ETSI TS 132 412 V12.0.0 (2014-10)

Digital cellular telecommunications system (Phase 2+);
Universal Mobile Telecommunications System (UMTS);
LTE;
Telecommunication management;
Performance Management (PM)
Integration Reference Point (IRP): Information Service (IS)
(3GPP TS 32.412 version 12.0.0 Release 12)

Reference
RTS/TSGS-0532412vc00

Keywords
GSM,LTE,UMTS

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

The present document can be downloaded from:
http://www.etsi.org

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status.
Information on the current status of this and other ETSI documents is available at
http://portal.etsi.org/tb/status/status.asp

If you find errors in the present document, please send your comment to one of the following services:
http://portal.etsi.org/chaircor/ETSI_support.asp

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.
The content of the PDF version shall not be modified without the written authorization of ETSI.
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2014.
All rights reserved.

DECT™, **PLUGTESTS**™, **UMTS**™ and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.
3GPP™ and **LTE**™ are Trade Marks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.
GSM® and the GSM logo are Trade Marks registered and owned by the GSM Association.

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

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

The present document may refer to technical specifications or reports using their 3GPP identities, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between GSM, UMTS, 3GPP and ETSI identities can be found under http://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**may not**", "**need**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

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

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 TS-family covering the 3rd Generation Partnership Project: Technical Specification Group Services and System Aspects; Telecommunication management, as identified below:

32.411: "Performance Management (PM) Integration Reference Point (IRP): Requirements"

32.412: "Performance Management (PM) Integration Reference Point (IRP): Information Service (IS)"

32.416: "Performance Management (PM) Integration Reference Point (IRP); Solution Set (SS) definitions"

The present document is part of a set of TSs which describes 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.

This IRP IS is aligned with ITU-T M.3704 [16] in terms of the definitions of operations for Performance management.

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 (IRP): Information Service (IS)".

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

[6] Void.

[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 CM Integration Reference Point (IRP): Information Service (IS)".

[10] 3GPP TS 32.342: "Telecommunication management; File Transfer (FT) Integration Reference Point (IRP): Information Service (IS)".

[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 (IS)".

[13] Void.

[14] 3GPP TS 32.40x: "Telecommunication management; Performance Management (PM)".

[15] 3GPP TS 32.150: "Telecommunication management; Integration Reference Point (IRP) Concept and definitions".

[16] ITU-T M.3704 (01/2012): 'Telecommunication Management, Including TMN and Network Maintenance; Integrated services digital networks; Common management services – Performance management – Protocol neutral requirements and analysis'.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions defined 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
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
PMIRP	Performance Management Integration Reference Point
UML	Unified Modelling Language

4 System Overview

4.1 System Context

The general definition of the System Context for the present IRP is found in 3GPP TS 32.150 [15] subclause 4.7.

In addition, the set of related IRP(s) relevant to the present IRP is shown in the two diagrams below.

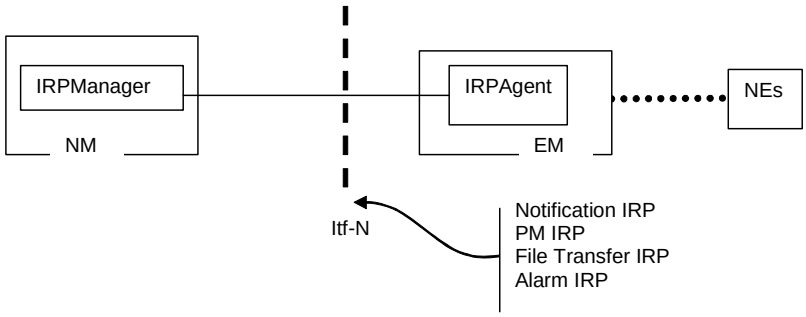


Figure 4.1: System Context A

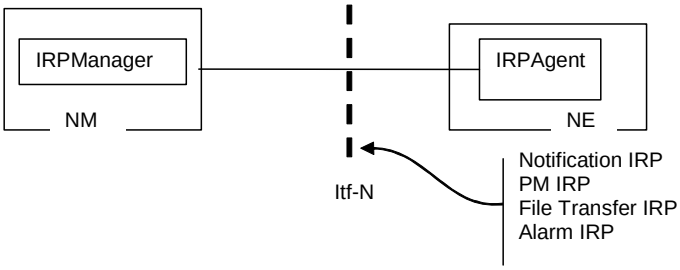


Figure 4.2: System Context B

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 (IOCs)

6.1 Imported information entities and local labels

Label reference	Local label
3GPP TS 32.312 [5], information object class, `ManagedGenericIRP`	ManagedGenericIRP
3GPP TS 32.602 [9], information object class, `ManagedEntity`	ManagedEntity
3GPP TS 32.302 [12], information object class, `NotificationIRP`	NotificationIRP
3GPP TS 32.111-2 [4], information object class, `AlarmIRP`	AlarmIRP
3GPP TS 32.342 [10], information object class, `FileTransferIRP`	FileTransferIRP

6.2 Class diagram

6.2.1 Attributes and relationships

This clause introduces the set of Support Information Object Classes (Support 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 clause provides the overview of all Support IOCs in UML. Subsequent clauses provide more detailed specification of various aspects of these support object classes.

