Source:	**SA5 (Telecom Management)**
Title:	**Rel-6 32.412-100 (Performance Management (PM) Integration Reference Point (IRP): Information Service) - for Information**
Document for:	**Information**
Agenda Item:	**7.5.3**

Presentation of Technical Specification to TSG SA

Presentation to:	**TSG SA Meeting #20**
Document for presentation:	**TS 32.412, Version 1.0.0**
Presented for:	**Information**

Abstract of document:

This is a Technical Specification on the Performance Management Integration Reference Point Information Service for 3GPP Release 6.

The new Performance Management IRP is intended for Release 6 and consists of a Requirements Specification (32.411; Stage 1), an Information Service Specification (32.412; Stage 2) and one or two Stage 3 Solution Set Specifications (32.413 CORBA or 32.414 CMIP).

The purpose of those specifications is to provide the essential Performance Management (PM) capabilities for measurement job administration, performance alarm handling and management of measurement file transfer over the Itf-N.

The work is done against the WID contained in SP-020499 (Work Item ID: OAM-PM), approved in SA#17.

Changes since last presentation to TSG-SA:
New.

Outstanding Issues:
Further minor alignments with the Generic File Transfer IRP might be required.

Contentious Issues:
No consensus could be reached so far on Optional/Mandatory statements for the PM IRP Operations and Notifications. More inputs, especially from Operators, on the operational use cases for the PM IRP are expected to ensure market requirements are met.

3GPP TS 32.412 V1.0.0 (2003-06)

Technical Specification

**3rd Generation Partnership Project;
Technical Specification Group Services and System Aspects;
Telecommunication management;
Performance Management (PM)
Integration Reference Point (IRP):
Information service
(Release 6)**

3GPP

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis
Valbonne - FRANCE
Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

3GPP

Contents

Foreword

This Technical Specification (TS) has been produced by the 3rd Generation Partnership Project (3GPP).

The present document is part the 32.41x-series covering the 3rd Generation Partnership Project: Technical Specification Group Services and System Aspects; Telecommunication Management; Performance Management, as identified below:

32.411: "Requirements";

32.412: "Information service";

32.413: "Common Object Request Broker Architecture (CORBA) solution set";

32.414: "Common Management Information Protocol (CMIP) solution set".

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

Introduction

The present document is part of a set of TSs which describe the requirements and information model necessary for the Telecommunication Management (TM) of 3G systems. The TM principles and TM architecture are specified in 3GPP TS 32.101 [1] and 3GPP TS 32.102 [2].

A 3G system is composed of a multitude of Network Elements (NE) of various types and, typically, different vendors, which inter-operate in a co-ordinated manner in order to satisfy the network users' communication requirements.

Any evaluation of PLMN-system behaviour will require performance data collected and recorded by its NEs according to a schedule established by the EM.

This aspect of the management environment is termed Performance Management. The purpose of any Performance Management activity is to collect performance related data, which can be used to locate potential problems in the network.

1 Scope

The present document specifies the Information Service for the Performance Management Integration Reference Point (PM IRP) as it applies to the Itf-N.

This IRP IS defines the semantics of operations (and their parameters) visible across the Itf-N in a protocol and technology neutral way. It does not define the syntax or encoding of the operations and their parameters.

2 References

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

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TS 32.101: "Telecommunication management; Principles and high level requirements".

[2] 3GPP TS 32.102: "Telecommunication management; Architecture".

[3] ITU-T Recommendation X.721 (1992): "Information technology - Open Systems Interconnection - Structure of management information: Definition of management information".

[4] 3GPP TS 32.111-2: "Telecommunication management; Fault management; Part 2: Alarm integration reference point: Information service".

[5] 3GPP TS 32.312: "Telecommunication management; Generic Integration Reference Point (IRP) management; Information service".

[6] 3GPP TS 32.622: "Telecommunication management; Configuration Management (CM); Generic network resources Integration Reference Point (IRP): Network Resource Model (NRM)".

[7] 3GPP TS 32.401: "Telecommunication management; Performance Management (PM); Concept and requirements".

[8] 3GPP TS 32.411: "Telecommunication management; Performance Management (PM) Integration Reference Point (IRP): Requirements".

[9] 3GPP TS 32.602: "Telecommunication management; Configuration Management (CM); Basic configuration management Integration Reference Point (IRP): Information service".

[10] 3GPP TS 32.342: "Telecommunication management; File transfer Integration Reference Point (IRP): Information service".

[11] 3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".

[12] 3GPP TS 32.302: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP): Information service".

[13] 3GPP TS 32.662: "Telecommunication management; Configuration Management (CM); Kernel CM information service".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TS 32.411 [8] and 3GPP TS 32.401 [7] apply.

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

CM	Configuration Management
CMIP	Common Management Information Protocol
CMIS	Common Management Information Service
CORBA	Common Object Request Broker Architecture
EM	Element Manager
IOC	Information Object Class
IRP	Integration Reference Point
NE	Network Element
NM	Network Manager
NRM	Network Resource Model
PM	Performance Management
UML	Unified Modelling Language

4 System overview

4.1 System context

Figures 4.1 and 4.2 identify system contexts of the IRP defined by the present specification in terms of its implementation called IRPAgent and the user of the IRPAgent, called IRPManager. For a definition of IRPManager and IRPAgent, see 3GPP TS 32.102 [2].

The IRPAgent implements and supports this IRP. The IRPAgent can reside in an Element Manager (EM) (see figure 4.1) or a Network Element (NE) (see figure 4.2). In the former case, the interfaces (represented by a thick dotted line) between the EM and the NEs are not the subject of this IRP.

An IRPAgent supports one of the two System Contexts defined here. By observing the interaction across this Itf-N, an IRPManager cannot deduce if EM and NE are integrated in a single system or if they run in separate systems.
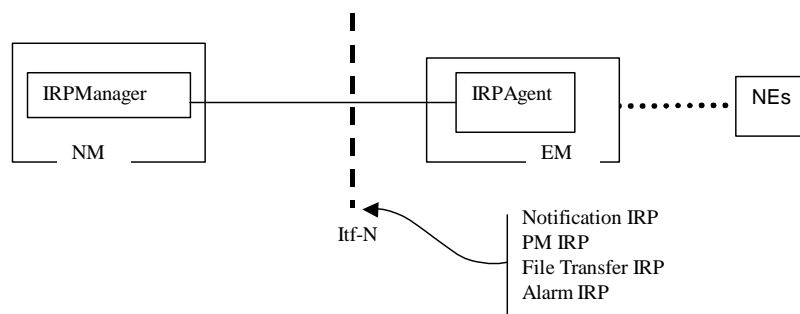
Figure 4.1: System Context A
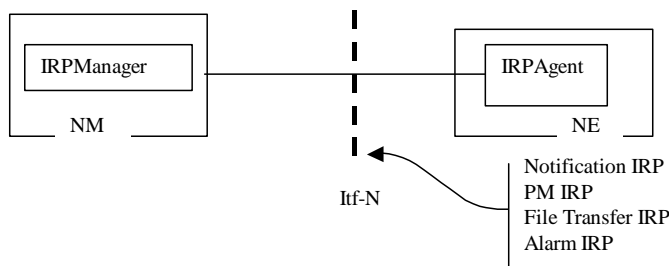
Figure 4.2: System Context B

4.2 Compliance rules

For general definitions of compliance rules related to qualifiers (Mandatory/Optional/Conditional) for *operations*, *notifications and parameters* (of operations and notifications) please refer to 3GPP TS 32.102 [2].

5 Void

6 Information Object Classes

6.1 Information entities imported and local labels

Label reference	Local label
3GPP TS 32.622 [6], information object class, Top	Top
3GPP TS 32.622 [6], information object class, IRPAgent	IRPAgent
3GPP TS 32.622 [6], information object class, GenericIRP	GenericIRP
3GPP TS 32.312 [5], information object class, ManagedGenericIRP	ManagedGenericIRP
3GPP TS 32.602 [9], information object class, ManagedEntity	ManagedEntity
3GPP TS 32.342 [10], information object class, FileDescriptor	FileDescriptor
3GPP TS 32.302 [12], information object class, NotificationIRP	NotificationIRP

6.2 Class Diagram

6.2.1 Attributes and relationships

This subclause introduces the set of Information Object Classes (IOCs) that encapsulate information within the IRPAgent. The intent is to identify the information required for the PMIRP Agent implementation of its operations and notification emission. This subclause provides the overview of all support object classes in UML. Subsequent subclauses provide more detailed specification of various aspects of these support object classes.
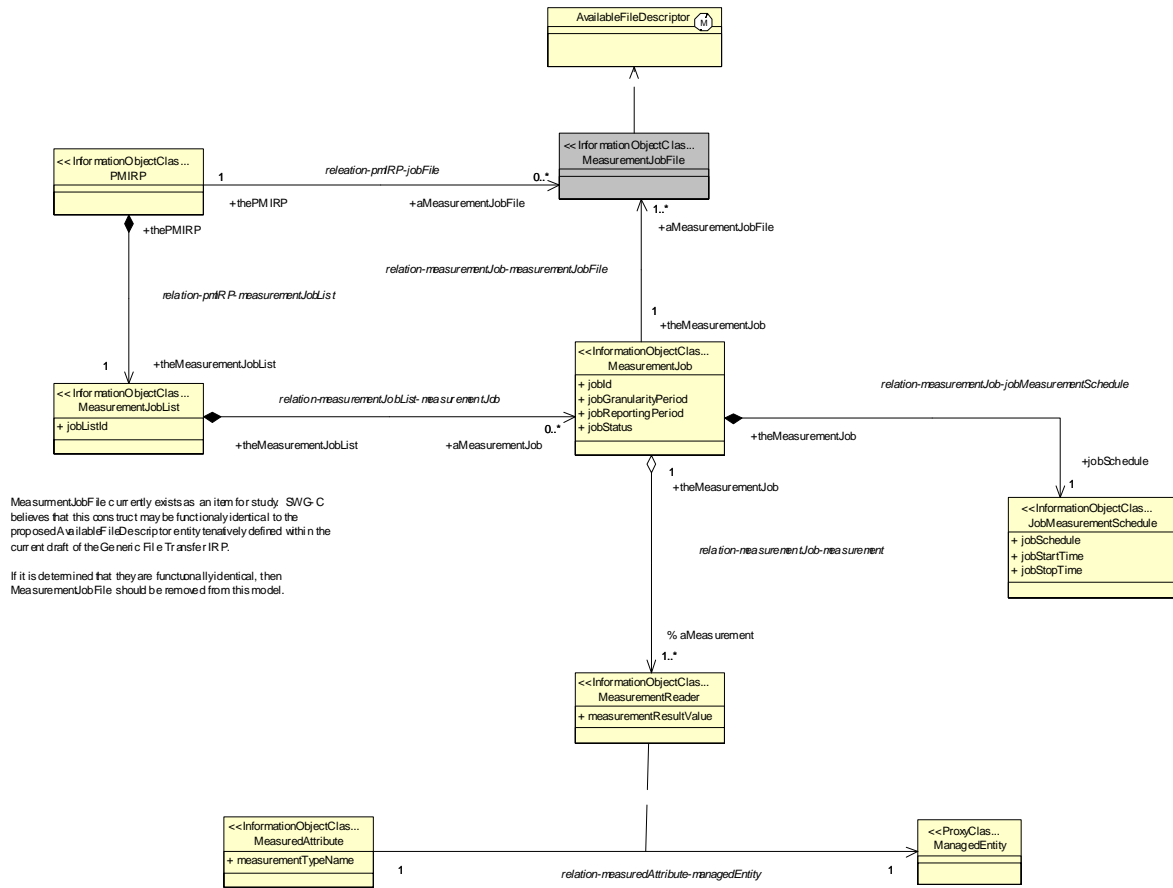
Figure 6.1(a): Information Object Class UML Diagram - Measurement
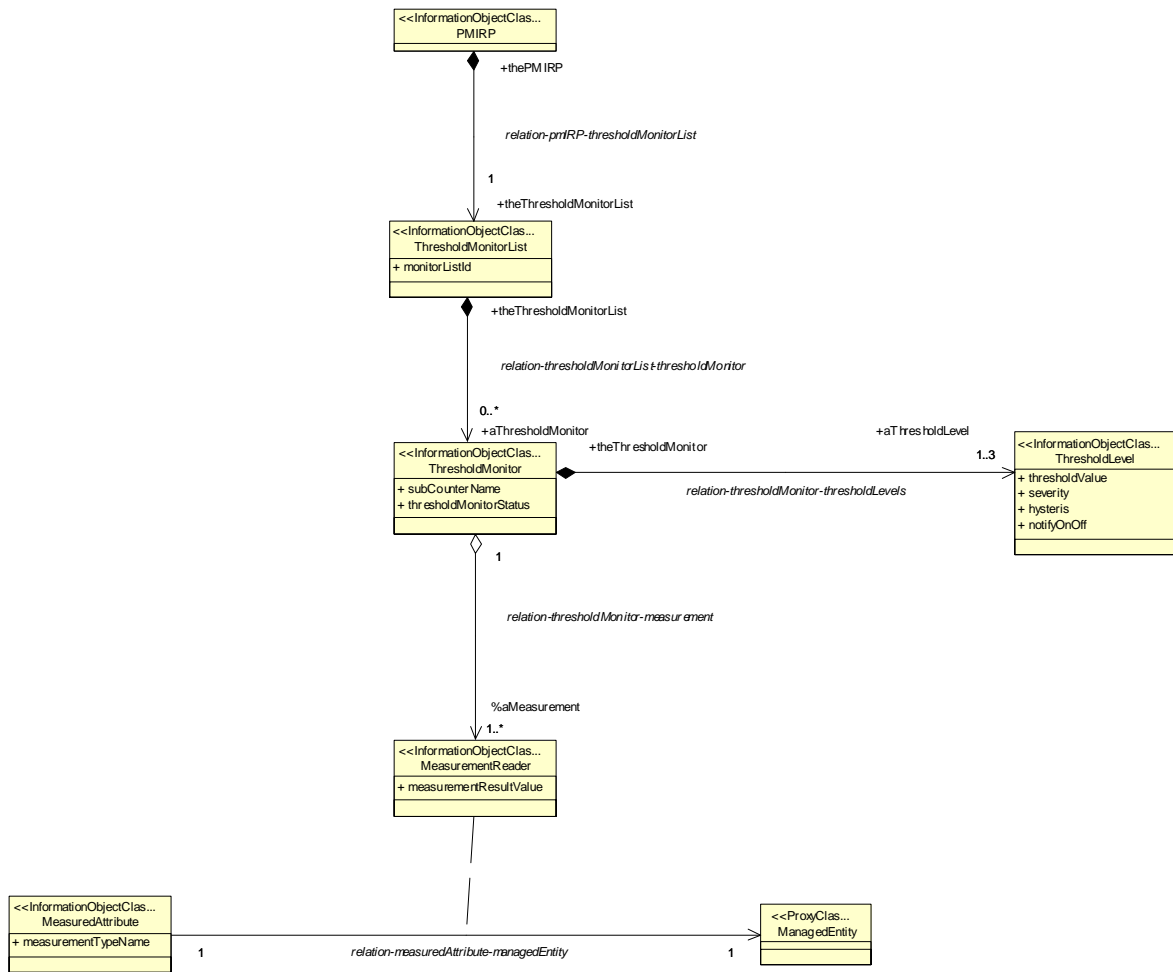
Figure 6.1(b): Information Object Class UML Diagram - Thresholding
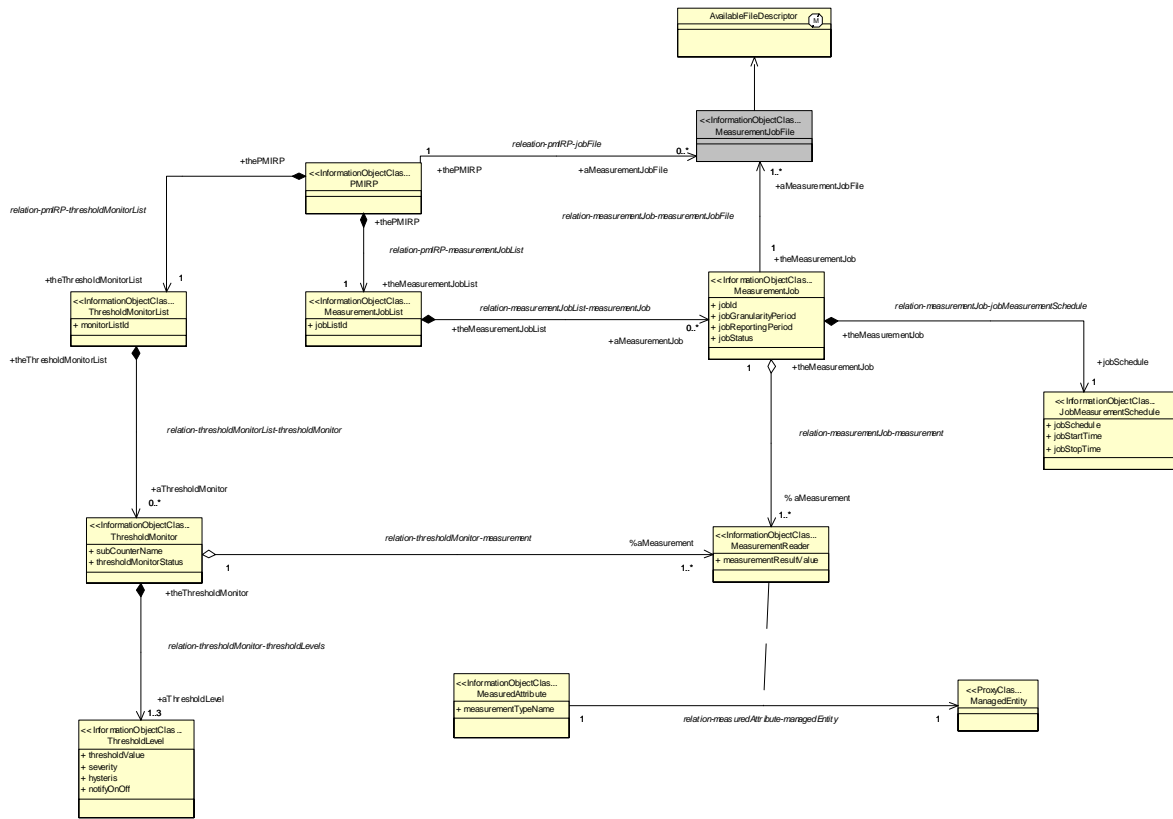
Figure 6.1(c): Information Object Class UML Diagram - Combined

