通知头中的域名（domain_name）、类型名（type_name）和事件名（event_name）3个域置空串 详细参数映射见下表
result	异常： GetAlarmList, ManagedGenericIRPSystem::ParameterNotSupported, ManagedGenericIRPSystem::InvalidParameter

表61 alarmInformationList 操作参数映射表

分析阶段操作参数	说　明
alarmId	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: ALARM_ID。 其值为字符串类型
objectClass， objectInstance	名值对的名字是一个字符串 NotificationIRPConstDefs::AttributeNameValue:: MANAGED_OBJECT_INSTANCE。 其值为字符串型
notificationId	名值对的名字是一个字符串 NotificationIRPConstDefs::AttributeNameValue:: NOTIFICATION_ID。 其值为long长整型
alarmRaisedTime	名值对的名字是一个字符串 NotificationIRPConstDefs::AttributeNameValue:: EVENT_TIME。 其值为OMG定义的UtcT类型
alarmClearedTime	名值对的名字是一个字符串 NotificationIRPConstDefs::AttributeNameValue:: EVENT_TIME。 其值为OMG定义的UtcT类型
alarmChangedTime	名值对的名字是一个字符串 NotificationIRPConstDefs::AttributeNameValue:: EVENT_TIME。 其值为OMG定义的UtcT类型
eventType	名值对的名字是一个字符串"event_name" 其值为字符串
probableCause	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: PROBABLE_CAUSE。 其值为短整型，由AlarmIRPConstDefs:: ProbableCause接口定义
perceivedSeverity	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: PERCEIVED_SEVERITY。 其值为短整型，由AlarmIRPConstDefs:: PerceivedSeverity定义
specificProblem	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: SPECIFIC_PROBLEM。 其值为字符串型
backedUpStatus	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: BACKED_UP_STATUS。 其值为boolean类型，由AlarmIRPConstDefs::BackedUpStatusType定义

表61（续）

分析阶段操作参数	说　明
backUpObject	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: BACKED_UP_OBJECT。 其值为string类型，携带back-up对象的DN
trendIndication	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: TREND_INDICATION。 其值为enum类型，由AlarmIRPConstDefs:: TrendIndicationType定义
thresholdInfo	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: THRESHOLD_INFO。 其值为AlarmIRPConstDefs::ThresholdInfoType类型
stateChangeDefinition	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: STATE_CHANGE_DEFINITION。 其值为AlarmIRPConstDefs::AttributeChangeSetType类型
monitoredAttributes	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: MONITORED_ATTRIBUTES。 其值为AlarmIRPConstDefs::AttributeSetType类型
proposedRepairActions	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: PROPOSED_REPAIR_ACTIONS。 其值为字符串类型
additionalText	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: ADDITIONAL_TEXT。 其值为字符串类型
additionalInformation	名值对的名字和值由厂家定义
ackTime	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: ACK_TIME。 其值为ManagedGenericIRPConstDefs::IRPTime类型
ackUserId	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: ACK_USER_ID。 其值为字符串类型
ackSystemId	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: ACK_SYSTEM_ID。 其值为字符串类型

表61（续）

分析阶段操作参数	说　明
ackState	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: ACK_STATE。 其值为短整型，在AlarmIRPConstDefs:: AckState 接口中定义
clearedUserId	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: CLEAR_USER_ID。 其值为字符串类型
clearedSystemId	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: CLEAR_SYSTEM_ID。 其值为字符串类型
serviceUser	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: SERVICE_USER。 其值为字符串类型
serviceProvider	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: SERVICE_PROVIDER。 其值为字符串类型
SecurityAlarmDetector	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: SECURITY_ALARM_DETECTOR。 其值为字符串类型

表62　acknowledgeAlarms 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
alarmInformationAnd SeverityReferenceList	AlarmIRPConstDefs::AlarmInformationIdAndSevSeq alarm_information_id_and_sev_list 其中，perceivedSeverity为可选 { alarmId - Mandatory; perceivedSeverity - Optional }
ackUserId	string ack_user_id
ackSystemId	ManagedGenericIRPConstDefs::StringTypeOpt ack_system_id
badAlarmInformation ReferenceList	AlarmIRPConstDefs::BadAcknowledgeAlarmInfoSeq bad_ack_alarm_info_list
Result	ManagedGenericIRPConstDefs::Signal 异常： AcknowledgeAlarms, ManagedGenericIRPSystem::ParameterNotSupported, ManagedGenericIRPSystem::InvalidParameter,

表63 unacknowledgeAlarms 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
alarmInformation ReferenceList	AlarmIRPConstDefs::AlarmInformationIdSeq alarm_information_id_list
ackUserId	string ack_user_id
ackSystemId	ManagedGenericIRPConstDefs::StringTypeOpt ack_system_id
badAlarmInformation ReferenceList	AlarmIRPConstDefs::BadAlarmInformationIdSeq bad_alarm_information_id_list
result	ManagedGenericIRPConstDefs::Signal 异常 UnacknowledgeAlarms, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::ParameterNotSupported, ManagedGenericIRPSystem::InvalidParameter

表64 getAlarmCount 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
alarmAckState, filter	ManagedGenericIRPConstDefs::StringTypeOpt filter
criticalCount, majorCount, minorCount, warningCount, indeterminateCount, clearedCount	long critical_count, long major_count, long minor_count, long warning_count, long indeterminate_count, long cleared_count
result	异常： GetAlarmCount, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::ParameterNotSupported, ManagedGenericIRPSystem::InvalidParameter

表65 getIRPVersion 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
versionNumberSet	返回值类型 ManagedGenericIRPConstDefs::VersionNumberSet
status	异常： GetAlarmIRPVersions

表66 setComment 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
alarmInformationReferenceList	AlarmIRPConstDefs::AlarmInformationIdSeq alarm_information_id_list
commentUserId	string comment_user_id
commentSystemId	ManagedGenericIRPConstDefs::StringTypeOpt comment_system_id
commentText	string comment_text
badAlarmInformation ReferenceList	AlarmIRPConstDefs::BadAlarmInformationIdSeq bad_alarm_information_id_list
result	ManagedGenericIRPConstDefs::Signal, 异常： CommentAlarms, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::ParameterNotSupported ManagedGenericIRPSystem::InvalidParameter

表67 getOperationProfile 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
irpVersion	ManagedGenericIRPConstDefs::VersionNumber alarm_irp_version
operationNameProfile, operationParameterProfile	返回值类型 ManagedGenericIRPConstDefs::MethodList
status	异常： GetAlarmIRPOperationsProfile, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

表68 getNotificationProfile 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
irpVersion	ManagedGenericIRPConstDefs::VersionNumber alarm_irp_version
notificationNameProfile, notificationParameterProfile	返回值类型 ManagedGenericIRPConstDefs::MethodList
result	异常： GetAlarmIRPNotificationProfile, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

表69 clearAlarms 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
alarmInformation ReferenceList	AlarmIRPConstDefs::AlarmInformationIdSeq alarm_information_id_list
clearUserId	string clear_user_id
clearSystemId	string clear_system_id
badAlarmInformation ReferenceList	AlarmIRPConstDefs::BadAlarmInformationIdSeq bad_alarm_information_id_list
result	ManagedGenericIRPConstDefs::Signal 异常： ClearAlarms, ManagedGenericIRPSystem::ParameterNotSupported, ManagedGenericIRPSystem::InvalidParameter

6.3 通知参数映射

在《接口分析》中定义了故障管理功能接口通知中各个参数的语义，表70~77中逐个给出了通知中各个参数到基于CORBA/IDL技术接口设计的映射。

表70 notifyNewAlarm 通知参数映射表（与安全无关告警）

分析阶段通知参数	OMG CORBA结构事件 参数映射	说明
分析阶段没有参数	域名（domain_name）	字符串，用来说明IRP的版本信息
notificationType	类型名（type_name）	字符串常量： AlarmIRPConstDefs:: NotificationType:: NOTIFY_FM_NEW_ALARM
alarmType	事件名（event_name）	AlarmIRPConstDefs::AlarmType中的“X1” ~ “X5”

表70（续）

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
分析阶段没有参数	可变事件头（variable Header）	
objectClass，objectInstance	可过滤体域（filterable_body_fields）的一个名值对	发出通知的网络资源对象类和对象实例。 名值对的名字是一个字符串 NotificationIRPConstDefs::AttributeNameValue:: MANAGED_OBJECT_INSTANCE。 其值为字符串型
notificationId	可过滤体域（filterable_body_fields）的一个名值对	通知标识符，用来唯一标识通知，可进行通知的关联。 名值对的名字是一个字符串 NotificationIRPConstDefs::AttributeNameValue:: NOTIFICATION_ID。 其值为long长整型
eventTime	可过滤体域（filterable_body_fields）的一个名值对	指明事件发生时间。使用 OMG定义的UtcT类型。 名值对的名称为字符串： 名值对的名字是一个字符串 NotificationIRPConstDefs::AttributeNameValue:: EVENT_TIME
systemDN	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 NotificationIRPConstDefs::AttributeNameValue:: SYSTEM_DN。 其值为字符串类型
probableCause	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: PROBABLE_CAUSE。 其值为短整型，由AlarmIRPConstDefs:: ProbableCause接口定义
perceivedSeverity	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: PERCEIVED_SEVERITY。 其值为短整型，由AlarmIRPConstDefs:: PerceivedSeverity接口定义
specificProblem	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: SPECIFIC_PROBLEM。 其值为字符串型
correlatedNotification	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: CORRELATED_NOTIFICATIONS。 其值为AlarmIRPConstDefs:: CorrelatedNotificationSetType类型
backedUpStatus	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: BACKED_UP_STATUS。 其值为boolean类型，由AlarmIRPConstDefs::BackedUpStatusType定义

表70（续）

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
backUpObject	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串AlarmIRPConstDefs::AttributeNameValue::BACKED_UP_OBJECT。 其值为string类型，携带back-up对象的DN
trendIndication	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串AlarmIRPConstDefs::AttributeNameValue::TREND_INDICATION。 其值为enum类型，由AlarmIRPConstDefs:: TrendIndicationType定义
thresholdInfo	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串AlarmIRPConstDefs::AttributeNameValue::THRESHOLD_INFO。 其值为AlarmIRPConstDefs::ThresholdInfoType类型
stateChangeDefinition	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串AlarmIRPConstDefs::AttributeNameValue::STATE_CHANGE_DEFINITION。 其值为AlarmIRPConstDefs::AttributeChangeSetType类型
monitoredAttributes	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串AlarmIRPConstDefs::AttributeNameValue::MONITORED_ATTRIBUTES。 其值为AlarmIRPConstDefs::AttributeSetType类型
proposedRepairActions	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串AlarmIRPConstDefs::AttributeNameValue::PROPOSED_REPAIR_ACTIONS。 其值为字符串类型
additionalText	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串AlarmIRPConstDefs::AttributeNameValue::ADDITIONAL_TEXT。 其值为字符串类型
additionalInformation	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字和值由厂商定义
alarmId	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串AlarmIRPConstDefs::AttributeNameValue::ALARM_ID。 其值为字符串类型。如果这个字串是一个零长度字串或此名值对不存在，缺省语义为alarmId由此结构化事件中的managedObjectInstance,eventType,probableCause和specificProblem拼接而成（如果他们存在）。因为probableCause编码为短整型，拼接之前应把它转化为字符串型。拼接所得的结果不能含有空格
分析阶段没有参数	remaining_ body	

表71 notifyNewAlarm 通知参数映射表（承载安全告警）

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
分析阶段没有参数	域名（domain_name）	字符串，用来说明IRP的版本信息
notificationType	类型名（type_name）	字符串常量： AlarmIRPConstDefs:: NotificationType:: NOTIFY_FM_NEW_ALARM
alarmType	事件名（event_name）	AlarmIRPConstDefs:: AlarmType中的“X6” ~ “X10”
分析阶段没有参数	可变事件头（variable Header）	
objectClass， objectInstance	可过滤体域（filterable_body_fields）的一个名值对	发出通知的网络资源对象类和对象实例。 名值对的名字是一个字符串 NotificationIRPConstDefs::AttributeNameValue:: MANAGED_OBJECT_INSTANCE。 其值为字符串型
notificationId	可过滤体域（filterable_body_fields）的一个名值对	通知标识符，用来唯一标识通知，可进行通知的关联。 名值对的名字是一个字符串 NotificationIRPConstDefs::AttributeNameValue:: NOTIFICATION_ID。 其值为long长整型
eventTime	可过滤体域（filterable_body_fields）的一个名值对	指明事件发生时间。使用 OMG定义的UtcT类型。 名值对的名字是一个字符串 NotificationIRPConstDefs::AttributeNameValue:: EVENT_TIME
systemDN	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 NotificationIRPConstDefs::AttributeNameValue:: SYSTEM_DN。 其值为字符串类型
probableCause	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: PROBABLE_CAUSE。 其值为短整型，由AlarmIRPConstDefs:: ProbableCause接口定义
perceivedSeverity	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: PERCEIVED_SEVERITY。 其值为短整型，由AlarmIRPConstDefs:: PerceivedSeverity接口定义
correlatedNotification	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: CORRELATED_NOTIFICATIONS。 其值为AlarmIRPConstDefs:: CorrelatedNotificationSetType类型
additionalText	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: ADDITIONAL_TEXT。 其值为字符串类型

表71（续）

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
additionalInformation	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字和值由厂家定义
alarmId	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串AlarmIRPConstDefs::AttributeNameValue::ALARM_ID。 其值为字符串类型。如果这个字串是一个零长度字串或此名值对不存在，缺省语义为alarmId由此结构化事件中的managedObjectInstance, eventType, probableCause 和specificProblem拼接而成（如果他们存在）。因为probableCause编码为短整型，拼接之前应把它转化为字符串型。拼接所得的结果不能含有空格
serviceUser	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串AlarmIRPConstDefs::AttributeNameValue::SERVICE_USER。 其值为字符串类型
serviceProvider	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串AlarmIRPConstDefs::AttributeNameValue::SERVICE_PROVIDER。 其值为字符串类型
securityAlarmDetector	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串AlarmIRPConstDefs::AttributeNameValue::SECURITY_ALARM_DETECTOR。 其值为字符串类型
分析阶段没有参数	remaining_ body	

表72　notifyAckStateChanged 通知参数映射表

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
分析阶段没有参数	域名（domain_name）	字符串，用来说明IRP的版本信息
NotificationType	类型名（type_name）	字符串常量：AlarmIRPConstDefs::NotificationType::NOTIFY_FM_ ACK_STATE_CHANGED
alarmType	事件名（event_name）	参见 notifyNewAlarm 中相关定义
分析阶段没有参数	可变事件头（variable Header）	
objectClass，objectInstance	可过滤体域（filterable_body_fields）的一个名值对	参见notifyNewAlarm中相关定义
notificationId	可过滤体域（filterable_body_fields）的一个名值对	参见notifyNewAlarm中相关定义
eventTime	可过滤体域（filterable_body_fields）的一个名值对	参见notifyNewAlarm中相关定义
systemDN	可过滤体域（filterable_body_fields）的一个名值对	参见notifyNewAlarm中相关定义

表72（续）

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
probableCause	可过滤体域（filterable_body_fields）的一个名值对	参见notifyNewAlarm中相关定义
perceivedSeverity	可过滤体域（filterable_body_fields）的一个名值对	参见notifyNewAlarm中相关定义
alarmId	可过滤体域（filterable_body_fields）的一个名值对	参见notifyNewAlarm中相关定义
ackTime	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: ACK_TIME。 其值为ManagedGenericIRPConstDefs:: IRPTime类型
ackUserId	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: ACK_USER_ID。 其值为字符串类型
ackSystemId	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: ACK_SYSTEM_ID。 其值为字符串类型
ackState	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: ACK_STATE。 其值为短整型，在AlarmIRPConstDefs:: AckState 接口中定义
分析阶段没有参数	remaining_body	

表73　notifyClearedAlarm 通知参数映射表

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
分析阶段没有参数	域名（domain_name）	字符串，用来说明IRP的版本信息
NotificationType	类型名（type_name）	字符串常量： AlarmIRPConstDefs::NotificationType:: NOTIFY_FM_ CLEARED_ALARM
alarmType	事件名（event_name）	参见 notifyNewAlarm 中相关定义
分析阶段没有参数	可变事件头（variable Header）	
objectClass，objectInstance	可过滤体域（filterable_body_fields）的一个名值对	参见notifyNewAlarm中相关定义
notificationId	可过滤体域（filterable_body_fields）的一个名值对	参见notifyNewAlarm中相关定义
eventTime	可过滤体域（filterable_body_fields）的一个名值对	参见notifyNewAlarm中相关定义
systemDN	可过滤体域（filterable_body_fields）的一个名值对	参见notifyNewAlarm中相关定义
probableCause	可过滤体域（filterable_body_fields）的一个名值对	参见notifyNewAlarm中相关定义

表73（续）

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
perceivedSeverity	可过滤体域（filterable_body_fields）的一个名值对	参见notifyNewAlarm中相关定义
correlatedNotifications	—	[注1]
alarmId	可过滤体域（filterable_body_fields）的一个名值对	参见notifyNewAlarm中相关定义
clearedUserId	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: CLEAR_USER_ID。 其值为字符串类型
clearedSystemId	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: CLEAR_SYSTEM_ID。 其值为字符串类型
分析阶段没有参数	remaining_body	
[注1]在CORBA接口中，correlatedNotifications没有使用；在CORBA接口中，一条notifyClearedAlarm通知只能清除一条告警		

表74　notifyPotentialFaultyAlarmList 通知参数映射表

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
分析阶段没有参数	域名（domain_name）	字符串，用来说明IRP的版本信息
NotificationType	类型名（type_name）	字符串常量： AlarmIRPConstDefs::NotificationType:: NOTIFY_FM_POTENTIAL_FAULTY_ALARM_LIST
分析阶段没有参数	事件名（event_name）	应置成空串
分析阶段没有参数	可变事件头（variable Header）	
objectClass， objectInstance	可过滤体域（filterable_body_fields）的一个名值对	发出通知的 AlarmIRP 对象或网络资源对象的类及实例。 此域用来指明部分还是全部告警列表有潜在的错误
notificationId	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
eventTime	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
systemDN	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
reason	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue::REASON。 其值为字符串类型
分析阶段没有参数	保留体(remaining_ body)	