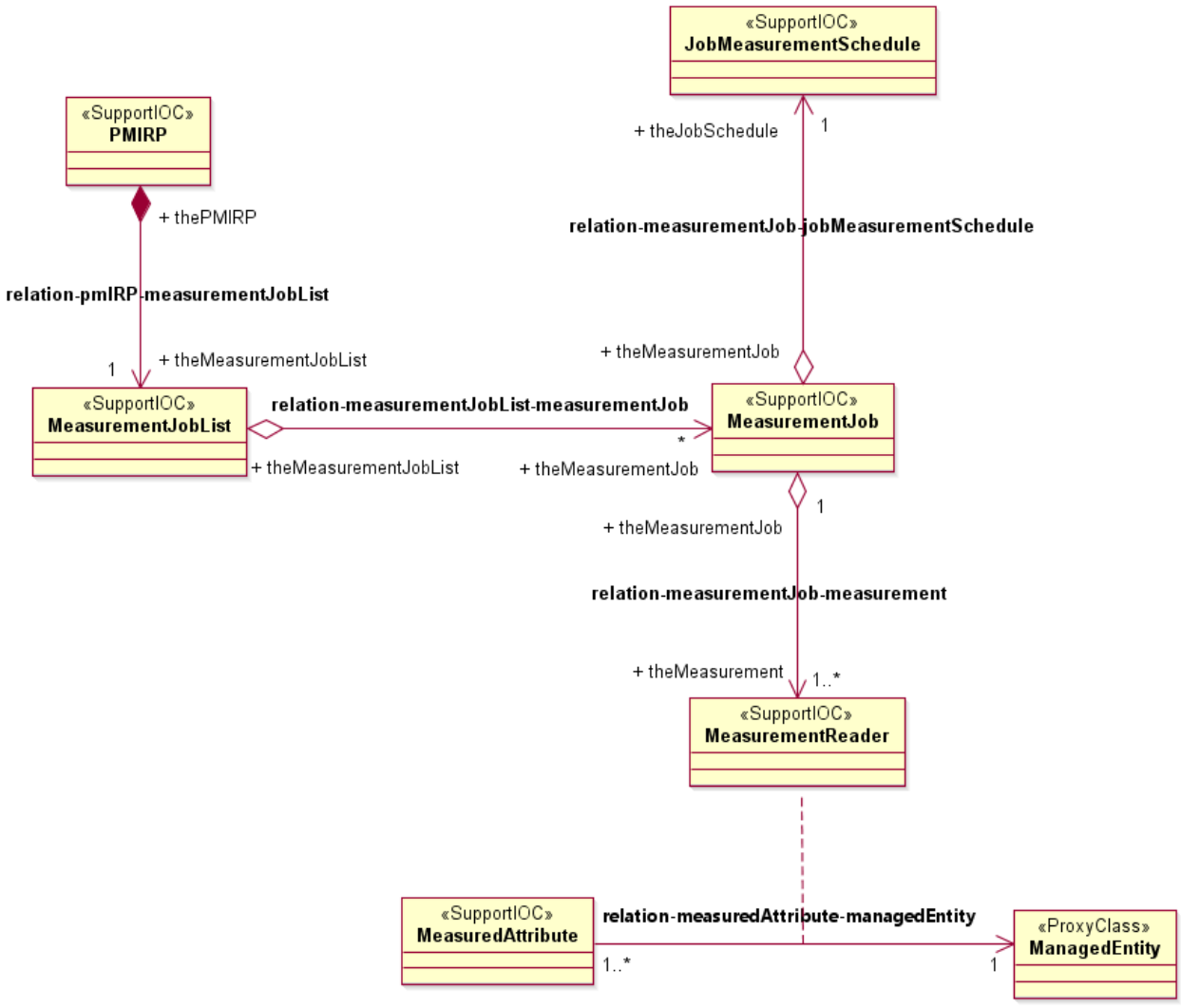


Figure 6.2.1(a): Information Object Class UML Diagram - Measurement

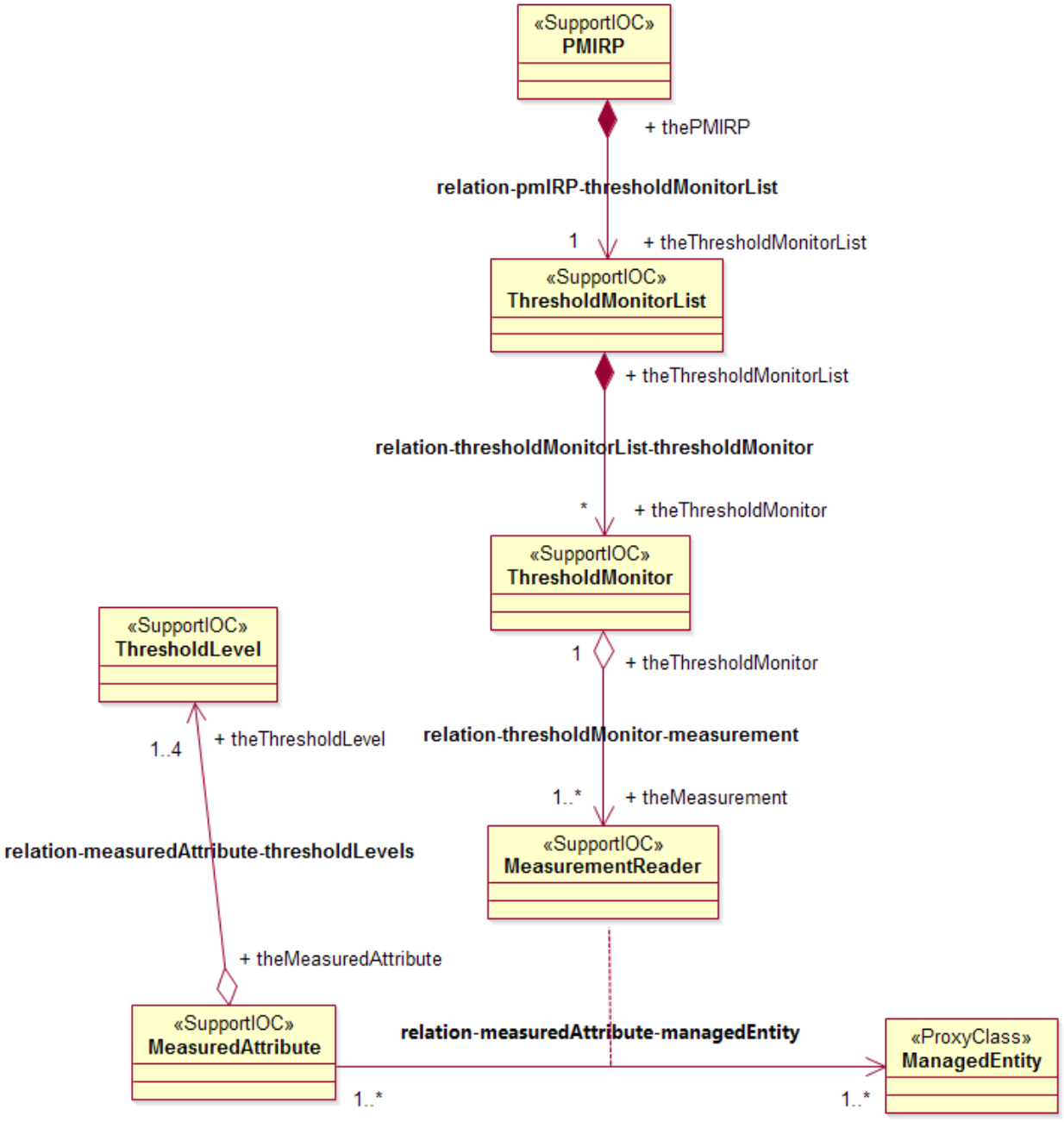


Figure 6.2.1(b): Information Object Class UML Diagram - Thresholding

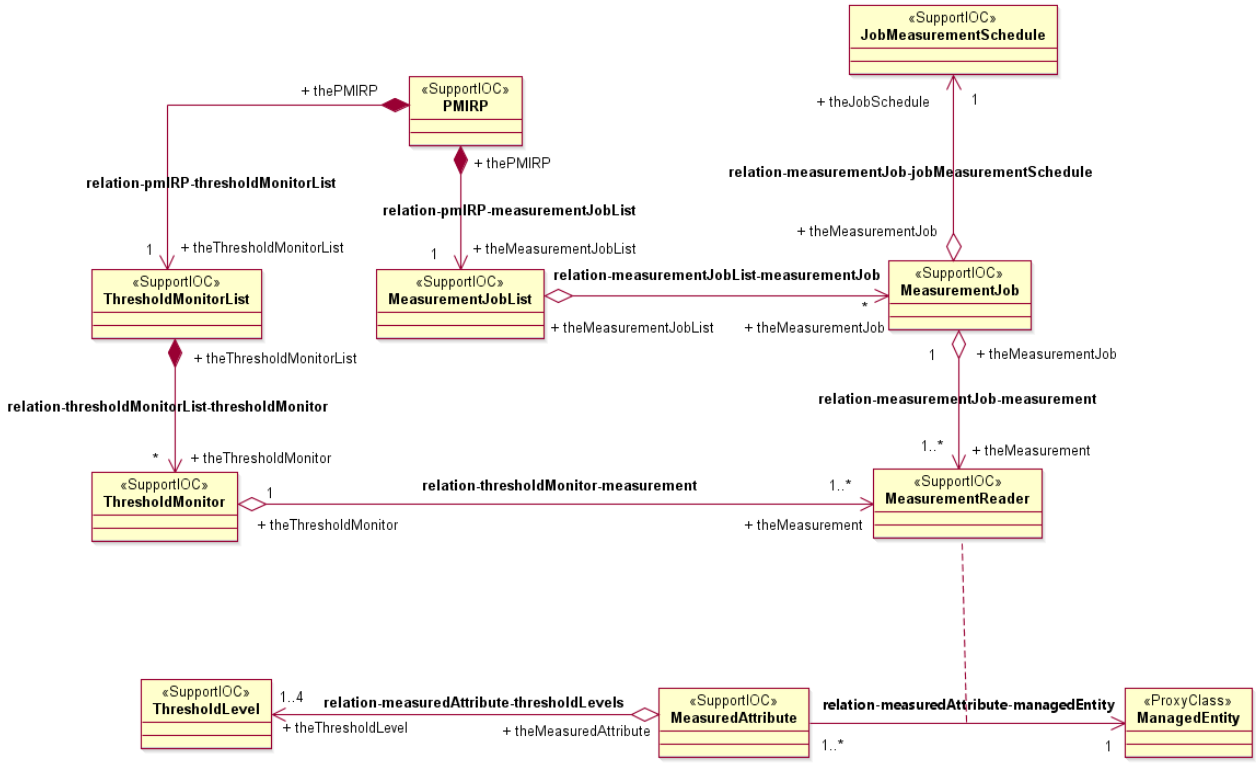


Figure 6.2.1(c): Information Object Class UML Diagram - Combined

6.2.2 Inheritance

This subclause depicts the inheritance relationships that exist between Support IOCs.

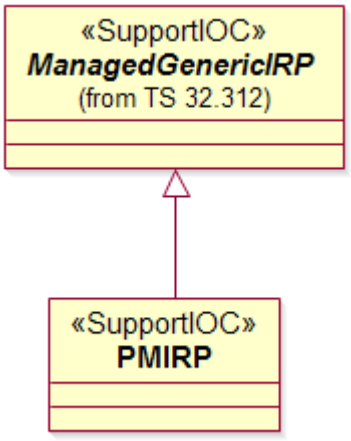


Figure 6.2.2: Information Object Class Inheritance UML Diagram

6.3 Information Object Class (IOCs) 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 defined in `JobMeasurementSchedule`.

The target measurementTypes are those measurementTypes, whose names are in the related `MeasuredAttribute.measurementTypeName`, of the related `ManagedEntity` instances. The attributes of `MeasurementJob`, `JobMeasurementSchedule` and `MeasuredAttribute` can not be modified (except jobStatus) once a measurement job is created.

The `MeasurementJob` shall use its information and the information of the related `JobMeasurementSchedule` and the information of `MeasuredAttribute`(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` should 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.

The `PMIRP` Agent may decide to stop a `MeasurementJob` because of a `PMIRP` Agent 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 `PMIRP` Agent and the related managed resource. In such case, the `PMIRP` Agent can stop the `MeasurementJob` resulting in the `MeasurementJob.jobStatus` == "Stopped". Additionally, " notifyMeasurementJobStatusChanged" notification shall be emitted to notify all subscribed `IRPManagers` about the stopping of a `MeasurementJob`.

The PMIRP Agent may also selectively suspend one or more MeasurementJobs without any operator"s action in case of overload. In such case, the PMIRP Agent suspends all the chosen MeasurementJob(s) resulting in the MeasurementJob.jobStatus == 'Suspended' and emit 'notifyMeasurementJobStatusChanged' notification for each MeasurementJob to notify all subscribed IRPManager(s) about the suspension. The behaviour on threshold monitors of PMIRP Agent after suspending is the same as the one after calling suspendMeasurementJob.

Then it should be possible, at any time, for the operator to resume a MeasurementJob suspended by the system. PMIRP agent may also resume the measurementJob by itself after detecting the end of the overload. When PMIRP Agent does not support the resumeMeasurementJob operation, PMIRP Agent shall resume the measurementJob automatically after detecting the end of overload. The behaviour of PMIRP Agent after resuming the measurementJob is the same as the one after calling resumeMeasurementJob. PMIRPAgent may suspend the corresponding measurementJob(s) when it can decide which one/ones cause(s) the overload; otherwise, PMIRP Agent can take jobPriority as a reference. The criterion for suspending one or another measurementJob of equal job priority is vendor specific.

The way to detect the beginning and the end of overload is out of the scope of the present document.

6.3.1.2 Attributes

Attribute name	Support Qualifier	Read Qualifier	Write Qualifier
`jobId`	M	M	-
`jobGranularityPeriod`	M	M	-
`jobReportingPeriod`	M	M	-
`jobStatus`	M	M	-
`jobPriority`	O	M	-

6.3.1.3 State diagram

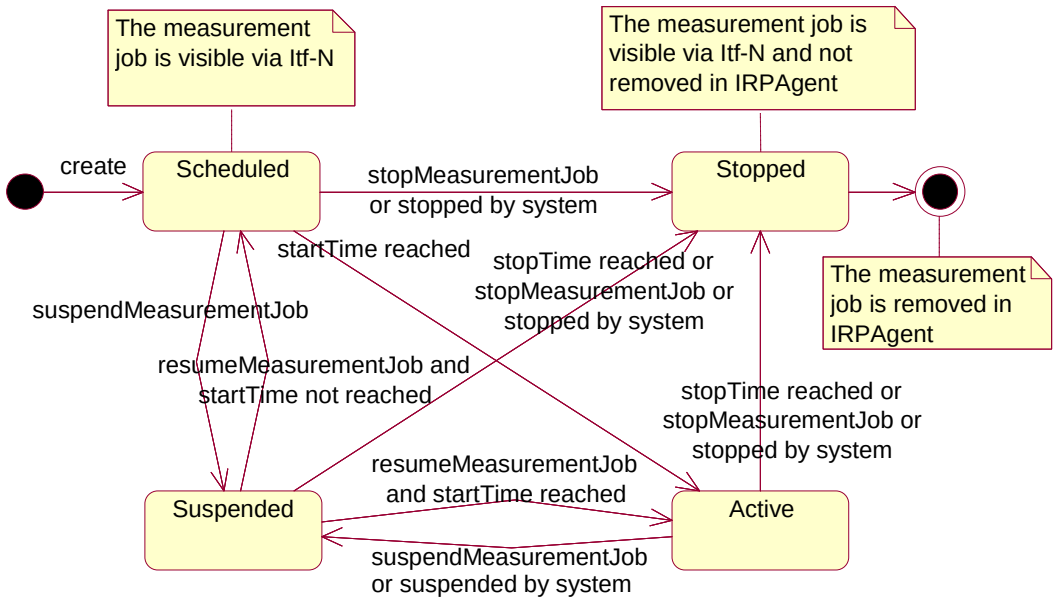


Figure 6.3.1.3.1: State Diagram for MeasurementJob

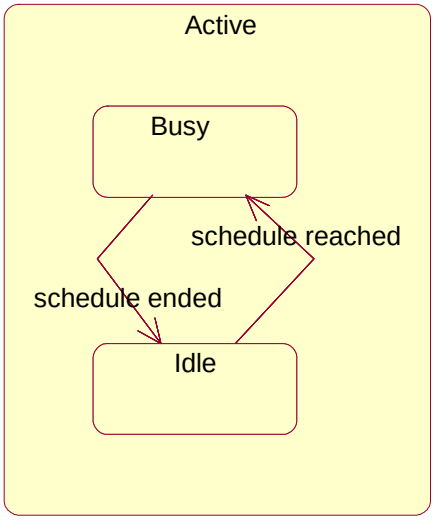


Figure 6.3.1.3.2: SubState Diagram of Active

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

6.3.2 JobMeasurementSchedule

6.3.2.1 Definition

`JobMeasurementSchedule` is the representation of the time intervals when the measurement job monitors and collects the performance measurement data. The attributes of `JobMeasurementSchedule` can not be modified once a measurement job is created.