6.2.2   Inheritance
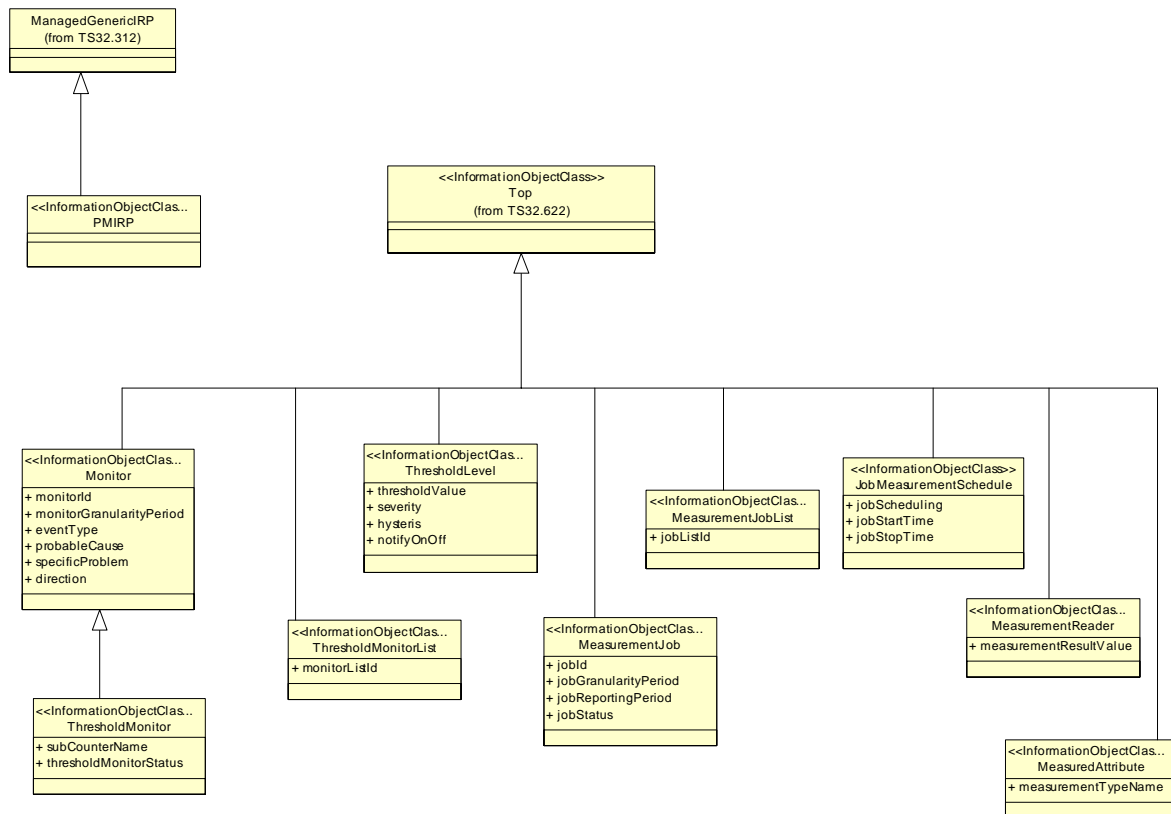
Figure 6.2: Information Object Class Inheritance UML Diagram

6.3   Information Object Class definitions

6.3.1   MeasurementJob

6.3.1.1   Definition

It represents a task that monitors and collects the performance measurement data, i.e. values of multiple measurementTypes of multiple ManagedEntity instances, at regular time intervals.

The target measurementTypes are those measurementTypes, whose names are in the related MeasurementTypeScanner. measurementTypeNames, of the related ManagedEntity instances. The attributes of MeasurementJob and measurementTypeScanner can not be modified (except jobStatus) once measurementJob is created.

The MeasurementJob shall use its information and that of the related measurementTypeScanner (s) to perform measurement data collection during the MeasurementJob life-time. At the time of data collection, if MeasurementJob suspects the validity of the collected monitored values, the MeasurementJob shall convey the fact to the IRPManager using the suspectFlag (3GPP TS 32.401 [7]) of the Report. The MeasurementJob will not emit any notification about this fact. Furthermore, the MeasurementJob shall continue to monitor the same target measurementTypes. Even in the worst possible case when the MeasurementJob cannot collect a single value from the target measurementTypes, the MeasurementJob must continue its activities according to the schedule created at MeasurementJob creation time.

[Editor's note: Ericsson wants to change "shall convey the fact to…" to "may" convey the fact to…."., because it would not be possible to test the suspect flag setting.]

[Editor's note: This paragraph needs further discussion about clear definition of "suspect flag". To clarify the conditions for setting the suspect flag. Compare with 3GPP TS 32.401 subclause 4.3.2.]

The PM IRPAgent may decide to stop a MeasurementJob because of a PM IRPAgent internal problem or other problems such as in the case when the related managed resource is not accessible (e.g., uninstalled) or there exist a prolonged communication link problem between the PM IRPAgent and the related managed resource. In such case, the PM IRPAgent can stop the MeasurementJob resulting in the MeasurementJob.jobStatus == "Stopped". And "notifyMeasurementJobStopped" notification shall be emitted to notify all subscribed IRPManagers about the stopping of a MeasurementJob.

[Editor's note: The last sentence is added by editor while add "notifyMeasurementJobStopped" notification.]

[Editor's note: It was discussed regarding the possibility to create several meas jobs on e.g. the same cell. "When deleting a cell, it must not exist any measurement cell on this cell.". New notification "PM-job stopped" needed? This notification needs to be described. (Added sentence in subclause 6.3.1.1, which need to be further reviewed).]

6.3.1.2   Attributes

Attribute name	Visibility	Support Qualifier	Read Qualifier	Write Qualifier
jobId	+	M	M	–
jobStartTime	+	M	M	–
jobStopTime	+	M	M	–
jobSchedule	+	O	M	–
jobGranularityPeriod	+	M	M	–
jobReportingPeriod	+	M	M	–
jobStatus	+	M	M	–

[Editor's note: Description of qualifiers in this table to be aligned with SWGC description when agreed.]

6.3.1.3   State Diagram

[Editor's note: should there be a 3gpp deleteMeasurementJob or a system parameter, time based, to govern the transition of MeasurementJob from Stopped to Exit.]

[Editor's note: "Measurement job visible via Itf-N (or not) after stop/delete operation" need further discussion. Verify the suitability of the word "Thoroughly".]

[Editor's note: are we missing a state for "prematurely" stopped measurement job?]
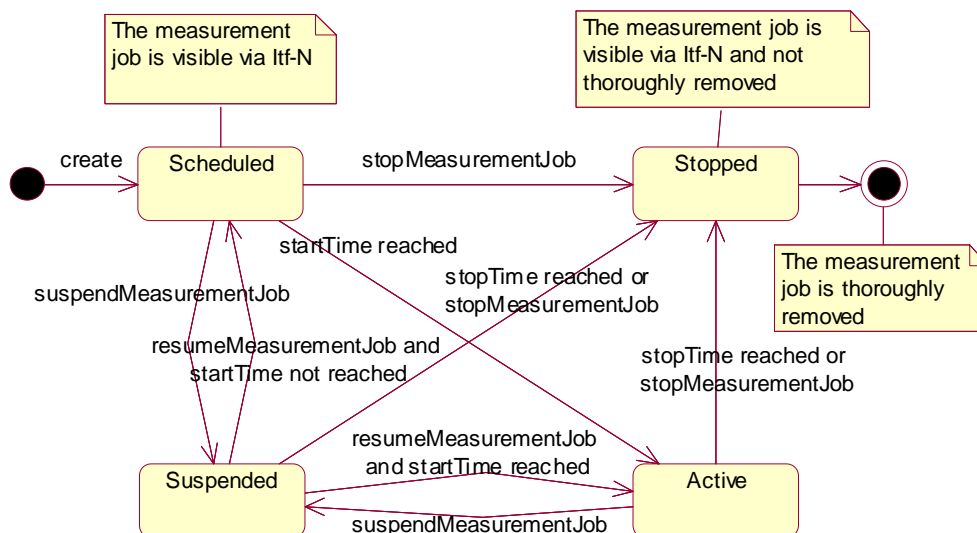
Figure 6.3: State Diagram for MeasurementJob
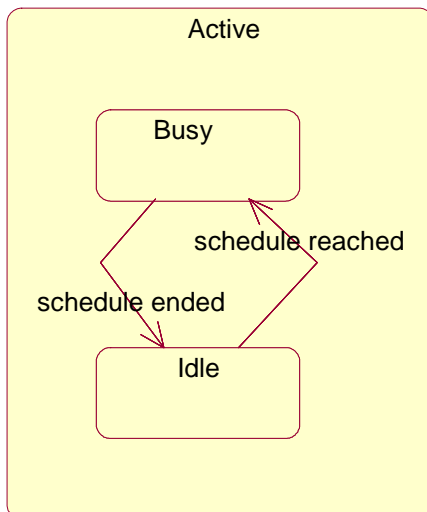
Figure 6.4: SubState Diagram of Active

The detailed description and state transition scenarios for MeasurementJob are in annex A.

6.3.2 PMIRP

6.3.2.1 Definition

PMIRP is the representation of the performance management capabilities specified by the present document. This IOC inherits from ManagedGenericIRP IOC specified in 3GPP TS 32.312 [5].

6.3.3 PMFileDescriptor

6.3.3.1 Definition

It represents the information about an existing closed file containing performance measurement data. The file is ready for retrieval by IRPManagers. The PM IRPAgent will not modify this file.

This inherits from File Transfer Descriptor IOC specified in 3GPP TS 32.342 [10].

[Editor's note: If the Requirement and IS of File Transfer IRP is available, the IOC PMFileDescriptor may be modified according to the File Transfer IRP.]

6.3.3.2 Attributes

Attribute name	Visibility	Support Qualifier	Read Qualifier	Write Qualifier
fileDirectory	+	M	M	-
fileName	+	M	M	-
fileSize	+	M	M	-
fileCompression	+	M	M	-
fileCreationTime	+	M	M	-
fileDeletionTime	+	M	M	-
fileServerAddress	+	M	M	-
fileAccessPassword(1)	+	M	M	-(2)
fileAccessUserName(1)	+	M	M	-(2)
fileFormat	+	M	M	-
fileURL	+	M	M	-

[Editor's note:

(1) Use of fileAccessPassword and fileAccessUserName is to be studied in conjunction with File Transfer IRP WT in SWGC.

(2) To be forwarded to Security Management IRP group.]

6.3.4 MeasurementJobList

6.3.4.1 Definition

MeasurementJobList is the representation of list of MeasurementJob.

6.3.5 MeasurementTypeScanner

6.3.5.1 Definition

It represents the role that can read the values of the measured measurementTypes of the related ManagedEntity instances. The names of the measured measurementTypes are captured by measurementTypeNames.

6.3.5.2 Attributes

Attribute name	Visibility	Support Qualifier	Read Qualifier	Write Qualifier
MeasurementTypeNames	+	M	M	–

6.3.6 ManagedEntity

6.3.6.1 Definition

The IOC ManagedEntity represents the role that can be played by an instance of an IOC defined in Network Resources Models, e.g. Generic Network Resource Model, Core Network Resource Model, UTRAN Network Resource Model or GERAN Network Resource Model. ManagedEntity is used in the specification of PM IRP operations to represent an instance of an IOC defined in these Network Resource Models.

6.3.7 Monitor

6.3.7.1 Definition

It represents a capability to determine the threshold-crossing and threshold-clearing. This class is abstract in that it cannot be instantiated. The ThresholdMonitor inherits this class.

The instance of this extended class, such as the ThresholdMonitor, shall emit notifyMonitorCreation when it was first created; and shall emit a notifyMonitorDeletion when it is deleted.

The instance of this extended class shall also emit notifyNewAlarm, notifyChangedAlarm and notifyClearedAlarm according to the rules specified in annex B.

6.3.7.2 Attribute

Attribute name	Visibility	Support Qualifier	Read Qualifier	Write Qualifier
monitorId	+	M	M	–
monitorGranularityPeriod	+	M	M	–
eventType	+	M	M	–
probableCause	+	M	M	–
specificProblem	+	M	M	–
direction	+	M	M	–

6.3.7.3 Notification

Notification name	Note
notifyNewAlarm	See subclause 7.9.1
notifyClearedAlarm	See subclause 7.9.2
notifyChangedAlarm	See subclause 7.9.3
notifyMonitorCreation	See subclause 7.9.4
notifyMonitorDeletion	See subclause 7.9.5

6.3.8 ThresholdMonitor

6.3.8.1 Definition

In order to monitor the overall health of the network, the thresholds are set by the authorized users to generate network performance related alarms. The ThresholdMonitor contains the values of the threshold settings for the PM parameters. If the threshold is crossed, a Quality of Service Alarm Notification will be emitted.

[Editor's note: whether the last sentence is "a Alarm Notification will be emitted." or "a Quality of Service Alarm Notification will be emitted" to be determined in future meeting.]

6.3.8.2 Attribute

Attribute name	Visibility	Support Qualifier	Read Qualifier	Write Qualifier
measurementTypeName	+	M	M	O
subCounterName	+	O	M	O
thresholdMonitorStatus	+	M	M	–

6.3.9 ThresholdLevel

6.3.9.1 Definition

It defines the thresholding criteria (via ThresholdLevel. measurementTypeValue and ThresholdLevel.hysteresis) for a measurementType or subCounter. It also specifies the severity level (via ThresholdLevel.severity) carried in the alarm triggered by the threshold crossing event.

This instance cannot be modified once created.

6.3.9.2 Attribute

Attribute name	Visibility	Support Qualifier	Read Qualifier	Write Qualifier
thresholdValue	+	M	M	O
severity	+	M	M	O
hysteresis	+	M	M	O
notifyOnOff	+	O	M	O

6.4 Information relationship definitions

[Editor's note: names of relations need to be aligned with those in the class diagram.]

6.4.1 Relation-pMIRP-MeasurementJobList (M)

6.4.1.1 Definition

This represents the relationship between PMIRP and MeasurementJobList.

6.4.1.2 Role

Name	Definition
The Measurement`JobList`	It represents the Measurement`JobList`.

6.4.1.3 Constraint

Name	Definition
unique`JobListId`	The `jobListIds` playing the role of the`Measurement`JobList`, are unique within a particular `PMIRP`.

6.4.2 Relation-MeasurementJobList-MeasurementJob (M)

6.4.2.1 Definition

This represents the relationship between Measurement`JobList` and Measurement`Job`.