表75 notifyAlarmListRebuilt 通知参数映射表

分析阶段通知参数	OMG CORBA结构事件参数映射	说明
分析阶段没有参数	域名（domain_name）	字符串，用来说明IRP的版本信息
NotificationType	类型名（type_name）	字符串常量： AlarmIRPConstDefs::NotificationType:: NOTIFY_FM_ ALARM_LIST_REBUILT
分析阶段没有参数	事件名（event_name）	应置成空串
分析阶段没有参数	可变事件头（variable Header）	
objectClass， objectInstance	可过滤体域（filterable_body_fields）的一个名值对	发出通知的 AlarmIRP 对象或网络资源对象的类及实例
notificationId	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
eventTime	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
systemDN	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
reason	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue::REASON。 其值为字符串类型
alarmListAlignment Requirement	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue:: ALARM_LIST_ALIGNMENT_REQUIREMENT。 其值为枚举类型 AlarmIRPConstDefs::AlarmListAlignmentRequirementType
分析阶段没有参数	remaining_ body	

表76 notifyComments 通知参数映射表

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
分析阶段没有参数	域名（domain_name）	字符串，用来说明IRP的版本信息
NotificationType	类型名（type_name）	字符串常量： AlarmIRPConstDefs::NotificationType:: NOTIFY_FM_ COMMENT_ADDED
分析阶段没有参数	事件名（event_name）	应置成空串
分析阶段没有参数	可变事件头（variable Header）	
objectClass， objectInstance	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
notificationId	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
eventTime	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
systemDN	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义

表76（续）

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
alarmId	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
probableCause	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
perceivedSeverity	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
comments	可过滤体域（filterable_body_fields）的一个名值对	名值对的名字是一个字符串 AlarmIRPConstDefs::AttributeNameValue::COMMENTS。其值为AlarmIRPConstDefs:: CommentSet
分析阶段没有参数	remaining_body	

表77　notifyChangedAlarm 通知参数映射表

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
分析阶段没有参数	域名（domain_name）	字符串，用来说明IRP的版本信息
NotificationType	类型名（type_name）	字符串常量：AlarmIRPConstDefs::NotificationType::NOTIFY_FM_ CHANGED_ALARM
alarmType	事件名（event_name）	参见 notifyNewAlarm 中相关定义
分析阶段没有参数	可变事件头（variable Header）	
objectClass，objectInstance	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
notificationId	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
eventTime	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
systemDN	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
alarmId	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
probableCause	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
perceivedSeverity	可过滤体域（filterable_body_fields）的一个名值对	参见 notifyNewAlarm 中相关定义
分析阶段没有参数	remaining_body	

7 性能管理接口设计

下面给出了性能管理接口分析到基于CORBA/IDL技术接口设计的映射表格。

7.1 操作和通知映射

在《接口分析》中定义了性能管理接口操作和通知的语义，表78给出了这些操作和通知到基于CORBA/IDL技术接口设计的映射。

表78 操作和通知映射表

分析阶段操作/通知	设计阶段映射
createMeasurementJob	create_measurement_job
stopMeasurementJob	stop_measurement_job
suspendMeasurementJob	suspend_measurement_job
resumeMeasurementJob	resume_measurement_job
listMeasurementJobs	list_measurement_jobs
createThresholdMonitor	create_threshold_monitor
deleteThresholdMonitor	delete_threshold_monitor
listThresholdMonitors	list_threshold_monitors
suspendThresholdMonitor	suspend_threshold_monitor
resumeThresholdMonitor	resume_threshold_monitor
getIRPVersion [注]	get_PM_IRP_versions
getIRPOperationsProfile [注]	get_PM_IRP_operations_profile
getIRPNotificationProfile [注]	get_PM_IRP_notification_profile
notifyMeasurementJobStatusChanged	push_structured_events
notifyThresholdMonitorStatusChanged	push_structured_events
注：这三个操作是从ManagedGenericIRP中继承下来的操作	

7.2 操作参数映射

在《接口分析》中定义了性能管理接口操作中各个参数的语义，表79~91中逐个给出了操作中各个参数到基于CORBA/IDL技术接口设计的映射。

表79 createMeasurementJob 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
iocName	PMIRPConstDefs::MOClassNameType moClass
iocInstanceList	PMIRPConstDefs::MOInstanceListType moInstanceList
measurementCategoryList	PMIRPConstDefs::MeasurementCategoryListType measurementCategoryList
granularityPeriod	PMIRPConstDefs:: GranularityPeriodType granularityPeriod
reportingPeriod	PMIRPConstDefs::ReportingPeriodType reportingPeriod
startTime	PMIRPConstDefs::StartTimeTypeOpt startTime
stopTime	PMIRPConstDefs::StopTimeTypeOpt stopTime
schedule	PMIRPConstDefs::ScheduleTypeOpt schedule
jobId	返回值类型 PMIRPConstDefs::JobIdType
unsupportedList	PMIRPConstDefs::JUnsupportedListType unsupportedList
result	ManagedGenericIRPConstDefs::Signal 异常：CreateMeasurementJob， ManagedGenericIRPSystem::InvalidParameter HighWorkLoad

表80 stopMeasurementJob 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
jobId	PMIRPConstDefs::JobIdType jobId
result	PMIRPConstDefs::ResultType 异常：StopMeasurementJob, UnknownJob, JobCannotBeStopped

表81 suspendMeasurementJob 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
jobId	PMIRPConstDefs::JobIdType jobId
result	PMIRPConstDefs::ResultType 异常：SuspendMeasurementJob， UnknownJob, JobAlreadySuspended

表82 resumeMeasurementJob 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
jobId	PMIRPConstDefs::JobIdType jobId
result	PMIRPConstDefs::ResultType 异常：ResumeMeasurementJob， UnknownJob, JobIsNotSuspended

表83 listMeasurementJobs 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
jobIdList	PMIRPConstDefs::JobIdListType JobIdList
jobInfoList	PMIRPConstDefs::JobInfoListType JobInfoList
result	PMIRPSystem::ResultType 异常：ListMeasurementJobs， ManagedGenericIRPSystem::InvalidParameter

表84 createThresholdMonitor 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
iOCName	PMIRPConstDefs::MOClassNameType moClass
iOCInstanceList	PMIRPConstDefs::MOInstanceListType moInstanceList
thresholdInfoList	PMIRPConstDefs::ThresholdInfoListType thresholdInfoList
monitorGranularityPeriod	PMIRPConstDefs:: MonitorGranularityPeriodType monitorGranularityPeriod
thresholdMonitorId	返回值类型 PMIRPConstDefs::MonitorIdType
unsupportedList	PMIRPConstDefs::MUnsupportedListType unsupportedList
result	ManagedGenericIRPConstDefs::Signal 异常：CreateThresholdMonitor， ManagedGenericIRPSystem::InvalidParameter

表85 deleteThresholdMonitor 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
thresholdMonitorId	PMIRPConstDefs:: MonitorIdType MonitorId
result	PMIRPConstDefs::ResultType 异常：DeleteThresholdMonitor, UnknownThresholdMonitor

表86 listThresholdMonitor 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
monitorIdList	PMIRPConstDefs:: MonitorIdListType monitorIdList
monitorInfoList	返回值类型 PMIRPConstDefs::MonitorInfoListType
result	PMIRPConstDefs::ResultType 异常：ListThresholdMonitor, ManagedGenericIRPSystem::InvalidParameter

表87 suspendThresholdMonitor 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
thresholdMonitorId	PMIRPConstDefs::JobIdType MonitorId
result	PMIRPConstDefs::ResultType 异常：SuspendThresholdMonitor, UnknownThresholdMonitor, ThresholdMonitorAlreadySuspended

表88 resumeThresholdMonitor 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
thresholdMonitorId	PMIRPConstDefs::JobIdType MonitorId
result	PMIRPConstDefs::ResultType 异常：ResumeThresholdMonitor, UnknownThresholdMonitor, ThresholdMonitorIsNotSuspended

表89 getIRPVersion 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
versionNumberSet	返回值类型 ManagedGenericIRPConstDefs::VersionNumberSet
result	异常：GetPMIRPVersions

表90 getOperationsProfile 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
IRPVersion	ManagedGenericIRPConstDefs::VersionNumber pm_irp_version
operationNameProfile, operationParameterPrfile	返回值类型 ManagedGenericIRPConstDefs::MethodList
result	异常：GetPMIRPOperationsProfile, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

表91 getNotificationProfile 操作参数映射表

分析阶段操作参数	设计阶段操作参数映射
IRPVersion	ManagedGenericIRPConstDefs::VersionNumber pm_irp_version
notificationNameProfile, notificationParameterPrfile	返回值类型 ManagedGenericIRPConstDefs::MethodList
result	异常：GetPMIRPNotificationProfile, ManagedGenericIRPSystem::OperationNotSupported, ManagedGenericIRPSystem::InvalidParameter

7.3 通知参数映射

在《接口分析》中定义了性能管理接口通知中各个参数的语义，表92和93中列出了通知中各个参数到基于CORBA/IDL技术接口设计的映射。

表92 notifyMeasurementJobStatusChange 通知格式

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
分析阶段没有相应参数	域名（domain_name）	字符串，为相应IRP信息模型的版本号
NotificationType	类型名（type_name）	PMIRPConstDefs模块中的 ET_MEASUREMENT_JOB_STATUS_CHANGED
--	事件名（event_name）	应置成空串
分析阶段没有相应参数	可变事件头（variable Header）	
objectClass， objectInstance	可过滤体域（filterable_body_fields）的一个名值对	发出通知的管理对象类和对象实例，即PMIRP的对象类和对象实例。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: MANAGED_OBJECT_INSTANCE 其值属字符串类型
notificationId	可过滤体域（filterable_body_fields）的一个名值对	通知标识符，用来唯一标识通知，可进行通知的关联。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: NOTIFICATION_ID 其值是长整形long
eventTime	可过滤体域（filterable_body_fields）的一个名值对	指明事件发生时间。使用 OMG定义的UtcT类型。 NotificationIRPConstDefs::AttributeNameValue:: EVENT_TIME
systemDN	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串 NotificationIRPConstDefs::AttributeNameValue:: SYSTEM_DN 其值属字符串类型
jobId	可过滤体域（filterable_body_fields）的一个名值对	PMIRPNotifDefs::notifyMeasurementJobStatusChanged::JOB_ID 其值属PMIRPConstDefs::JobIdType类型
jobStatus	可过滤体域（filterable_body_fields）的一个名值对	PMIRPNotifDefs::notifyMeasurementJobStatusChanged::JOB_STATUS 其值属PMIRPConstDefs::JobStatusType类型
reason	可过滤体域（filterable_body_fields）的一个名值对	PMIRPNotifDefs::notifyMeasurementJobStatusChanged::REASON 其值属字符串类型
分析阶段没有相应参数	保留体（remaining_ body）	

表93 notifyMonitorStatusChange 通知格式

分析阶段通知参数	OMG CORBA结构事件参数映射	说　明
分析阶段没有相应参数	域名（domain_name）	字符串，为相应IRP信息模型的版本号
NotificationType	类型名（type_name）	PMIRPConstDefs模块中的 ET_THRESHOLD_MONITOR_STATUS_CHANGED
--	事件名（event_name）	应置成空串
分析阶段没有相应参数	可变事件头（variable Header）	
objectClass， objectInstance	可过滤体域（filterable_body_fields）的一个名值对	发出通知的管理对象类和对象实例，即PMIRP的对象类和对象实例。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: MANAGED_OBJECT_INSTANCE 其值属字符串类型
notificationId	可过滤体域（filterable_body_fields）的一个名值对	通知标识符，用来唯一标识通知，可进行通知的关联。 名值对的名称为字符串： NotificationIRPConstDefs::AttributeNameValue:: NOTIFICATION_ID 其值是长整形long
eventTime	可过滤体域（filterable_body_fields）的一个名值对	指明事件发生时间。使用 OMG定义的UtcT类型。 NotificationIRPConstDefs::AttributeNameValue:: EVENT_TIME
systemDN	可过滤体域（filterable_body_fields）的一个名值对	名值对的名称为字符串 NotificationIRPConstDefs::AttributeNameValue:: SYSTEM_DN 其值属字符串类型
thresholdMonitorId	可过滤体域（filterable_body_fields）的一个名值对	PMIRPNotifDefs::notifyThresholdMonitorStatusChanged::MONITOR_ID 其值属PMIRPConstDefs::MonitorIdType类型
monitorStatus	可过滤体域（filterable_body_fields）的一个名值对	PMIRPNotifDefs::notifyThresholdMonitorStatusChanged:: MONITOR _STATUS 其值属PMIRPConstDefs:: MonitorStatusType类型
reason	可过滤体域（filterable_body_fields）的一个名值对	PMIRPNotifDefs::notifyThresholdMonitorStatusChanged::REASON 其值属字符串类型
分析阶段没有相应参数	保留体（remaining_ body）	

附 录 A
（规范性附录）
基于 CORBA/IDL 的接口信息模型

A.1 通用部分接口信息模型

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

#ifndef _MANAGEDGENERICIRPCONSTDEFS_IDL_
#define _MANAGEDGENERICIRPCONSTDEFS_IDL_

#include <TimeBase.idl>

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

/* ## Module: ManagedGenericIRPConstDefs
This module contains definitions commonly used among all IRPs such as Alarm IRP.
================================================================
*/
module ManagedGenericIRPConstDefs
{
    /*
    Definition imported from CosTime.
    The time refers to time in Greenwich Time Zone.
    It also consists of a time displacement factor in the form of minutes of
    displacement from the Greenwich Meridian.
    */
    typedef TimeBase::UtcT IRPTime;

    enum Signal {OK, FAILURE, PARTIALFAILURE};

    /*
    The VersionNumber is a string that identifies the IRP specification name
    and its version number.  See definition "IRP document version number
    string" or "IRPVersion".

    The VersionNumberSet is a sequenece of such VersionNumber.  It is returned
```

```
by get_XXX_IRP_versions().  The sequence order has no significance.
*/
typedef string VersionNumber;
typedef sequence <VersionNumber> VersionNumberSet;

typedef string MethodName;
typedef string ParameterName;
typedef sequence <ParameterName> ParameterList;

/*
The Method defines the structure to be returned as part of
get_supported_operations_profile().  The name shall be the actual method
name (ex. "attach_push", "change_subscription_filter", etc.)
The parameter_list contains a list of strings.  Each string shall be
the actual parameter name (ex. "manager_reference", "filter", etc.)
*/
struct Method
{
    MethodName name;
    ParameterList parameter_list;
};

/*
List of all methods and their associated parameters.
*/
typedef sequence <Method> MethodList;

/*
StringTypeOpt is a type carrying an optional parameter.
If the boolean is TRUE, then the value is present.
Otherwise the value is absent.
*/
union StringTypeOpt switch (boolean)
{
    case TRUE: string value;
};
```

```
/*
ShortTypeOpt is a type carrying an optional parameter.
If the boolean is TRUE, then the value is present.
Otherwise the value is absent.
*/
union ShortTypeOpt switch (boolean)
{
    case TRUE: short value;
};

/*
UnsignedShortTypeOpt is a type carrying an optional parameter.
If the boolean is TRUE, then the value is present.
Otherwise the value is absent.
*/
union UnsignedShortTypeOpt switch (boolean)
{
    case TRUE: unsigned short value;
};

/*
LongTypeOpt is a type carrying an optional parameter.
If the boolean is TRUE, then the value is present.
Otherwise the value is absent.
*/
union LongTypeOpt switch (boolean)
{
    case TRUE: long value;
};

/*
UnsignedLongTypeOpt is a type carrying an optional parameter.
If the boolean is TRUE, then the value is present.
Otherwise the value is absent.
*/
union UnsignedLongTypeOpt switch (boolean)
{
```

```
        case TRUE: unsigned long value;
    };
};

#endif   // _MANAGEDGENERICIRPCONSTDEFS_IDL_

//File "ManagedGenericIRPSystem.idl"
//The IRP document version number is "Generic IRP V1.0"

#ifndef _MANAGEDGENERICIRPSYSTEM_IDL_
#define _MANAGEDGENERICIRPSYSTEM_IDL_

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

/* ## Module: ManagedGenericIRPSystem
This module contains definitions commonly used among all IRPs such as Alarm IRP.
================================================================
*/
module ManagedGenericIRPSystem
{
    /*
    Exception thrown when an unsupported optional parameter
    is passed with information.
    The parameter shall be the actual unsupported parameter name.
    */
    exception ParameterNotSupported { string parameter; };

    /*
    Exception thrown when an invalid parameter value is passed.
    The parameter shall be the actual parameter name.
    */
    exception InvalidParameter { string parameter; };

    /*
    Exception thrown when a valid but unsupported parameter value is passed.
    The parameter shall be the actual parameter name.
```

```
    */
    exception ValueNotSupported { string parameter; };

    /*
    Exception thrown when an unsupported optional method is called.
    */
    exception OperationNotSupported {};
};

#endif  // _MANAGEDGENERICIRPSYSTEM_IDL_
```

A.2 访问入口点接口信息模型

```
//File "EPIRPSystem.idl"
//The IRP document version number is "EP IRP V1.0"

#ifndef EPIRPSystem_idl
#define EPIRPSystem_idl

#include "NotificationIRPConstDefs.idl"
#include "ManagedGenericIRPConstDefs.idl"
#include "ManagedGenericIRPSystem.idl"

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

/* ## Module: EPIRPSystem
This module implements capabilities of EPIRP.
==========================================================================
*/
module EPIRPSystem
{
  enum ResultType {OK, Failure};

  typedef string IRPIdType;
  typedef string SystemDNType;
  typedef sequence<string> DNListType;

  /*
```

```
 IRPManagementScopeOpt is a type carrying an optional parameter.
 If the boolean is TRUE, then the value is present.
 Otherwise the value is absent.