6.3.2.2 Attributes

Attribute name	Support Qualifier	Read Qualifier	Write Qualifier
`jobStartTime`	M	M	-
`jobStopTime`	M	M	-
`jobSchedule`	M	M	-

6.3.3 PMIRP

6.3.3.1 Definition

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

6.3.3.2 Attribute

There is no additional attribute defined for this Support IOC besides those inherited.

6.3.3.3 Notification

Name	Qualifier	Notes
`notifyMeasurementJobStatusChanged`	M	See clause 7.6.1.
`notifyThresholdMonitorObjectCreation`	M (see note)	See clause 7.7.1.
`notifyThresholdMonitorObjectDeletion`	M (see note)	See clause 7.7.2.
`notifyThresholdMonitorStatusChanged`	O (see note)	See clause 7.7.3.

NOTE: This notification is defined within the PMIRPNotification_2 Interface (see clause 7.7) which is qualified as optional.
If the Interface is supported, then the qualifier defined here is relevant.
Otherwise, the qualifier is irrelevant.

6.3.4 MeasurementJobList

6.3.4.1 Definition

`MeasurementJobList` is the representation of the list of `MeasurementJobs`.

6.3.4.2 Attributes

Attribute name	Support Qualifier	Read Qualifier	Write Qualifier
`jobListId`	M	M	-

6.3.5 MeasuredAttribute

6.3.5.1 Definition

It represents the name of the measurementType of the related `ManagedEntity` instance whose value is to be monitored and collected. The name of the monitored and collected measurementType is captured by measurementTypeName.

6.3.5.2 Attributes

Attribute name	Support Qualifier	Read Qualifier	Write Qualifier
`measurementTypeName`	M	M	-

6.3.6 MeasurementReader

6.3.6.1 Definition

It represents the capability to read the value of the measurementType of the related `ManagedEntity` instance. The value of the monitored and collected measurementType is captured by measurementResultValue.

6.3.6.2 Attributes

Attribute name	Support Qualifier	Read Qualifier	Write Qualifier
`measurementResultValue`	M	M	-
`probableCause`	M	M	-
`specificProblem`	M	M	-
`direction`	M	M	-

6.3.7 ManagedEntity

6.3.7.1 Definition

The IOC `ManagedEntity` represents 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. It also can represent VSE subclass of 3GPP defined NRM IOCs or VSE defined NRM IOCs. Besides, it can also represent a 'private Object Class' as defined in TS 32.404, e.g. RA. `ManagedEntity` is used in the specification of `PMIRP` operations to represent an IOC defined in these Network Resource Models and private Object Classes.

6.3.8 Monitor

6.3.8.1 Definition

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

It is the IRPAgent"s choice to support (a) or (b) (but not both). The support is on an IRPAgent system wide basis and is not on a per threshold basis. The IRPAgent"s behaviour regarding which approach (i.e., (a) or (b) above) to use, shall be the same for emitting alarms and for clearing alarms.

The instances of a class derived from this abstract class shall emit notifyThresholdMonitorObjectCreation when they are first created; and shall emit a notifyThresholdMonitorObjectDeletion when deleted.

The instances of a class derived from this abstract class shall also emit notifyNewAlarm, notifyChangedAlarm and notifyClearedAlarm according to the rules specified in Annex B: Threshold Alarm Triggering Events.
The `objectClass` and `objectInstance` parameter of these notifications carry the class and DN of the `ManagedEntity` whose `measurementType` is being monitored and whose threshold condition has been triggered. The eventType of these notifications carry "Quality of Service Alarm". See 3GPP TS 32.111-2 [4].

6.3.8.2 Attributes

Attribute name	Support Qualifier	Read Qualifier	Write Qualifier
`monitorId`	M	M	-
`monitorGranularityPeriod`	M	M	-

6.3.8.3 Notification

Notification name	Note
`notifyThresholdMonitorCreation`	See clause 7.1 (class diagram).
`notifyThresholdMonitorDeletion`	See clause 7.1 (class diagram).

6.3.9 ThresholdMonitorList

6.3.9.1 Definition

`ThresholdMonitorList` is the representation of the list of `ThresholdMonitors`.

6.3.9.2 Attributes

Attribute name	Support Qualifier	Read Qualifier	Write Qualifier
`monitorListId`	M	M	-

6.3.10 ThresholdMonitor

6.3.10.1 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 (a) crossed or (b) reached (see 6.3.8.1 for clarification of the alternatives (a) and (b)), a performance alarm will be emitted (see 3GPP TS 32.401 [7]). The name of the monitored measurementType is captured by `MeasuredAttribute.measurementTypeName` and the value of it is captured by `MeasurementReader.measurementResultValue`.

6.3.10.2 Attribute

Attribute name	Support Qualifier	Read Qualifier	Write Qualifier
`thresholdMonitorStatus`	M	M	-

6.3.11 ThresholdLevel

6.3.11.1 Definition

It defines the thresholding criteria (via `ThresholdLevel.thresholdValue` and `ThresholdLevel.hysteresis`) for a measurementType. It also specifies the thresholdSeverity level (via `ThresholdLevel.thresholdSeverity`) carried in the alarm triggered by the (a) threshold crossing or (b) threshold reaching event. (See 6.3.8.1 for clarification of the alternatives (a) and (b)).

6.3.11.2 Attribute

Attribute name	Support Qualifier	Read Qualifier	Write Qualifier
`thresholdValue`	M	M	-
`thresholdSeverity`	M	M	-
`hysteresis`	M	M	-

6.4 Information relationship definitions

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
theMeasurementJobList	It represents the MeasurementJobList.
thePMIRP	It represents the PMIRP.

6.4.1.3 Constraint

Name	Definition
uniqueJobListId	The jobListIds playing the role of theMeasurementJobList, are unique within a particular PMIRP.

6.4.2 relation-measurementJobList-measurementJob (M)

6.4.2.1 Definition

This represents the relationship between MeasurementJobList and MeasurementJob.

6.4.2.2 Role

Name	Definition
theMeasurementJob	It represents the MeasurementJob.
theMeasurementJobList	It represents the MeasurementJobList.

6.4.2.3 Constraint

Name	Definition
uniqueJobId	The jobIds of all MeasurementJobs, playing the role of theMeasurementJob, are unique within a particular PMIRP and MeasurementJobList.

6.4.3 relation-measurementJob-jobMeasurementSchedule (M)

6.4.3.1 Definition

This represents the relationship between MeasurementJob and JobMeasurementSchedule.

6.4.3.2 Role

Name	Definition
theJobSchedule	It represents the JobMeasurementSchedule.
theMeasurementJob	It represents the MeasurementJob.

6.4.4 relation-measurementJob-measurement (M)

6.4.4.1 Definition

This represents the relationship between `MeasurementJob` and `MeasurementReader`.

6.4.4.2 Role

Name	Definition
`theMeasurement`	It represents the `MeasurementReader`.
`theMeasurementJob`	It represents the `MeasurementJob`.

6.4.5 relation-measuredAttribute-managedEntity (M)

6.4.5.1 Definition

This represents the relationship between `MeasuredAttribute` and `ManagedEntity`.

6.4.5.2 Role

Name	Definition
`theMeasuredAttribute`	The `MeasuredAttribute`, when playing this role, reads the monitored measurement types of the related `ManagedEntity` instances. The names of measurement types are captured by the `MeasuredAttribute.MeasurementTypeName` . When playing this role, the `MeasuredAttribute` is aware of the class name and DN of the `ManagedEntity`.
`theME`	The `ManagedEntity`, when playing this role, represents the actual network resource instance under measurement or whose measurementTypes are being monitored for (a) threshold-crossing or (b) threshold-reaching and threshold-clearing. (See 6.3.8.1 for clarification of the alternatives (a) and (b))

6.4.5.3 Constraint

Name	Definition
`applyToSameObjectClass`	The `MeasuredAttribute`, which playing the role of "theMeasuredAttribute", 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".

6.4.6 relation-pmIRP-thresholdMonitorList (M)

6.4.6.1 Definition

This represents the relationship between PMIRP and ThresholdMonitorList.

6.4.6.2 Role

Name	Definition
theThresholdMonitorList	It represents the ThresholdMonitorList.
thePMIRP	It represents the PMIRP.

6.4.6.3 Constraint

Name	Definition
uniqueMonitorListId	The monitorListIds playing the role of theThresholdMonitorList, are unique within a particular PMIRP.

6.4.7 relation-thresholdMonitorList-thresholdMonitor (M)

6.4.7.1 Definition

This represents the relationship between ThresholdMonitorList and ThresholdMonitor.

6.4.7.2 Role

Name	Definition
theThresholdMonitor	It represents the ThresholdMonitor.
theThresholdMonitorList	It represents the ThresholdMonitorList.

6.4.7.3 Constraint

Name	Definition
uniqueMonitorId	The monitorIds of all ThresholdMonitors, playing the role of theThresholdMonitor, are unique within a particular ThresholdMonitorList.

6.4.8 relation-thresholdMonitor-measurement (M)

6.4.8.1 Definition

This represents the relationship between ThresholdMonitor and MeasurementReader.

6.4.8.2 Role

Name	Definition
theMeasurement	It represents the MeasurementReader.
theThresholdMonitor	It represents the ThresholdMonitor.

6.4.9 relation-measuredAttribute-thresholdLevels (M)

6.4.9.1 Definition

This represents the relationship between `MeasuredAttribute` and `ThresholdLevel`.

6.4.9.2 Role

Name	Definition
`theMeasuredAttribute`	The `MeasuredAttribute`, when playing this role, can have 1..4 `ThresholdLevel` associates with it to set the (a) threshold-crossing or (b) threshold reaching and the clearing criteria (i.e. `ThresholdLevel.thresholdValue`, `ThresholdLevel.hysteresis`) and associated thresholdSeverity (i.e. `ThresholdLevel.thresholdSeverity`) of the monitored measurementType. (see 6.3.8.1 for clarification of the alternatives (a) and (b))
`theThresholdLevel`	When playing this role, the `ThresholdLevel` captures the (a) threshold-crossing or (b) threshold-reaching and the threshold-clearing criteria and its associated thresholdSeverity level. (see 6.3.8.1 for clarification of the alternatives (a) and (b))

6.4.9.3 Constraint

Name	Definition
`noMoreThanFourLevels`	A `MeasuredAttribute` may have no more than four `ThresholdLevels` associated with it.