6.4.2.2 Role

Name	Definition
the`Job`	It represents the Measurement`Job`.

6.4.2.3 Constraint

Name	Definition
unique`JobId`	The `jobIds` of all Measurement`Jobs`, playing the role of `theJob`, are unique within a particular `PMIRP` and Measurement`JobList`.

6.4.3 Relation-MeasurementJobList-pMFileDescriptor (M)

6.4.3.1 Definition

This represents the relationship between `MJobList` and PMFileDescriptor.

6.4.3.2 Role

Name	Definition
the`PMFile`	It represents the PMFileDescriptor.

6.4.3.3 Constraint

Name	Definition
unique`FileName`	The concatenation of `fileDirectory`, `filename` of all `PMFileDescriptor` instances, playing the role of the `PMFile` with the same `PMIRP`, must be unique among all `PMFileDescriptors` or the `URL`, playing the same role, must be unique among all `PMFileDescriptors`.

6.4.4 Relation-MeasurementJob-MeasurementTypeScanner (M)

6.4.4.1 Definition

This represents the relationship between Measurement`Job` and `MeasurementTypeScanner`.

6.4.4.2 Role

Name	Definition
the`MTS`	It represents the `MeasurementTypeScanner`.

6.4.5 Relation-MeasurementTypeScanner-managedEntity (M)

6.4.5.1 Definition

This represents the relationship between `MeasurementTypeScanner` and ManagedEntity.

6.4.5.2 Role

Name	Definition
the`MTS`	The `MeasurementTypeScanner`, when playing this role, reads the monitored measurement types of the related `ManagedEntity instances`. The names of measurement types are captured by the `MeasurementTypeScanner.measurementTypeNames`. When playing this role, the `MeasurementTypeScanner` is aware of the class name and DN of the `ManagedEntity`.
the`ME`	The `managedEntity`, when playing this role, represents the actual network resource instance under measurement.

6.4.6 Relation-pMIRP-tMonitor (M)

6.4.6.1 Definition

This represents the relationship between `PMIRP` and ThresholdMonitor.

6.4.6.2 Role

Name	Definition
`thresholdMonitor-1`	It represents the ThresholdMonitor.

6.4.6.3 Constraint

Name	Definition
`uniqueMonitorId`	The monitor`Ids` of all thresholdMonitor`s`, playing the role of `theThresholdMonitor-1`, are unique within a particular `PMIRP`.

6.4.7 Relation-tMonitor-mE (M)

6.4.7.1 Definition

This represents the relationship between ThresholdMonitor and ManagedEntity.

6.4.7.2 Role

Name	Definition
thresholdMonitor-2	The ThresholdMonitor reads the monitored measurementTypes or subCounters of the related ManagedEntity instances. When playing this role, the ThresholdMonitor is aware of the class name and DN of the ManagedEntity whose measurementTypes or subCounters are being monitored.
theME	TheME, when playing this role, represents the actual network resource instance whose measurementTypes or subCounters are being monitored for threshold-crossing and threshold-clearing.

6.4.7.3 Constraint

Name	Definition
applyToSameObjectClass	The ThresholdMonitor, which playing the role of 'theThresholdMonitor-2', can monitor the thresholds of the measurementTypes of one or multiple managed object instances of the same managed object class, which playing the role of "theME".

6.4.8 Relation-tMonitor-tLevel (M)

6.4.8.1 Definition

This represents the relationship between ThresholdMonitor and ThresholdLevel.

6.4.8.2 Role

Name	Definition
thresholdMonitor-3	When playing this role, the ThresholdMonitor can access the threshold-crossing and clearing criteria (i.e. ThresholdLevel.thresholdValue, ThresholdLevel.hysteresis) of the monitored measurementType or subCounter. It can also access the associated severity (i.e. ThresholdLevel.severity).
thresholdLevel	When playing this role, the ThresholdLevel captures the threshold-crossing and threshold-clearing criteria and its associated severity level.

6.4.8.3 Constraint

Name	Definition
noMoreThanThreeLevels	A ThresholdMonitor can only have no more than three ThresholdLevels.

6.5 Information attribute definition

6.5.1 Definition and legal values

Attribute Name	Definition	Legal Values
jobId	It identifies the measurementJob from all other existing and stopped measurementJob instances.	Its value is string.
jobStartTime	It specifies the begin time from which the MeasurementJob will be active.	All values that indicate valid time. Default value for start is "start now"
jobStopTime [Editor's note: To check whether "Atribute Name", "Definition" and "Legal Values" of "jobStopTime" is aligned with corresponding definition in 3GPP TS 32.401.]	It specifies the end time after which the MeasurementJob will be stopped.	All values that indicate valid time and it should be later than jobStartTime. It's not necessary that jobStartTime and jobStopTime specifies time within the same day. Default value for stop is "never stop".

Attribute Name	Definition	Legal Values
jobSchedule	It specifies the detailed time frames during which the measurementJob.jobStatus = Active and its substate = Busy.	Its value is only one of the following, dailyScheduling or weeklyScheduling. The legal values for them refer to ITU-T Recommendation X.721 [3].
jobGranularityPeriod	It specifies the period between two successive measurements.	The value can be 5 minutes, 15 minutes, 30 minutes, 1 hours, 12 hours and 24 hours, etc. The minimum granularity period is 5 minutes in most cases, but for some measurements it may only make sense to collect data in a larger granularity period.
jobReportingPeriod	It specifies the period between two successive emissions of notifyPMJobFileReady. The two notifications are related to the same Job. See constraints reportTime in subclause 6.5.2.	Its value should be one or multiple of jobGranularityPeriod.
jobStatus	It specifies the status of measurementJob.	Its value should be one of the following: Scheduled, Active, Suspended Stopped"
fileFormat	It identifies the encoding technique used by the file.	Its value should indicate the IRPVersion of the PM file format specification (3GPP TS 32.401 [7]) plus to indicate if "ASN1" or "XML-DTD" or "XML-schema" is used.
FileURL [Editor's note: To be clarified whether 4 fields are needed (fileURL, fileDirectory, filename, fileServerAddress)]	It identifies the URL of the file described by PMFileDescriptor.	String. E.g.: ftp://nms.telecom_org.com/datastore/<xxx> where <xxx> is the fileName defined in 3GPP TS 32.401 [7] subclause B.1.2.
fileDirectory	It identifies the file directory under which the related file, described by PMFileDescriptor, is stored.	String. E.g: "D:\user\performanceFiles\"
fileName	It identifies the name of file containing performance measurement data and is described by PMFileDescriptor	String. The file naming convention is defined in 3GPP TS 32.401 [7] subclause B.1.2.
fileSize	It identifies the size of a file containing performance measurement data and is described by PMFileDescriptor.	Its value is positive Integer (the unit is byte).
fileCompression	It identifies the name of a file compression algorithm. The file contains performance measurement data and is described by PMFileDescriptor.	An empty fileCompression means that there is no compression on the file. Choice of compression algorithm is vendor-specific but is encouraged to use industrial standard algorithm such as GZIP. [Editor's note: Choice of compression algorithm might be modified later to align with FT IRP.]
fileCreationTime	It identifies the file creation date and time.	All values that indicate valid time and should not be earlier than the related MeasurementJob creation time.
fileDeletionTime	It identifies the date and time beyond which the file may be deleted.	All values that indicate valid time. It should be later than fileCreationTime. An empty value means "file remains forever".
fileServerAddress	It identifies the address of the server holding the file.	String.
fileAccessUserName	It identifies the username for fetching the file.	A string using valid characters defined for use by Distinguished Name 3GPP TS 32.300 [11].
fileAccessPassword	It identifies the password for fetching the file.	A string using valid characters defined for use by Distinguished Name 3GPP TS 32.300 [11].
measurementTypeNames	It identifies names of measurement types under monitoring.	List of measurement type name.

Attribute Name	Definition	Legal Values
monitorId	It specifies the unique ThresholdMonitor in the PMIRP Agent.	String.
measurementTypeName	It identifies a name of one measurement type whose value is being monitored for threshold-crossing and threshold-clearing.	String
subCounterName	It identifies a name of one subCounter whose value is being monitored for threshold-crossing and threshold-clearing.	String

thresholdMonitorStatus	It specifies the current status of the ThresholdMonitor.	ENUM of <
		    Active - ThresholdMonitor is working;
		    Suspended - ThresholdMonitor is
		              suspended.
		>

monitorGranularityPeriod	It specifies the period between two successive reading of the thresholdValue to determine threshold-crossing and threshold-clearing.	It can be 5 minutes, 15 minutes, 30 minutes, 1 hour, 12 hours or 24 hours.
eventType	It identifies the event type carried by the threshold crossing alarm.	It is "Quality of Service Alarm" 3GPP TS 32.111-2 [4]
probableCause	It identifies the probable cause (of the threshold crossing) carried by the threshold crossing alarm.	String.
specificProblem	It identifies the specific problem (causing the threshold crossing) carried by the threshold crossing alarm.	String
direction	Some measurementType, the higher its thresholdValue, the higher is the severity. For others, the lower its thresholdValue, the lower is its severity. This attribute identifies if the measurementType is of the former (i.e. "Increasing") or latter type (i.e. "Decreasing"). If it is "Increasing", the threshold is said to be crossed when the value rises (when compared against the last read value) across one or more threshold values. The threshold is said to be cleared when the measurementType value falls (when compared against the last read value) across one or more threshold values. If it is "Decreasing", the threshold is said to be crossed when the measurementType value falls across one or more threshold values. The threshold is said to be cleared when the measurementType rises across one or more threshold values and is now above all threshold values.	ENUM <"Increasing", "Decreasing">

Attribute Name	Definition	Legal Values
thresholdValue	It defines the threshold value of the monitored measurementTypes or subCounters. If the value is crossed, the performance alarm may be emitted.	If the monitored measurementType or subCounter is of Gauge type, this thresholdValue shall be of the same type. If the monitored measurementType or subCounter is of counter type, then this value should be expressed as a rate, i.e., the number of units of type of the monitored measurementType or subCounter over unit of time. Note this rate is independent from the monitorGranularityPeriod. This means that changes in the monitorGranularityPeriod should not impact the rate used for threshold monitoring.
hysteresis	A threshold has a value. It can have a hysteresis. A threshold with a hysteresis has a threshold-high and a threshold-low values that are different than the threshold value. A threshold without a hysteresis can be considered as a threshold whose threshold-high and threshold-low values are equal to the threshold value. A hysteresis, therefore, defines the threshold-high and threshold-low levels within which the measurementType value is allowed to oscillate without triggering a threshold-crossing or threshold-clearing condition. See annex B.	Integer or Real
severity	It identifies the severity of the threshold crossing event.	ENUM <Minor, Major, Critical>
notifyOnOff	It specifies whether the performance alarm will be emitted when threshold is crossed.	Boolean

6.5.2   Constraints

Name	Definition
measurementTypeReadTimes	The measurementType reading times are determined by the following rules. - If granularityPeriod is 5 minutes, then read on every 5th minute of the hour. - If granularityPeriod is 15 minutes, then read on every 15th minute of the hour. - If granularityPeriod is 30 minutes, then read on every 30th minute of the hour - If granularityPeriod is 1 hour, then read on every hour of the day - If granularityPeriod is 12 hours, then the first read shall be as soon as possible on the hour and subsequent reads shall be at 12 hours interval. - If granularityPeriod is 24 hours, then the first read shall be as soon as possible on the hour and subsequent reads shall be at 24 hours interval.
firstReportTime	If the reportingPeriod is n times the granularityPeriod, then the first report time shall be immediately after the completion of the n-th successful measurementType reading.
numberOfThresholdLevelPerThresholdMonitor	In relation-tMonitor-tLevel, the multiplicity of theThresholdLevel is "1..3". If PMIRP supports multi-level severity, then there are at most 3 ThresholdLevels relating to one ThresholdMonitor. All ThresholdLevel.severity instances shall hold a different value that is chosen from the following: 'Critical', 'Major' and 'Minor'. If PMIRP does not support multi-level severity, then there is only one ThresholdLevel relating to one ThresholdMonitor. The sole ThresholdLevel.severity shall hold one of the following: 'Critical', 'Major' or 'Minor'.
multiLevelSeverityOrder	The multiple ThresholdLevel(s), related to one ThresholdMonitor, shall be of the following order. If ThresholdLevel.severity of one instance is higher than that of the other instance and the direction is 'Increasing', then the ThresholdLevel.thresholdValue is higher than that of the other. If ThresholdLevel.severity of one instance is higher than that of the other instance and the direction is 'Decreasing', then the ThresholdLevel.thresholdValue is lower than that of the other.
multiLevelSeverityHysteresisOverLap	The values of ThresholdLevel.hysteresis, relating to one ThresholdMonitor, shall not overlap.

7 Interface definition

[Editor's note: M & O qualifiers are still under discussion.]

7.1 Class diagram

[Editor's note:

(1) to be replaced by contribution Motorola's S5-038224 pending agreement.

(2) Removal of maxJobAllowed from the diagram.]
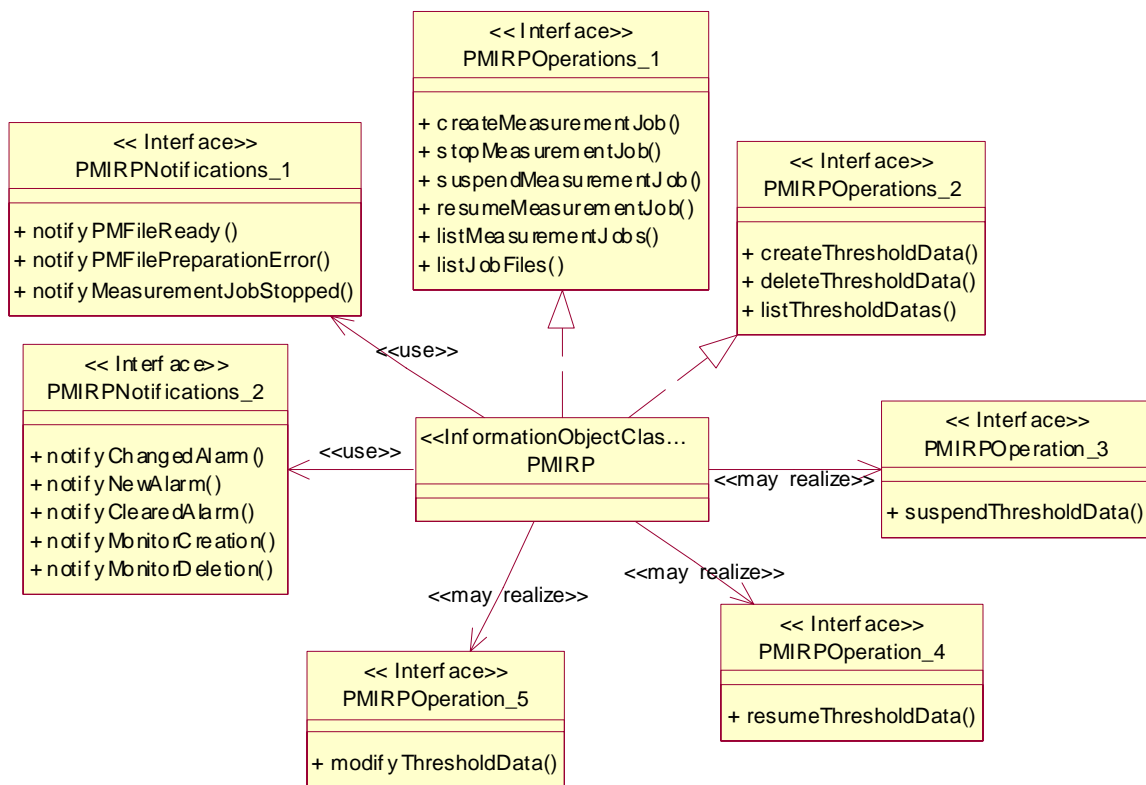
Figure 7.1: Class Diagram

7.2 Generic rules

- **Rule 1:** each operation with at least one input parameter supports a pre-condition valid_input_parameter which indicates that all input parameters shall be valid with regards to their information type. Additionally, each such operation supports an exception operation_failed_invalid_input_parameter which is raised when pre-condition valid_input_parameter is false. The exception has the same entry and exit state.

- **Rule 2:** each operation with at least one optional input parameter supports a set of pre-conditions supported_optional_input_parameter_xxx where "xxx" is the name of the optional input parameter and the pre-condition indicates that the operation supports the named optional input parameter. Additionally, each such operation supports an exception operation_failed_unsupported_optional_input_parameter_xxx which is raised when (a) the pre-condition supported_optional_input_parameter_xxx is false and (b) the named optional input parameter is carrying information. The exception has the same entry and exit state.