*/
union IRPManagementScopeOpt switch (boolean)
{
   case TRUE: DNListType value;
};

/*
The IRPElement defines the structure to be returned as part of
getIRPOutline().
*/
struct IRPElement
{
   IRPIdType irpId;
   ManagedGenericIRPConstDefs::VersionNumberSet irpVersions;
   IRPManagementScopeOpt irpManagementScope;
};

/*
 List of all IRPElement and their associated parameters.
*/
typedef sequence<IRPElement> IRPListType;

struct SupportedIRPListTypeElement
{
   SystemDNType systemDN;
   IRPListType irpList;
};

typedef sequence<SupportedIRPListTypeElement> SupportedIRPListType;

typedef string ManagerIdentifierType;

typedef string IRPDnType;
```

```
enum ChangeModeType {REGISTER, DEREGISTER, MODIFY};

/*
Define the parameters specified in
the notifyEpInfoChanges notification.
*/
interface AttributeNameValue
{
    const string IRP_DN = "IRP_DN";
    const string CHANGE_MODE = "CHANGE_MODE";
    const string ADDITIONAL_TEXT = "ADDITIONAL_TEXT";
};


const string ET_IRPINFO_CHANGES = "notifyIrpInfoChanges";

exception InvalidIRPVersion { string reason; };
exception InvalidRequestedParameters { string reason; };
exception UnknownIRPReference { string reason; };

/*
 System fails to complete the operation.  System can provide reason
 to qualify the exception.  The semantics carried in reason
 is outside the scope of this IRP.
*/
exception GetIRPOutline { string reason; };
exception GetIRPReference { string reason; };
exception ReleaseIRPReference { string reason; };
exception GetEPIRPVersions { string reason; };
exception GetEPIRPOperationsProfile { string reason; };
exception GetEPIRPNotificationProfile { string reason; };

/*
*/
 interface EPIRP
 {
   /**
```

```
* The IRPManager uses this operation to request the EPIRP to
* return the outline information of the supported IRPs. The EPIRP
* shall return the outline information of all the IRPs supported by the
* IRPAgent that contains the EPIRP. The EPIRP may
* additionally return the outline information of all the IRPs supported
* by other IRPAgents.
*/
ResultType      get_IRP_outline(
   in ManagedGenericIRPConstDefs::VersionNumber iRPVersion,
   out SupportedIRPListType supportedIRPList
)
raises (GetIRPOutline,InvalidIRPVersion);

/**
* The IRPManager uses this operation to request the EPIRP to return
* the stringified IOR of the IRP identified by systemDn and irpId.
*/
ResultType      get_IRP_reference(
   in ManagerIdentifierType managerIdentifier,
   in SystemDNType systemDn,
   in IRPIdType irpId,
   out string iRPReference
)
raises (GetIRPReference,
    InvalidRequestedParameters);

/**
* The IRPManager uses this operation to request the IRPAgent to
* release a specific IRP reference. Whether the IRP reference
* is really released or not in the IRPAgent is outside the
* scope of this document.
*/
ResultType      release_IRP_reference(
   in ManagerIdentifierType managerIdentifier,
   in string iRPReference
)
raises (ReleaseIRPReference,
```

```
        UnknownIRPReference);

    /**
    * Return the list of all supported EPIRP versions.
    */
    ManagedGenericIRPConstDefs::VersionNumberSet get_EP_IRP_versions (
    )
    raises (GetEPIRPVersions);


    /**
    * Return the list of all supported operations and their supported
    * parameters for a specific EPIRP version.
    */
    ManagedGenericIRPConstDefs::MethodList get_EP_IRP_operations_profile (
        in ManagedGenericIRPConstDefs::VersionNumber iRPVersion
    )
    raises (GetEPIRPOperationsProfile,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);

    /**
    * Return the list of all supported notifications and their supported
    * parameters for a specific EPIRP version.
    */
    ManagedGenericIRPConstDefs::MethodList get_EP_IRP_notification_profile
    (
        in ManagedGenericIRPConstDefs::VersionNumber iRPVersion
    )
    raises (GetEPIRPNotificationProfile,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);
};


/**
*   Constant definitions for the EPInfoChanges notification
```

```
*/
module notifyIRPInfoChanges
{
  const string MANAGED_OBJECT_CLASS =
    NotificationIRPConstDefs::AttributeNameValue::MANAGED_OBJECT_CLASS;
  const string MANAGED_OBJECT_INSTANCE =
    NotificationIRPConstDefs::AttributeNameValue::MANAGED_OBJECT_INSTANCE;
  const string NOTIFICATION_ID =
    NotificationIRPConstDefs::AttributeNameValue::NOTIFICATION_ID;
  const string EVENT_TIME =
    NotificationIRPConstDefs::AttributeNameValue::EVENT_TIME;
  const string SYSTEM_DN =
    NotificationIRPConstDefs::AttributeNameValue::SYSTEM_DN;
  const string EVENT_TYPE = ET_IRPINFO_CHANGES;

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

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

  /**
  * This constant defines the name of the additionalText property,
  * which is transported in the filterable_body fields.
  * The data type for the value of this property is string.
  */
  const string ADDITIONAL_TEXT = AttributeNameValue::ADDITIONAL_TEXT;
  };
```

```
};

#endif
```

A.3 通知管理接口信息模型

本节采用IDL定义了通知管理接口的信息模型。通知格式定义基于CORBA通知服务中的CosNotification模块中的相关定义。

```
//File "NotificationIRPConstDefs.idl"
//The IRP document version number is "Notification IRP V1.0"

#ifndef _NOTIFICATIONIRPCONSTDEFS_IDL_
#define _NOTIFICATIONIRPCONSTDEFS_IDL_

#include "ManagedGenericIRPConstDefs.idl"

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


/* ## Module: NotificationIRPConstDefs
This module contains definitions specific for Notification IRP.
=================================================================
*/
module NotificationIRPConstDefs
{


    /*
    Define the parameters (in the notification header) specified in
    the Notification IRP: IS.
    */
    interface AttributeNameValue
    {
        const string NOTIFICATION_ID = "a";
        const string EVENT_TIME = "b";
        const string SYSTEM_DN = "c";
        const string MANAGED_OBJECT_CLASS = "d";
```

```
        const string MANAGED_OBJECT_INSTANCE = "e";
    };

    /*
    It defines the notification categories.
    A notification category is identified by the IRP name and its version number.
    */
    typedef ManagedGenericIRPConstDefs::VersionNumberSet NotificationCategorySet;

    /*
    It defines the notification types of a particular notification category.
    */
    typedef sequence <string> NotificationTypePerNotificationCategory;

    /*
    This sequence identifies all notification types of all notification
    categories identified by NotificationCategorySet.  The number of elements
    in this sequence shall be identical to that of NotificationCategorySet.
    */
    typedef sequence <NotificationTypePerNotificationCategory>
        NotificationTypesSet;

    /*
    It defines a sequence of SubscriptionIds.
    */
    typedef string SubscriptionId;
    typedef sequence <SubscriptionId> SubscriptionIdSet;

    /*
    This indicates if the subscription is Active (not suspended), Suspended,
    or Invalid.
    */
    enum SubscriptionState {ACTIVE, SUSPENDED, INVALID};
};

#endif  // _NOTIFICATIONIRPCONSTDEFS_IDL_
```

```
//File "NotificationIRPSystem.idl"
//The IRP document version number is "Notification IRP V1.0"

#ifndef _NOTIFICATIONIRPSYSTEM_IDL_
#define _NOTIFICATIONIRPSYSTEM_IDL_

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

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

/* ## Module: NotificationIRPSystem
This module implements capabilities of Notification IRP.
================================================================
*/
module NotificationIRPSystem
{
    /*
    System fails to complete the operation.  System can provide reason
    to qualify the exception.  The semantics carried in reason
    is outside the scope of this IRP.
    */
    exception GetNotificationIRPVersions { string reason; };
    exception GetNotificationIRPOperationsProfile { string reason; };
    exception GetNotificationIRPNotificationProfile { string reason; };
    exception Attach { string reason; };
    exception DetachException { string reason; };
    exception GetSubscriptionStatus { string reason; };
    exception ChangeSubscriptionFilter { string reason; };
    exception GetNotificationCategories { string reason; };

    exception GetSubscriptionIds { string reason; };

    exception AlreadySubscribed {};
```

```
exception AtLeastOneNotificationCategoryNotSupported {};

interface NotificationIRP
{
    /*
    Return the list of all supported Notification IRP versions
    Each IRPVersion is defined by the rule in TS 32.311 clause titled
    "IRP document version number string"
    */
    ManagedGenericIRPConstDefs::VersionNumberSet get_notification_IRP_versions
    (
    )
    raises (GetNotificationIRPVersions);

    /*
    Return the list of all supported operations and their supported
    parameters for a specific Notification IRP version.
    */
    ManagedGenericIRPConstDefs::MethodList
        get_notification_IRP_operations_profile (
            in ManagedGenericIRPConstDefs::VersionNumber
                notification_irp_version
    )
    raises (GetNotificationIRPOperationsProfile,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);

    /*
    Return the list of all supported notifications.
    Agent should always throw a ManagedGenericIRPSystem::OperationNotSupported
    exception.
    Similar method, such as get_alarm_IRP_notification_profile,
    is supported in other IRP versions such as Alarm IRP.
    */
    ManagedGenericIRPConstDefs::MethodList
        get_notification_IRP_notification_profile (
            in ManagedGenericIRPConstDefs::VersionNumber
```

```
                notification_irp_version
    )
    raises (GetNotificationIRPNotificationProfile,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);

    /*
    Obtain the list of all supported notification categories.
    */
    NotificationIRPConstDefs::NotificationCategorySet
        get_notification_categories (
            out NotificationIRPConstDefs::NotificationTypesSet
                notification_type_list
    )
    raises (GetNotificationCategories,
        ManagedGenericIRPSystem::OperationNotSupported);

    NotificationIRPConstDefs::SubscriptionId attach_push (
        in string manager_reference,
        in unsigned long time_tick,
        in NotificationIRPConstDefs::NotificationCategorySet
            notification_categories,
        in ManagedGenericIRPConstDefs::StringTypeOpt filter
    )
    raises (Attach, ManagedGenericIRPSystem::ParameterNotSupported,
            ManagedGenericIRPSystem::InvalidParameter, AlreadySubscribed,
            AtLeastOneNotificationCategoryNotSupported);

    NotificationIRPConstDefs::SubscriptionId attach_push_b (
        in string manager_reference,
        in unsigned long time_tick,
        in NotificationIRPConstDefs::NotificationCategorySet
            notification_categories,
        in ManagedGenericIRPConstDefs::StringTypeOpt filter,
        out CosNotifyChannelAdmin::SequenceProxyPushSupplier system_reference
    )
    raises (Attach, ManagedGenericIRPSystem::OperationNotSupported,
```

```
            ManagedGenericIRPSystem::ParameterNotSupported,
            ManagedGenericIRPSystem::InvalidParameter,
            AlreadySubscribed, AtLeastOneNotificationCategoryNotSupported);

NotificationIRPConstDefs::SubscriptionId attach_pull (
    in string manager_reference,
    in unsigned long time_tick,
    in NotificationIRPConstDefs::NotificationCategorySet
        notification_categories,
    in ManagedGenericIRPConstDefs::StringTypeOpt filter,
    out CosNotifyChannelAdmin::SequenceProxyPullSupplier system_reference
)
raises (Attach, ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::ParameterNotSupported,
            ManagedGenericIRPSystem::InvalidParameter,
            AlreadySubscribed, AtLeastOneNotificationCategoryNotSupported);

/*
Replace the present filter constraint with the one provided.
*/
void change_subscription_filter (
    in NotificationIRPConstDefs::SubscriptionId subscription_id,
    in string filter
)
raises (ChangeSubscriptionFilter,
      ManagedGenericIRPSystem::OperationNotSupported,
      ManagedGenericIRPSystem::InvalidParameter);

/*
Check the current state of the subscription.
*/
NotificationIRPConstDefs::NotificationCategorySet get_subscription_status
(
    in NotificationIRPConstDefs::SubscriptionId subscription_id,
    out ManagedGenericIRPConstDefs::StringTypeOpt filter_in_effect,
    out NotificationIRPConstDefs::SubscriptionState subscription_state,
    out unsigned long time_tick
```

```
        )
        raises (GetSubscriptionStatus,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);

        NotificationIRPConstDefs::SubscriptionIdSet get_subscription_ids (
            in string manager_reference
        )
        raises (GetSubscriptionIds,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);

        /*
        Terminates the subscription with the agent.
        */
        void detach (
            in string manager_reference,
            in NotificationIRPConstDefs::SubscriptionId subscription_id
        )
        raises (DetachException,
            ManagedGenericIRPSystem::ParameterNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);
    };
};

#endif  // _NOTIFICATIONIRPSYSTEM_IDL_
```

A.4 链路监视接口信息模型

```
//File "CSIRPConstDefs.idl"
//The IRP document version number is "CS IRP V1.0"

#ifndef CSIRPConstDefs_idl
#define CSIRPConstDefs_idl

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

/* ## Module: CSIRPConstDefs
```

```
This module contains commonly used definitions for CSIRP.
================================================================
*/
module CSIRPConstDefs
{

  typedef unsigned short HeartbeatPeriodType;

  /*
  If ET_HEARTBEAT notification is triggered by NM positively by invoking
  trigger_heartbeat operation, the value of this parameter shall be IRPManager,
  otherwise, it shall be IRPAgent.
  */
  enum TriggerFlagType {IRPManager, IRPAgent};

  typedef string ManagerIdentifierType;

  typedef string ChannelIdType;

  /*
  It specifies whether the operation is success or failed.
  */
  enum ResultType { Success, Failure };

   /**
   * This block identifies attributes which are included as part of the
   * CommunicationSurveillanceIRP. These attribute values should not
   * clash with those defined for the attributes of notification
   * header (see IDL of Notification IRP).
   */
   module AttributeNameValue
   {
      const string HEARTBEAT_PERIOD = "HEARTBEAT_PERIOD";
      const string CHANNEL_ID = "CHANNEL_ID";
      const string TRIGGER_FLAG = "TRIGGER_FLAG";
      const string MANAGER_IDENTIFIER = "MANAGER_IDENTIFIER";
```

```
    };

};
#endif

//File "CSIRPSystem.idl"
//The IRP document version number is "CS IRP V1.0"

#ifndef CSIRPSystem_idl
#define CSIRPSystem_idl

#include "ManagedGenericIRPSystem.idl"
#include "ManagedGenericIRPConstDefs.idl"
#include "CSIRPConstDefs.idl"

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

/* ## Module: CSIRPSystem
This module implements capabilities of CSIRP.
================================================================
*/
module CSIRPSystem
{

  /**
  * The InvalidHeartbeatPeriod exception is used when the period
  * value to be set by IRPManager is not a reasonable in IRPAgent's
  * implementation. A very short period may cause IRPAgent to
  * send many heartbeat notification in a short time, which may
  * decrease the performance of IRPAgent. To prevent this,
  * IRPAgent may set the lower limit period in its system
  * implemntation. When the period to be set is shorter the
  * lower limit period, IRPAgent may throw this exception
  * and reject to set the period to new value.
  * Note: set the period to zero must be allowed. The behaviour of
  * setting period to zero pls see definition for Period.
```

```
*/
exception InvalidHeartbeatPeriod
{
   unsigned short periodLowerLimit;
   string reason;
};



exception InvalidManagerIdentifier { string reason; };
exception ConflictingHeartbeatPeriod { string reason; };

/*
 System fails to complete the operation.   System can provide reason
 to qualify the exception.   The semantics carried in reason
 is outside the scope of this IRP.
 */
 exception GetHeartbeatPeriod { string reason; };
 exception SetHeartbeatPeriod { string reason; };
 exception TriggerHeartbeat { string reason; };
 exception GetCSIRPVersions { string reason; };
 exception GetCSIRPOperationsProfile { string reason; };
 exception GetCSIRPNotificationProfile { string reason; };

 interface CSIRP
 {
    /**
    * IRPManager invokes this operation to obtain the current
    * heartbeat period.
    */
    CSIRPConstDefs::ResultType       get_heartbeat_period(
       out CSIRPConstDefs::HeartbeatPeriodType heartbeatPeriod
    )
    raises (GetHeartbeatPeriod);

    /**
    * IRPManager invokes this operation to set the heartbeatPeriod.
```

```
* If the heartbeatPeriod is modified by one IRPManager, a
* Communication Surveillance notification should be emitted
* immediately to all the subscribed IRPManagers to indicate
* the new heartbeatPeriod. If the heartbeatPeriod is set to
* zero, one Communication Surveillance notification will be
* emitted immediately and no more Communication Surveillance
* notifications unless the heartbeatPeriod is modified again.
*/
CSIRPConstDefs::ResultType      set_heartbeat_period(
   in CSIRPConstDefs::HeartbeatPeriodType heartbeatPeriod
)
raises (SetHeartbeatPeriod,
            ConflictingHeartbeatPeriod,
            InvalidHeartbeatPeriod);

/*
* IRPManager invoke this operation to trigger ET_HEARTBEAT
* notification positively.
*/
CSIRPConstDefs::ResultType trigger_heartbeat(
   in CSIRPConstDefs::ManagerIdentifierType managerIdentifier
)
raises (TriggerHeartbeat, InvalidManagerIdentifier);

/**
* Return the list of all supported CSIRP versions.
*/
ManagedGenericIRPConstDefs::VersionNumberSet get_CS_IRP_versions (
)
raises (GetCSIRPVersions);

/**
* Return the list of all supported operations and their supported
* parameters for a specific CSIRP version.
*/
ManagedGenericIRPConstDefs::MethodList get_CS_IRP_operations_profile (
      in ManagedGenericIRPConstDefs::VersionNumber iRPVersion
```

```
        )
        raises (GetCSIRPOperationsProfile,
                ManagedGenericIRPSystem::OperationNotSupported,
                ManagedGenericIRPSystem::InvalidParameter);

        /**
        * Return the list of all supported notifications and their supported
        * parameters for a specific CSIRP version.
        */
        ManagedGenericIRPConstDefs::MethodList get_CS_IRP_notification_profile (
            in ManagedGenericIRPConstDefs::VersionNumber iRPVersion
        )
        raises (GetCSIRPNotificationProfile,
                ManagedGenericIRPSystem::OperationNotSupported,
                ManagedGenericIRPSystem::InvalidParameter);


    };

};
#endif

//File "CSIRPNotifDefs.idl"
//The IRP document version number is "CS IRP V1.0"

#ifndef CSIRPNotifDefs_idl
#define CSIRPNotifDefs_idl

#include "CSIRPConstDefs.idl"
#include "NotificationIRPConstDefs.idl"


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

/* ## Module: CSIRPNotifDefs
This module contains the specification of all notifications of CS IRP Agent.
```

```
================================================================
*/
module CSIRPNotifDefs
{

  const string ET_HEARTBEAT = "notifyHeartbeat";

   /**
   *   Constant definitions for the FileReady notification
   */
   interface notifyHeartbeat: NotificationIRPConstDefs::AttributeNameValue
   {
     const string EVENT_TYPE = ET_HEARTBEAT;

     /**
     * This constant defines the name of the    period property,
     * which is transported in the filterable_body fields.
     * The data type for the value of this property
     * is CSIRPConstDefs::HeartbeatPeriodType.
     */
     const string HEARTBEAT_PERIOD = CSIRPConstDefs::AttributeNameValue::HEARTBEAT_PERIOD;

     /*
     * This constant defines the name of the
     * channelId property,
     * which is transported in the filterable_body
     * fields.
     * The data type for the value of this property
     * is CSIRPConstDefs::ChannelIdType.
     */
     const string CHANNEL_ID = CSIRPConstDefs::AttributeNameValue::CHANNEL_ID;

     /*
     * This constant defines the name of the
     * triggerFlag property,
     * which is transported in the filterable_body
     * fields.
```

```
        * The data type for the value of this property
        * is CSIRPConstDefs::TriggerFlagType.
        */
         const string TRIGGER_FLAG = CSIRPConstDefs::AttributeNameValue::TRIGGER_FLAG;