6.5 Information attribute definition

6.5.1 Definition and legal values

Attribute Name	Definition	Legal Values
direction	For some measurementType, the higher its thresholdValue, the higher is the thresholdSeverity. For others, the lower its thresholdValue, the higher is its thresholdSeverity. 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 event is triggered when the value first equals or exceeds (when compared against the last read value) a threshold value. The threshold is said to be cleared when the measurementType value falls below (when compared against the last read value) one or more threshold values. If it is "Decreasing", the threshold event is triggered when the measurementType value first equals or falls below one or more threshold values. The threshold is said to be cleared when the measurementType value rises above the threshold value. See annex B (Threshold Related Performance Alarm Triggering Events) for details of the behaviour of multiple thresholds.	Possible values are: "Increasing", "Decreasing"
hysteresis	A threshold has a value. It can have a hysteresis. A threshold with a hysteresis has a threshold-high and a threshold-low value that are different from 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 (a) threshold-crossing or (b) threshold-reaching or a threshold-clearing condition. (See 6.3.8.1 for clarification of the alternatives (a) and (b)) threshold-high = threshold + hysteresis threshold-low = threshold - hysteresis See annex B (Threshold Related Performance Alarm Triggering Events).	Any positive value
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. 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.
jobId	It identifies the MeasurementJob instance (and distinguishes it from all other existing and stopped MeasurementJob instances of the PMIRP Agent).	Any identifier except: 1. Those that identify MeasurementJob instances whose MeasurementJob.jobStatus(s)are Scheduled, Active, Suspended or Stopped; and 2. Those that appear in filenames of files ready for IRPManager retrieval.
jobListId	It identifies the singleton MeasurementJobList of the PMIRP Agent.	Any identifier.
jobReportingPeriod	It specifies the period between two successive emissions of notifyFileReady or notifyFilePreparationError [10]. The two notifications are related to the same Job. See constraints reportTime in clause 6.5.2.	Its value should be one or multiple of jobGranularityPeriod.

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]. The legal values for them are as follows. dailyScheduling: {{ intervalStart {hour 0, minute 0}, intervalEnd {hour 23, minute 59}}} weeklyScheduling: {{ daysOfWeek '1111111'B, intervalsOfDay dailyScheduling}}
jobStartTime	It specifies the begin time from which the MeasurementJob will be active.	All values that indicate valid timestamp.
jobStatus	It specifies the status of MeasurementJob.	Its value should be one of the following: Scheduled, Active, Suspended Stopped
jobPriority	It specifies the priority of MeasurementJob	Its value should be one of the following: Low, Medium, High
jobStopTime	It specifies the end time after which the MeasurementJob will be stopped.	All values that indicate valid timestamp and it should be later than jobStartTime.It's not necessary that jobStartTime and jobStopTime specifies time within the same day. This attribute may carry the value "indefinitely".
measurementResultValue	It identifies the value of a measurement type.	Any valid measurement result value.
measurementTypeName	It identifies a name of one measurement type whose value is being collected and monitored.	Any valid measurement type name as defined by the measurement definition template in 3GPP TS 32.40x [14].
monitorGranularityPeriod	It specifies the period between two successive reading of the thresholdValue to determine (a) threshold-crossing or (b) threshold-reaching and threshold-clearing. (See 6.3.8.1 for clarification of the alternatives (a) and (b))	It can be 5 minutes, 15 minutes, 30 minutes, 1 hour, 12 hours or 24 hours. It has to be a multiple of the jobGranularityPeriod if the MeasurementJob monitoring the same measurementType exists.
monitorId	It identifies the ThresholdMonitor instance (and distinguishes it from all other existing ThresholdMonitor instances of the PMIRP Agent).	Any identifier except those that are currently used.
monitorListId	It identifies the singleton ThresholdMonitorList in the PMIRP Agent.	Any identifier.
probableCause	It identifies the probable cause (of the (a) threshold crossing or (b) threshold reaching) carried by the threshold crossing or reaching alarm. (See 6.3.8.1 for clarification of the alternatives (a) and (b))	"Threshold Crossed"
thresholdSeverity	It identifies the thresholdSeverity of the (a) threshold crossing or (b) threshold reaching event. (See 6.3.8.1 for clarification of the alternatives (a) and (b))	Warning, Minor, Major, Critical
specificProblem	It identifies the specific problem (causing the (a) threshold crossing or (b) threshold reaching) carried by the threshold crossing or reaching alarm. (See 6.3.8.1 for clarification of the alternatives (a) and (b))	Any valid specificProblem as defined by 3GPP TS 32.111-2 [4].
thresholdMonitorStatus	It specifies the current status of the ThresholdMonitor.	Active - ThresholdMonitor is working; Suspended - ThresholdMonitor is suspended.

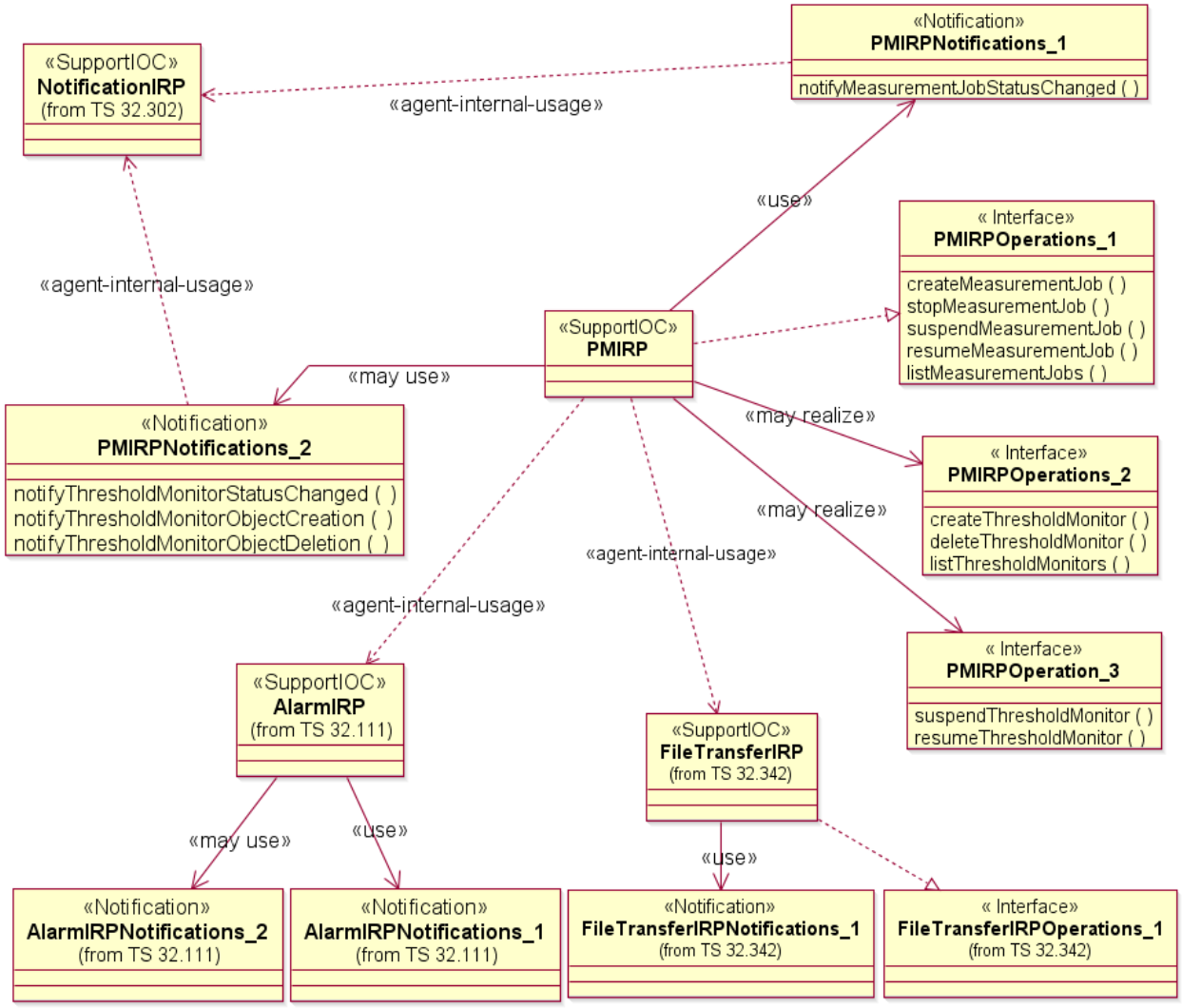
Attribute Name	Definition	Legal Values
thresholdValue	It defines the threshold value of the monitored measurementTypes. If the value is (a) crossed or (b) reached, the performance alarm shall be emitted depending on the value of the thresholdMonitorStatus. (See 6.3.8.1 for clarification of the alternatives (a) and (b))	If the monitored measurementType is of Gauge type, this thresholdValue shall be of the same type. If the monitored measurementType 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 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.

6.5.2 Constraints

Name	Definition
firstReportTime	If the jobReportingPeriod is n times the jobGranularityPeriod, then the first report time shall be immediately after the completion of the n-th successful measurementType reading.
measurementTypeReadTimes	The measurementType reading times are determined by the following rules. - If jobGranularityPeriod is 5 minutes, then read on every 5th minute of the hour. - If jobGranularityPeriod is 15 minutes, then read on every 15th minute of the hour. - If jobGranularityPeriod is 30 minutes, then read on every 30th minute of the hour - If jobGranularityPeriod is 1 hour, then read on every hour of the day - If jobGranularityPeriod 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 jobGranularityPeriod 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.
multiLevelSeverity HysteresisOverlap	For each ThresholdLevel.thresholdSeverity, there may be a ThresholdLevel.hysteresis value corresponding to it. These multi-level values of ThresholdLevel.hysteresis, relating to one ThresholdMonitor, shall not overlap.
multiLevelSeverityOrder	The multiple ThresholdLevel(s), related to one ThresholdMonitor, shall be of the following order. If ThresholdLevel.thresholdSeverity 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.thresholdSeverity 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.
numberOfThresholdLevelPerThresholdMonitor	In relation-tMonitor-tLevel, the multiplicity of theThresholdLevel is "1..4". If PMIRP supports multi-level thresholdSeverity, then there are at most 4 ThresholdLevels relating to one ThresholdMonitor. All ThresholdLevel.thresholdSeverity instances shall hold a different value that is chosen from the following: 'Critical', 'Major', "Warning' and 'Minor'. If PMIRP does not support multi-level thresholdSeverity, then there is only one ThresholdLevel relating to one ThresholdMonitor. The sole ThresholdLevel.thresholdSeverity shall hold one of the following: 'Critical', 'Major', "Warning' or 'Minor'.

7 Interface definition

7.1 Class diagram



NOTE 1: PMIRPNotifications_2 is mandatory if PMIRPOperations_2 is supported.
NOTE 2: The suspendMeasurementJob and resumeMeasurementJob operations are optional.

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 (M)

7.3.1 Operation createMeasurementJob (M)

7.3.1.1 Definition

This operation supports `IRPManager`'s request to create a `MeasurementJob` through Itf-N.