- **Rule 3:** each operation shall support a generic exception operation_failed_internal_problem which is raised when an internal problem occurs and that the operation cannot be completed. The exception has the same entry and exit state.

NOTE: These rules are mapped at the solution set level. Pre-conditions and exceptions, generated by these rules, need not appear explicitly in the present document.

7.3 PMIRPOperations_1 Interface

7.3.1 Operation createMeasurementJob (M)

7.3.1.1 Definition

IRPManager invokes this operation to request IRPAgent to create a MeasurementJob through Itf-N.

Once created, the attributes of MeasurementJob (except MeasurementJob.jobStatus) and the related measurementTypeScanner will not be modified during the life-time of the MeasurementJob.

One MeasurementJob can monitor and collect the value of one or multiple measurementTypes.

When a measurementType is monitored or collected by one MeasurementJob, another MeasurementJob which wants to monitor or collect the same measurementType with different granularity period will be rejected.

[Editor's note:

(1) The original editor's note "One pMAttribute can only be monitored or collected by just one measurement" with comments "clarify that it is for a given object instance" was replaced by the above two paragraphs.

(2) add "instance" after "measurementType in 4th paragraph ?

(3) add "or same" after "different" in 4th paragraph ?

(4) all attribute names should begin with a small letter.]

7.3.1.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
moClass	M	ManagedEntity.objectClass	It specifies one managed entity class name. IRPManager requests that one or more measurementType (s) of the instances of this class to be monitored.
moInstanceList	M	List of ManagedEntity.objectInstance	It specifies the list of managed object instances to be measured. MoInstanceList can be absent, if so, the MeasurementJob will be applied to all known (to PMIRP) instances of the specified MO class. [Editor's note: To be reconsidered. At least, one instance should be specified ???]
measurementCategoryList	M	LIST of MeasurementTypeScanner. measurementTypeName	MeasurementTypeScanner. MeasurementTypeName specifies the corresponding name of measurementType to be measured. This field can be absent, if so the MeasurementJob will be applied to all the measurementTypes of the specified MOC or MOI. [Editor's note: To be reconsidered: at least one pMAttributes should be specified here ???]
granularityPeriod	M	MeasurementJob.jobGranularityPeriod	See subclause 6.5.1.

Parameter Name	Qualifier	Information type	Comment
reportingPeriod	M	MeasurementJob.jobReportingPeriod	See subclause 6.5.1. EM/NE will periodically scan the performance data according to the "reportingPeriod" parameter and produces an aggregated report in one or more files. These files can be transferred through file transfer mechanism which could eventually be covered by a separate File Transfer IRP (refer to 3GPP TS 32.xxx [x]). [Editor's note: To be addressed again when FT IRP documents are available.]
startTime	O	MeasurementJob.jobStartTime	See subclause 6.5.1.
stopTime	O	MeasurementJob.jobStopTime	See subclause 6.5.1.
Schedule	O	MeasurementJob.jobSchedule	See subclause 6.5.1.

7.3.1.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
jobId	M	MeasurementJob.jobId	Unique identifier of the MeasurementJob from all the existing and stopped MeasurementJobs in a PMIRP Agent.

unsupportedList	M	List of STRUCT <
		    ManagedEntity.objectInstance,
		    MeasurementTypeScanner.measurementTypeName,
		    reason
		>
			To create a MeasurementJob, best-effort is required. The parameter of 'unsupportedList' must be returned if status = PartialSuccess.
			The reason can be:
			(a) measurementType is unknown to the PMIRP or
			(b) It is invalid (e.g., the name contains illegal characters) or
			(c) It is not supported in the specific implementation.
			(d) The related monstrance is unknown (e.g. it does not exist at the time of this operation invocation) to the PMIRP.(e) Insufficient capacity to monitor the related monstrance(s).

Status	M	ENUM (Success, Failure, PartialSuccess)	An operation may fail because of a specified or unspecified reason.

7.3.1.4 Pre-condition

validMeasurementInfoList AND validSchedule AND validGranularityPeriod AND validReportingPeriod AND sufficientCapacity.

Assertion Name	Definition
atLeastOneValidMeasurementType	At least one input measurementType is valid. The word valid means that the PMIRP is aware of this measurementType name, is aware of the ManagedEntity holding such measurementType and that the MeasurementJob created can monitor its value according to the monitoring criteria, e.g. granularity period, given in the input parameter.
validStartTime	The startTime is valid.
ValidStopTime	The stopTime is valid. StopTime should be larger than startTime.
validSchedule	The schedule is valid.
validGranularityPeriod	The granularityPeriod is valid.
validReportingPeriod	The reportingPeriod is valid.
sufficentCapacity	The resource of EM and NE can support the requested measurement.
noConflictingGranularityPeriod	All the specified measurementTypes have not been monitored with another granularity period.

7.3.1.5 Post-condition

measurementJobIsCreated AND (someMeasurementTypesMonitored OR allMeasurementTypesMonitored).

Assertion Name	Definition
measurementJobIsCreated.	"The measurementJob identified by jobId exists. All MeasurementJob.measurementTypeNames are known. All related ManagedEntity instances are known. A 'known' measurementType name implies that PMIRP have confidence that it is a defined measurementType for the related ManagedEntity. A 'known' ManagedEntity implies that PMIRP have confidence that the instance exist in the managed network. The word 'known' does not imply that, at future data collection time, these known entities still exist or functioning well to respond to data collection. Depending on the current time and the input schedule, the MeasurementJob.jobStatus will be set accordingly. EXAMPLE: If input schedule indicates to "start now", then the MeasurementJob.jobStatus is set to Active. If input schedule indicates to start at 02:00 PM and the current time is 1:30 PM, the MeasurementJob.jobStatus is set to Scheduled.
someMeasurementTypesMonitored	At least one but not all input measurementTypes are monitored by the newly created MeasurementJob.
allmeasurementTypesMonitored	All input measurementTypes are monitored by the newly created MeasurementJob.

7.3.1.6 Exceptions

Exception Name	Definition
InvalidMeasurementInfo	**Condition:** (MeasurementInfoListIsValid) not verified. **Returned information:** output parameter status is set to 'failure'. **Exit state:** Entry State.
invalidStartTime	**Condition:** validStartTime is false. **Returned Information:** Name of the exception; status is set to 'failure'. **Exit state:** Entry state.
invalidStopTime	**Condition:** validStopTime is false. **Returned Information:** Name of the exception; status is set to 'failure'. **Exit state:** Entry state.
invalidSchedule	**Condition:** validSchedule is false. **Returned Information:** Name of the exception; status is set to 'failure'. **Exit state:** Entry state.
invalidGranularityPeriod	**Condition:** validGranularityPeriod is false. **Returned Information:** Name of the exception; status is set to 'failure'. **Exit state:** Entry state.
invalidReportingPeriod	**Condition:** validReportingPeriod is false. **Returned Information:** Name of the exception; status is set to 'failure'. **Exit state:** Entry state.
highWorkLoad	**Condition:** sufficientCapacity is false. **Returned Information:** Name of the exception and the detailed reason specifies whether EM or NE is high workload; status is set to 'failure'. **Exit state:** Entry state.
conflictingGranularityPeriod	**Condition:** noConflictingGranularityPeriod is false. **Returned Information:** Name of the exception; status is set to 'failure'. **Exit state:** Entry state.

7.3.2 Operation stopMeasurementJob (M)

7.3.2.1 Definition

This operation supports NM to stop a MeasurementJob through Itf-N, after which, the MeasurementJob will still be visible via Itf-N. Whether the MeasurementJob is thoroughly removed from the managed system is vendor specific and out of scope of the present document.

[Editor's note: whether delete "still" or not is to be discussed.]

7.3.2.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
jobId	M	MeasurementJob.jobId	It specifies the MeasurementJob to be stopped.

7.3.2.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
status	M	ENUM (Success, Failure)	An operation may fail because of a specified or unspecified reason.

7.3.2.4 Pre-condition

measurementJobExists AND measurementJobCanBeStopped.

Assertion Name	Definition
measurementJobExists	"MeasurementJob specified in the input parameter exists."
measurementJobCanBeStopped	"MeasurementJob specified in the input parameter can be stopped now"

7.3.2.5 Post-condition

measurementJobIsStopped.

Assertion Name	Definition
measurementJobIsStopped.	"the MeasurementJob identified by JobId is still visible from Itf-N". [Editor's note: whether delete "still" or not is to be discussed.]

7.3.2.6 Exceptions

Exception Name	Definition
unknownJob	**Condition:** (MeasurementJobExists) not verified. **Returned information:** output parameter status is set to 'failure'. **Exit state:** Entry State.
jobCannotBeStopped	**Condition:** (measurementJobCanBeStopped not verified. **Returned information:** output parameter status is set to 'failure'. **Exit state:** Entry State.

7.3.3 Operation suspendMeasurementJob (M)

7.3.3.1 Definition

This operation supports NM to suspend a MeasurementJob through Itf-N. When the MeasurementJob is suspended, the collection of measurement result data by the MeasurementJob will stop, regardless of its schedule, but the MeasurementJob still exists. The suspend operation is necessary in following situation:

- High work load experienced by managed system.

- The specified measurement data is not needed in a specific duration.

- Other specific requirement.

7.3.3.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
jobId	M	MeasurementJob.jobId	It specifies the MeasurementJob to be suspended.

7.3.3.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
status	M	ENUM (Success, Failure)	An operation may fail because of a specified or unspecified reason.

7.3.3.4 Pre-condition

measurementJobExists AND measurementJobIsNotSuspended.

Assertion Name	Definition
measurementJobExists	"The MeasurementJob specified in the input parameter exists."
measurementJobIsNotSuspended	"The MeasurementJob specified in the input parameter is not suspended."

7.3.3.5 Post-condition

measurementJobIsSuspended.

Assertion Name	Definition
measurementJobIsSuspended	"The MeasurementJob identified by measurementJobId is suspended. It means the following. (a) All files open for capturing measurement results are closed. (b) If one of the files closed contain measurement data, a notifyPMFileReady is emitted indicating the availability of the file(s). (c) No more reading of measurementType values. (d) The MeasurementJob.jobStatus is set to 'Suspended'."

7.3.3.6 Exceptions

Exception Name	Definition
unknownJob	**Condition:** (measurementJobExists) not verified. **Returned information:** output parameter status is set to 'failure'. **Exit state:** Entry State.
jobHasBeenSuspended	**Condition:** (measurementJobIsNotSuspended) not verified. **Returned information:** output parameter status is set to 'failure'. **Exit state:** Entry State.

7.3.4 Operation resumeMeasurementJob (M)

7.3.4.1 Definition

This operation supports NM to resume a suspended MeasurementJob. When the MeasurementJob is resumed, it will work according to criteria (e.g. granularity period, startTime, stopTime, schedule) set up by the corresponding createMeasurementJob.

7.3.4.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
jobId	M	MeasurementJob.jobId	It specifies the MeasurementJob to be resumed.

7.3.4.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
status	M	ENUM (Success, Failure)	An operation may fail because of a specified or unspecified reason.

7.3.4.4   Pre-condition

`measurementJobExists` AND `measurementJobIsSuspended`.

Assertion Name	Definition
measurementJobExists	"MeasurementJob specified by the input parameter exists."
measurementJobIsSuspended	"MeasurementJob specified by the input parameter is suspended."

7.3.4.5   Post-condition

`measurementJobIsResumed`.

Assertion Name	Definition
measurementJobIs Resumed	"The `MeasurementJob` identified by `MeasurementJobId` is resumed, It means the following. This resumed MeasurementJob shall behave as if it is first created by `createMeasurementJob`. Depending on the current time and the `MeasurementJob.startTime`, the `MeasurementJob.jobStatus` will be set accordingly."

7.3.4.6   Exceptions

Exception Name	Definition
unknownJob	**Condition:** (measurementJobExists) not verified. **Returned information:** output parameter status is set to 'failure'. **Exit state:** Entry State.
jobIsNotSuspended	**Condition:** (measurementJobIsSuspended) not verified. **Returned information:** output parameter status is set to 'failure'. **Exit state:** Entry State.

7.3.5   Operation listMeasurementJobs (M)

7.3.5.1   Definition

This operation allows `IRPManager` to list the information of all or of specified current `MeasurementJobs`.

7.3.5.2   Input parameters

Parameter Name	Qualifier	Information type	Comment
retrieveInfo	M	List of MeasurementJob.jobId	This parameter specifies the criteria to list the `MeasurementJobs`. If the parameter specifies the list of jobId to be retrieved, then the corresponding information of jobs will be returned. If the parameter contains no information, all the `MeasurementJobs` are retrieved.

7.3.5.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment

jobInfoList	M	LIST OF STRUCT <
		    attributes of `MeasurementJob`,
		    (refer to subclause 6.5.1)
		    moClass,
		    moInstanceList,
		    `measurementCategoryList`
		    (refer to subclause 7.3.1.2)
		 >
			Returned information of corresponding MeasurementJobs matching the input criteria. If no match, then the length of the list will be 0 (with status == Success).

status	M	ENUM (Success, Failure)	An operation may fail because of a specified or unspecified reason.

7.3.5.4 Pre-condition

retrieveInfoIsValid.

Assertion Name	Definition
retrieveInfoIsValid	"retrieveInfo specified in the input parameter is valid."

7.3.5.5 Post-condition

measurementJobsAreReturned AND measurementJobNotAffected.

Assertion Name	Definition
measurementJobsAreReturned	"the information of required `MeasurementJobs` are returned in the output parameters".
measurementJobNotAffected	"The current `MeasurementJob(s)` are not affected by the operation".

7.3.5.6 Exceptions

Exception Name	Definition
errorCriteria	**Condition:** (retrieveInfoIsValid) not verified. **Returned information:** output parameter status is set to 'failure'. **Exit state:** Entry State.

7.3.6 Operation listJobFiles (M)

7.3.6.1 Definition

This operation allows IRPManager to list all or specified measurement data files stored in the NE/EM. These performance data files are collected for the specified MeasurementJob in the specified file creation time range.

A Solution Set may choose to split this operation in several operations (e.g. operations to get "iterator" which fulfill the criteria and other operations to retrieve the detailed information of the files from the "iterator").

7.3.6.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
jobId	M	MeasurementJob.jobId	It specifies the MeasurementJob whose measurement data files will be returned.
beginTime	O	---	The IRPManager wants to list information about the stored file(s) whose creation times are between (on or after) T1 and (before) T2. This parameter specifies T1. It indicates date and time. If this parameter is not supported or if this parameter is supported but it contains no information, it means the IRPManager places no restriction on the T1.

Parameter Name	Qualifier	Information type	Comment
endTime	O	---	This parameter specifies T2 (see above). It indicates date and time. If this parameter is not supported or if this parameter is supported but it contains no information, it means T2 is the same as the invocation time of this operation.

7.3.6.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment

jobFileInfoList	M	STRUCT <
		    PMFileDescriptor.fileServerAddress(M),
		    PMFileDescriptor.fileUserName(M),
		    PMFileDescriptor.filePassword(M),
		    PMFileDescriptor.fileCompression(M),
		    PMFileDescriptor.fileFormat(M),
		    RelatedFileList (M)
		        LIST of STRUCT <
		            PMFileDescriptor.fileDirectory(M),
		            FileInfoList (M)
		               LIST OF STRUCT <
		                  PMFileDescriptor.filename(M)
		                  PMFileDescriptor.fileSize(M)
		                  PMFileDescriptor.fileCreationTime(M)
		                  PMFileDescriptor.fileDeletionTime(M)
		    >
		      >
		    >
		[Editor's note: This type should keep consistency with that of File Transfer IRP. When to define File Transfer IRP, the information defined here should be taken into consideration.]
		[Editor's note: The optionality qualifier will be discussed.]
			The output parameter specifies the required file information.
			Alternative of jobFileReference.
			JobFileInfoList is more efficient than jobFileReference.

jobFileReference	M	STRUCT <
		  PMFileDescriptor.fileUserName(M),
		  PMFileDescriptor.filePassword(M),
		  PMFileDescriptor,fileCompression(M),
		  PMFileDescriptor.fileFormat(M),
		  RelatedFileList(M)
		    LIST of STRUCT <
		        PMFileDescriptor.fileURL(M),
		        PMFileDescriptor.fileSize(M)
		        PMFileDescriptor.fileCreationTime(M)
		        PMFileDescriptor.fileDeletionTime(M)
		    >
		>
		
		[Editor's note: This type should keep consistency with that of File Transfer IRP. When to define File Transfer IRP, the information defined here should be taken into consideration.]
		[Editor's note: The optionality qualifier will be discussed.]
			Alternative of jobFileInfoList.

status	M	ENUM (Success, Failure)	An operation may fail because of a specified or unspecified reason.