        /*
        * This constant defines the name of the
        * managerIdentifier property,
        * which is transported in the filterable_body
        * fields.
        * The data type for the value of this property
        * is CSIRPConstDefs::ManagerIdentifierType.
        */
        const string MANAGER_IDENTIFIER = CSIRPConstDefs::AttributeNameValue::MANAGER_IDENTIFIER;
    };

};
#endif
```

A.5 通知日志管理接口信息模型

待补充

A.6 文件传输接口信息模型

```
//File "FileTransferIRPConstDefs.idl"
//The IRP document version number is "FileTransfer IRP V1.0"

#ifndef FileTransferIRPConstDefs_idl
#define FileTransferIRPConstDefs_idl

#include <TimeBase.idl>

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

/* ## Module: FileTransferIRPConstDefs
This module contains commonly used definitions for FileTransferIRP.
================================================================
*/
```

```
module FileTransferIRPConstDefs
{
  enum ResultType {OK, Failure};

  typedef TimeBase::UtcT UTCTimeType;
  typedef UTCTimeType BeginTimeType;
  typedef UTCTimeType EndTimeType;

  enum LocationChoice {Directory, URL};

  //The FileLocationType may be a directory path or a URL
  union FileLocationType switch (LocationChoice)
  {
    case Directory:   string fileLocationD;
    /* e.g. \\202.112.101.1\D:\user\performanceFiles\<fileName> */
    case URL: string fileLocationU;
    /* e.g. ftp://nms.telecom_org.com/datastore/<fileName> */
  };

  typedef unsigned long FileSizeType; //the unit is byte
  typedef string FileCompressionType;
  typedef string FileFormatType;

  struct FileInfoType
  {
    FileLocationType fileLocation;
    FileSizeType fileSize;
    UTCTimeType fileReadyTime;
    UTCTimeType fileExpirationTime;
    FileCompressionType fileCompression;
    FileFormatType fileFormat;
  };

  typedef sequence<FileInfoType> FileInfoListType;

  enum ManagementDataTypeType {PM, CM, IM, TE, CT, NL, CG, OT};
```

```
   /*
   Define the parameters specified in the notifyFileReady
   and notifyFilePreparationError notification.
   */
   interface AttributeNameValue
   {
       const string FILE_INFO_LIST = "FILE_INFO_LIST";
       const string ADDITIONAL_TEXT = "ADDITIONAL_TEXT";
       const string REASON = "REASON";
   };

};

#endif

//File "FileTransferIRPSystem.idl"
//The IRP document version number is "FileTransfer IRP V1.0"

#ifndef FileTransferIRPSystem_idl
#define FileTransferIRPSystem_idl

#include "ManagedGenericIRPConstDefs.idl"
#include "ManagedGenericIRPSystem.idl"
#include "FileTransferIRPConstDefs.idl"

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

/* ## Module: FileTransferIRPSystem
This module implements capabilities of FileTransferIRP.
================================================================
*/
module FileTransferIRPSystem
{

   exception InvalidTimes { string reason; };
   exception InvalidFileInfoList { string reason; };
```

```
/*
 System fails to complete the operation.  System can provide reason
 to qualify the exception.  The semantics carried in reason
 is outside the scope of this IRP.
*/
exception ListAvailableFiles { string reason; };
exception FileDownloadIndication { string reason; };
exception GetFileTransferIRPVersions { string reason; };
exception GetFileTransferIRPOperationsProfile { string reason; };
exception GetFileTransferIRPNotificationsProfile { string reason; };

/*
*/
 interface FileTransferIRP
 {
   /*
   * IRPManager invoke this operation to get the files information
   */
   FileTransferIRPConstDefs::ResultType      list_available_files(
     in FileTransferIRPConstDefs::ManagementDataTypeType managementDataType,
     in FileTransferIRPConstDefs::BeginTimeType beginTime,
     in FileTransferIRPConstDefs::EndTimeType endTime,
     out FileTransferIRPConstDefs::FileInfoListType fileInfoList
   )
   raises (ListAvailableFiles,InvalidTimes);

   /*
   * IRPManager invoke this operation to indicate completion of downloading files to IRPAgent
   */
   FileTransferIRPConstDefs::ResultType      file_download_indication(
     in FileTransferIRPConstDefs::FileInfoListType fileInfoList
   )
   raises (FileDownloadIndication,InvalidFileInfoList);

    /**
    * Return the list of all supported FileTransferIRP versions.
```

```
        */
        ManagedGenericIRPConstDefs::VersionNumberSet get_FileTransfer_IRP_versions (
        )
        raises (GetFileTransferIRPVersions);


        /**
        * Return the list of all supported operations and their supported
        * parameters for a specific FileTransferIRP version.
        */
        ManagedGenericIRPConstDefs::MethodList get_FileTransfer_IRP_operations_profile (
            in ManagedGenericIRPConstDefs::VersionNumber iRPVersion
        )
        raises (GetFileTransferIRPOperationsProfile,
                    ManagedGenericIRPSystem::OperationNotSupported,
                    ManagedGenericIRPSystem::InvalidParameter);

        /**
        * Return the list of all supported notifications and their supported
        * parameters for a specific FileTransferIRP version.
        */
        ManagedGenericIRPConstDefs::MethodList get_FileTransfer_IRP_notifications_profile
        (
            in ManagedGenericIRPConstDefs::VersionNumber iRPVersion
        )
        raises (GetFileTransferIRPNotificationsProfile,
                    ManagedGenericIRPSystem::OperationNotSupported,
                    ManagedGenericIRPSystem::InvalidParameter);
    };

};

#endif

//File "FileTransferIRPNotifications.idl"
//The IRP document version number is "FileTransfer IRP V1.0"
```

```
#ifndef _FILETRANSFERIRPNOTIFICATIONS_IDL_
#define _FILETRANSFERIRPNOTIFICATIONS_IDL_

#include "FileTransferIRPConstDefs.idl"
#include "NotificationIRPConstDefs.idl"


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

/* ## Module: FileTransferIRPNotifDefs
This module contains the specification of all notifications of FileTransfer IRP Agent.
================================================================
*/
module FileTransferIRPNotifications
{

  const string ET_FILE_READY = "notifyFileReady";
  const string ET_FILE_PREPARATION_ERROR = "notifyFilePreparationError";

   /**
   *   Constant definitions for the FileReady notification
   */
   interface notifyFileReady: NotificationIRPConstDefs::AttributeNameValue
   {
     const string EVENT_TYPE = ET_FILE_READY;

     /**
     * This constant defines the name of the fileInfoList property,
     * which is transported in the filterable_body fields.
     * The data type for the value of this property is
     * FileTransferIRPConstDefs::FileInfoListType.
     */
     const string FILE_INFO_LIST =
          FileTransferIRPConstDefs::AttributeNameValue::FILE_INFO_LIST;

     /**
```

```
    * This constant defines the name of the additionalText property,
    * which is transported in the filterable_body fields.
    * The data type for the value of this property is string.
    */
    const string ADDITIONAL_TEXT =
        FileTransferIRPConstDefs::AttributeNameValue::ADDITIONAL_TEXT;
    };

/**
*   Constant definitions for the FilePreparationError notification
*/
interface notifyFilePreparationError: NotificationIRPConstDefs::AttributeNameValue
{
    const string EVENT_TYPE = ET_FILE_PREPARATION_ERROR;

    /**
    * This constant defines the name of the fileInfoList property,
    * which is transported in the filterable_body fields.
    * The data type for the value of this property is
    * FileTransferIRPConstDefs::FileInfoListType.
    */
    const string FILE_INFO_LIST =
        FileTransferIRPConstDefs::AttributeNameValue::FILE_INFO_LIST;

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

    /**
    * This constant defines the name of the additionalText property,
    * which is transported in the filterable_body fields.
    * The data type for the value of this property is string.
    */
```

```
        const string ADDITIONAL_TEXT =
            FileTransferIRPConstDefs::AttributeNameValue::ADDITIONAL_TEXT;
        };

};
#endif //_FILETRANSFERIRPNOTIFICATIONS_IDL_
```

A.7 公共配置管理接口信息模型

KernelCM IRP

```
//File "KernelCmIRPSystem.idl"
//The IRP document version number is "KernelCM IRP V1.0"

#ifndef KernelCmIRPSystem_idl
#define KernelCmIRPSystem_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

//File "KernelCmNotifDefs.idl"
//The IRP document version number is "KernelCM IRP V1.0"

#ifndef KernelCmNotifDefs_idl
#define KernelCmNotifDefs_idl

#include <TimeBase.idl>              // CORBA Time Service
#include "NotificationIRPConstDefs.idl"
#include "StateManagementIRPConstDefs.idl"

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

module KernelCmNotifDefs
{
        /**
         * Definition of ITU-T defined semantics.
         * These constants are used in the type_name
         * (header.fixed_header.event_type.type_name)
         * field to denote the notification type
         * Note all values are unique among themselves.   Other IRP documents
         * cannot use the same values.
         */

        const string ET_OBJECT_CREATION = "x6";
        const string ET_OBJECT_DELETION = "x7";
        const string ET_ATTRIBUTE_VALUE_CHANGE = "x8";
```

```
  const string ET_CM_SYNCHRONIZATION_RECOMMENDED = "x9";
  const string ET_STATE_CHANGE = "xA";

  /**
   * 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 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
     */
    /*
    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";
    };

    interface NotificationCommon
    {
        /**
         * This constant defines a field in the filterable
         * information in a StructuredEvent.
         * This string is mapped to the name part of a
         * Property in the event and the value part will
         * carry the MO class name represented
         * as a string.
         */
```

```
const string MANAGED_OBJECT_CLASS =
  NotificationIRPConstDefs::AttributeNameValue::MANAGED_OBJECT_CLASS;


/**
 * This constant defines a field in the filterable
 * information in a StructuredEvent.
 * This string is mapped to the name part of a
 * Property in the event and the value part will
 * carry the MO distinguished name represented
 * as a string.
 */
const string MANAGED_OBJECT_INSTANCE =
  NotificationIRPConstDefs::AttributeNameValue::MANAGED_OBJECT_INSTANCE;

/**
 *  This constant defines the name of the notification
 *  ID property, which is transported in the
 *  filterable_body_fields
 */
const string NOTIFICATION_ID =
  NotificationIRPConstDefs::AttributeNameValue::NOTIFICATION_ID;

/**
 *  This constant defines the name of the
 *  event time property, which is transported in the
 *  filterable_body_fields.
 *  The data type for the value of this property
 *  is defined by datatype CommonIRPConstDefs::IRPTime
 */
const string EVENT_TIME =
  NotificationIRPConstDefs::AttributeNameValue::EVENT_TIME;

/**
 *  This constant defines the name of the
 *  system name property, which is transported in the
 *  filterable_body_fields
```

```
 */
const string SYSTEM_DN =
   NotificationIRPConstDefs::AttributeNameValue::SYSTEM_DN;

/**
 *   This constant defines the name of the
 *   source indicator    property, which is transported in the
 *   filterable_body_fields
 */
const string SOURCE_INDICATOR =
   KernelCmNotifDefs::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,
 *   which is transported in the filterable_body
 *   fields.
 *   The data type for the value of this property
 *   is a string.
 */
const string ADDITIONAL_TEXT =
   KernelCmNotifDefs::AttributeNameValue::ADDITIONAL_TEXT;


/**
 *   This constant defines the name of the
 *   correlated notifications property,
 *   which is transported in the
 *   filterable_body_fields
```

```
        *   The value part of the property is
        *       KernelCmNotifDefs::CorrelatedNotificationSetType
        */
    const string CORRELATED_NOTIFICATIONS =
      KernelCmNotifDefs::AttributeNameValue::CORRELATED_NOTIFICATIONS;
};


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

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


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

    /**
     * This information mapped into the remainder_of_body
     * in the StructuredEvent
     */
```

```
    typedef MOAttributeSet InitialAttributeValues;
};


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

    /**
     * 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
{
    const string EVENT_TYPE =
      KernelCmNotifDefs::ET_CM_SYNCHRONIZATION_RECOMMENDED;

   /**
     * This constant defines a field in the filterable
     * information in a StructuredEvent.
     * This string is mapped to the name part of a
     * Property in the event and the value part will
     * carry the MO class name represented
     * as a string.
     */
    const string MANAGED_OBJECT_CLASS =
      NotificationIRPConstDefs::AttributeNameValue::MANAGED_OBJECT_CLASS;

    /**
     * This constant defines a field in the filterable
     * information in a StructuredEvent.
     * This string is mapped to the name part of a
     * Property in the event and the value part will
     * carry the MO distinguished name represented
     * as a string.
     */
    const string MANAGED_OBJECT_INSTANCE =
      NotificationIRPConstDefs::AttributeNameValue::MANAGED_OBJECT_INSTANCE;

    /**
     *   This constant defines the name of the notification
     *   ID property, which is transported in the
     *   filterable_body_fields
```

```
 */
const string NOTIFICATION_ID =
  NotificationIRPConstDefs::AttributeNameValue::NOTIFICATION_ID;

/**
 *  This constant defines the name of the
 *  event time property, which is transported in the
 *  filterable_body_fields.
 *  The data type for the value of this property
 *  is defined by datatype CommonIRPConstDefs::IRPTime
 */
const string EVENT_TIME =
  NotificationIRPConstDefs::AttributeNameValue::EVENT_TIME;

/**
 *  This constant defines the name of the
 *  system name property, which is transported in the
 *  filterable_body_fields
 */
const string SYSTEM_DN =
  NotificationIRPConstDefs::AttributeNameValue::SYSTEM_DN;


/**
 *  This constant defines the name of the
 *  additional text property,
 *  which is transported in the filterable_body
 *  fields.
 *  The data type for the value of this property
 *  is a string.
 */
const string ADDITIONAL_TEXT =
  KernelCmNotifDefs::AttributeNameValue::ADDITIONAL_TEXT;

 /**
 *  This constant defines the name of the
 *  base MO class property,
```

```
        *   which is transported in the filterable_body
        *   fields.
        *   The value part of this property will carry
        *   the base MO class name as a string.
        */
       const string BASE_MO_CLASS =
         KernelCmNotifDefs::AttributeNameValue::BASE_MO_CLASS;

       /**
        *   This constant defines the name of the
        *   base MO instance property,
        *   which is transported in the filterable_body
        *   fields.
        *   The value part of this property will carry
        *   the base MO distinguished name as a string.
        */
       const string BASE_MO_INSTANCE =
         KernelCmNotifDefs::AttributeNameValue::BASE_MO_INSTANCE;

       /**
        *   This constant defines the name of the
        *   scope property,
        *   which is transported in the filterable_body
        *   fields.
        *   The data type for the value of this property
        *   is KernelCmNotifDefs::ScopePara.
        */
       const string SCOPE =
         KernelCmNotifDefs::AttributeNameValue::SCOPE;

   };

   /**
    *   Constant definitions for the State Change notification
    */
   interface StateChange : NotificationCommon
```

```
        {
            const string EVENT_TYPE =
              KernelCmNotifDefs::ET_STATE_CHANGE;
        };


};

#endif
```

StateManagement

```
//File "StateManagementIRPConstDefs.idl"
//The IRP document version number is "StateM IRP V1.0"

#ifndef StateManagementIRPConstDefs_idl
#define StateManagementIRPConstDefs_idl

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

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

/* ## Module: StateManagementIRPConstDefs
This module contains commonly used definitions for State Management IRP
================================================================================
*/
module StateManagementIRPConstDefs
{
/*
   Constant definitions for state management notifications uses when populating the
   Cos::Structured event.
   The "name" party of the structured event carries the following constant definitions
   appropriate to the state being notified.
   Refer to TS 32.663 regarding how to populate the structured event
*/
    interface AttributeNameValue {
```

```
        const string OPERATIONAL_STATE      = "operationalState";
        const string USAGE_STATE            = "usageState";
        const string ADMINISTRATIVE_STATE = "administrativeState";
        const string ALARM_STATUS           = "alarmStatus";
        const string PROCEDURAL_STATUS      = "proceduralStatus";
        const string AVAILABILITY_STATUS    = "availabilityStatus";
        const string CONTROL_STATUS         = "controlStatus";
        const string STANDBY_STATUS         = "standbyStatus";
        const string UNKNOWN_STATUS         = "unknownStatus";
    };


/*
    The following structures provide the new state value,
    and the optional old state value
    The structures are passed in the value part of the cos structured event
*/

    /*
    Definition of Operational State based on X.721 [3], if mandatory.
    */
    enum OperationalState
    {
        Disabled, Enabled
    };

    union OperationalStateTypeOpt switch(boolean)
    {
        case TRUE: StateManagementIRPConstDefs::OperationalState operational_state;
    };

    struct OperationalStateOldNewValue{
        OperationalState new;
        StateManagementIRPConstDefs::OperationalStateTypeOpt old;
    };
```

```
/*
Definition of Usage State based on X.721 [3], if mandatory.
*/
enum UsageState
{
    Idle, Active, Busy
};

union UsageStateTypeOpt switch(boolean)
{
    case TRUE: StateManagementIRPConstDefs::UsageState usage_state;
};

struct UsageStateOldNewValue{
    UsageState new;
    StateManagementIRPConstDefs:: UsageStateTypeOpt old;
};


/*
Definition of Administrative State based on X.721 [3], if mandatory.
*/
enum AdministrativeState
{
    Locked, Unlocked, ShuttingDown
};

union AdministrativeStateTypeOpt switch(boolean)
{
    case TRUE: StateManagementIRPConstDefs::AdministrativeState administrative_state;
};

struct AdministrativeStateOldNewValue{
    AdministrativeState new;
    StateManagementIRPConstDefs:: AdministrativeStateTypeOpt old;
};
```

```
/*
Definition of Alarm Status based on M.3100 [4], if mandatory.
*/
enum AlarmStatusValues
{
    CLEARED, INDETERMINATE, WARNING, MINOR, MAJOR, CRITICAL
};
typedef sequence <AlarmStatusValues,5> AlarmStatus;

union AlarmStatusTypeOpt switch(boolean)
{
    case TRUE: StateManagementIRPConstDefs::AlarmStatus alarm_status;
};

struct AlarmStatusOldNewValue{
    AlarmStatusValues new;
    StateManagementIRPConstDefs:: AdministrativeStateTypeOpt old;
};

/*
Definition of Procedural Status based on X.721 [3], if mandatory.
*/
enum ProceduralStatusValues
{
    InitializationRequired, NotInitialized, Initializing, Reporting,
    Terminating
};
typedef sequence <ProceduralStatusValues,5> ProceduralStatus;

union ProceduralStatusTypeOpt switch(boolean)
{
    case TRUE: StateManagementIRPConstDefs::ProceduralStatus procedural_status;
};

struct ProceduralStatusOldNewValue{
```

```
    ProceduralStatusValues new;
    StateManagementIRPConstDefs:: ProceduralStatusTypeOpt old;
};