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

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

When a measurementType is collected by one `MeasurementJob` for a given instance, another `MeasurementJob` which wants to collect the same measurementType for the same instance with different or the same jobGranularityPeriod may be rejected. This behaviour shall be consistent for a given implementation by a specific vendor.

7.3.1.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
iOCName	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.
iOCInstanceList	M	List of <ManagedEntity.objectInstance>	It specifies the list of DNs of ManagedEntity instances whose measurementType(s) are to be collected. ManagedEntity instance(s) can be added in the network regardless of whether a MeasurementJob exists or not. An empty list means that all instances known by PMIRP shall be monitored. The support of the use of empty list to indicate all instances is optional. If the MeasurementJob is created using empty list, there are two possible implementations: 1) PMIRP shall monitor all instances known by PMIRP at the time of MeasurementJob creation, and it will not monitor instances added later. 2) PMIRP shall monitor all instances known by PMIRP at the time of MeasurementJob creation, and it shall monitor instances added later.
measurementCategoryList	M	List of < MeasuredAttribute. measurementTypeName>	MeasuredAttribute. MeasurementTypeName specifies the corresponding name of measurementType to be measured. The element of measurementCategoryList could be one of the following: - The form "family.measurementName.subcounter" can be used in order to retrieve a specified subcounter of a MeasurementType. - The form "family.measurementName" can be used in order to retrieve a specific MeasurementType. In case the MeasurementType includes subcounters, all subcounters will be retrieved. - The form "family" can be used in order to retrieve all MeasurementTypes in this family. At least, one element of measurementCategoryList shall be specified.
granularityPeriod	M	MeasurementJob.jobGranularityPeriod	See clause 6.5.1 (definition and legal values).
reportingPeriod	M	MeasurementJob.jobReportingPeriod	See clause 6.5.1 (definition and legal values). PMIRP will periodically monitor the performance data according to the "reportingPeriod" parameter and produces an aggregated report in one or more files. These files shall be transferred through file transfer mechanism covered by the File Transfer IRP IS (see TS 32.342 [10]).
startTime	O	JobMeasurementSchedule.jobStartTime	See clause 6.5.1 (definition and legal values). Default value is "start now". If startTime is in the past, the current time will be used and the job will start immediately.
stopTime	O	JobMeasurementSchedule.jobStopTime	See clause 6.5.1 (definition and legal values). Default value for jobStopTime is to run indefinitely. StopTime shall be later than startTime and current time.
schedule	O	JobMeasurementSchedule.jobSchedule	See clause 6.5.1 (definition and legal values). Default value is "daily".
priority	O	JobMeasurementJob.jobPriority	See clause 6.5.1 (definition and legal values). Default value is 'Medium'

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 < ManagedEntity.objectInstance, MeasuredAttribute.measurementTypeName, reason >	To create a MeasurementJob, best-effort is required. The parameter of 'unsupportedList' must be returned if status = PartialSuccess. The reason can be any of: measurementTypeName is unknown to the PMIRP. measurementTypeName is invalid. measurementTypeName is not supported in the specific implementation. The related IOC Instance is unknown (e.g. it does not exist at the time of this operation invocation) to the PMIRP. Insufficient capacity to monitor the related IOC Instance(s). measurementTypeName is already monitored for the IOC instance with the same or another granularity period. When the input parameter iOCInstanceList is empty, the unsupportedList shall be empty.
status	M	ENUM (Success, Failure, PartialSuccess)	An operation may fail because of a specified or unspecified reason.

7.3.1.4 Pre-condition

atLeastOneValidMeasurementType AND validStartTime AND validStopTime AND validSchedule AND validGranularityPeriod AND validReportingPeriod AND sufficientCapacity AND validPriority.

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.
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.
validPriority	The priority is valid.

7.3.1.5 Post-condition

someMeasurementTypeNamesMonitored AND notifyMeasurementJobStatusChangedMayBeEmitted.

Assertion Name	Definition
someMeasurementTypeNamesMonitored	At least one input measurementTypeName is monitored by the newly created MeasurementJob.
notifyMeasurementJobStatusChangedMayBeEmitted	If start time is "start now", the MeasurementJob.jobStatus turns "Active" and a notifyMeasurementJobStatusChanged is emitted.

7.3.1.6 Exceptions

Exception Name	Definition
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 which is one of: emCpuBusy; emHDShortage, emLowMemory, {neCpuBusy, neObjectInstList}, {neHDShortage neObjectInstList}, {neLowMemory, neObjectInstList}, maxJobReached, otherReason; status is set to 'Failure'. **Exit state:** Entry state.
invalidPriority	**Condition:** validPriority is false. **Returned Information:** Name of the exception; status is set to "Failure". **Exit state:** Entry state.
noValidMeasurementType	**Condition:** (atLeastOneValidMeasurementType) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.

7.3.2 Operation stopMeasurementJob (M)

7.3.2.1 Definition

This operation supports `IRPManager`'s request to stop a `MeasurementJob` through Itf-N, after which, the `MeasurementJob` may still be visible via Itf-N. Whether the `MeasurementJob` is removed from the managed system is vendor specific and out of scope of the present document.

The behaviour of the IRPAgent when the job is stopped is vendor specific, which means the job could be stopped at the end of the GranularityPeriod or immediately.

After the job has been stopped, the notifyFileReady or notifyFilePreparationError notification shall be emitted immediately or when the next reporting period is reached.

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` AND `notifyMeasurementJobStatusChangedEmitted`.

Assertion Name	Definition
`measurementJobIsStopped.`	The `MeasurementJob` identified by jobId is stopped but may still be visible via Itf-N. The Job state is "Stopped".
`notifyMeasurementJobStatusChangedEmitted`	The notifyMeasurementJobStatusChanged is emitted.

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 (O)

7.3.3.1 Definition

This operation supports `IRPManager`'s request 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.

After the job has been suspended, the notifyFileReady or notifyFilePreparationError notification shall be emitted immediately or when the next reporting period is reached.

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` AND notifyThresholdMonitorStatusChangedEmitted AND `notifyMeasurementJobStatusChangedEmitted`.

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 notifyFileReady is emitted indicating the availability of the file(s). (c) No more reading of measurementType values. (d) The `MeasurementJob.jobStatus` is set to 'Suspended'. (e) For an IRPAgent using the measurement result data for threshold monitoring, the detection of the related performance alarm conditions is suspended.
`notifyThresholdMonitorStatusChangedEmitted`	The notifyThresholdMonitorStatusChanged is emitted, if the ThresholdMonitor behaviour is dependent on the availability of the measurement result.
`notifyMeasurementJobStatusChangedEmitted`	The notifyMeasurementJobStatusChanged is emitted.

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.
`jobAlreadySuspended`	**Condition:** (measurementJobIsNotSuspended) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.

7.3.4 Operation resumeMeasurementJob (O)

7.3.4.1 Definition

This operation supports `IRPManager`'s request 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` operation.

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` AND `sufficientCapacity`.

Assertion Name	Definition
`measurementJobExists`	`MeasurementJob` specified by the input parameter exists.
`measurementJobIsSuspended`	`MeasurementJob` specified by the input parameter is suspended.
`sufficientCapacity`	`The resource of EM and NE can support the requested measurement.`

7.3.4.5 Post-condition

`measurementJobIsResumed` AND `notifyMeasurementJobStatusChangedEmitted`.

Assertion Name	Definition
`measurementJobIsResumed`	The `MeasurementJob` identified by MeasurementJobId is resumed., It means the following. This resumed `MeasurementJob` shall behave as it did when it was first created by `createMeasurementJob`. Depending on the current time and the `JobMeasurementSchedule.startTime`, the `MeasurementJob.jobStatus` will be set accordingly.
`notifyMeasurementJobStatusChangedEmitted`	The notifyMeasurementJobStatusChanged is emitted.

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.
`highWorkLoad`	**Condition:** sufficientCapacity is false. **Returned Information:** Name of the exception and the detailed reason which is one of: emCpuBusy; emHDShortage, emLowMemory, {neCpuBusy, neObjectInstList}, {neHDShortage neObjectInstList}, {neLowMemory, neObjectInstList}, maxJobReached, otherReason; status is set to 'Failure'. **Exit state:** Entry state.

7.3.5 Operation listMeasurementJobs (M)

7.3.5.1 Definition

This operation supports IRPManager's request to list the information of all or a set of specified current MeasurementJobs.

7.3.5.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
jobIdList	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< attributes of MeasurementJob, attributes of JobMeasurementSchedule, (refer to definition and legal values in clause 6.5.1) iOCName, iOCInstanceList, measurementCategoryList (refer to input parameter of createMeasurementJob in clause 7.3.1.2) >	Returned information of corresponding MeasurementJobs matching the input criteria. If no match, then the length of the jobInfolist will be 0 (with status == Success). Jobs are created by successful invocation of createMeasurementJob (see 7.3.1.2). If the job is created using non-empty iOCInstanceList in createMeasurementJob, then iOCInstanceList here shall contain the DNs of the supported IOC instances. If the job is created using empty iOCInstanceList, then iOCInstanceList here shall be empty as well. This case is valid only if IRPAgent support the use of empty list. (see clause 7.3.1.2)
status	M	ENUM (Success, Failure)	An operation may fail because of a specified or unspecified reason.

7.3.5.4 Pre-condition

jobIdListIsValid.

Assertion Name	Definition
jobIdListIsValid	jobIdList 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
invalidJobIdList	**Condition:** (jobIdListIsValid) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.

7.4 PMIRPOperations_2 Interface (O)

7.4.1 Operation createThresholdMonitor (M)

7.4.1.1 Definition

This operation supports IRPManager's request to create a ThresholdMonitor that defines the thresholds for some specific measurementTypes. If the threshold defined is (a) crossed or (b) reached, the related performance alarms 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 (a) threshold-crossing or (b) threshold reaching or for clearing if that measurementType is not already subject to a MeasurementJob monitoring. This kind of PMIRP will only determine the (a) threshold crossing or (b) threshold reaching 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). ThresholdMonitor is independent of any existing MeasurementJob(s) and no new MeasurementJob(s) are created for threshold monitoring purposes.

(See 6.3.8.1 for clarification of the alternatives (a) and (b))

7.4.1.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
iOCName	M	ManagedEntity.objectClass	It specifies the ManagedEntity class name of the instances specified by iOCInstanceList.
iOCInstanceList	M	List of <ManagedEntity.objectInstance>	It specifies the DNs of ManagedEntity instances whose measurementTypes are to be monitored. At least, one instance shall be specified.
thresholdInfoList	M	List of < MeasuredAttribute.measurementTypeName, MeasurementReader.probableCause, MeasurementReader .specificProblem, MeasurementReader .direction, thresholdPack > where thresholdPack is list of thresholdPackElement. The thresholdPackElement is STRUCT < ThresholdLevel.thresholdValue, ThresholdLevel.thresholdSeverity, ThresholdLevel.hysteresis, >	It specifies the thresholds to specific measurementTypes. See clause 6.5.1 (definition and legal values). The number of ThresholdLevel in thresholdLevelList must be 1, 2, 3 or 4.
monitorGranularityPeriod	M	Monitor.monitorGranularityPeriod	See clause 6.5.1 (definition and legal values).