7.3.6.4 Pre-condition

validTimes

Assertion Name	Definition
validTimes	"The beginTime is before current time. The endTime is after beginTime".

7.3.6.5 Post-condition

`measurementJobIsNotAffected` AND `requiredFileInfoIsReturned`.

Assertion Name	Definition
`measurementJobIsNotAffected.`	"The MeasurementJob identified by JobId is not affected by the operation".
`requiredFileInfoIsReturned`	"The required file information is returned".

7.3.6.6 Exceptions

Exception Name	Definition
`invalidTimes`	**Condition:** (validTimes) not verified. **Returned information:** output parameter status is set to Failure. **Exit state:** Entry State.

7.4 PMIRPOperations_2 Interface

7.4.1 Operation createThresholdMonitor (M)

7.4.1.1 Definition

IRPManager can request the NE/EM to create a ThresholdMonitor to define the threshold for some specific measurementTypes or subCounters. If the threshold defined is crossed, the related notification will be emitted to subscribed IRPManager(s).

Two cases are allowed. One case only accepts threshold monitoring of measurementType (s) that are already under monitoring by an existing MeasurementJob. This kind of PMIRP will not monitor a measurementType for threshold-crossing and clearing if that measurementType is not already subject to a MeasurementJob monitoring. This kind of PMIRP will only determine the threshold crossing and clearing events when the related MeasurementJob(s) are in Active states. The IRPManager, when interacting with this kind of PMIRP, must first start a MeasurementJob to monitor the measurementTypes and then invoke this operation for the same measurementTypes.

The other case is that it can accept threshold monitoring of measurementType (s) regardless if they are already under monitoring by existing MeasurementJob(s).

7.4.1.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
`moClass`	M	`ManagedEntity.objectClass`	It specifies the managed entity class name.
`moInstanceList`	M	List of `ManagedEntity.DistinguishedName`	It specifies the list of managed object instances whose measurementTypes or subCounters will be monitored. MoInstance can be absent. If so, the threshold of all known (to PMIRP) instances of the specified MO class will be monitored.

Parameter Name	Qualifier	Information type	Comment
thresholdInfoList	O	LIST of STRUCT < ThresholdMonitor. measurementTypeName, ThresholdMonitor.subCounterName, [Editor's note: "subCounterName" may be considered in measurement job.] Monitor.probableCause, Monitor.specificProblem, Monitor.direction, thresholdPack } where thresholdPack is list of thresholdPackElement. The thresholdPackElement is STRUCT < ThresholdLevel.thresholdValue, ThresholdLevel.severity, ThresholdLevel.hysteresis ThresholdLevel.notifyOnOff > [Editor's note] The notifyOnOff need to be discussed.	It specifies the thresholds to specific measurementTypes or subCounters. see subclause 6.5.1.
monitorGranularityPeriod	O	Monitor.monitorGranularityPeriod	See subclause 6.5.1

7.4.1.3   Output parameters

Parameter Name	Qualifier	Matching Information	Comment
monitorId	M	Identifier of the ThresholdMonitor.	It specifies the unique identifier of the ThresholdMonitor in the PM IRP Agent.
unsupportedList	M	List of STRUCT < ManagedEntity.objectClass, ManagedEntity.objectInstance, ThresholdMonitor. measurementTypeName, ThresholdMonitor.subCounterName, ThresholdLevel.reason >	To create a ThresholdMonitor, best-effort is required. This parameter identifies the unsupported but requested measurementType (s)/subCounter(s). The reason can be: (a) The PMIRP has trouble starting monitoring the threshold of this measurementType/subCounter. (b) The measurementType/subCounter is illegal. (c) The measurementType/subCounter exists but it is not currently under monitoring by any MeasurementJob and that the PMIRP requires that it be under monitoring by MeasurementJob (before it can be monitored for thresholding). (d) Hysteresis is overlapped. This parameter is used only when the operation returns 'PartialSuccess'.
status	M	ENUM (Success, Failure, PartialSuccess)	An operation may fail because of a specified or unspecified reason.

7.4.1.4   Pre-condition

validGranularityPeriod AND someValidMeasurementType AND validDirection AND
validNumberOfThresholdPackElements AND validOrderOfThresholdPackElements.

Assertion Name	Definition
validGranularityPeriod	The monitorGranularityPeriod is valid.
someValidMeasurementType	At least one requested measurementType/subCounter can be monitored for thresholding.
validDirection	The direction is valid.
validNumberOfThresholdPack Elements	If PMIRP supports multi-level severity, it is valid if the number of elements in thresholdPack is 1, 2 or 3. If PMIRP does not support multi-level severity, it is valid if the number is 1.
validOrderOfThresholdPack Elements	This assertion is applicable if PMIRP supports multi-level severity. It is valid if it satisfies the constraint defined by multiLevelSeverityOrder of subclause 6.5.2.

7.4.1.5 Post-condition

thresholdMonitorIsCreated.

Assertion Name	Definition
thresholdMonitorIsCreated	"The ThresholdMonitor identified by monitorId exists. And the performance alarm way be emitted if the thresholds are crossed. The ThresholdMonitor.thresholdMonitorStatus will be Active.

7.4.1.6 Exceptions

Exception Name	Definition
invalidClassOrInstances	**Condition:** (validClassAndInstances) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.
invalidGranularityPeriod	**Condition:** (validGranularityPeriod) not verified. **Returned Information:** Name of the exception; status is set to 'Failure'. **Exit state:** Entry state.
noValidMeasurementType	**Condition:** (someValidMeasurementType) not verified. **Returned information:** output parameter status is set to 'failure'. **Exit state:** Entry State.
invalidMultiLevelSeverityOrder	**Condition:** (validMultiLevelSeverityOrder) not verified. **Returned information:** output parameter status is set to 'failure'. **Exit state:** Entry State.
invalidMultiLevelSeverityHysteresis	**Condition:** (validMultiLevelSeverityHysteresis) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.
invalidDirection	**Condition:** (validDirection) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.

7.4.2 Operation deleteThresholdMonitor (M)

7.4.2.1 Definition

IRPManager can request NE/EM to delete a specified ThresholdMonitor through Itf-N.

At the time of the removal, all outstanding threshold-crossing alarms will stay (i.e. the FM IRPAgent's AlarmList will contain an AlarmInformation indicating threshold-crossing). The IRPManager needs to use other means to remove the AlarmInformation in the FMIRP AlarmList.

7.4.2.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
monitorId	M	ThresholdMonitor.monitorId	It specifies the ThresholdMonitor to be stopped.

7.4.2.3   Output parameters

Parameter Name	Qualifier	Matching Information	Comment
status	M	ENUM (Success, Failure)	An operation may fail because of a specified or unspecified reason.

7.4.2.4   Pre-condition

thresholdMonitorExists.

Assertion Name	Definition
thresholdMonitorExists	"ThresholdMonitor specified exists."

7.4.2.5   Post-condition

thresholdMonitorIsDeleted.

Assertion Name	Definition
threhsoldMonitorDeleted	"the thresholdMonitor identified by monitorId doesn't exist."

7.4.2.6   Exceptions

Exception Name	Definition
unknownMonitor	**Condition:** (thresholdMonitorExists) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.

7.4.3   Operation listThresholdMonitors (M)

7.4.3.1   Definition

This operation allows NM to list detailed information about all or specified ThresholdMonitors.

7.4.3.2   Input parameters

Parameter Name	Qualifier	Information type	Comment
retrieveInfo	M	LIST of ThresholdMonitor.monitorId	This parameter specifies the criteria to list the ThresholdMonitors. If the parameter specifies the list of monitorIds to be retrieved, then the corresponding information of ThresholdMonitors will be returned. If the parameter contains no information, all the ThresholdMonitors in the PMIRP Agent are retrieved.

7.4.3.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment

monitorInfoList	M	LIST OF STRUCT <
		
		    ManagedEntity.objectClass,
		    LIST of
		ManagedEntity.DistinguishedName,
		    attributes of ThresholdMonitor (see subclause 6.5.2),
		    thresholdInfoList (See subclause 7.5.1.2)
		>
			Returned information of corresponding ThresholdMonitors. If no ThresholdMonitor satisfied the criteria, then the length of the list will be 0.

status	M	ENUM (Success, Failure)	An operation may fail because of a specified or unspecified reason.

7.4.3.4 Pre-condition

retrieveInfoIsValid.

Assertion Name	Definition
retrieveInfoIsValid	"retrieveInfo specified in the input parameter is valid."

7.4.3.5 Post-condition

thresholdMonitorsAreReturned AND thresholdsAreUnaffected.

Assertion Name	Definition
thresholdMonitorsAreReturned	"the information of required ThresholdMonitors are returned in the output parameters. "
thresholdsAreUnaffected	"The ThresholdMonitors are not affected."

7.4.3.6 Exceptions

Exception Name	Definition
errorCriteria	**Condition:** (retrieveInfoIsValid) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.

7.5 PMIRPOperations_3 Interface

7.5.1 Operation suspendThresholdMonitor (O)

[Editor's note: There may be problem supporting this operation in multiple IRPManager environment since the suspension, requested by one IRPManager, will not be known by other IRPManagers. This note also applies to similar operations for measurementJobs.]

7.5.1.1 Definition

A ThresholdMonitor can be suspended, and then the NE/EM will not emit related performance notification when the thresholds are crossed. But the ThresholdMonitor still exists, and it will work again when it is resumed.

7.5.1.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
thresholdMonitorId	M	ThresholdMonitor.monitorId	It specifies the ThresholdMonitor to be suspended.

7.5.1.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
status	M	ENUM (Success, Failure)	An operation may fail because of a specified or unspecified reason.

7.5.1.4 Pre-condition

thresholdMonitorExists AND thresholdMonitorIsNotSuspended.

Assertion Name	Definition
thresholdMonitorExists	"ThresholdMonitor specified in the input parameter exists."
thresholdMonitorIsNotSuspended	"ThresholdMonitor specified in the input parameter is not suspended."

7.5.1.5 Post-condition

thresholdMonitorIsSuspended.

Assertion Name	Definition
thresholdMonitorIsSuspended	"the ThresholdMonitor identified by monitorId is suspended, that means it will not work until it is resumed. The ThresholdMonitor.thresholdMonitorStatus will be 'suspended'.

7.5.1.6 Exceptions

Exception Name	Definition
unknownThresholdMonitor	**Condition:** (thresholdMonitorExists) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.
ThresholdMonitorHasBeen Suspended	**Condition:** (thresholdMonitorIsNotSuspended) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.

7.6 PMIRPOperations_4 Interface

7.6.1 Operation resumeThresholdMonitor (O)

[Editor's note: See editor's note on suspendThresholdMonitor.]

7.6.1.1 Definition

A suspended ThresholdMonitor can be resumed by IRPManager. When the ThresholdMonitor is resumed, NE/EM may emit performance notifications when the thresholds are crossed.

7.6.1.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
thresholdMonitorId	M	ThresholdMonitor.monitorId	It specifies the ThresholdMonitor to be resumed.

7.6.1.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
status	M	ENUM (Success, Failure)	An operation may fail because of a specified or unspecified reason.

7.6.1.4 Pre-condition

thresholdMonitorExists AND thresholdMonitorIsSuspended.

Assertion Name	Definition
thresholdMonitorExists	"ThresholdMonitor specified in the input parameter exists."
thresholdMonitorIsSuspended	"ThresholdMonitor specified in the input parameter is suspended."

7.6.1.5 Post-condition

thresholdMonitorIsResumed.

Assertion Name	Definition
thresholdMonitorIs Resumed.	"the ThresholdMonitor identified by monitorId is resumed, that means it will work again. The ThresholdMonitor.ThresholdMonitor.thresholdMonitorStatus will be Active."

7.6.1.6 Exceptions

Exception Name	Definition
unknownThresholdMonitor	**Condition:** (thresholdMonitorExists) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.
thresholdMonitorIsNotSuspended	**Condition:** (thresholdMonitorIsSuspended) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.

7.7 PMIRPOperations_5 Interface

7.7.1 Operation modifyThresholdMonitor (O)

[Editor's note: Since the modification of the threshold data by one IRPManager is not known by other IRPManager, there may be some problem in a multi IRPManager environment (similar problem with suspend/resume operations.)]

7.7.1.1 Definition

The parameters of ThresholdMonitor, such as 'thresholdInfoCounterList', 'thresholdInfoGaugeList' and 'monitorGranularityPeriod', can be modified by IRPManager. The ThresholdMonitor should be suspended before it is modified. Any change on a threshold value will only be taken into account for the next granularity period, see 3GPP TS 32.401 [7] (requirement of threshold management).

7.7.1.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
thresholdMonitorId	M	ThresholdMonitor.monitorId	It specifies the ThresholdMonitor to be modified.
moClass	O	Name of ManagedEntity	It specifies the Managed Object Class to which the threshold will be applied.
moInstanceList	O	LIST of DistinguishedName	It specifies the Managed Object Instances to which the threshold will be applied.
thresholdInfoList	O	See the same parameter defined in subclause 7.5.1.2.	If the thresholdInfoList is to be modified, the parameter carries the new value of thresholdInfoList.
monitorGranularityPeriod	O	ThresholdMonitor.monitorGranularityPeriod	If the monitorGranularityPeriod is to be modified, the parameter carries the new value of monitorGranularityPeriod.

7.7.1.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
status	M	ENUM (Success, Failure)	An operation may fail because of a specified or unspecified reason.

7.7.1.4 Pre-condition

thresholdMonitorExists AND thresholdMonitorIsSuspended.

Assertion Name	Definition
thresholdMonitorExists	"ThresholdMonitor specified in the input parameter exists."
thresholdMonitorIsSuspended	"ThresholdMonitor specified in the input parameter is suspended."

7.7.1.5 Post-condition

thresholdMonitorIsModified

Assertion Name	Definition
thresholdMonitorIsModified	"the ThresholdMonitor identified by monitorId is modified.

7.7.1.6 Exceptions

Exception Name	Definition
unknownThresholdMonitor	**Condition:** (thresholdMonitorExists) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.
thresholdMonitorIsNotSuspended	**Condition:** (thresholdMonitorIsSuspended) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.

7.8 PmIRPNotification_1 Interface

7.8.1 notifyPMFileReady (M)

7.8.1.1 Definition

After MeasurementJob have captured the scanned measurement data in file(s) according to the MeasurementJob.jobReportingPeriod, MeasurementJob will emit notification to subscribed IRPManager(s) to notify the availability of the file(s).

7.8.1.2   Input Parameters

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M, F	--	This parameter and objectInstance together carry the same semantics of systemDN.
objectInstance	M, F	--	This parameter and objectClass together carry the same semantics of systemDN.
notificationId	M		
eventTime	M, F	--	
notificationType	M, F	"notifyPMFileReady".	
systemDN	C, F	IRPAgent.systemDN.	It carries the DN of the IRPAgent that emits this notification.
jobId	M	MeasurementJob.jobId	
jobFileInfoList	M	See subclause 7.3.6.3.	The parameter specifies the information of the available file.
jobFileReference	M	See subclause 7.3.6.3	The parameter specifies the information of the available file
additionalText	O	String	It carries vendor-specific semantics not defined in the present document.

7.8.1.3   Triggering Event

7.8.1.3.1   From-state

Assertion Name	Definition
filesAreReadyAndClosed	EM/NE have scanned the measurement data according to the reporting period defined in MeasurementJob and have stored the data into one or more files.

7.8.1.3.2   To-state

Assertion Name	Definition
newNotification Reported	The 'notifyPmFileReady' notification is emitted to the subscribed IRPManager(s). Based on the information contained in the notification, the IRPManager can get the file(s) through file transfer mechanism which is defined in File Transfer IRP [10].

7.8.2   notifyPMFilePreparationError (M)

7.8.2.1   Definition

The subscribed IRPManagers are notified regarding the occurrence of an error during the preparation of the file. This notification is an event and will not be treated as alarms defined in Alarm IRP (3GPP TS 32.111-2 [4]).

7.8.2.2   Input parameters

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M, F	--	See objectClass of subclause 7.8.1.2
objectInstance	M, F	--	See objectInstance of subclause 7.8.1.2.
notificationId	M	--	See notificationId of subclause 7.8.1.2.
eventTime	M, F	--	It registers the time when file preparation failure is detected.
systemDN	C, F	IRPAgent.systemDN	
notificationType	M, F	"notifyPMFilePreparationError"	
jobId	M	MeasurementJob.jobId	
additionalText	M		It carries vendor-specific semantics not defined in the present document.