/*
Definition of Availability Status based on X.721 [3], if mandatory.
*/
enum AvailabilityStatusValues
{
    InTest, Failed, PowerOff, OffLine, OffDuty, Dependency, Degraded,
    NotInstalled, LogFull
};
typedef sequence <AvailabilityStatusValues,9> AvailabilityStatus;

union AvailabilityStatusTypeOpt switch(boolean)
{
    case TRUE: StateManagementIRPConstDefs::AvailabilityStatus availability_status;
};

struct AvailabilityStatusOldNewValue{
    AvailabilityStatusValues new;
    StateManagementIRPConstDefs:: AvailabilityStatusTypeOpt old;
};


/*
Definition of Control Status based on X.721 [3], if mandatory.
*/
enum ControlStatusValues
{
    SubjectToTest, PartOfServicesLocked, ReservedForTest, Suspended
};
typedef sequence <ControlStatusValues,4> ControlStatus;

union ControlStatusTypeOpt switch(boolean)
{
```

```
    case TRUE: StateManagementIRPConstDefs::ControlStatus control_status;
};

struct ControlStatusOldNewValue{
    ControlStatusValues new;
    StateManagementIRPConstDefs:: ControlStatusTypeOpt old;
};


/*
Definition of Standby Status based on X.721 [3], if mandatory.
*/
enum StandbyStatus
{
    HotStandby, ColdStandby, ProvidingService
};

union StandbyStatusTypeOpt switch(boolean)
{
    case TRUE: StateManagementIRPConstDefs::StandbyStatus standby_status;
};

struct StandbyStatusOldNewValue{
    StandbyStatus new;
    StateManagementIRPConstDefs:: StandbyStatusTypeOpt old;
};


/*
Definition of Unknown Status based on X.721 [3], if mandatory
(if switch is TRUE then value equal to TRUE implies "unknown status").
*/
union UnknownStatus switch(boolean)
{
    case TRUE: boolean value;
};
```

```
    union UnknownStatusTypeOpt switch(boolean)
    {
        case TRUE: StateManagementIRPConstDefs::UnknownStatus unknown_status;
    };

    struct UnknownStatusOldNewValue{
        UnknownStatus new;
        StateManagementIRPConstDefs:: UnknownStatusTypeOpt old;
    };
};
#endif
```

A.8 基本配置管理接口信息模型

BasicCM IRP

```
//File "BasicCmIRPSystem.idl"
//The IRP document version number is "BasicCM IRP V1.0"

#ifndef BasicCmIRPSystem_idl
#define BasicCmIRPSystem_idl

#include "ManagedGenericIRPConstDefs.idl"
#include "ManagedGenericIRPSystem.idl"

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

module BasicCmIRPSystem
{

    /**
     * Defines the name of a Managed Object Class
     */
    typedef string MOClass;


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

```
 */
typedef string DN;


/**
 * Defines the name of an attribute of a Managed Object
 */
typedef string MOAttributeName;


/**
 * Defines the value of an attribute of a Managed Object in form of a CORBA
 * Any. Apart from basic datatypes already defined in CORBA, the allowed
 * attribute value types are defined in the AttributeTypes module.
 */
typedef any MOAttributeValue;


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


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

};

 exception IllegalFilterFormatException {
     string reason;
 };
```

```
 exception IllegalDNFormatException {
     string reason;
 };
 exception IllegalScopeTypeException {
     string reason;
 };
 exception IllegalScopeLevelException {
     string reason;
 };
 exception UndefinedMOException {
     string reason;
 };

exception UndefinedScopeException {
     string reason;
};

exception FilterComplexityLimit {
    string reason;
};

exception DuplicateMO {};

exception CreateNotAllowed {};

exception ObjectClassMismatch {};

exception NoSuchObjectClass {
    MOClass objectClass;
};

 exception ParentObjectDoesNotExist {};

/**
 * System otherwise fails to complete the operation.  System can provide
 * reason to qualify the exception. The semantics carried in reason
 * is outside the scope of this IRP.
```

```
 */
exception NextBasicCmInformations { string reason; };
exception NextDeleteErrors { string reason; };
exception NextModifyErrors { string reason; };
exception DestroyException { string reason; };
exception GetBasicCmIRPVersion { string reason; };
exception GetBasicCmIRPOperationProfile { string reason; };
exception GetBasicCmIRPNotificationProfile { string reason; };
exception FindManagedObjects { string reason; };
exception CreateManagedObject { string reason; };
exception DeleteManagedObjects { string reason; };
exception ModifyManagedObjects { string reason; };


typedef string FilterType;

/**
 * ResultContents is used to tell how much information to get back
 * from the find_managed_objects operation.
 *
 * NAMES: Used to get only Distinguished Name
 *        for MOs.
 *        The name contains both the MO class
 *        and the names of all superior objects in the naming
 *        tree.
 *
 * NAMES_AND_ATTRIBUTES: Used to get both NAMES plus
 *     MO attributes (all or selected).
 */
enum ResultContents
{
   NAMES,
   NAMES_AND_ATTRIBUTES
};

/**
 * ScopeType defines the kind of scope to use in a search
```

```
 * together with SearchControlType.level, in a SearchControlType value.
 *
 * SearchControlType.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
};

/**
 * SearchControlType controls the find_managed_object search,
 * and contains:
 * the type of scope ("type" field),
 * the level of scope ("level" field), level 0 means the "baseObject",
 *       level 1 means baseobject including its sub-ordinates etc..
 * the filter ("filter" field),
 * the resu!t type ("contents" field).
 * The type, level and contents fields are all mandatory.
 * The filter field contains the filter expression.
 *   The string "TRUE" indicates "no filter",
 * i.e. a filter that matches everything.
 */
struct SearchControlType
{
    ScopeType type;
    unsigned long level;
```

```
        FilterType filter;
        ResultContents contents;
    };

    /**
     * Represents an attribute: "name" is the attribute name
     * and "value" is the attribute value.
     */
    struct MOAttribute
    {
        MOAttributeName name;
        MOAttributeValue value;
    };

    typedef sequence<MOAttribute> MOAttributeSet;


    struct Result
    {
        DN mo;
        MOAttributeSet attributes;
    };

    typedef sequence<Result> ResultSet;


    /**
     * AttributeErrorCategory defines the categories of errors, related to
     * attributes, that can occur during creation or modification of MOs.
     *
     * NO_SUCH_ATTRIBUTE: The specified attribute does not exist.
     * INVALID_ATTRIBUTE_VALUE: The specified attribute value is not valid.
     * MISSING_ATTRIBUTE_VALUE: An attribute value is required but none was
     *     provided and no default value is defined for the attribute.
     * INVALID_MODIFY_OPERATOR: The specified modify operator is not valid
     *     (e.g. operator ADD_VALUES applied to a non multi-valued attribute
```

```
 *     or operator SET_TO_DEFAULT applied where no default value is defined).
 * MODIFY_NOT_ALLOWED: The modification of the attribute is not allowed.
 * MODIFY_FAILED: The modification failed because of an unspecified reason.
 */
enum AttributeErrorCategory
{
    NO_SUCH_ATTRIBUTE,
    INVALID_ATTRIBUTE_VALUE,
    MISSING_ATTRIBUTE_VALUE,
    INVALID_MODIFY_OPERATOR,
    MODIFY_NOT_ALLOWED,
    MODIFY_FAILED
};


/**
 * DeleteErrorCategory defines the categories of errors that can occur
 * during deletion of MOs.
 *
 * SUBORDINATE_OBJECT: The MO cannot be deleted due to subordinate MOs.
 * DELETE_NOT_ALLOWED: The deletion of the MO is not allowed.
 * DELETE_FAILED: The deletion failed because of an unspecified reason.
 */
enum DeleteErrorCategory
{
    SUBORDINATE_OBJECT,
    DELETE_NOT_ALLOWED,
    DELETE_FAILED
};


/**
 * AttributeError represents an error, related to an attribute, that occured
 * during creation or modification of MOs.
 * It contains:
 * - the name of the indicted attribute ("name" field),
 * - the category of the error ("error" field),
```

```
 * - optionally, the indicted attribute value ("value" field),
 * - optionally, additional details on the error ("reason" field).
 */
struct AttributeError
{
    MOAttributeName name;
    AttributeErrorCategory error;
    MOAttributeValue value;
    string reason;
};

typedef sequence<AttributeError> AttributeErrorSeq;


/**
 * DeleteError represents an error that occured during deletion of MOs.
 * It contains:
 * - the distinguished name of the indicted MO ("objectName" field),
 * - the category of the error ("error" field),
 * - optionally, additional details on the error ("reason" field).
 */
struct DeleteError
{
    DN objectName;
    DeleteErrorCategory error;
    string reason;
};

typedef sequence<DeleteError> DeleteErrorSeq;


/**
 * ModifyAttributeErrors represents errors that occured during
 * modification of attributes of a MO.
 * It contains:
 * - the distinguished name of the indicted MO ("objectName" field),
 * - a sequence containing the attribute errors ("errors" field).
```

```
 */
struct ModifyAttributeErrors
{
    DN objectName;
    AttributeErrorSeq errors;
};

typedef sequence<ModifyAttributeErrors> ModifyAttributeErrorsSeq;


/**
 The BasicCmInformationIterator is used to iterate through a snapshot of
  Managed Object Information when IRPManager invokes find_managed_objects.
  IRPManager uses it to pace the return of Managed Object Information.

 IRPAgent controls the life-cycle of the iterator. However, a destroy
  operation is provided to handle the case where IRPManager wants to stop
  the iteration procedure before reaching the last iteration.
 */
interface BasicCmInformationIterator
{

    /**
     This method returns between 1 and "how_many" Managed Object information.
      The IRPAgent may return less than "how_many" items even if there are
      more items to return. "how_many" must be non-zero. Return TRUE if there
      may be more Managed Object information to return. Return FALSE if there
      are no more Managed Object information to be returned.

      If FALSE is returned, the IRPAgent will automatically destroy the
       iterator.

      @parm how_many how many elements to return in the "fetchedElements" out
       parameter.
      @parm fetchedElements the elements.
      @returns A boolean indicating if any elements are returned.
       "fetchedElements" is empty when the BasicCmInformationIterator is
```

```
        empty.
      */

      boolean next_basicCmInformations (
        in unsigned short how_many,
        out ResultSet fetchedElements
      )
      raises (NextBasicCmInformations,
              ManagedGenericIRPSystem::InvalidParameter);

      /**
      This method destroys the iterator.
      */

      void destroy ()
      raises (DestroyException);

};// end of BasicCmInformationIterator


/**
 The DeleteResultIterator is used to iterate through the list of deleted MOs
  when IRPManager invokes method "delete_managed_objects".
  IRPManager uses it to pace the return of Managed Object Information.

 IRPAgent controls the life-cycle of the iterator. However, a destroy
  operation is provided to handle the case where IRPManager wants to stop
  the iteration procedure before reaching the last iteration.
 */
interface DeleteResultIterator : BasicCmInformationIterator
{

    /**
      Inherited method "next_basicCmInformations" has the same behaviour as
       for interface BasicCmInformationIterator, except that:
       - The Managed Object information returned in parameter
         "fetchedElements" contains only the DNs of the deleted MOs
```

```
        (no attributes are returned).
      - If FALSE is returned, the IRPAgent will not automatically destroy the
        iterator.
    */

  /**
    This method returns between 0 and "how_many" deletion errors. The
    IRPAgent may return less than "how_many" items even if there are more
    items to return. "how_many" must be non-zero. Return TRUE if there are
    more deletion errors to return. Return FALSE if there are no more
    deletion errors to be returned.

    If FALSE is returned and last call to inherited method
    "next_basicCmInformations" also returned FALSE (i.e. no more Managed
    Object information to be returned), the IRPAgent will automatically
    destroy the iterator.

    @parm how_many: how many deletion errors to return in the
     "fetchedDeleteErrors" out parameter.
    @parm fetchedDeleteErrors: the deletion errors.
    @returns: a boolean indicating if any deletion errors are returned.
    */

    boolean next_deleteErrors (
      in unsigned short how_many,
      out DeleteErrorSeq fetchedDeleteErrors
    )
    raises (NextDeleteErrors,
            ManagedGenericIRPSystem::InvalidParameter);

}; // end of DeleteResultIterator


/**
  The ModifyResultIterator is used to iterate through the list of modified
    MOs when IRPManager invokes method "modify_managed_objects".
    IRPManager uses it to pace the return of Managed Object Information.
```

```
  IRPAgent controls the life-cycle of the iterator. However, a destroy
    operation is provided to handle the case where IRPManager wants to stop
    the iteration procedure before reaching the last iteration.
  */
interface ModifyResultIterator : BasicCmInformationIterator
{

    /**
     Inherited method "next_basicCmInformations" has the same behaviour as
      for interface BasicCmInformationIterator, except that:
      - The Managed Object information returned in parameter
        "fetchedElements" contains DNs and attributes of the modified MOs.
      - If FALSE is returned, the IRPAgent will not automatically destroy the
        iterator.
     */

    /**
     This method returns between 0 and "how_many" modification errors. The
      IRPAgent may return less than "how_many" items even if there are more
      items to return. "how_many" must be non-zero. Return TRUE if there are
      more modification errors to return. Return FALSE if there are no more
      modification errors to be returned.

      If FALSE is returned and last call to inherited method
      "next_basicCmInformations" also returned FALSE (i.e. no more Managed
      Object information to be returned), the IRPAgent will automatically
      destroy the iterator.

      @parm how_many: how many modification errors to return in the
       "fetchedModifyErrors" out parameter.
      @parm fetchedModifyErrors: the modification errors.
      @returns: a boolean indicating if any modification errors are returned.
     */

     boolean next_modificationErrors (
        in unsigned short how_many,
```

```
        out ModifyAttributeErrorsSeq fetchedModifyErrors
    )
    raises (NextModifyErrors,
            ManagedGenericIRPSystem::InvalidParameter);

}; // end of ModifyResultIterator


typedef sequence<MOAttributeName> AttributeNameSet;


/**
 * ModifyOperator defines the way in which an attribute value is to be
 * applied to an attribute in a modification of MO attributes.
 *
 * REPLACE: replace the current value with the provided value
 * ADD_VALUES: for a multi-valued attribute, add the provided values to the
 *     current list of values
 * REMOVE_VALUES: for a multi-valued attribute, remove the provided values
 *     from the current list of values
 * SET_TO_DEFAULT: set the attribute to its default value
 */
enum ModifyOperator
{
    REPLACE,
    ADD_VALUES,
    REMOVE_VALUES,
    SET_TO_DEFAULT
};


/**
 * AttributeModification defines an attribute value and the way it is to
 * be applied to an attribute in a modification of MO attributes.
 * It contains:
 * - the name of the attribute to modify ("name" field),
 * - the value to apply to this attribute ("value" field),
```

```
     * - the way the attribute value is to be applied to the attribute
     *     ("operator" field).
     */
    struct AttributeModification
    {
        MOAttributeName name;
        MOAttributeValue value;
        ModifyOperator operator;
    };

    typedef sequence<AttributeModification> AttributeModificationSet;


/**
 * The BasicCmIrpOperations interface.
 * Supports a number of Resource Model versions.
 */
interface BasicCmIrpOperations
{
    /**
     * Get the version(s) of the interface
     *
     * @raises GetBasicCmIRPVersion when the system for some reason
     *     can not return the supported versions.
     * @returns all supported versions.
     */
    ManagedGenericIRPConstDefs::VersionNumberSet get_basicCm_IRP_version()
        raises (GetBasicCmIRPVersion);

    /**
     * Return the operation profile for a specific Basic CM IRP version.
     *
     * @raises GetBasicCmIRPOperationProfile when the system for some reason
     *     cannot return the supported operations and parameters.
     * @returns the list of all supported operations and their supported
     *     parameters for the specified version.
```

```
 */
ManagedGenericIRPConstDefs::MethodList get_basicCm_IRP_operation_profile
(
    in ManagedGenericIRPConstDefs::VersionNumber basicCm_IRP_version
)
raises (GetBasicCmIRPOperationProfile,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);

/**
 * Return the notification profile for a specific Basic CM IRP version.
 *
 * @raises GetBasicCmIRPNotificationProfile when the system for some
 *     reason cannot return the supported notifications and parameters.
 * @returns the list of all supported notifications and their supported
 *     parameters for the specified version.
 */
ManagedGenericIRPConstDefs::MethodList
    get_basicCm_IRP_notification_profile (
        in ManagedGenericIRPConstDefs::VersionNumber basicCm_IRP_version
)
raises (GetBasicCmIRPNotificationProfile,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);

/**
 * Performs a containment search, using a SearchControlType to
 * control the search and the returned results.
 *
 * All MOs in the scope constitute a set that the filter works on.
 * The result BasicCmInformationIterator contains all matched MOs,
 * with the amount of detail specified in the SearchControlType.
 * For the special case when no managed objects are matched in
 * find_managed_objects, the BasicCmInformationIterator will be returned.
 * Executing the next_basicCmInformations in the
 * BasicCmInformationIterator will return FALSE for
 * completion.
```

```
        *
        * @parm baseObject The start MO in the containment tree.
        * @parm searchControl the SearchControlType to use.
        * @parm requestedAttributes defines which attributes to get.
        *     If this parameter is empty (""),    all attributes shall
        *     be returned. In this version this is the only supported semantics.
        *     Note that this argument is only
        *     relevant if ResultContents in the search control is
        *     specifed to NAMES_AND_ATTRIBUTES.
        *
        *
        * @raises ManagedGenericIRPSystem::ValueNotSupported if a valid but
        * unsupported parameter value is passed. E.g. the contents
        * field in the searchcontrol parameter contains the value NAMES and
        * the optional getContainment IS operation is not supported.
        * @raises UndefinedMOException The MO does not exist.
        * @raises IllegalDNFormatException The dn syntax string is
        * malformed.
        * @raises IllegalScopeTypeException The ScopeType in scope contains
        * an illegal value.
        * @raises IllegalScopeLevelException The scope level is negative
        * (<0).
        * @raises IllegalFilterFormatException The filter string is
        * malformed.
        * @raises FilterComplexityLimit if the filter syntax is correct,
        *     but the filter is too complex to be processed by the IRP agent.
        * @see SearchControlType
        * @see BasicCmInformationIterator
        */
      BasicCmInformationIterator find_managed_objects(in DN baseObject,
                                                      in SearchControlType searchControl,
                                                      in AttributeNameSet requestedAttributes)
          raises (FindManagedObjects,
                  ManagedGenericIRPSystem::ParameterNotSupported,
                  ManagedGenericIRPSystem::InvalidParameter,
                  ManagedGenericIRPSystem::ValueNotSupported,
                  UndefinedMOException,
```

```
                IllegalDNFormatException,
                UndefinedScopeException,
                IllegalScopeTypeException,
                IllegalScopeLevelException,
                IllegalFilterFormatException,
                FilterComplexityLimit);

/**
 * Performs the creation of a MO instance in the MIB maintained
 * by the IRPAgent.
 *
 * @parm objectName: the distinguished name of the MO to create.
 * @parm referenceObject: the distinguished name of a reference MO.
 * @parm attributes: in input, initial attribute values for the MO to
 *     create; in output, actual attribute values of the created MO.
 * @parm attributeErrors: errors, related to attributes, that caused the
 *     creation of the MO to fail.
 *
 * @raises ManagedGenericIRPSystem::OperationNotSupported: The operation
 *     is not supported.
 * @raises ManagedGenericIRPSystem::ParameterNotSupported: An optional
 *     parameter is not supported.
 * @raises ManagedGenericIRPSystem::InvalidParameter: An invalid
 *     parameter value has been provided.
 * @raises UndefinedMOException: The MO does not exist.
 * @raises IllegalDNFormatException: The DN syntax string is malformed.
 * @raises DuplicateMO: A MO already exist with the same DN as the one
 *     to create.
 * @raises CreateNotAllowed: The creation of the MO is not allowed.
 * @raises ObjectClassMismatch: The object class of the MO to create does
 *     not match with the object class of the provided reference MO.
 * @raises NoSuchObjectClass: The class of the object to create is not
 *     recognized.
 * @raises ParentObjectDoesNotExist: The parent MO instance of the
 *    ManagedEntity specified to be created does not exist.
 */
void create_managed_object (
```

```
        in DN objectName,
        in DN referenceObject,
        inout MOAttributeSet attributes,
        out AttributeErrorSeq attributeErrors
    )
    raises (CreateManagedObject,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::ParameterNotSupported,
            ManagedGenericIRPSystem::InvalidParameter,
            UndefinedMOException,
            IllegalDNFormatException,
            DuplicateMO,
            CreateNotAllowed,
            ObjectClassMismatch,
            NoSuchObjectClass,
            ParentObjectDoesNotExist);

    /**
     * Performs the deletion of one or more MO instances from the MIB
     * maintained by the IRPAgent, using a SearchControlType to control the
     * instances to be deleted.
     *
     * All MOs in the scope constitute a set that the filter works on.
     * All matched MOs will be deleted by this operation.
     * The returned DeleteResultIterator is used to retrieve the DNs of the
     * MOs deleted and the errors that may have occurred preventing deletion
     * of some MOs.
     * For the special case when no managed objects are matched in
     * delete_managed_objects, the DeleteResultIterator will be returned.
     * Executing the next_basicCmInformations in the DeleteResultIterator
     * will return FALSE for completion.
     *
     * @parm baseObject: the start MO in the containment tree.
     * @parm searchControl: the SearchControlType to use; field "contents" has no
     *      meaning here and shall be ignored.
      @returns: a DeleteResultIterator (see above).
     *
```

```
 * @raises ManagedGenericIRPSystem::OperationNotSupported: The operation
 *     is not supported.
 * @raises ManagedGenericIRPSystem::InvalidParameter: An invalid
 *     parameter value has been provided.
 * @raises UndefinedMOException: The MO does not exist.
 * @raises IllegalDNFormatException: The DN syntax string is malformed.
 * @raises IllegalScopeTypeException: The ScopeType in scope contains
 *     an illegal value.
 * @raises IllegalScopeLevelException: The scope level is negative (<0).
 * @raises IllegalFilterFormatException: The filter string is malformed.
 * @raises FilterComplexityLimit: The filter syntax is correct,
 *     but the filter is too complex to be processed by the IRPAgent.
 */
DeleteResultIterator delete_managed_objects (
    in DN baseObject,
    in SearchControlType searchControl
)
raises (DeleteManagedObjects,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter,
            UndefinedMOException,
            IllegalDNFormatException,
            UndefinedScopeException,
            IllegalScopeTypeException,
            IllegalScopeLevelException,
            IllegalFilterFormatException,
            FilterComplexityLimit);

/**
 * Performs the modification of MO attributes. One or more MOs attributes
 * may be modified according to a SearchControlType.
 *
 * All MOs in the scope constitute a set that the filter works on.
 * All matched MOs will have their attributes modified by this operation.
 * The returned ModifyResultIterator is used to retrieve the DNs of the
 * modified MOs together with the values of the modified attributes, and
 * the errors that may have occurred preventing modification of some
```

```
     * attributes.
     * For the special case when no managed objects are matched in
     * modify_managed_objects, the ModifyResultIterator will be returned.
     * Executing the next_basicCmInformations in the ModifyResultIterator
     * will return FALSE for completion.
     *
     * @parm baseObject: the start MO in the containment tree.
     * @parm searchControl: the SearchControlType to use; field "contents" has no
           meaning here and shall be ignored.
     * @parm modifications: the values for the attributes to modify and
           the way those values are to be applied to the attributes.
      @returns: a ModifyResultIterator (see above).
     *
     * @raises ManagedGenericIRPSystem::OperationNotSupported: The operation
     *     is not supported
     * @raises ManagedGenericIRPSystem::InvalidParameter: An invalid
     *     parameter value has been provided
     * @raises UndefinedMOException: The MO does not exist.
     * @raises IllegalDNFormatException: The DN syntax string is malformed.
     * @raises IllegalScopeTypeException: The ScopeType in scope contains
     *     an illegal value.
     * @raises IllegalScopeLevelException: The scope level is negative (<0).
     * @raises IllegalFilterFormatException: The filter string is malformed.
     * @raises FilterComplexityLimit: The filter syntax is correct,
     *     but the filter is too complex to be processed by the IRPAgent.
     */
    ModifyResultIterator modify_managed_objects (
        in DN baseObject,
        in SearchControlType searchControl,
        in AttributeModificationSet modifications
    )
    raises (ModifyManagedObjects,
                ManagedGenericIRPSystem::OperationNotSupported,
                ManagedGenericIRPSystem::InvalidParameter,
                UndefinedMOException,
                IllegalDNFormatException,
                UndefinedScopeException,
```

```
                IllegalScopeTypeException,
                IllegalScopeLevelException,
                IllegalFilterFormatException,
                FilterComplexityLimit);

    };
};
#endif
```