7.4.1.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
monitorId	M	ThresholdMonitor.monitorId.	It specifies the unique identifier of the ThresholdMonitor in the PMIRP Agent.
unsupportedList	M	List of < ManagedEntity.objectClass, ManagedEntity.objectInstance, MeasuredAttribute. measurementTypeName, reason >	To create a ThresholdMonitor, best-effort is required. This parameter identifies the unsupported but requested measurementType(s). The reason can be: (a) The PMIRP has trouble starting monitoring the threshold of this measurementType. (b) The measurementType is illegal. (c) The measurementType 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 can be monitored for thresholding.
validDirection	The direction is valid.
validNumberOfThresholdPackElements	If PMIRP supports multi-level thresholdSeverity, it is valid if the number of elements in thresholdPack is 1, 2, 3 or 4. If PMIRP does not support multi-level thresholdSeverity, it is valid if the number is 1.
validOrderOfThresholdPackElements	This assertion is applicable if PMIRP supports multi-level thresholdSeverity. It is valid if it satisfies the constraint defined by multiLevelSeverityOrder of clause 6.5.2 (constrains).

7.4.1.5 Post-condition

thresholdMonitorIsCreated AND notifyObjectCreationEmitted.

Assertion Name	Definition
thresholdMonitorIsCreated	The ThresholdMonitor identified by monitorId exists. And the performance alarm shall be emitted if the thresholds are (a) crossed or (b) reached and the thresholdMonitorStatus is "Active". (See 6.3.8.1 for clarification of the alternatives (a) and (b)) The ThresholdMonitor.thresholdMonitorStatus will be Active.
notifyObjectCreationEmitted.	The notifyObjectCreation is emitted to indicate the creation of a ThresholdMonitor.

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.
invalidNumberOfThresholdPackElements	**Condition:** 4(validNumberOfThresholdPackElements) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.
invalidOrderOfThresholdPackElements	**Condition:** (validOrderOfThresholdPackElements) 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

This operation supports `IRPManager`'s request to delete a specified `ThresholdMonitor`.

At the time of the removal, all outstanding (a) threshold-crossing or (b) threshold reaching alarms will stay (i.e. the `FMIRP` Agent's `AlarmList` will contain an `AlarmInformation` indicating (a) threshold-crossing or (b) threshold reaching). The `IRPManager` needs to use other means to remove the `AlarmInformation` in the `FMIRP` `AlarmList`. (See 6.3.8.1 for clarification of the alternatives (a) and (b))

7.4.2.2 Input parameters

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

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 in the input parameters exists.

7.4.2.5 Post-condition

`thresholdMonitorIsDeleted` AND `notifyObjectDeletionEmitted`..

Assertion Name	Definition
`thresholdMonitorIsDeleted`	`ThresholdMonitor` identified by monitorId ceases to exist.
`notifyObjectDeletionEmitted`	The notifyObjectDeletion is emitted to indicate the deletion of a ThresholdMonitor.

7.4.2.6 Exceptions

Exception Name	Definition
`unknownThresholdMonitor`	**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 supports `IRPManager`'s request to list detailed information about all or specified `ThresholdMonitors`.

7.4.3.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
`monitorIdList`	M	List of <`ThresholdMonitor.monitorId`>	This parameter specifies the `ThresholdMonitors` to be listed. If the parameter specifies the list of monitorIds, then the corresponding information of `ThresholdMonitors` will be returned. If the parameter contains no information, all the `ThresholdMonitors` in the `PMIRP` Agent will be returned.

7.4.3.3 Output parameters

Parameter Name	Qualifier	Matching Information	Comment
`monitorInfoList`	M	List of < attributes of `ThresholdMonitor` (see definition and legal values in clause 6.5.1), iOCName, iOCInstanceList, thresholdInfoList (See clause 7.4.1.2 input parameters of create thresholdMonitor operation) >	Returned information of corresponding `ThresholdMonitors` matching the input criteria. If no match, then the length of the monitorInfolist will be 0 (with status == Success).
`status`	M	ENUM (Success, Failure)	An operation may fail because of a specified or unspecified reason.

7.4.3.4 Pre-condition

`monitorIdListIsValid`.

Assertion Name	Definition
`monitorIdListIsValid`	monitorIdList 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
`invalidMonitorIdList`	**Condition:** (monitorIdListIsValid) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.

7.5 PMIRPOperations_3 Interface (O)

7.5.1 Operation suspendThresholdMonitor (M)

7.5.1.1 Definition

This operation supports `IRPManager`'s request to suspend the `ThresholdMonitor`. If the operation succeed, its `thresholdMonitorStatus` shall be set to "Suspended". `PMIRP` shall not emit performance alarms related to this `ThresholdMonitor`. The `ThresholdMonitor` shall still exist.

7.5.1.2 Input parameters

Parameter Name	Qualifier	Information type	Comment
`monitorId`	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` AND `notifyThresholdMonitorStatusChangedEmitted`.

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'.
`notifyThresholdMonitorStatusChangedEmitted`	The notifyThresholdMonitorStatusChanged is emitted.

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.
`thresholdMonitorAlreadySuspended`	**Condition:** (thresholdMonitorIsNotSuspended) not verified. **Returned information:** output parameter status is set to 'Failure'. **Exit state:** Entry State.

7.5.2 Operation resumeThresholdMonitor (M)

7.5.2.1 Definition

This operation supports `IRPManager`'s request to resume a suspended `ThresholdMonitor.`

7.5.2.2 Input parameters

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

7.5.2.3 Output parameters

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

7.5.2.4 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.

7.5.2.5 Post-condition

`thresholdMonitorIsResumed` AND `notifyThresholdMonitorStatusChangedEmitted`.

Assertion Name	Definition
`thresholdMonitorIsResumed.`	The `ThresholdMonitor` identified by monitorId is resumed, that means it will work again. The `ThresholdMonitor.thresholdMonitorStatus` will be "Active'.
`notifyThresholdMonitorStatusChangedEmitted`	The notifyThresholdMonitorStatusChanged is emitted.

7.5.2.6 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.

7.6 PMIRPNotification_1 Interface (M)

7.6.1 notifyMeasurementJobStatusChanged (M)

7.6.1.1 Definition

The PMIRP Agent notifies all subscribed IRPManagers about the status changes of a MeasurementJob. The status changes in that case include Suspended=>Scheduled, Active=>Suspended, Scheduled=>Suspended, Suspended=>Active, Scheduled=>Active, Active=>Stopped, Suspended=>Stopped, Scheduled=>Stopped.

7.6.1.2 Input parameters

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M, Y	PMIRP.objectClass	Notification header - see [12]. It shall carry the PMIRP class name.
objectInstance	M, Y	PMIRP.objectInstance	Notification header - see [12]. It shall carry the DN of the PMIRP.
eventTime	M, Y	--	Notification header - see [12].
notificationType	M, Y	"notifyMeasurementJobStatusChanged"	Notification header - see [12].
systemDN	C, Y	--	Notification header - see [12].
notificationId	M, N	--	Notification header - see [12].
jobId	M, Y	MeasurementJob.jobId	
jobStatus	M, N	MeasurementJob.jobStatus	The new status of the MeasurementJob.
reason	O, N	--	It carries one or several of the assertion names of the From-state of Triggering Event.

7.6.1.3 Triggering Event

7.6.1.3.1 From-state

failToReadMeasurementTypesForExtendedProlongPeriod OR internalProblem OR stopMeasurementJob OR stopTimeReached OR resumeMeasurementJob OR suspendMeasurementJob OR startTimeReached OR suspendMeasurementJobBySystem

Assertion Name	Definition
failToReadMeasurementTypesForExtendedProlongPeriod	Because the PMIRP Agent have failed to read the monitored measurementType (s) from managed resources for one or more times, the PMIRP Agent 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, PMIRP Agent decides that it no longer can maintain the MeasurementJob in any jobStatus but "Stopped".
stopMeasurementJob	The stopMeasurementJob returns success.
stopTimeReached	The stop time for MeasurementJob has been reached.
resumeMeasurementJob	The resumeMeasurementJob returns success.
suspendMeasurementJob	The suspendMeasurementJob returns success.
startTimeReached	The start time for MeasurementJob has been reached.
suspendMeasurementJobBySystem	The MeasurementJob has been suspended by the system in case of overload

7.6.1.3.2 To-state

`measurementJobStatusChanged`

Assertion Name	Definition
`measurementJobStatusChanged`	The `MeasurementJob.jobStatus` changed to a new value.

7.6.2 Void

7.7 PMIRPNotification_2 Interface (O)

7.7.1 notifyThresholdMonitorObjectCreation (M)

7.7.1.1 Definition

IRPAgent notifies the subscribed IRPManager that a new Monitor instance has been created and that the notification satisfies the filter constraint expressed in IRPManager"s `subscribe` operation (see TS 32.302 [12]).

7.7.1.2 Input Parameters

Parameter Name	Qualifier	Matching Information	Comment
`objectClass`	M, Y	PMIRP.objectClass	See Table 7.6.1.2.
`objectInstance`	M, Y	PMIRP.objectInstance	See Table 7.6.1.2.
`notificationId`	M. N	--	See Table 7.6.1.2.
`eventTime`	M, Y	ManagedEntity.creationTime	See Table 7.6.1.2.
`systemDN`	C, Y	--	See Table 7.6.1.2.
`notificationType`	M, Y	'notifyThresholdMonitorObjectCreation'	Notification header - see [12].
`monitorId`	M, N	Monitor.monitored	See monitorId definition in clause 6.5.1.
`monitorGranularityPeriod`	M, N	Monitor.monitorGranularityPeriod	See monitorGranularityPeriod definition in clause 6.5.1.
`thresholdMonitorStatus`	M, N	ThresholdMonitor.thresholdMonitorStatus	See thresholdMonitorStatus definition in clause 6.5.1.

7.7.1.3 Triggering Event

7.7.1.3.1 From-state

`stateBeforeObjectCreation`.

Assertion Name	Definition
`stateBeforeObjectCreation`	The number of instances of the IOC ManagedEntity is equal to N.

7.7.1.3.2 To-state

`stateAfterObjectCreation`.

Assertion Name	Definition
`stateAfterObjectCreation`	The number of instances of the IOC ManagedEntity is equal to N + 1.

7.7.2 notifyThresholdMonitorObjectDeletion (M)

7.7.2.1 Definition

IRPAgent notifies the subscribed IRPManager of a deleted Monitor instance. The IRPAgent invokes this notification because the subject notification satisfies the filter constraint expressed in the IRPManager `subscribe` operation (see TS 32.302 [12]).