7.8.2.3 Triggering Event

7.8.2.3.1 From-state

errorInPreparation OR hardDiskFull OR hardDiskFailure OR tooManyFiles OR collectionTimeOut OR incompleteTruncatedFile OR corruptedFile OR lowMemory OR dataNotAvailable.

Assertion Name	Definition
errorInPreparation	There is unspecified error when EM/NE prepares the PM files.
hardDiskFull	The file system is full and no more files can be opened.
hardDiskFailure	The hard disk has failed and normal input/output cannot be performed.
tooManyFiles	The file system lacks resource to open a new file to capture measurement data.
collectionTimeOut	The reading of monitored measurementTypes takes too long.
incompleteTruncatedFile	The file is truncated for unspecified reason. The suspect flag (3GPP TS 32.401 [7] annex A) should be set.
corruptedFile	The file is corrupted for unspecified reason. The suspect flag (3GPP TS 32.401 [7] annex A) should be set.
lowMemory	The system lacks sufficient memory to open a new file to capture measurement data.
dataNotAvailable	Measurement data is not available when the monitored measurement Type (s) is being read.

7.8.2.3.2 To-state

fileIsKept.

Assertion Name	Definition
fileIsKept	The file, whose preparation provokes an error, is kept (if possible) for a period after which the file may be removed. The period is vendor specific.

7.8.3 notifyMeasurementJobStopped (M)

7.8.3.1 Definition

The PM IRPAgent notifies all subscribed IRPManagers about the stopping of a MeasurementJob.

7.8.3.2 Input Parameters

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M, F	--	This parameter and objectInstance together carry the same semantics of systemDN.
objectInstance	M, F	--	This parameter and objectClass together carry the same semantics of systemDN.
notificationId	M		
eventTime	M, F	--	
notificationType	M, F	"notifyMeasurementJobStopped".	
systemDN	C, F	IRPAgent.systemDN.	It carries the DN of the IRPAgent that emits this notification.
jobId	M	MeasurementJob.jobId	
reason	O	String	It carries one of the assertion names of the From-state of Triggering Event.

7.8.3.3 Triggering Event

7.8.3.3.1 From-state

Assertion Name	Definition
`failToRead measurementType sForExtendedProlon gPeriod`	Because the PM IRPAgent have failed to read the monitored measurementType (s) from managed resources for one or more times, the PM IRPAgent decides that it will not try to read in the future and place the MeasurementJob in "Stopped" state.
`internalProblem`	Because of an unspecified internal problem, PM IRPAgent decides that it no longer can maintain the MeasurementJob in any jobStatus but "Stopped".
`stopMeasurementJob`	The stopMeasurementJob returns success.
`stopTimeExpired`	The current time is equal to MeasurementJob.jobStopTime.

7.8.3.3.2 To-state

Assertion Name	Definition
`measurementJobStopped`	The MeasurementJob.jobStatus == "Stopped".

7.9 PmIRPNotifications_2 Interface

7.9.1 notifyNewAlarm (M)

7.9.1.1 Definition

Refer to the notifyNewAlarm notification of Alarm IRP: IS (3GPP TS 32.111-2 [4]).

The PM IRPAgent emits this notification to the Alarm IRPAgent and Notification IRPAgent for processing. Then, the notification is observable across the Itf-N.

7.9.1.2 Input parameters

See the corresponding table in Alarm IRP: IS (3GPP TS 32.111-2 [4]).

Note that the parameters objectClass and objectInstance shall relate to the ThresholdMonitor instance.

> NOTE: A new parameter threshold is used instead of using the thresholdInfo parameter defined in Alarm IRP: IS. The parameter threshold contains ThresholdMonitor. measurementTypeName, ThresholdMonitor.subCounterName, Monitor.direction, ThresholdLevel.thresholdValue, ThresholdLevel.hysteresis and the observed value.

7.9.1.3 Triggering Event

Refer to annex B.

7.9.2 notifyChangedAlarm (C)

7.9.2.1 Definition

The PM IRPAgent emits this notification to the Alarm IRPAgent and Notification IRPAgent for processing. Then, the notification is observable across the Itf-N.

7.9.2.2 Input parameters

See the corresponding table in Alarm IRP: IS (3GPP TS 32.111-2 [4]).

Note that the parameters objectClass and objectInstance shall relate to the ThresholdMonitor instance.

NOTE: A new parameter threshold is used. See subclause 7.9.1 for its definition and usage.

7.9.2.3 Triggering Event

Refer to annex B.

7.9.3 notifyClearedAlarm (M)

7.9.3.1 Definition

Refer to the specification of notifyClearedAlarm Notification of Alarm IRP: IS (3GPP TS 32.111-2 [4]).

The PM IRPAgent emits this notification to the Alarm IRPAgent and Notification IRPAgent for processing. Then, the notification is observable across the Itf-N.

7.9.3.2 Input parameters

Refer to the specification of notifyClearedAlarm Notification of Alarm IRP: IS (3GPP TS 32.111-2 [4]).

Note that the parameters objectClass and objectInstance shall relate to the ThresholdMonitor instance.

NOTE: A new parameter threshold is used. See subclause 7.9.1 for its definition and usage.

7.9.3.3 Triggering event

Refer to annex B.

7.9.4 notifyMonitorCreation (M)

7.9.4.1 Definition

Refer to the specification of notifyObjectCreation Notification of Kernel CM IRP: IS (3GPP TS 32.662 [13]).

The PM IRPAgent emits this notification to the Notification IRPAgent for processing. Then, the notification is observable across the Itf-N.

7.9.4.2 Input parameters

Refer to the specification of notifyObjectCreation Notification of Kernel CM IRP: IS (3GPP TS 32.662 [13]).

The parameters objectClass and objectInstance shall relate to the ThresholdMonitor instance.

7.9.4.3 Triggering event

The ThresholdMonitor is created.

7.9.5 notifyMonitorDeletion (M)

7.9.5.1 Definition

Refer to the specification of notifyObjectDeletion Notification of Kernel CM IRP: IS (3GPP TS 32.662 [13]).

The PM IRPAgent emits this notification to the Notification IRPAgent for processing. Then, the notification is observable across the Itf-N.

7.9.5.2 Input parameters

Refer to the specification of notifyObjectDeletion Notification of Kernel CM IRP: IS (3GPP TS 32.662 [13]).

The parameters objectClass and objectInstance shall relate to the ThresholdMonitor instance.

7.9.5.3 Triggering event

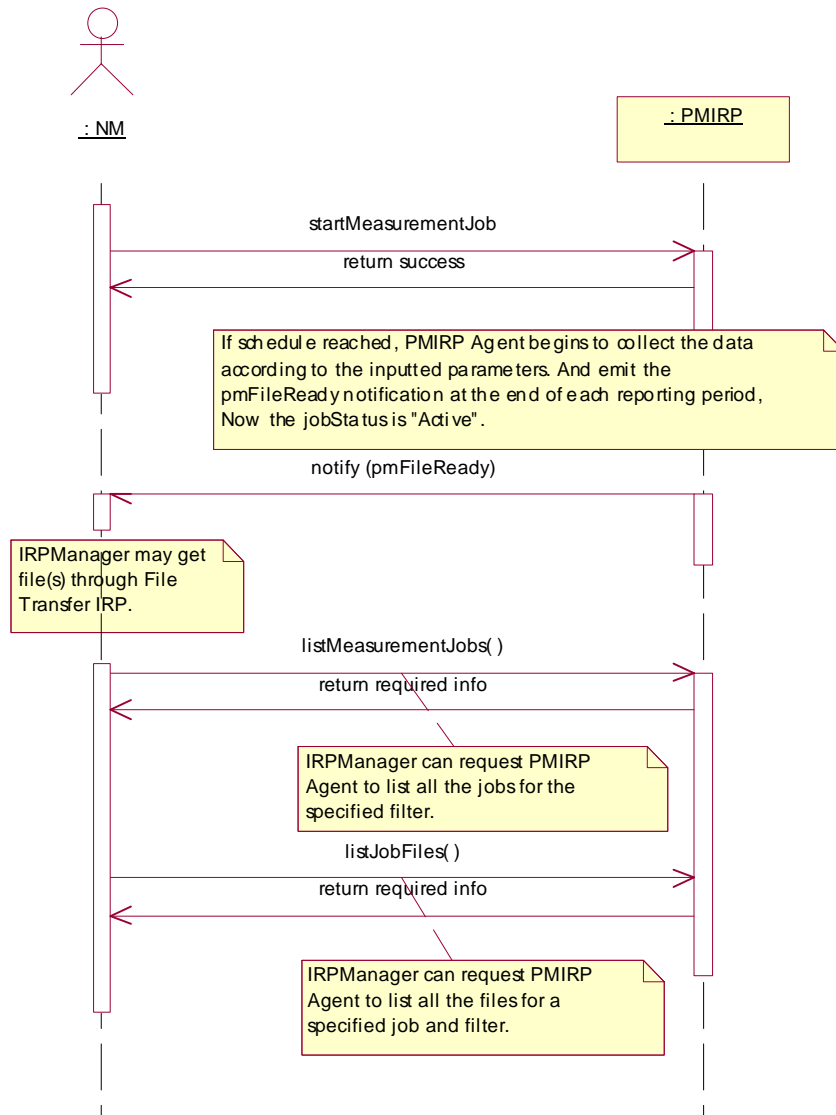The ThresholdMonitor is created.

8 Scenarios

8.1 createMeasurementJob

Figure 8.1: Scenario1 of Performance Management
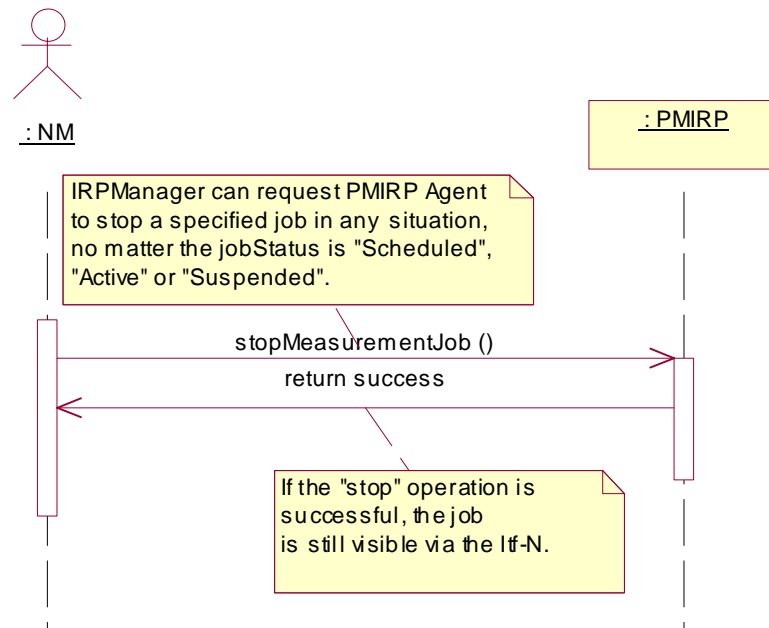
8.2   stopMeasurementJob

Figure 8.2: Scenario2 of Performance Management

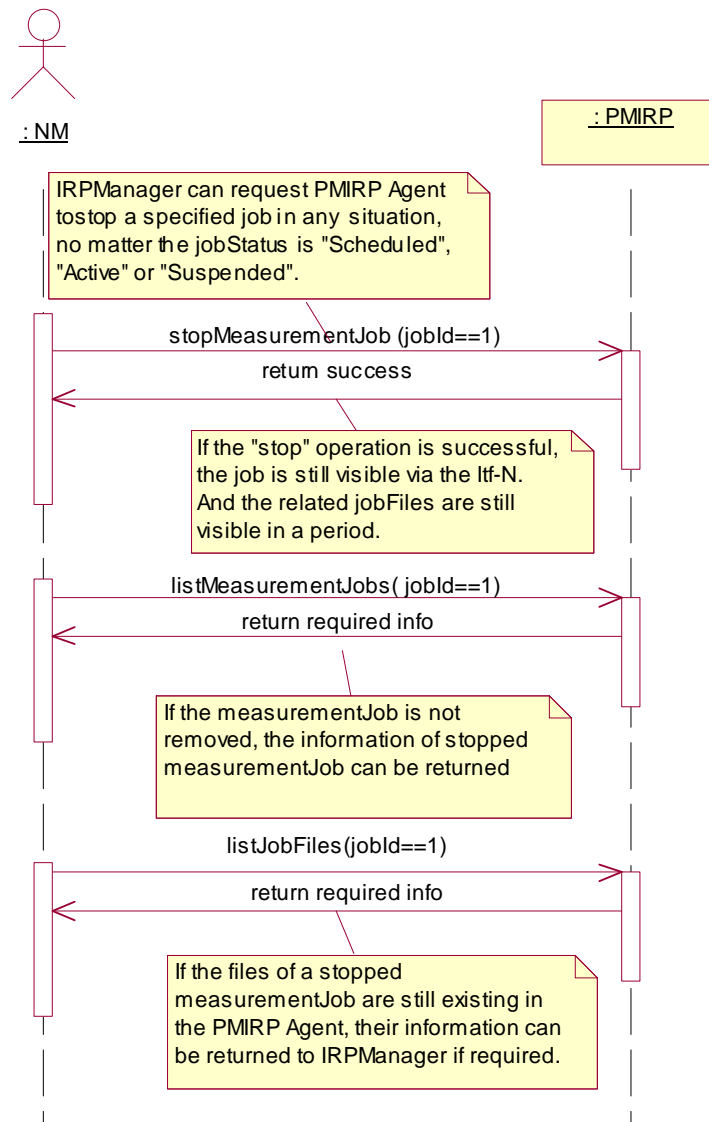## 8.3 stopMeasurementJob/listMeasurementJobs/listJobFiles

Figure 8.3: Scenario3 of Performance Management
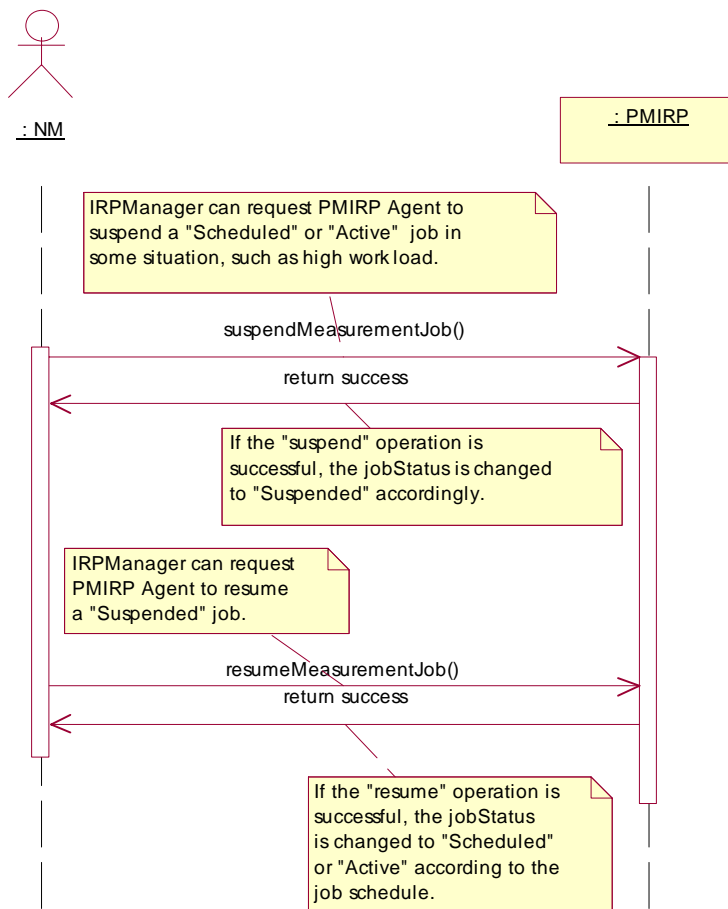
8.4 suspendMeasurementJob/resumeMeasurementJob

Figure 8.4: Scenario4 of Performance Management

Annex A (normative):
Illustration of the state described in the state Diagram

A.1 Definition of state

A.1.1 Scheduled

A.1.1.1 Definition

"Scheduled" means that the job has been created and the start time has not reached.

A.1.1.2 From state

Two cases for a MeasurementJob becomes Scheduled are as follows:

- When IRPManager invokes createMeasurementJob operation successfully, a MeasurementJob will be created and the state of the job is "Scheduled". If the starTime is "start now", then the MeasurementJob.jobStatus goes from "Scheduled" to "Active" immediately without further triggers.