A.9 故障管理接口信息模型

```
//File "AlarmIRPConstDefs.idl"
//The IRP document version number is "Alarm IRP V1.0"

#ifndef AlarmIRPConstDefs_idl
#define AlarmIRPConstDefs_idl

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

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

/* ## Module: AlarmIRPConstDefs
This module contains commonly used definitions for Alarm IRP
================================================================
*/
module AlarmIRPConstDefs
{
    /*
    The format of Distinguished Name (DN) is specified in 3GPP TS 32.300
    "Name Conventions for Managed Objects".
    */
    typedef string DN;

    /* DNTypeOpt is an optional type.
    If the discriminator is true the value is present.
    Otherwise the value is null.
    */
```

```
union DNTypeOpt switch (boolean)
{
    case TRUE: DN value;
};

/*
This block identifies the alarm types specified for this IRP version.
These types carry the same semantics as the TMN ITU-T defined event
types of the same name.
Their encodings for this version of Alarm IRP are defined here.  Other IRP
documents, or other versions of Alarm IRP, shall identify their own
alarm types for their use.  They shall define their encodings
as well.  Values defined here are unique among themselves.
*/
interface AlarmType
{
    const string COMMUNICATIONS_ALARM = "x1";
    const string PROCESSING_ERROR_ALARM = "x2";
    const string ENVIRONMENTAL_ALARM = "x3";
    const string QUALITY_OF_SERVICE_ALARM = "x4";
    const string EQUIPMENT_ALARM = "x5";
    const string INTEGRITY_VIOLATION = "x6";
    const string OPERATIONAL_VIOLATION = "x7";
    const string PHYSICAL_VIOLATION = "x8";
    const string SECURITY_SERVICE_OR_MECHANISM_VIOLATION = "x9";
    const string TIME_DOMAIN_VIOLATION = "x10";
};

/*
This block identifies the notification types defined by this
Alarm IRP version.
*/
interface NotificationType
{
    const string NOTIFY_FM_NEW_ALARM = "x1";
    const string NOTIFY_FM_CHANGED_ALARM = "x2";
    const string NOTIFY_FM_ACK_STATE_CHANGED = "x3";
```

```
    const string NOTIFY_FM_COMMENT_ADDED = "x4";
    const string NOTIFY_FM_CLEARED_ALARM = "x5";
    const string NOTIFY_FM_ALARM_LIST_REBUILT = "x6";
    const string NOTIFY_FM_POTENTIAL_FAULTY_ALARM_LIST = "x7";
};

/*
This block identifies the levels of severity.
*/
interface PerceivedSeverity
{
    const short INDETERMINATE = 1;
    const short CRITICAL = 2;
    const short MAJOR = 3;
    const short MINOR = 4;
    const short WARNING = 5;
    const short CLEARED = 6;
};

/*
This block identifies the probable cause of a reported alarm.
*/
interface ProbableCause
{
    /*
    Probable causes originating from M.3100.
    Values below correspond to M.3100 values.
    */
    const short INDETERMINATE = 0;
    const short ALARM_INDICATION_SIGNAL = 1;
    const short CALL_SETUP_FAILURE = 2;
    const short DEGRADED_SIGNAL = 3;
    const short FAR_END_RECEIVER_FAILURE = 4;
    const short FRAMING_ERROR = 5;
    const short LOSS_OF_FRAME = 6;
    const short LOSS_OF_POINTER = 7;
    const short LOSS_OF_SIGNAL = 8;
```

```
const short PAYLOAD_TYPE_MISMATCH = 9;
// Values 10 correspond to a duplicated probable cause
const short REMOTE_ALARM_INTERFACE = 11;
const short EXCESSIVE_BIT_ERROR_RATE = 12;
const short PATH_TRACE_MISMATCH = 13;
const short UNAVAILABLE = 14;
const short SIGNAL_LABEL_MISMATCH = 15;
const short LOSS_OF_MULTI_FRAME = 16;
const short COMMUNICATIONS_RECEIVE_FAILURE = 17;
const short COMMUNICATIONS_TRANSMIT_FAILURE = 18;
const short MODULATION_FAILURE = 19;
const short DEMODULATION_FAILURE = 20;
// Values 21-26 correspond to duplicated probable causes
// Values 27-50 are reserved for M.3100 potential future extensions
const short BACK_PLANE_FAILURE = 51;
const short DATA_SET_PROBLEM = 52;
const short EQUIPMENT_IDENTIFIER_DUPLICATION = 53;
const short EXTERNAL_IF_DEVICE_PROBLEM = 54;
const short LINE_CARD_PROBLEM = 55;
const short MULTIPLEXER_PROBLEM = 56;
const short NE_IDENTIFIER_DUPLICATION = 57;
const short POWER_PROBLEM = 58;
const short PROCESSOR_PROBLEM = 59;
const short PROTECTION_PATH_FAILURE = 60;
const short RECEIVER_FAILURE = 61;
const short REPLACEABLE_UNIT_MISSING = 62;
const short REPLACEABLE_UNIT_TYPE_MISMATCH = 63;
const short SYNCHRONIZATION_SOURCE_MISMATCH = 64;
const short TERMINAL_PROBLEM = 65;
const short TIMING_PROBLEM = 66;
const short TRANSMITTER_FAILURE = 67;
const short TRUNK_CARD_PROBLEM = 68;
const short REPLACEABLE_UNIT_PROBLEM = 69;
const short REAL_TIME_CLOCK_FAILURE = 70;
// Values 71-80 correspond to duplicated probable causes
const short PROTECTION_MECHANISM_FAILURE = 81;
const short PROTECTING_RESOURCE_FAILURE = 82;
```

```
// Values 83-100 are reserved for M.3100 potential future extensions
const short AIR_COMPRESSOR_FAILURE = 101;
const short AIR_CONDITIONING_FAILURE = 102;
const short AIR_DRYER_FAILURE = 103;
const short BATTERY_DISCHARGING = 104;
const short BATTERY_FAILURE = 105;
const short COMMERCIAL_POWER_FAILURE = 106;
const short COOLING_FAN_FAILURE = 107;
const short ENGINE_FAILURE = 108;
const short FIRE_DETECTOR_FAILURE = 109;
const short FUSE_FAILURE = 110;
const short GENERATOR_FAILURE = 111;
const short LOW_BATTERY_THRESHOLD = 112;
const short PUMP_FAILURE = 113;
const short RECTIFIER_FAILURE = 114;
const short RECTIFIER_HIGH_VOLTAGE = 115;
const short RECTIFIER_LOW_F_VOLTAGE = 116;
const short VENTILATION_SYSTEM_FAILURE = 117;
const short ENCLOSURE_DOOR_OPEN = 118;
const short EXPLOSIVE_GAS = 119;
const short FIRE = 120;
const short FLOOD = 121;
const short HIGH_HUMIDITY = 122;
const short HIGH_TEMPERATURE = 123;
const short HIGH_WIND = 124;
const short ICE_BUILD_UP = 125;
const short INTRUSION_DETECTION = 126;
const short LOW_FUEL = 127;
const short LOW_HUMIDITY = 128;
const short LOW_CABLE_PRESSURE = 129;
const short LOW_TEMPERATURE = 130;
const short LOW_WATER = 131;
const short SMOKE = 132;
const short TOXIC_GAS = 133;
// Values 134-135 correspond to duplicated probable causes
const short EXTERNAL_POINT_FAILURE = 136;
// Values 137-150 are reserved for potential M.3100 future extensions
```

```
const short STORAGE_CAPACITY_PROBLEM = 151;
const short MEMORY_MISMATCH = 152;
const short CORRUPT_DATA = 153;
const short OUT_OF_CPU_CYCLES = 154;
const short SOFTWARE_ENVIRONMENT_PROBLEM = 155;
const short SOFTWARE_DOWNLOAD_FAILURE = 156;
const short LOSS_OF_REAL_TIME = 157;
const short REINITIALIZED = 158;
// Values 159-167 correspond to duplicated probable causes
// Values 168-200 are reserved for potential M.3100 future extensions
// Values 201-202 correspond to duplicated probable causes
const short EXCESSIVE_ERROR_RATE = 203;
// Values 204-207 correspond to duplicated probable causes
// Values 208-300 are reserved for potential M.3100 future extensions
/*
Probable causes originating from X.721.
Values below correspond to X.721 values with an offset of 300.
*/
const short ADAPTER_ERROR = 301;
const short APPLICATION_SUBSYSTEM_FAILURE = 302;
const short BANDWIDTH_REDUCED = 303;
// Value 304 corresponds to a duplicated probable cause
const short COMMUNICATIONS_PROTOCOL_ERROR = 305;
const short COMMUNICATIONS_SUBSYSTEM_FAILURE = 306;
const short CONFIGURATION_OR_CUSTOMIZATION_ERROR = 307;
const short CONGESTION = 308;
// Value 309 corresponds to a duplicated probable cause
const short CPU_CYCLES_LIMIT_EXCEEDED = 310;
const short DATA_SET_OR_MODEM_ERROR = 311;
// Value 312 corresponds to a duplicated probable cause
const short DTE_DCE_INTERFACE_ERROR = 313;
// Value 314 corresponds to a duplicated probable cause
const short EQUIPMENT_MALFUNCTION = 315;
const short EXCESSIVE_VIBRATION = 316;
const short FILE_ERROR = 317;
// Values 318-320 correspond to duplicated probable causes
const short HEATING_OR_VENTILATION_OR_COOLING_SYSTEM_PROBLEM = 321;
```

```
const short HUMIDITY_UNACCEPTABLE = 322;
const short INPUT_OUTPUT_DEVICE_ERROR = 323;
const short INPUT_DEVICE_ERROR = 324;
const short LAN_ERROR = 325;
const short LEAK_DETECTED = 326;
const short LOCAL_NODE_TRANSMISSION_ERROR = 327;
// Values 328-329 correspond to duplicated probable causes
const short MATERIAL_SUPPLY_EXHAUSTED = 330;
// Value 331 corresponds to a duplicated probable cause
const short OUT_OF_MEMORY = 332;
const short OUTPUT_DEVICE_ERROR = 333;
const short PERFORMANCE_DEGRADED = 334;
// Value 335 corresponds to a duplicated probable cause
const short PRESSURE_UNACCEPTABLE = 336;
// Values 337-338 correspond to duplicated probable causes
const short QUEUE_SIZE_EXCEEDED = 339;
const short RECEIVE_FAILURE = 340;
// Value 341 corresponds to a duplicated probable cause
const short REMOTE_NODE_TRANSMISSION_ERROR = 342;
const short RESOURCE_AT_OR_NEARING_CAPACITY = 343;
const short RESPONSE_TIME_EXCESSIVE = 344;
const short RETRANSMISSION_RATE_EXCESSIVE = 345;
const short SOFTWARE_ERROR = 346;
const short SOFTWARE_PROGRAM_ABNORMALLY_TERMINATED = 347;
const short SOFTWARE_PROGRAM_ERROR = 348;
// Value 349 corresponds to a duplicated probable cause
const short TEMPERATURE_UNACCEPTABLE = 350;
const short THRESHOLD_CROSSED = 351;
// Value 352 corresponds to a duplicated probable cause
const short TOXIC_LEAK_DETECTED = 353;
const short TRANSMIT_FAILURE = 354;
// Value 355 corresponds to a duplicated probable cause
const short UNDERLYING_RESOURCE_UNAVAILABLE = 356;
const short VERSION_MISMATCH = 357;
// Values 358-500 are reserved for potential X.721 future extensions
/*
Probable causes for 2G & 3G wireless systems.
```

```
*/
const short A_BIS_TO_BTS_INTERFACE_FAILURE = 501;
const short A_BIS_TO_TRX_INTERFACE_FAILURE = 502;
const short ANTENNA_PROBLEM = 503;
const short BATTERY_BREAKDOWN = 504;
const short BATTERY_CHARGING_FAULT = 505;
const short CLOCK_SYNCHRONIZATION_PROBLEM = 506;
const short COMBINER_PROBLEM = 507;
const short DISK_PROBLEM = 508;
// Value 509 corresponds to a duplicated probable cause
const short EXCESSIVE_RECEIVER_TEMPERATURE = 510;
const short EXCESSIVE_TRANSMITTER_OUTPUT_POWER = 511;
const short EXCESSIVE_TRANSMITTER_TEMPERATURE = 512;
const short FREQUENCY_HOPPING_DEGRADED = 513;
const short FREQUENCY_HOPPING_FAILURE = 514;
const short FREQUENCY_REDEFINITION_FAILED = 515;
const short LINE_INTERFACE_FAILURE = 516;
const short LINK_FAILURE = 517;
const short LOSS_OF_SYNCHRONIZATION = 518;
const short LOST_REDUNDANCY = 519;
const short MAINS_BREAKDOWN_WITH_BATTERY_BACKUP = 520;
const short MAINS_BREAKDOWN_WITHOUT_BATTERY_BACKUP = 521;
const short POWER_SUPPLY_FAILURE = 522;
const short RECEIVER_ANTENNA_FAULT = 523;
// Value 524 corresponds to a duplicated probable cause
const short RECEIVER_MULTICOUPLER_FAILURE = 525;
const short REDUCED_TRANSMITTER_OUTPUT_POWER = 526;
const short SIGNAL_QUALITY_EVALUATION_FAULT = 527;
const short TIMESLOT_HARDWARE_FAILURE = 528;
const short TRANSCEIVER_PROBLEM = 529;
const short TRANSCODER_PROBLEM = 530;
const short TRANSCODER_OR_RATE_ADAPTER_PROBLEM = 531;
const short TRANSMITTER_ANTENNA_FAILURE = 532;
const short TRANSMITTER_ANTENNA_NOT_ADJUSTED = 533;
// Value 534 corresponds to a duplicated probable cause
const short TRANSMITTER_LOW_VOLTAGE_OR_CURRENT = 535;
const short TRANSMITTER_OFF_FREQUENCY = 536;
```

```
const short DATABASE_INCONSISTENCY = 537;
const short FILE_SYSTEM_CALL_UNSUCCESSFUL = 538;
const short INPUT_PARAMETER_OUT_OF_RANGE = 539;
const short INVALID_PARAMETER = 540;
const short INVALID_POINTER = 541;
const short MESSAGE_NOT_EXPECTED = 542;
const short MESSAGE_NOT_INITIALIZED = 543;
const short MESSAGE_OUT_OF_SEQUENCE = 544;
const short SYSTEM_CALL_UNSUCCESSFUL = 545;
const short TIMEOUT_EXPIRED = 546;
const short VARIABLE_OUT_OF_RANGE = 547;
const short WATCH_DOG_TIMER_EXPIRED = 548;
const short COOLING_SYSTEM_FAILURE = 549;
const short EXTERNAL_EQUIPMENT_FAILURE = 550;
const short EXTERNAL_POWER_SUPPLY_FAILURE = 551;
const short EXTERNAL_TRANSMISSION_DEVICE_FAILURE = 552;
// Values 553-560 correspond to duplicated probable causes
const short REDUCED_ALARM_REPORTING = 561;
const short REDUCED_EVENT_REPORTING = 562;
const short RECUCED_LOGGING_CAPABILITY = 563;
const short SYSTEM_RESOURCES_OVERLOAD = 564;
const short BROADCAST_CHANNEL_FAILURE = 565;
const short CONNECTION_ESTABLISHMENT_ERROR = 566;
const short INVALID_MESSAGE_RECEIVED = 567;
const short INVALID_MSU_RECEIVED = 568;
const short LAPD_LINK_PROTOCOL_FAILURE = 569;
const short LOCAL_ALARM_INDICATION = 570;
const short REMOTE_ALARM_INDICATION = 571;
const short ROUTING_FAILURE = 572;
const short SS7_PROTOCOL_FAILURE = 573;
const short TRANSMISSION_ERROR = 574;
// Value 575 corresponds to a duplicated probable cause
// Values 576-700 are reserved for potential future extensions
// for 2G & 3G wireless systems
/*
Probable causes originating from M.3100 security alarm causes.
Values below correspond to M.3100 values with an offset of 700.
```

```
    */
    const short AUTHENTICATION_FAILURE   = 701;
    const short BREACH_OF_CONFIDENTIALITY = 702;
    const short CABLE_TAMPER = 703;
    const short DELAYED_INFORMATION = 704;
    const short DENIAL_OF_SERVICE = 705;
    const short DUPLICATE_INFORMATION = 706;
    const short INFORMATION_MISSING = 707;
    const short INFORMATION_MODIFICATION_DETECTED = 708;
    const short INFORMATION_OUT_OF_SEQUENCE = 709;
    // Value 710 corresponds to a duplicated probable cause
    const short KEY_EXPIRED = 711;
    const short NON_REPUDIATION_FAILURE = 712;
    const short OUT_OF_HOURS_ACTIVITY = 713;
    const short OUT_OF_SERVICE = 714;
    const short PROCEDURAL_ERROR = 715;
    const short UNAUTHORISED_ACCESS_ATTEMPT = 716;
    const short UNEXPECTED_INFORMATION = 717;
    const short UNSPECIFIED_REASON = 718;
    // Values 719-800 are reserved for potential M.3100 future extensions
};

/*
This block identifies the acknowledgement state of a reported alarm.
*/
interface AckState
{
    const short ACKNOWLEDGED = 1;
    const short UNACKNOWLEDGED = 2;
};

/*
This block identifies attributes which are included as part of the Alarm 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 ALARM_ID = "f";
    const string PROBABLE_CAUSE = "g";
    const string PERCEIVED_SEVERITY = "h";
    const string SPECIFIC_PROBLEM = "i";
    const string ADDITIONAL_TEXT = "j";
    const string ACK_TIME = "k";
    const string ACK_USER_ID = "l";
    const string ACK_SYSTEM_ID = "m";
    const string ACK_STATE = "n";
    const string COMMENTS = "o";
    const string BACKED_UP_STATUS = "p";
    const string BACK_UP_OBJECT = "q";
    const string THRESHOLD_INFO = "r";
    const string TREND_INDICATION = "s";
    const string STATE_CHANGE_DEFINITION = "t";
    const string MONITORED_ATTRIBUTES = "u";
    const string PROPOSED_REPAIR_ACTIONS = "v";
    const string CORRELATED_NOTIFICATIONS = "w";
    const string REASON = "x";
    const string CLEAR_USER_ID = "y";
    const string CLEAR_SYSTEM_ID = "z";
    const string ALARM_LIST_ALIGNMENT_REQUIREMENT = "ff";
    const string SERVICE_USER = "gg";
    const string SERVICE_PROVIDER = "hh";
    const string SECURITY_ALARM_DETECTOR = "ii";
    const string ALARM_RAISED_TIME = "kk";
    const string ALARM_CLEARED_TIME = "ll";
    const string ALARM_CHANGED_TIME = "mm";
};

/*
Defines the content of a Comment
*/
struct Comment
{
    ManagedGenericIRPConstDefs::IRPTime comment_time;
```

```
    string comment_text;
    string user_id;
    string system_id;
};

/*
Defines a set of comments which are placed in the COMMENTS attribute
of a structured event.
*/
typedef sequence <Comment> CommentSet;

/*
It indicates if an object has a back up.
True implies backed up.   False implies not backed up.
*/
typedef boolean BackedUpStatusType;

/*
It indicates if the threshold crossed was in the up or down direction.
*/
enum ThresholdIndicationType {Up, Down};

/*
It indicates if the AlarmList alignment is required.
*/
enum AlarmListAlignmentRequirementType {Required, NotRequired};

/* FloatTypeOpt is an optional type.
If the discriminator is true the value is present.
Otherwise the value is null.
*/
union FloatTypeOpt switch (boolean)
{
    case TRUE: float value;
};

/* ThresholdLevelIndType describes multi-level
```

```
threshold crossings.
Up is the only permitted choice for a counter.
If indication is "up", low value is optional.
@member indication: indicates up or down direction
  of crossing.
@member low: the low observed value.
@member high: the high observed value.
*/
struct ThresholdLevelIndType
{
    ThresholdIndicationType indication;
    FloatTypeOpt low;
    float high;
};

/* ThresholdLevelIndTypeOpt is an optional type.
If the discriminator is true the value is present.
Otherwise, the value is null.
*/
union ThresholdLevelIndTypeOpt switch (boolean)
{
    case TRUE: ThresholdLevelIndType value;
};

/* ThresholdInfoType indicates some gauge or counter
attribute passed a set threshold.
@member attributeID: identifies the attribute that
  crossed the threshold.
@member observedValue: attributes that are of type
  integer will be converted to floats.
@member thresholdlevel: This parameter is for
  multi-level thresholds. Optional.
@member armTime: May contain empty string.
*/
struct ThresholdInfoType
{
    string attributeID;
```

```
    float observedValue;
    ThresholdLevelIndTypeOpt thresholdLevel;
    string armTime;
};

/*
It indicates if some observed condition is getting better, worse,
or not changing.
*/
enum TrendIndicationType {LessSevere, NoChange, MoreSevere};

/*
It is used to report a changed attribute value.
*/
struct AttributeValueChangeType
{
    string attribute_name;
    any     old_value;   // type depends on attribute
    any     new_value;   // type depends on attribute
};

typedef sequence <AttributeValueChangeType> AttributeChangeSetType;

/*
It is used to report an attribute and its value.
*/
struct AttributeValueType
{
    string attribute_name;
    any     value;   // type depends on the attribute
};

typedef sequence <AttributeValueType> AttributeSetType;

typedef sequence <long> NotifIdSetType;

/*
```

```
This holds identifiers of notifications that are correlated.
*/
struct CorelatedNotification
{
    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 <CorelatedNotification> CorrelatedNotificationSetType;

/*
Define the structure of Alarm ID and Perceived Severity used within the
alarm acknowledgment operation. Note: perceived_severity is an optional
parameter. If this value is present, it must have one of the defined values
of Interface PerceivedSeverity.
*/
struct AlarmInformationIdAndSev
{
    string alarm_information_reference;
    ManagedGenericIRPConstDefs::ShortTypeOpt perceived_severity;
};

/*
Define set of the above structure of Alarm ID and Perceived Severity.
*/
typedef sequence <AlarmInformationIdAndSev> AlarmInformationIdAndSevSeq;

/*
```

```
It indicates the reason for an alarm acknowledgement to have failed:
  - The specified Alarm Information is absent from the Alarm List
  - The Perceived Severity to be acknowledged has changed and/or is different
    within the Alarm List
  - The acknowledgement failed for some other reason
*/
enum AcknowledgeFailureCategories
{
    UnknownAlarmId,
    WrongPerceivedSeverity,
    AcknowledgmentFailed
};

/*
Define the structure returned when an operation fails for a set of alarm ids.
A reason is provided in order to indicate why the operation failed.
*/
struct BadAlarmInformationId
{
    string alarm_information_reference;
    string reason;
};

/*
Define the structure returned when the acknowledge operation fails for a set
of alarm ids.
A failure category and a reason are provided in order to indicate why the
operation failed.
*/
struct BadAcknowledgeAlarmInfo
{
    string alarm_information_reference;
    AcknowledgeFailureCategories failure_category;
    string reason;
};

typedef sequence <BadAlarmInformationId> BadAlarmInformationIdSeq;
```

```
    typedef sequence <BadAcknowledgeAlarmInfo> BadAcknowledgeAlarmInfoSeq;
    typedef sequence <string> AlarmInformationIdSeq;
    typedef CosNotification::EventBatch AlarmInformationSeq;
};
#endif

//File "AlarmIRPSystem.idl"
//The IRP document version number is "Alarm IRP V1.0"

#ifndef AlarmIRPSystem_idl
#define AlarmIRPSystem_idl

#include "AlarmIRPConstDefs.idl"
#include "ManagedGenericIRPSystem.idl"

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

/* ## Module: AlarmIRPSystem
This module contains the specification of all operations of Alarm IRP Agent.
================================================================
*/
module AlarmIRPSystem
{
    /*
    System fails to complete the operation.  System can provide reason
    to qualify the exception.  The semantics carried in reason
    is outside the scope of this IRP.
    */
    exception GetAlarmIRPVersions { string reason; };
    exception GetAlarmIRPOperationsProfile { string reason; };
    exception GetAlarmIRPNotificationProfile { string reason; };
    exception AcknowledgeAlarms { string reason; };
    exception UnacknowledgeAlarms { string reason; };
    exception CommentAlarms { string reason; };
    exception ClearAlarms { string reason; };
    exception GetAlarmList { string reason; };
```

```
exception GetAlarmCount { string reason; };
exception NextAlarmInformations { string reason; };

/*
The AlarmInformationIterator is used to iterate through a snapshot of
Alarm Informations taken from the Alarm List when IRPManager invokes
get_alarm_list. IRPManager uses it to pace the return of Alarm
Informations.
IRPAgent controls the life-cycle of the iterator. However, a destroy
operation is provided to handle the case where IRPManager wants to stop
the iteration procedure before reaching the last iteration.
*/
interface AlarmInformationIterator
{
    /*
    This method returns between 1 and "how_many" Alarm Informations. The
    IRPAgent may return less than "how_many" items even if there are more
    items to return. "how_many" must be non-zero. Return TRUE if there may
    be more Alarm Information to return. Return FALSE if there are no more
    Alarm Information to be returned.
    If FALSE is returned, the IRPAgent will automatically destroy the
    iterator.
    */
    boolean next_alarmInformations (
        in unsigned short how_many,
        out AlarmIRPConstDefs::AlarmInformationSeq alarm_informations
    )
    raises (NextAlarmInformations, ManagedGenericIRPSystem::InvalidParameter);

    /*
    This method destroys the iterator.
    */
    void destroy();
};

interface AlarmIRP
{
```

```
/*
Return the list of all supported Alarm IRP versions.
Implementations are to provide a return value consisting of one or more
IRPVersions.
Each IRPVersion is defined by the rule in the clause titled
"IRP document version number string"
*/
ManagedGenericIRPConstDefs::VersionNumberSet get_alarm_IRP_versions (
)
raises (GetAlarmIRPVersions);

/*
Return the list of all supported operations and their supported
parameters for a specific Alarm IRP version.
*/
ManagedGenericIRPConstDefs::MethodList get_alarm_IRP_operations_profile (
    in ManagedGenericIRPConstDefs::VersionNumber alarm_irp_version
)
raises (GetAlarmIRPOperationsProfile,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);

/*
Return the list of all supported notifications and their supported
parameters for a specific Alarm IRP version.
*/
ManagedGenericIRPConstDefs::MethodList get_alarm_IRP_notification_profile
(
    in ManagedGenericIRPConstDefs::VersionNumber alarm_irp_version
)
raises (GetAlarmIRPNotificationProfile,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);

/*
Request to acknowledge one or more alarms.
*/
```

```
ManagedGenericIRPConstDefs::Signal acknowledge_alarms (
    in AlarmIRPConstDefs::AlarmInformationIdAndSevSeq
        alarm_information_id_and_sev_list,
    in string ack_user_id,
    in ManagedGenericIRPConstDefs::StringTypeOpt ack_system_id,
    out AlarmIRPConstDefs::BadAcknowledgeAlarmInfoSeq
        bad_ack_alarm_info_list
)
raises (AcknowledgeAlarms, ManagedGenericIRPSystem::ParameterNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);

/*
Request to remove acknowledgement information of one or more alarms.
*/
ManagedGenericIRPConstDefs::Signal unacknowledge_alarms (
    in AlarmIRPConstDefs::AlarmInformationIdSeq alarm_information_id_list,
    in string ack_user_id,
    in ManagedGenericIRPConstDefs::StringTypeOpt ack_system_id,
    out AlarmIRPConstDefs::BadAlarmInformationIdSeq
        bad_alarm_information_id_list
)
raises (UnacknowledgeAlarms,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::ParameterNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);

/*
Make comment to one or more alarms.
*/
ManagedGenericIRPConstDefs::Signal comment_alarms (
    in AlarmIRPConstDefs::AlarmInformationIdSeq alarm_information_id_list,
    in string comment_user_id,
    in ManagedGenericIRPConstDefs::StringTypeOpt comment_system_id,
    in string comment_text,
    out AlarmIRPConstDefs::BadAlarmInformationIdSeq
        bad_alarm_information_id_list
)
```

```
raises (CommentAlarms, ManagedGenericIRPSystem::OperationNotSupported,
        ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
Request to clear one or more alarms.
*/
ManagedGenericIRPConstDefs::Signal clear_alarms (
    in AlarmIRPConstDefs::AlarmInformationIdSeq alarm_information_id_list,
    in string clear_user_id,
    in ManagedGenericIRPConstDefs::StringTypeOpt clear_system_id,
    out AlarmIRPConstDefs::BadAlarmInformationIdSeq
        bad_alarm_information_id_list
)
raises (ClearAlarms, ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
This method returns Alarm Informations.
If flag is TRUE, all returned Alarm Informations shall be
in AlarmInformationSeq that contains 0 or more Alarm Informations.
Output parameter iter shall be useless.
If flag is FALSE, no Alarm Informations shall be in AlarmInformationSeq.
IRPAgent needs to use iter to retrieve them.
*/
AlarmIRPConstDefs::AlarmInformationSeq get_alarm_list (
    in ManagedGenericIRPConstDefs::StringTypeOpt filter,
    in AlarmIRPConstDefs::DNTypeOpt base_object,
    out boolean flag,
    out AlarmInformationIterator iter
)
raises (GetAlarmList, ManagedGenericIRPSystem::ParameterNotSupported,
        ManagedGenericIRPSystem::InvalidParameter);

/*
This method returns the count of Alarm Informations.
*/
```

```
        void get_alarm_count (
            in ManagedGenericIRPConstDefs::StringTypeOpt filter,
            out unsigned long critical_count,
            out unsigned long major_count,
            out unsigned long minor_count,
            out unsigned long warning_count,
            out unsigned long indeterminate_count,
            out unsigned long cleared_count
        )
        raises (GetAlarmCount, ManagedGenericIRPSystem::OperationNotSupported,
                ManagedGenericIRPSystem::ParameterNotSupported,
                ManagedGenericIRPSystem::InvalidParameter);
    };
};

#endif
```