7.7.2.2 Input Parameters

Parameter Name	Qualifier	Matching Information	Comment
`objectClass`	M, Y	PMIRP.objectClass	See Table 7.6.1.2.
`objectInstance`	M, Y	PMIRP.objectInstance	See Table 7.6.1.2.
`notificationId`	M, N	--	See Table 7.6.1.2.
`eventTime`	M, Y	ManagedEntity.deletionTime	See Table 7.6.1.2.
`systemDN`	C, Y	--	See Table 7.6.1.2.
`notificationType`	M, Y	'notifyThresholdMonitorObjectDeletion'	Notification header - see [12].
`monitorId`	M, Y	Monitor.monitored	See monitorId definition in clause 6.5.1.

7.7.2.3 Triggering Event

7.7.2.3.1 From-state

`stateBeforeObjectDeletion`.

Assertion Name	Definition
`stateBeforeObjectDeletion`	The number of instances of the IOC ManagedEntity is equal to N.

7.7.2.3.2 To-state

`stateAfterObjectDeletion`.

Assertion Name	Definition
`stateAfterObjectDeletion`	The number of instances of the IOC ManagedEntity is equal to N - 1.

7.7.3 notifyThresholdMonitorStatusChanged (O)

7.7.3.1 Definition

The PMIRP Agent notifies all subscribed IRPManagers about the status changes of a ThresholdMonitor. The status changes in that case include Suspended=>Active, Active=>Suspended.

NOTE: The notifyThresholdMonitorStatusChanged notification is mandatory if PMIRPOperations_2 is supported.

7.7.3.2 Input Parameters

Parameter Name	Qualifier	Matching Information	Comment
objectClass	M, Y	PMIRP.objectClass	See Table 7.6.1.2.
objectInstance	M, Y	PMIRP.objectInstance	See Table 7.6.1.2.
eventTime	M, Y	--	See Table 7.6.1.2.
notificationType	M, Y	"notifyThresholdMonitorStatusChanged".	See Table 7.6.1.2.
systemDN	C, Y	--	See Table 7.6.1.2.
notificationId	M, N	--	See Table 7.6.1.2.
monitorId	M, Y	Monitor.monitorId	
monitorStatus	M, N	ThresholdMonitor.thresholdMonitorStatus	The new status of the ThresholdMonitor.
reason	O, N	--	It carries one or several of the assertion names of the From-state of Triggering Event.

7.7.3.3 Triggering Event

7.7.3.3.1 From-state

resumeThresholdMonitor OR suspendThresholdMonitor

Assertion Name	Definition
resumeThresholdMonitor	The resumeThresholdMonitor returns success.
suspendThresholdMonitor	The suspendThresholdMonitor returns success.

7.7.3.3.2 To-state

thresholdMonitorStatusChanged

Assertion Name	Definition
thresholdMonitorStatusChanged	The ThresholdMonitor.thresholdMonitorStatus changed to a new value.