- When IRPManager invokes resumeMeasurementJob operation on a suspended job successfully and the start time of the job has not reached, the state of the job will be "Scheduled".

A.1.1.3 To state

Two transition for a MeasurementJob go from Scheduled are as follows:

- IRPManager can suspend the scheduled MeasurementJob for some reason.

- IRPManager can stop the job positively, and the job will be stopped.

A.1.2 Suspended

A.1.2.1 Definition

"Suspended" means that the job has been Suspended for some reason.

A.1.2.2 From state

The case for a MeasurementJob becomes Suspended is as follows.

- IRPManager invokes suspendMeasurementJob operation on a scheduled or active job successfully, the state of the job will be "Suspended".

A.1.2.3 To state

Three transition for a MeasurementJob go from Suspended are as follows:

- IRPManager can resume the suspended MeasurementJob. When the job is resumed, the measurementJob.jobStatus will be placed in such state, according to the initial startTime/stopTime/schedule, as if it is never been suspended.

- When stop time reaches, the MeasurementJob will be stopped.

- IRPManager stops the job positively, the MeasurementJob will be stopped.

A.1.3 Active

A.1.3.1 Definition

"Active" means that the start time of a MeasurementJob has reached. There are two sub states: "Busy" and "Idle".

"Busy" means that the MeasurementJob is scanning the measurementTypes. When a MeasurementJob becomes active, it does not mean that the job is always busy. IRPManager can set the detailed time frame (e.g. dailySchedule or weeklySchedule) for a MeasurementJob to be busy. If there is no particular time frame schedule for a job, it works when it is active.

"Idle" means that the MeasurementJob is not scanning the measurementTypes.

A.1.3.2 From state

Two cases for a MeasurementJob becomes Active are as follows.

- When start time of a scheduled MeasurementJob has reached, the state of the job becomes "Active".
- IRPManager invokes resumeMeasurementJob operation on a suspended job successfully and the start time of the job has reached, the state of the job will be "Active".

A.1.3.3 To state

Two transition for a MeasurementJob go from Active are as follows:

- IRPManager can suspend the active MeasurementJob for some reason.
- When stop time reaches, the job will be stopped.
- IRPManager stops the job positively, the job will be stopped.

A.1.4 Stopped

A.1.4.1 Definition

"Stopped" means that the MeasurementJob is deleted via the Itf-N. But Whether the managed system thoroughly removes the MeasurementJob immediately or later is vendor specific.

A.1.4.2 From state

Two cases for a MeasurementJob becomes Stopped are as follows.

- When stop time of a suspended/active MeasurementJob has reached, the state of the job becomes "Stopped".
- IRPManager invokes stopMeasurementJob operation on a scheduled/suspended/active job successfully, the state of the job will be "Stopped".
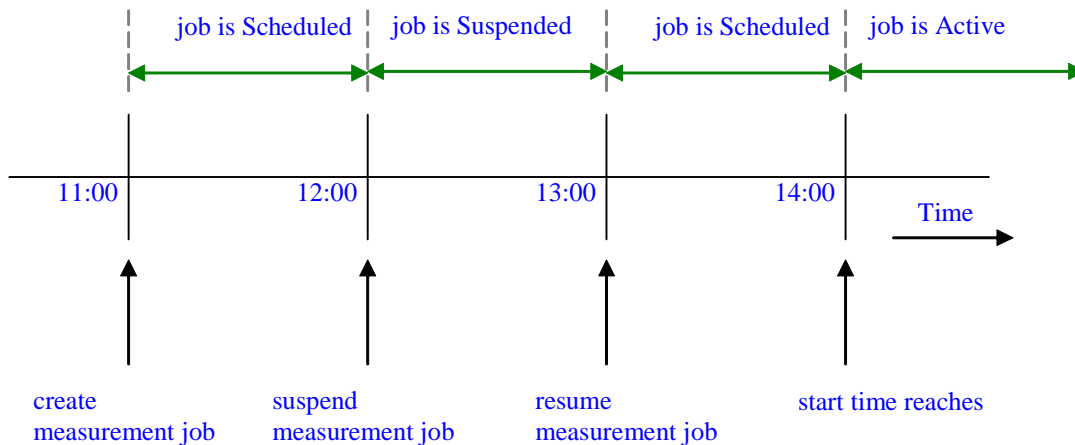
A.1.4.3 To state

The MeasurementJob is thoroughly removed from the managed system.

A.2 State transition scenarios

The following Scenarios give examples of states.

A.2.1 Scenario 1

A MeasurementJob is created at 11:00, and the startTime is 14:00, stopTime is 18:00. At 12:00, the MeasurementJob is suspended, and at 13:00, the MeasurementJob is resumed.

At 11:00: the MeasurementJob becomes scheduled after it was created.
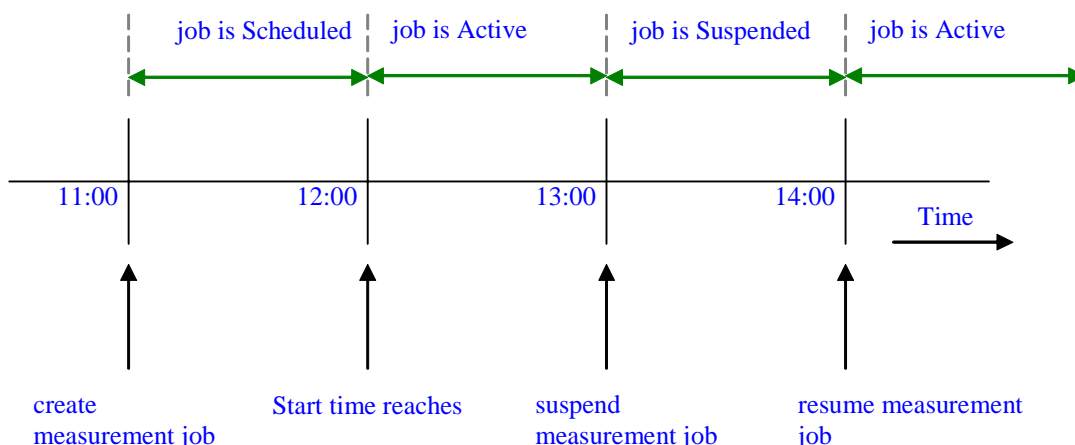At 12:00: the MeasurementJob is suspended and the jobStatus is "Suspended".
At 13:00: the MeasurementJob is resumed, and the jobStatus is "Scheduled" because the start time doesn't reach.
At 14:00: the MeasurementJob becomes active because start time reaches.

Figure A.1: Scenario 1

A.2.2 Scenario 2

A MeasurementJob is created at 11:00, and the startTime is 12:00, stopTime is 18:00. At 13:00, the MeasurementJob is suspended, and at 14:00, the MeasurementJob is resumed.

At 11:00: the MeasurementJob becomes scheduled after it was created.
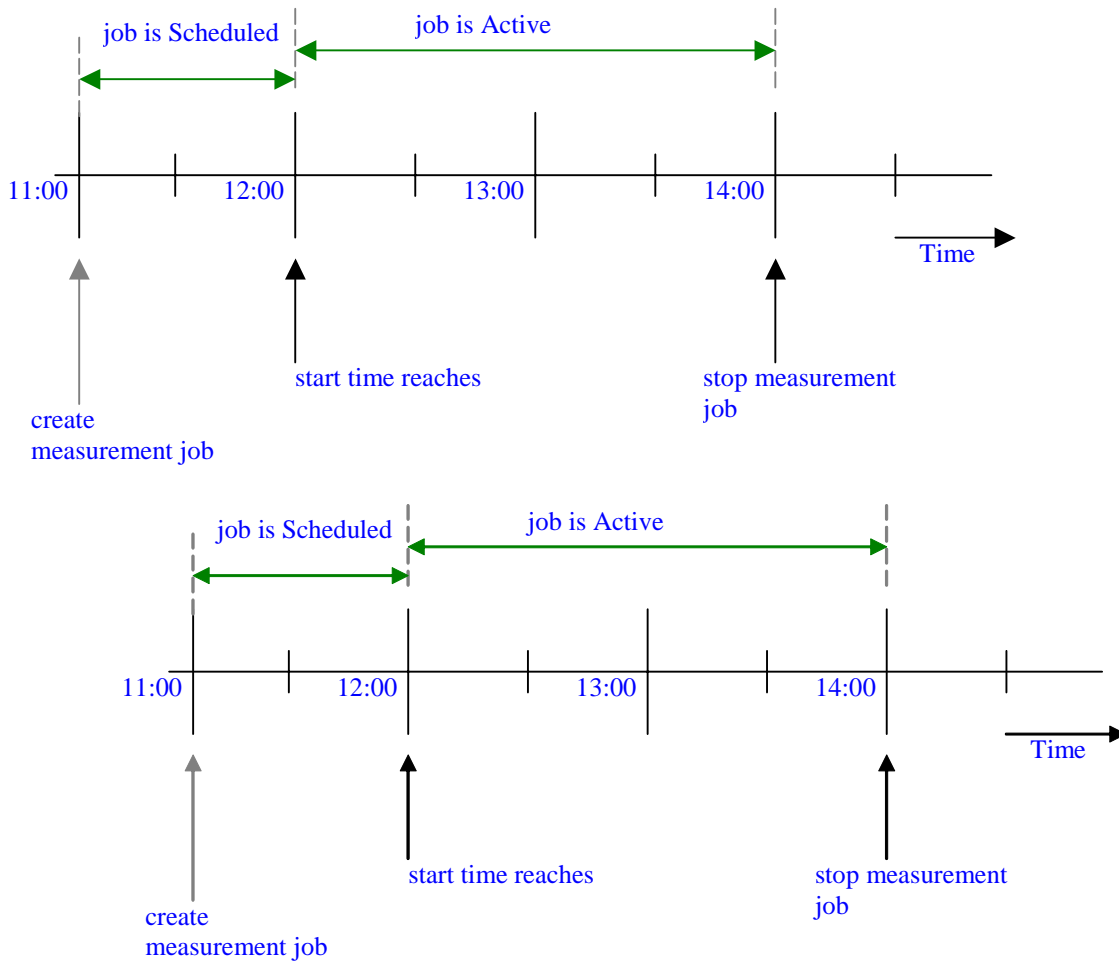At 12:00: the start time reaches, and the jobStatus is "Active".
At 13:00: the MeasurementJob is suspended, and the jobStatus is "Suspended".
At 14:00: the MeasurementJob is resumed, and the jobStatus is "Active".

Figure A.2: Scenario 2

A.2.3 Scenario 3

A MeasurementJob is created at 11:00 and the startTime is 12:00, stopTime is never stop. There is no other schedule defined. IRPManager stops the job positively at 14:00.
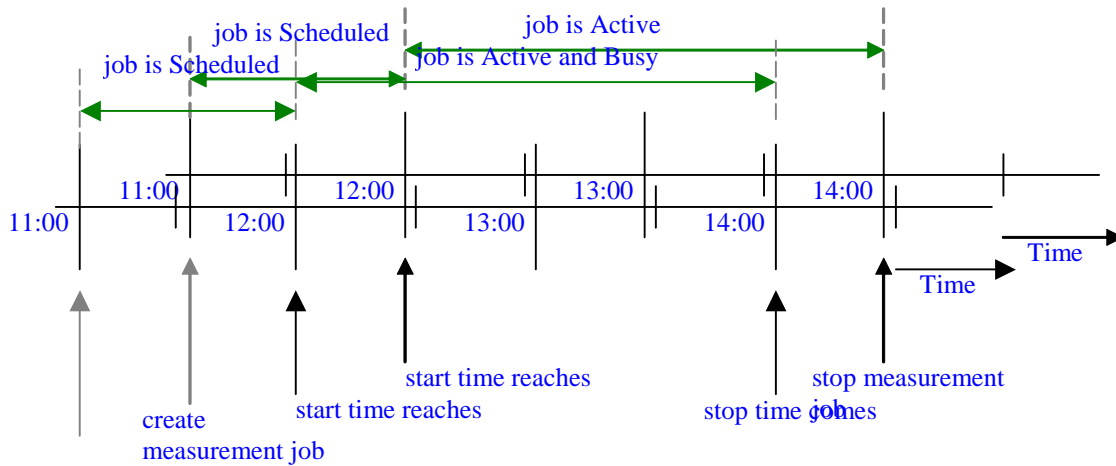

At 12:00: the MeasurementJob becomes active when start time reaches. Since there is no detailed time frame for the job, the job begins to work at the same time.

At 14:00: the MeasurementJob is stopped when IRPManager stops the job positively.

Figure A.3: Scenario 3

A.2.4 Scenario 4

A MeasurementJob is created at 11:00 and the startTime is 12:00, stopTime is 14:00. There is no other schedule defined.
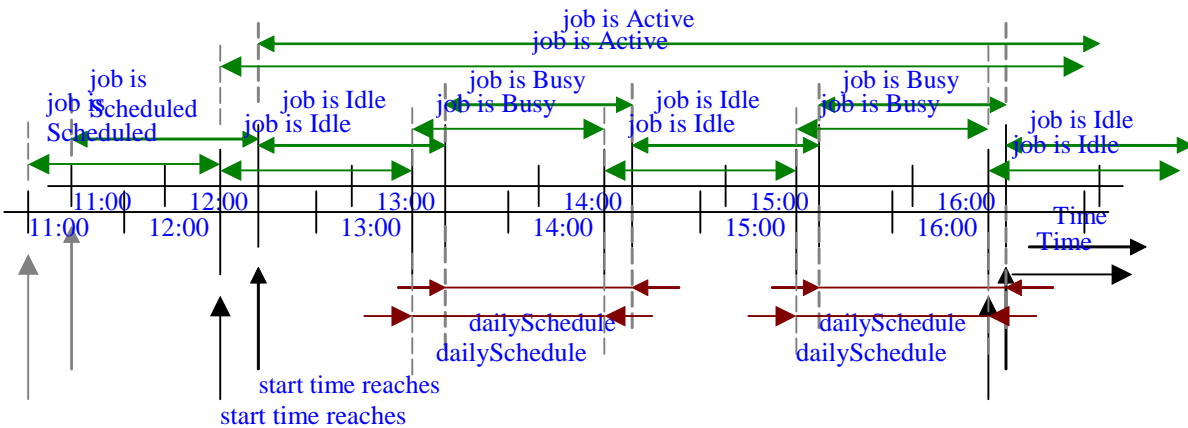
At 12:00: the MeasurementJob becomes active when start time reaches. Since there is no detailed time frame for the job, the job begins to work at the same time.

At 14:00: the MeasurementJob is stopped automatically when stop time comes.