A.10 性能管理接口信息模型

```
//File "PMIRPConstDefs.idl"
//The IRP document version number is "PM IRP V1.0"

#ifndef PMIRPConstDefs_idl
#define PMIRPConstDefs_idl

#include <TimeBase.idl>

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

/* ## Module: PMIRPConstDefs
This module contains commonly used definitions for PM IRP
================================================================
*/
module PMIRPConstDefs
{

    enum ResultType {OK, Failure};
```

```
typedef    string    MOClassNameType;
typedef    string    MOInstanceType;
typedef    sequence<MOInstanceType>    MOInstanceListType;
typedef    string    MeasurementCategoryType;
typedef    sequence<MeasurementCategoryType>    MeasurementCategoryListType;
typedef    unsigned long    GranularityPeriodType; //The unit is minute.
typedef    unsigned long    ReportingPeriodType; //The unit is minute.
typedef    TimeBase::UtcT UTCTimeType;

union StartTimeTypeOpt switch(boolean)
{
  case TRUE: UTCTimeType value;
};

union StopTimeTypeOpt switch(boolean)
{
  case TRUE: UTCTimeType value;
};

typedef string IntervalTimeType;
// the IntervalTimeType with format ab:cd (from 00:00 to 23:59 of one day)
struct IntervalOfDayType
{
  IntervalTimeType intervalStartTime;
  IntervalTimeType intervalStopTime;
};
typedef sequence<IntervalOfDayType> DailySchedulingType;
typedef string DayOfWeekType;
//The value of DayOfWeekType is a BIT STRING (SIZE(7))
//{sunday(0),monday(1),tuesday(2),wednesday(3),
//thursday(4), friday(5), saturday(6)}
typedef sequence<DayOfWeekType> DaysOfWeekType;
struct WeeklySchedulingElement
{
  DaysOfWeekType dayOfWeek;
  DailySchedulingType intervalsOfDay;
};
```

```
typedef sequence<WeeklySchedulingElement> WeeklySchedulingType;
enum scheduleTypeChoice { Daily, Weekly };
union ScheduleType switch (scheduleTypeChoice)
{
  case Daily:   DailySchedulingType dailyScheduling;
  case Weekly: WeeklySchedulingType weeklyScheduling;
};
union ScheduleTypeOpt switch(boolean)
{
  case TRUE: ScheduleType value;
};


typedef    unsigned long    JobIdType;
typedef sequence<JobIdType> JobIdListType;
struct JUnsupportedType
{
  MOInstanceType   moInstance;
  MeasurementCategoryType measurementCategory;
  string reason;
};
typedef   sequence<JUnsupportedType>   JUnsupportedListType;


/**
 * Defines the name of an attribute of a Managed Object
 */
typedef string MOAttributeName;

enum   JobStatusType {Scheduled, Active, Suspended, Stopped};
struct JobInfoType
{
 JobIdType   jobId;
 MOClassNameType   moClass;
 MOInstanceListType   moInstanceList;
 MeasurementCategoryListType   measurementCategoryList;
 GranularityPeriodType   granularityPeriod;
```

```
 ReportingPeriodType   reportingPeriod;
 StartTimeTypeOpt   startTime;
 StopTimeTypeOpt   stopTime;
 ScheduleTypeOpt   schedule;
 JobStatusType jobStatus;
};
typedef sequence<JobInfoType> JobInfoListType;

typedef string MeasurementTypeNameType;
typedef string SubCounterNameType;
typedef string ProbableCauseType;
typedef string SpecificProblemType;
typedef any ThresholdValueType;
enum SeverityType {Warning, Minor, Major, Critical};
union HysteresisType switch(boolean)
{
   case TRUE:   long longValue;
   case FALSE: float floatValue;
};
enum DirectionType { Increasing, Decreasing};
struct ThresholdPackElementType
{
   ThresholdValueType thresholdValue;
   SeverityType severity; // the value shall be
   // one of Warning, Minor, Major or Critical.
   HysteresisType hysteresis;
};
typedef sequence<ThresholdPackElementType> ThresholdPackType;
struct ThresholdInfoType
{
   MeasurementTypeNameType measurementTypeName;
   SubCounterNameType subCounterName;
   ProbableCauseType probableCause;
   SpecificProblemType specificProblem;
   DirectionType direction;
   ThresholdPackType thresholdPack;
};
```

```
typedef sequence<ThresholdInfoType> ThresholdInfoListType;
typedef GranularityPeriodType MonitorGranularityPeriodType;
typedef unsigned long MonitorIdType;
struct MUnsupportedType
{
  MOInstanceType   moInstance;
  MeasurementTypeNameType measurementTypeName;
  SubCounterNameType subCounterName;
  string reason;
};
typedef   sequence<MUnsupportedType>   MUnsupportedListType;
enum   MonitorStatusType {MSuspended, MActive};

typedef sequence<MonitorIdType> MonitorIdListType;
typedef string EventTypeType; // The value is "Quality of Service Alarm"
struct MonitorInfoType
{
 MonitorIdType   monitorId;
 MOClassNameType   moClass;
 MOInstanceListType   moInstanceList;
 MonitorGranularityPeriodType monitorGranularityPeriod;
 ThresholdInfoListType thresholdInfoList;
 MonitorStatusType thresholdMonitorStatus;
 EventTypeType eventType;
};
typedef sequence<MonitorInfoType> MonitorInfoListType;


/**
* This block identifies attributes which are included as part of the
* PMIRP. These attribute values should not
* clash with those defined for the attributes of notification
* header (see IDL of Notification IRP).
*/
interface AttributeNameValue
{
 const string JOB_ID = "JOB_ID";
```

```
    const string JOB_STATUS = "JOB_STATUS";
    const string REASON = "REASON";
    const string MONITOR_ID = "MONITOR_ID";
    const string MONITOR_STATUS = "MONITOR_STATUS";

    const string MONITOR_GRANULARITYPERIOD = "MONITOR_GRANULARITYPERIOD";
    const string THRESHOLD_INFO_LIST = "THRESHOLD_INFO_LIST";
  };

};

#endif

//File "PMIRPSystem.idl"
//The IRP document version number is "PM IRP V1.0"

#ifndef PMIRPSystem_idl
#define PMIRPSystem_idl

#include "ManagedGenericIRPSystem.idl"
#include "ManagedGenericIRPConstDefs.idl"
#include "PMIRPConstDefs.idl"

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

/* ## Module: PMIRPSystem
This module contains the specification of all operations of PM IRP Agent.
================================================================
*/
module PMIRPSystem
{

  /**
  * the reason specifies whether EM or NE is high workload. The value shall be one
  * of following: emCpuBusy; emHDShortage, emLowMemory, {neCpuBusy, neObjectInstList},
  * {neHDShortage neObjectInstList}, {neLowMemory, neObjectInstList}, maxJobReached,
```

```
    * otherReason.
    */
    exception HighWorkLoad { string reason; };

    exception UnknownJob { string reason; };
    exception JobCannotBeStopped { string reason; };
    exception JobAlreadySuspended { string reason; };
    exception JobIsNotSuspended { string reason; };
    exception UnknownThresholdMonitor { string reason; };
    exception ThresholdMonitorAlreadySuspended { string reason; };
    exception ThresholdMonitorIsNotSuspended { string reason; };

    /**
    * System fails to complete the operation.  System can provide reason
    * to qualify the exception.  The semantics carried in reason
    * is outside the scope of this IRP.
    */
    exception GetPMIRPVersions { string reason; };
    exception GetPMIRPOperationsProfile { string reason; };
    exception GetPMIRPNotificationProfile { string reason; };
    exception CreateMeasurementJob { string reason; };
    exception StopMeasurementJob { string reason; };
    exception SuspendMeasurementJob { string reason; };
    exception ResumeMeasurementJob { string reason; };
    exception ListMeasurementJobs { string reason; };

    exception CreateThresholdMonitor { string reason; };
    exception DeleteThresholdMonitor { string reason; };
    exception ListThresholdMonitors { string reason; };
    exception SuspendThresholdMonitor { string reason; };
    exception ResumeThresholdMonitor { string reason; };

    interface PMIRP
    {

        readonly attribute string iRPId;
```

```
/**
* Return the list of all supported PM IRP versions.
*/
ManagedGenericIRPConstDefs::VersionNumberSet get_PM_IRP_versions (
)
raises (GetPMIRPVersions);


/**
* Return the list of all supported operations and their supported
* parameters for a specific PM IRP version.
*/
ManagedGenericIRPConstDefs::MethodList get_PM_IRP_operations_profile (
    in ManagedGenericIRPConstDefs::VersionNumber pm_irp_version
)
raises (GetPMIRPOperationsProfile,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);

/**
* Return the list of all supported notifications and their supported
* parameters for a specific PM IRP version.
*/
ManagedGenericIRPConstDefs::MethodList get_PM_IRP_notification_profile
(
    in ManagedGenericIRPConstDefs::VersionNumber pm_irp_version
)
raises (GetPMIRPNotificationProfile,
            ManagedGenericIRPSystem::OperationNotSupported,
            ManagedGenericIRPSystem::InvalidParameter);

/**
* Request to create a MeasurementJob through Itf-N.
*/
ManagedGenericIRPConstDefs::Signal create_measurement_job (
   in PMIRPConstDefs::MOClassNameType    moClass,
   in PMIRPConstDefs::MOInstanceListType    moInstanceList,
```

```
    in PMIRPConstDefs::MeasurementCategoryListType    measurementCategoryList,
    in PMIRPConstDefs::GranularityPeriodType    granularityPeriod,
    in PMIRPConstDefs::ReportingPeriodType    reportingPeriod,
    in PMIRPConstDefs::StartTimeTypeOpt    startTime,
    in PMIRPConstDefs::StopTimeTypeOpt    stopTime,
    in PMIRPConstDefs::ScheduleTypeOpt    schedule,
    out PMIRPConstDefs::JobIdType jobId,
    out PMIRPConstDefs::JUnsupportedListType unsupportedList
)
raises (CreateMeasurementJob,
            ManagedGenericIRPSystem::InvalidParameter,
            HighWorkLoad);

/**
* Request to stop a MeasurementJob through Itf-N, after which,
* the MeasurementJob will still be visible via Itf-N. Whether
* the MeasurementJob is thoroughly removed immediately from
* the managed system is vendor specific.
*/
PMIRPConstDefs::ResultType stop_measurement_job (
    in PMIRPConstDefs::JobIdType jobId)
raises (StopMeasurementJob,
            UnknownJob,
            JobCannotBeStopped);

/**
* Request to suspend a MeasurementJob
*/
PMIRPConstDefs::ResultType suspend_measurement_job (
    in PMIRPConstDefs::JobIdType jobId)
raises (SuspendMeasurementJob,
            UnknownJob,
            JobAlreadySuspended);

/**
* Request to resume a MeasurementJob
*/
```

```
PMIRPConstDefs::ResultType resume_measurement_job (
   in PMIRPConstDefs::JobIdType jobId)
raises (ResumeMeasurementJob,
            UnknownJob,
            JobIsNotSuspended);

/**
 * Request to list the information of all or of specified
 * MeasurementJobs
 */
PMIRPConstDefs::ResultType list_measurement_jobs (
   in PMIRPConstDefs::JobIdListType jobListId,
   out PMIRPConstDefs::JobInfoListType jobInfoList)
raises (ListMeasurementJobs,
            ManagedGenericIRPSystem::InvalidParameter);

/**
 * Request to create a ThresholdMonitor to define the threshold
 * for some specific measurementTypes or subCounters
 */
ManagedGenericIRPConstDefs::Signal create_threshold_monitor (
   in PMIRPConstDefs::MOClassNameType    moClass,
   in PMIRPConstDefs::MOInstanceListType    moInstanceList,
   in PMIRPConstDefs::ThresholdInfoListType thresholdInfoList,
   in PMIRPConstDefs::MonitorGranularityPeriodType monitorGranularityPeriod,
   out PMIRPConstDefs::MonitorIdType monitorId,
   out PMIRPConstDefs::MUnsupportedListType unsupportedList)
raises (CreateThresholdMonitor,
            ManagedGenericIRPSystem::InvalidParameter);

/**
 * Request to delete a specified ThresholdMonitor
 */
PMIRPConstDefs::ResultType delete_threshold_monitor (
   in PMIRPConstDefs::MonitorIdType monitorId)
raises (DeleteThresholdMonitor,
            UnknownThresholdMonitor);
```

```
        /**
        * Request to list detailed information about all or
        * specified ThresholdMonitors
        */
        PMIRPConstDefs::ResultType list_threshold_monitors (
           in PMIRPConstDefs::MonitorIdListType monitorIdList,
           out PMIRPConstDefs::MonitorInfoListType monitorInfoList)
        raises (ListThresholdMonitors,
                    ManagedGenericIRPSystem::InvalidParameter);

        /**
        * Request to suspend a ThresholdMonitor
        */
        PMIRPConstDefs::ResultType suspend_threshold_monitor (
           in PMIRPConstDefs::MonitorIdType monitorId)
        raises (SuspendThresholdMonitor,
                    UnknownThresholdMonitor,
                    ThresholdMonitorAlreadySuspended);

        /**
        * Request to resume a ThresholdMonitor
        */
        PMIRPConstDefs::ResultType resume_threshold_monitor (
           in PMIRPConstDefs::MonitorIdType monitorId)
        raises (ResumeThresholdMonitor,
                    UnknownThresholdMonitor,
                    ThresholdMonitorIsNotSuspended);

    };

};

#endif

//File "PMIRPNotifDefs.idl"
//The IRP document version number is "PM IRP V1.0"
```

```
#ifndef PMIRPNotifDefs_idl
#define PMIRPNotifDefs_idl

#include "PMIRPConstDefs.idl"
#include "NotificationIRPConstDefs.idl"


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

/* ## Module: PMIRPNotifDefs
This module contains the specification of all notifications of PM IRP Agent.
================================================================
*/
module PMIRPNotifDefs
{

    const string ET_MEASUREMENT_JOB_STATUS_CHANGED = "notifyMeasurementJobStatusChanged";
    const string ET_THRESHOLD_MONITOR_STATUS_CHANGED = "notifyThresholdMonitorStatusChanged";


    /**
    * Constant definitions for the notifyMeasurementJobStatusChanged notification
    */
    interface notifyMeasurementJobStatusChanged: NotificationIRPConstDefs::AttributeNameValue
    {
      const string EVENT_TYPE = ET_MEASUREMENT_JOB_STATUS_CHANGED;

      /**
      * This constant defines the name of the jobId property,
      * which is transported in the filterable_body fields.
      * The data type for the value of this property
      * is PMIRPConstDefs::JobIdType.
      */
      const string JOB_ID = PMIRPConstDefs::AttributeNameValue::JOB_ID;
```

```
    /**
    * This constant defines the name of the jobStatus property,
    * which is transported in the filterable_body fields.
    * The data type for the value of this property
    * is PMIRPConstDefs::JobStatusType.
    */
    const string JOB_STATUS = PMIRPConstDefs::AttributeNameValue::JOB_STATUS;

    /**
    * This constant defines the name of the reason property,
    * which is transported in the filterable_body fields.
    * The data type for the value of this property is string.
    */
    const string REASON = PMIRPConstDefs::AttributeNameValue::REASON;
};

/**
* Constant definitions for the notifyThresholdMonitorStatusChanged notification
*/
interface notifyThresholdMonitorStatusChanged: NotificationIRPConstDefs::AttributeNameValue
{
    const string EVENT_TYPE = ET_THRESHOLD_MONITOR_STATUS_CHANGED;

    /**
    * This constant defines the name of the monitorId property,
    * which is transported in the filterable_body fields.
    * The data type for the value of this property
    * is PMIRPConstDefs::MonitorIdType.
    */
    const string MONITOR_ID = PMIRPConstDefs::AttributeNameValue::MONITOR_ID;

    /**
    * This constant defines the name of the monitorStatus property,
    * which is transported in the filterable_body fields.
    * The data type for the value of this property
    * is PMIRPConstDefs::MonitorStatusType.
    */
```

```
        const string MONITOR_STATUS = PMIRPConstDefs::AttributeNameValue::MONITOR_STATUS;

        /**
        * This constant defines the name of the reason property,
        * which is transported in the filterable_body fields.
        * The data type for the value of this property is string.
        */
        const string REASON = PMIRPConstDefs::AttributeNameValue::REASON;
    };

};

#endif
```

附 录 B
（资料性附录）
OMG 定义的 UtcT 格式时间的说明

B.1 说明

OMG的Time service里有对UtcT的定义，其时间是当前时间与15，Oct 1582，00:00:00的差值，单位基于100ns。其能够表示的范围很大，且精度可控。

起始时间“15，Oct 1582，00:00:00”是由罗马教皇Gregory XIII首先施行，称为格里高利历（Gregorian Calendar），这也就是现在的公历真正应用的起点。

B.2 取值示例

现在的计算机一般的起始参考时间是1970-01-01，所以两者之间存在的是一个固定的时间差，可事先算出来(略，假定为X，单位100ns)。

举例：

1） 如：格林威治时间（GMT）：1970-01-01，00:00:00（绝对精确）

用UtcT可表示如下：

```
struct UtcT {
    time = X; file://X将由事先计算好的值来进行替换
    inacclo = 0;
    inacchi = 0;
    tdf = 0;
};
```


2） 如当前时间为北京时间2004-09-03,12:45:23.069（精确到秒）

通过time(&ltime)；可以获得本地时间 ltime（用time_t表示）为：1094186723（s）

（在已知当前时间的字符表示法的前提下,也可以通过赋值给struct tm *的变量,然后再用mktime（struct tm *）的方法达到上述目的）；

本地与GMT的时差为：8h=480min=28800s，

因此，GMT的当前时间为：1094186723 – 28800=1094157923（s）

这也就是与GMT，1970-01-01，00:00:00时间相差的秒数，再加上69ms。

（用ftime（struct timeb *）可以获得精确到毫秒的本地时间）；

用UtcT可表示如下：

```
struct UtcT {
    time = X + 1094157923.069*10,000,000; file://计算结果即为TimeT形式的表示法
    inacclo = 10,000; // 低位，表示精度为ms。(小于32位，不需要高位inacchi)
    inacchi = 0; file://高位
    tdf = 480;    file://时区，单位：min
};
```

对于已知UtcT格式的时间，将上述过程反向进行即可。

参 考 文 献

[1]3GPP TS 32.413 v6.2.0：

Telecommunication management; Performance Management IRP; CORBA Solution Set

[2]3GPP TS 32.303 v6.1.0：

Telecommunication management; Configuration Management; Notification Integration Reference Point (IRP): CORBA Solution Set

[3]3GPP TS 32.663 v6.2.0：

Telecommunication management;Kernel CM Integration Reference Point (IRP): CORBA Solution Set

[4]3GPP TS 32.111-3 v6.0.0：

Telecommunication management;Part 2: Alarm Integration Reference Point (IRP); CORBA Solution Set

[5]3GPP TS 32.603 v6.1.0：

Telecommunication management;Basic CM Integration Reference Point (IRP): CORBA Solution Set

[6]3GPP TS 32.363 v6.1.0：

Telecommunication management;Entry Point Integration Reference Point (IRP); CORBA Solution Set

[7]3GPP TS 32.343 v6.0.0：

Telecommunication management;File Transfer IRP; CORBA Solution Set

[8]3GPP TS 32.353 v6.0.0：

Telecommunication management;Communication Surveillance IntegrationReference Point (IRP); CORBA Solution Set

[9]3GPP2 S.S0028-A (Version 1.0)：

OAM&P for cdma2000(3GPP R4 Delta Specification)

中华人民共和国
通信行业标准
2GHz数字蜂窝移动通信网网络管理通用技术要求
第4部分 基于CORBA技术的接口设计
YD/T 1584.4-2007
*
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
北京新瑞铭印刷有限公司印刷

*
本书如有印装质量问题，请与本社联系 电话：(010)67114922