Figure A.4: Scenario 4

A.2.5 Scenario 5

A MeasurementJob is created at 11:00 and the startTime is 12:00, stopTime is never stop. There is a dailySchedule defined, which is 13:00~14:00 and 15:00~16:00 every day.
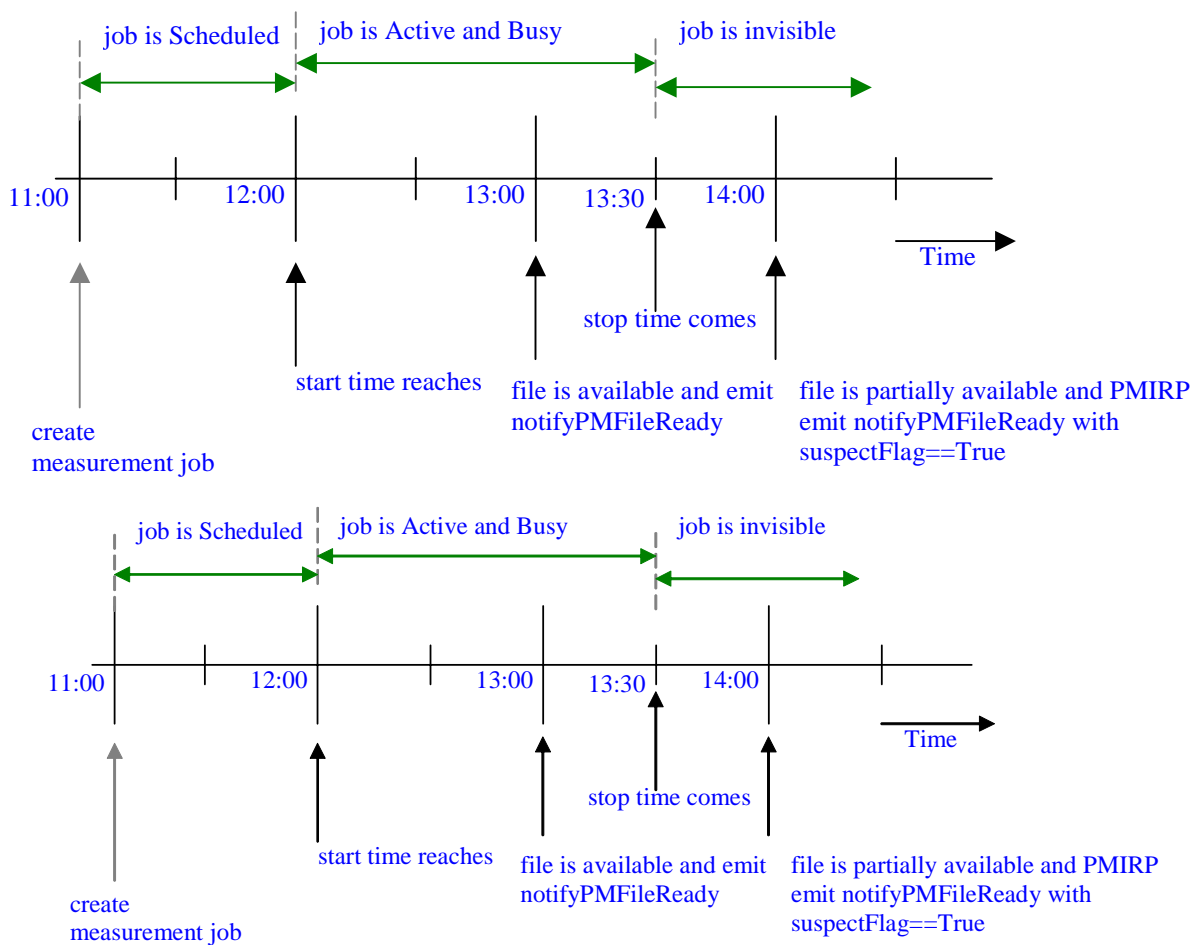

At 12:00: the MeasurementJob becomes active when start time reaches. Since there is detailed time frame schedule for the job, the job works only during the specified time frame, during other time, the job is idle.

At 13:00: the MeasurementJob becomes busy when the daily schedule is reaches.
and 15:00

Figure A.5: Scenario 5

A.2.6 Scenario 6

A MeasurementJob is created at 11:00 and the startTime is 12:00, stopTime is 13:30. The MeasurementJob.granularityPeriod is 30 minutes and the MeasurementJob.reportingPeriod is 1 hour. There is no other schedule defined.

At 12:00: the MeasurementJob becomes active when start time reaches. Since there is no detailed time frame for the job, the job begins to work at the same time.

At 13:00: the file(s) of the MeasurementJob are available, and PMIRP emits the notifyPMFileReady notification to subscripted IRPManager.

At 13:30: the MeasurementJob is automatically stopped when stop time comes. The corresponding performance file(s) are partially finished.

At 14:00: PMIRP will emit the last notifyPMFileReady notification with suspectFlag == True.

Figure A.6: Scenario 6

Annex B (normative):
Threshold Notification Triggering Events

This annex defines the Triggering Events for the various notifications. The Triggering Events normally are defined by each notification type, such as notifyChangedAlarm, under the subclause "Triggering Events". The Triggering Events related to thresholding are defined here collectively for ease of reference.

The Triggering Events are defined in a state transition table. The column labels (e.g. 0, 1, 2 of table in clause B.1) denote the to-state while the row labels denote the from-state. The cell denotes the emission action.

There are two clauses. Clause B.1 defines the Triggering Events for PMIRP that supports notifyChangedAlarm. Clause B.2 defines those for PMIRP that does not support notifyChangedAlarm.Change history.

Clause B.3 provides samples of triggering events.

B.1 For PMIRP supporting notifyChangedAlarm

Table B.1

	To state 0	**To state 1**	**To state 2**	**To state 3**
From State 0	no emission	1 notifyNewAlarm	1 notifyNewAlarm	1 notifyNewAlarm
From State 1	1 notifyClearedAlarm	no emission	1 notifyChangedAlarm	1 notifyChangedAlarm
From State 2	1 notifyClearedAlarm	1 notifyChangedAlarm	no emission	1 notifyChangedAlarm
From State 3	1 notifyClearedAlarm	1 notifyChangedAlarm	1 notifyChangedAlarm	no emission

The horizontal '0', '1', '2' and '3' are the to-states indicating the current measurementType value with respect to the different threshold levels. Threshold levels are associated with severity level. Higher threshold level associates with higher severity level.

> EXAMPLE: If direction = 'Increasing', state-0 means the measurementType value is below all thresholds. State-2 means that the measurementType value is above threshold level 2 but below threshold level 3. If direction = 'Decreasing', state-0 means the measurementType value is above all thresholds. State-2 means that the measurementType value is below threshold level 2 but above threshold level 3.

The vertical '0', '1', '2' and '3' are the from-states indicating the last-read measurementType value with respect to the threshold levels.

A threshold has a value. It can has a hysteresis. A threshold with a hysteresis has a threshold-high and a threshold-low values that are different than the threshold value. A threshold without a hysteresis can be considered as a threshold whose threshold-high and threshold-low values are equal to the threshold value.

For the direction = 'Increasing', the shaded cells indicate transitions caused by measurementType values rising across one or more threshold-high values. The non-shaded cells indicate transition caused by measurementType values falling across one or more threshold-low values.

For the direction = 'Decreasing', the shaded cells indicate transitions caused by measurementType values falling across one or more threshold-low values. The non-shaded cells indicate transition caused by measurementType values rising across one or more threshold-high values.

Each cell indicates the notification emission when from-state transits to to-state. The notifyNewAlarm and notifyChangedAlarm shall carry the severity level associated with the to-state.

The Monitor determines, at GP=X, if a threshold has been crossed by comparing the value read at GP==X with the value read at GP==X-1. In initial condition (i.e. when the Monitor reads the value at GP=1), the (hypothetical) value read at GP==X-1 is assumed to be at state-0.

B.2 For PMIRP Not Supporting notifyChangedAlarm

Table B.2

	To State 0	**To State 1**	**To State 2**	**To State 3**
From State 0	no emission	1 notifyNewAlarm	2 notifyNewAlarm(s)	3 notifyNewAlarm(s)
From State 1	1 notifyClearedAlarm	no emission	1 notifyNewAlarm	2 notifyNewAlarm(s)
From State 2	2 notifyClearedAlarm(s)	1 notifyClearedAlarm	no emission	1 notifyNewAlarm
From State 3	3 notifyClearedAlarm(s)	2 notifyClearedAlarm(s)	1 notifyClearedAlarm	no emission

If multiple notifyNewAlarm(s) are emitted, the different notifications shall carry different severity levels associated with the crossed threshold levels. In cell to-state-2/from-state-0, two notifyNewAlarm(s) are emitted. One carry severity level associated with threshold level 1. The other carry security level associated with threshold level 2.

The Monitor determines, at GP=X, if a threshold has been crossed by comparing the value read at GP==X with the value read at GP==X-1. In initial condition (i.e. when the Monitor reads the value at GP=1), the (hypothetical) value read at GP==X-1 is assumed to be at state-0.

B.3 Examples

B.3.1 Example 1

Figure B.1 illustrates a multi-level severity threshoding behaviour. The horizontal axis indicates the time intervals specified by the SimpleMonitor.monitorGranularityPeriod (GP). The vertical axis indicates the monitored measurementType values. The measurementType of this example is of type gauge. The SimpleMonitor.direction is set to "Increasing".

Further suppose that the measurementType is monitored for a two-level-severity-threshold-crossings.

The threshold for the lower-level-severity is set to 3 with a hysteresis low and high levels set to 2 and 4 respectively. The severity level is Minor.

The threshold for the higher-level-severity is set to 7 with a hysteresis low and high levels set to 6 and 8 respectively. The severity level is Major.

The notification triggering events are:

1. At GP interval 2, trigger a notifyNewAlarm with perceivedSeverity (ps) = Minor.

2. At GP interval 5, if the PMIRP supports notifyChangedAlarm, trigger a notifyChangedAlarm with ps = Major; else a notifyNewAlarm with ps = Major.

3. At GP interval 8, if the PMIRP supports notifyChangedAlarm, trigger a notifyChangedAlarm with ps = Minor; else a notifyClearedAlarm clearing the last emitted Major alarm.

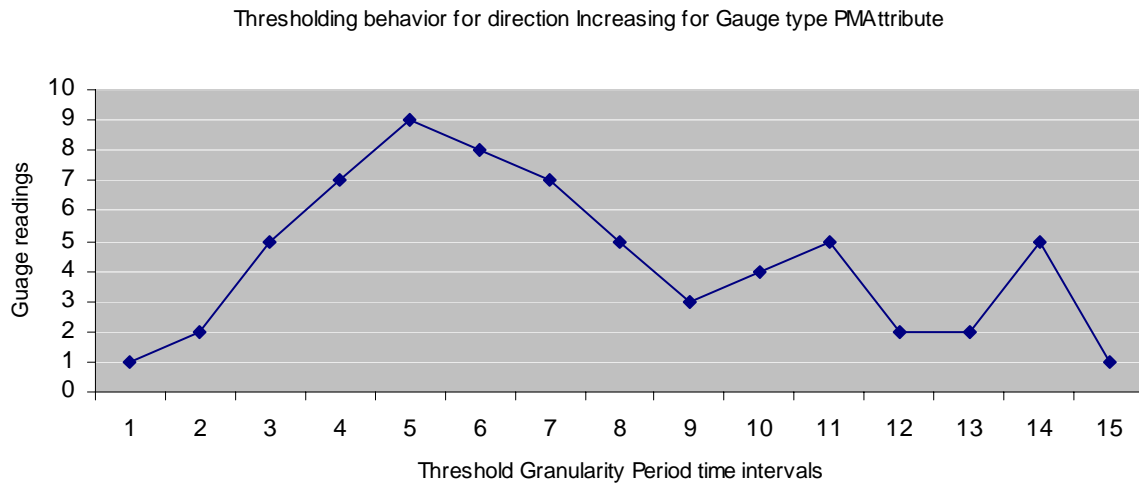4. At GP interval 15, trigger a notifyClearedAlarm clearing the last emitted Minor alarm.

Thresholding behavior for direction Increasing for Gauge type PMAttribute

Figure B.1

B.3.2   Example 2

Figure B.2 illustrates another multi-level severity threshoding behaviour for PMIRP supporting notifyChangedAarm. The horizontal axis indicates the time intervals specified by the GP. The vertical axis indicates the monitored measurementType values. The measurementType of this example is of type gauge. The SimpleMonitor.direction is set to "Decreasing".

Further suppose that the measurementType is monitored for a two-level severity threshold-crossings.

The threshold for the major-level severity is set to 7 with a hysteresis high and low levels set to 6 and 8 respectively.

The threshold for the minor-level severity is set to 9 with a hysteresis high and low levels set to 8 and 10 respectively.

The notification triggering events are:

1. At GP interval 1, trigger a notifyNewAlarm with ps = Major. See table B.1 table for transition State-0 to State-2.

2. At GP interval 5, trigger a notifyChangedAlarm with ps = Minor. See table B.1 for transition State-2 to State-1.

3. At GP interval 8, trigger a notifyChangedAlarm with ps = Major. See table B.1 for transition State-1 to State-2.
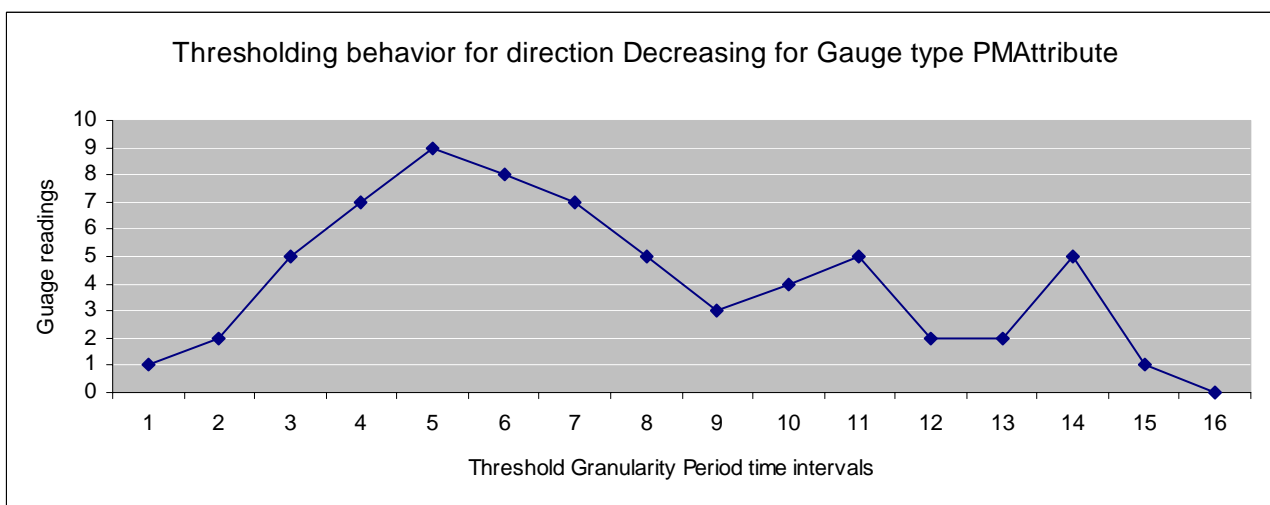

Figure B.2

B.3.3   Example 3

Figure B.3 illustrates a single-level threshoding behaviour. The horizontal axis indicates the time intervals specified by the GP. The vertical axis indicates the monitored measurementType values. The measurementType of this example is of type gauge. The SimpleMonitor.direction is set to "Increasing".

The threshold is set to 4 with a hysteresis high and low levels set to 5 and 3 respectively. The severity level is Critical.

The notification triggering events are:

1. At GP interval 3, trigger a notifyNewAlarm with ps = Critical.

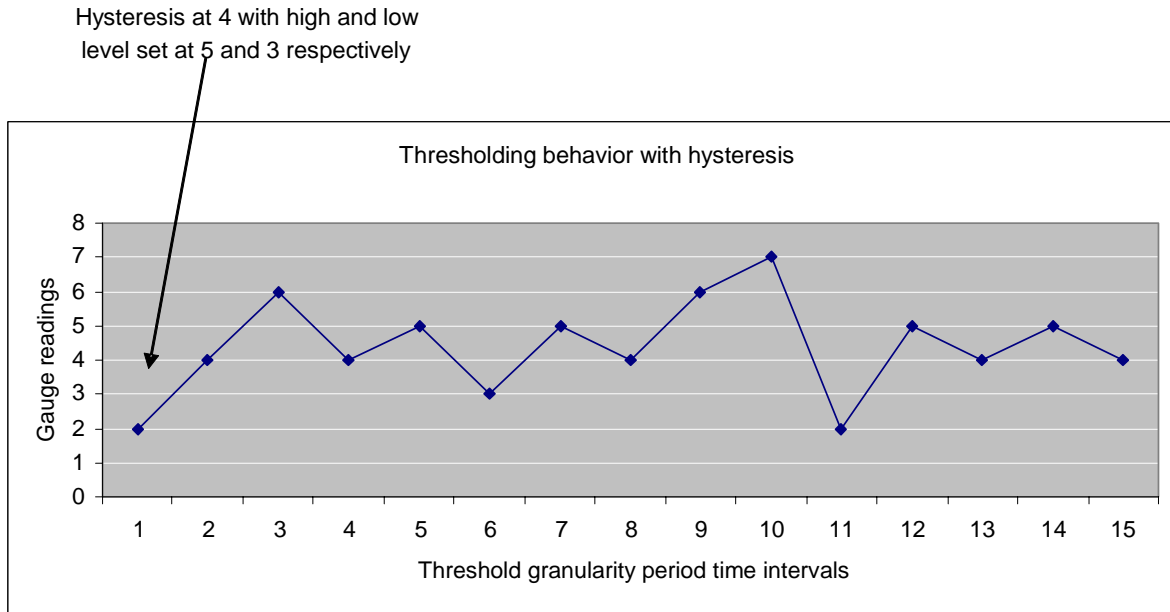2. At GP interval 11, trigger a notifyClearedAlarm with ps = Cleared.

Figure B.3

Annex C (informative):
Change history

Change history							
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Old**	**New**
Jun 2003	S_20	SP-030295	--	--	Submitted to TSG SA#20 for information	1.0.0	