8 Scenarios

8.1 createMeasurementJob

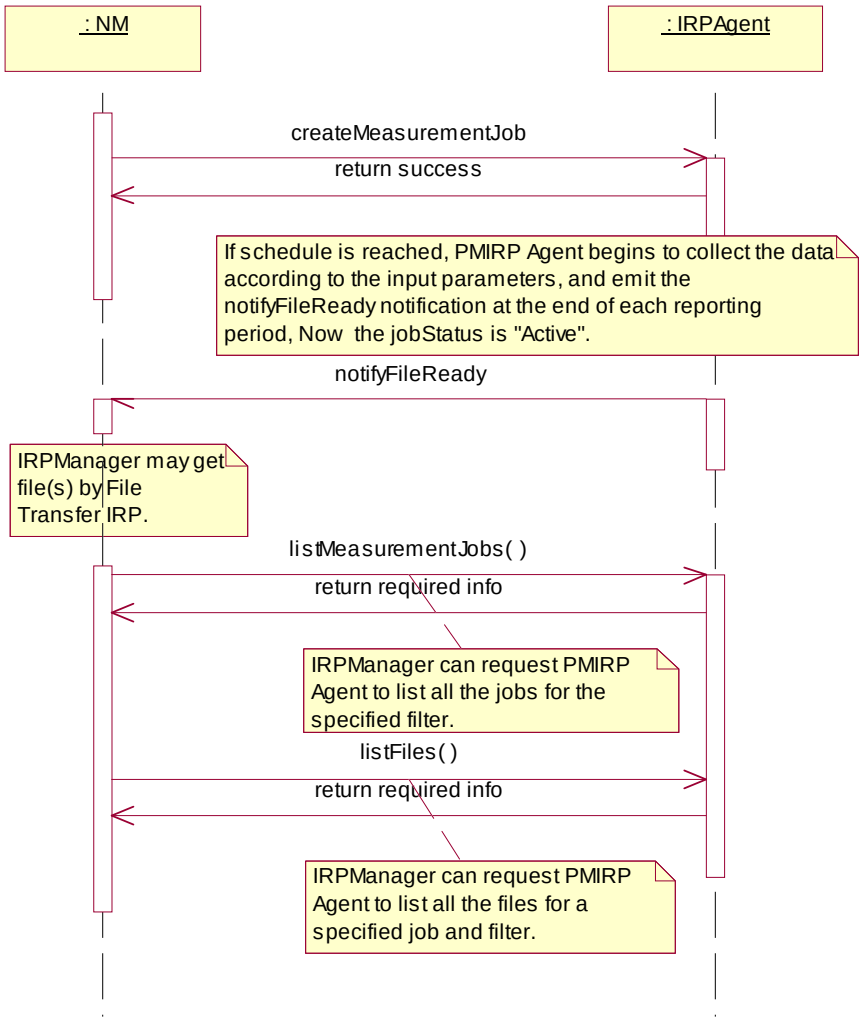


Figure 8.1: Scenario1 of Performance Management

8.2 stopMeasurementJob

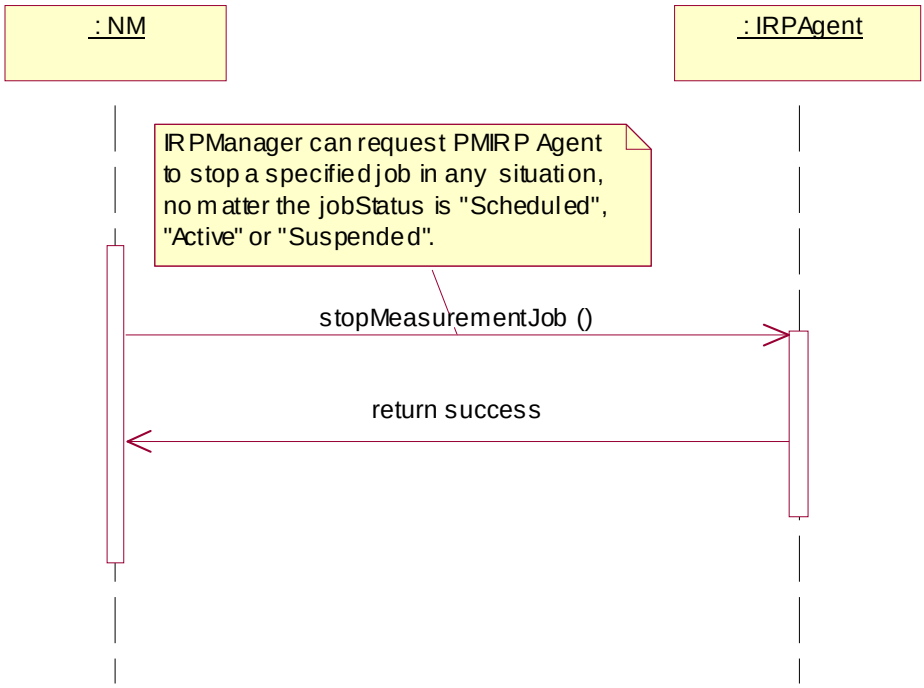


Figure 8.2: Scenario2 of Performance Management

8.3 stopMeasurementJob/listMeasurementJobs/listFiles

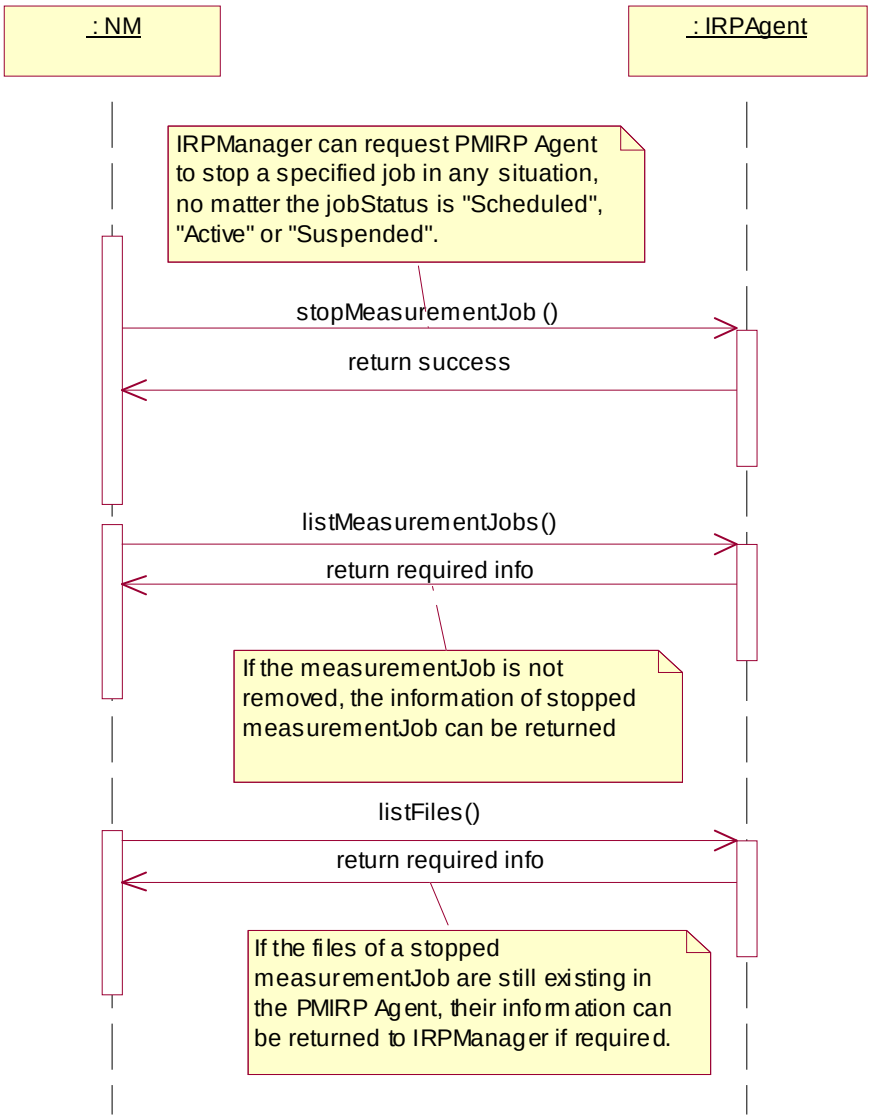


Figure 8.3: Scenario3 of Performance Management

8.4 suspendMeasurementJob/resumeMeasurementJob

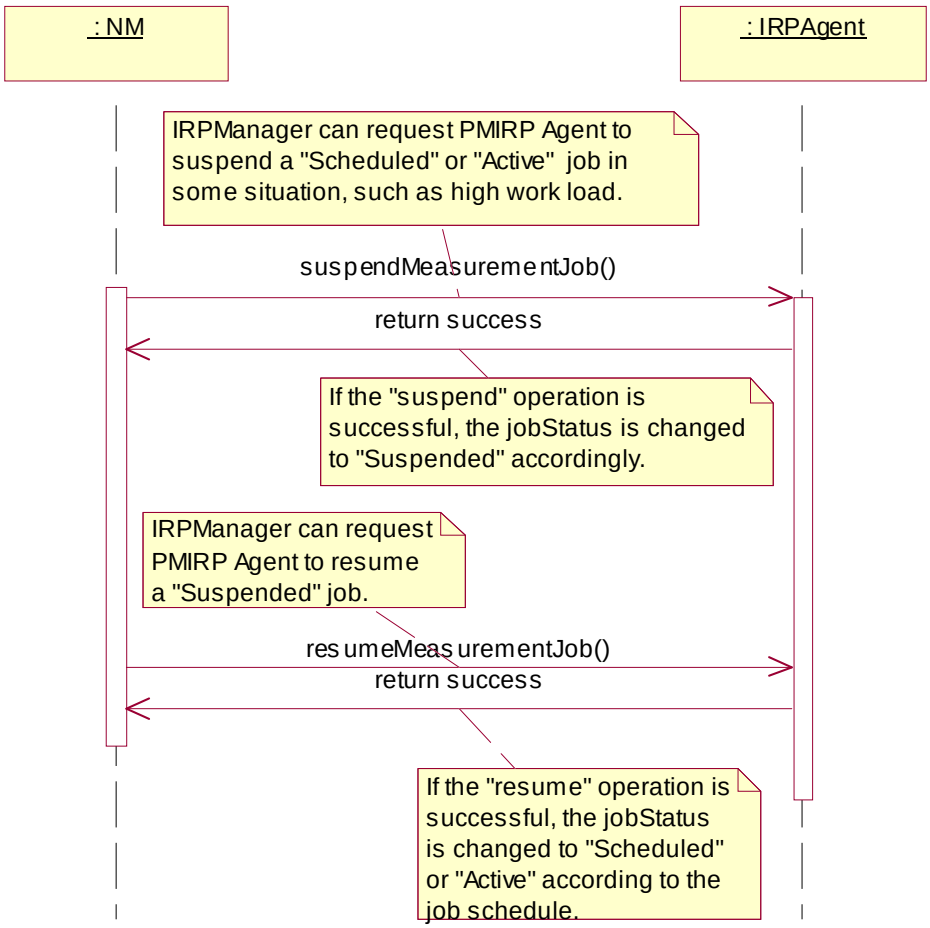


Figure 8.4: Scenario4 of Performance Management

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

A.1 Definition of state

A.1.1 Scheduled

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

A.1.2 Suspended

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

A.1.3 Active

"Active" means that the start time of a `MeasurementJob` has been reached.

There are two sub-states: "Busy" and "Idle".

"Busy" means that the `MeasurementJob` is monitoring the measurementTypes.

"Idle" means that the `MeasurementJob` is not monitoring the measurementTypes.

When a `MeasurementJob` becomes active, it does not mean that the job is immediately busy. `IRPManager` can set the detailed time frame (e.g. dailySchedule or weeklySchedule) for a `MeasurementJob` to be busy. If there is no time frame schedule for a job, it immediately monitors the measurement types when it becomes active.

A.1.4 Stopped

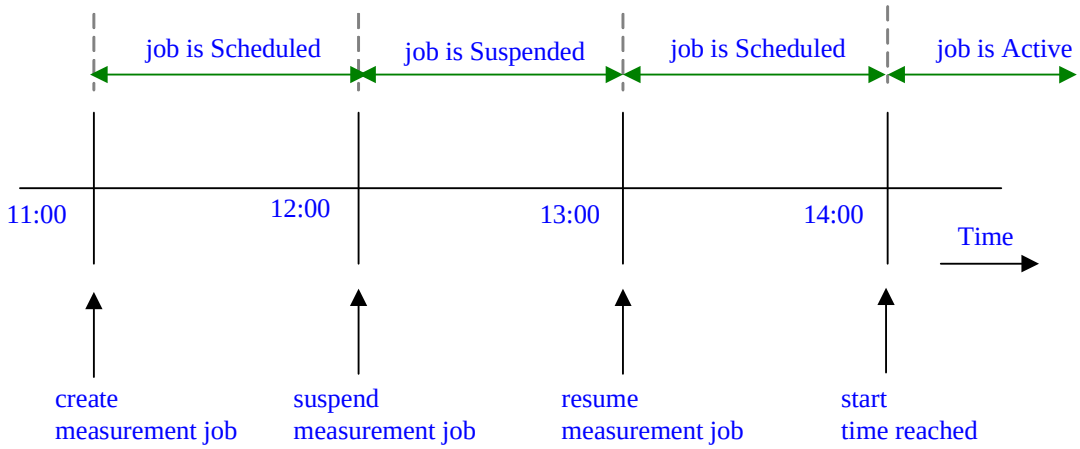
"Stopped" means that the `MeasurementJob` is deleted via the Itf-N. Whether the managed system removes the `MeasurementJob` immediately or later is vendor specific.

A.2 State transition scenarios

The following scenarios give examples of state transitions.

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.

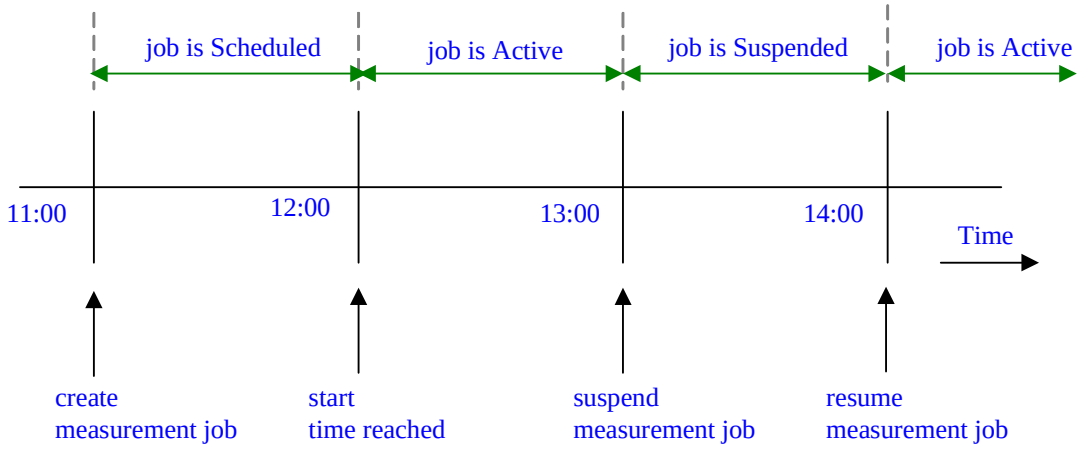


At 11:00: the `MeasurementJob` becomes scheduled after it was created.
At 12:00: the `MeasurementJob` is suspended and the jobStatus is "Suspended".
At 13:00: the `MeasurementJob` is resumed, and the jobStatus is "Scheduled" because the start time has not been reached.
At 14:00: the `MeasurementJob` becomes active because start time has been reached.

Figure A.2.1: 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.

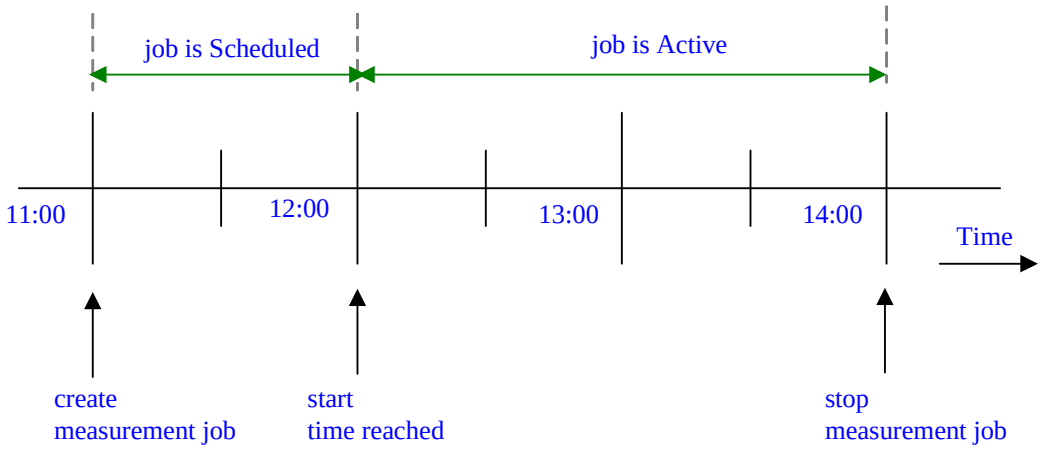


At 11:00: the `MeasurementJob` becomes scheduled after it was created.
At 12:00: the start time is reached, and the jobStatus is "Active".
At 13:00: the `MeasurementJob` is suspended, and the jobStatus is "Suspended".
At 14:00: the `MeasurementJob` is resumed, and the jobStatus is "Active".

Figure A.2.2: 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 at 14:00.

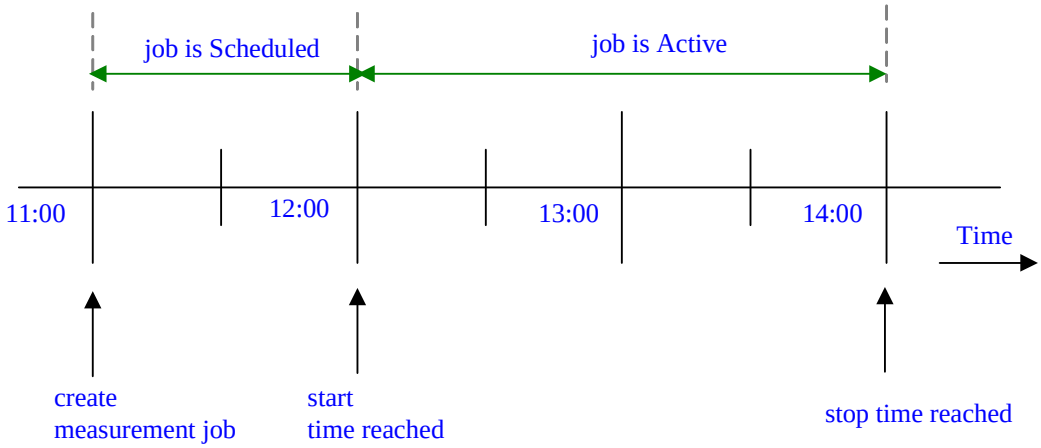


At 12:00: the `MeasurementJob` becomes active when start time is reached. Since there is no schedule for the job, it starts monitoring measurement types immediately.
At 14:00: the `MeasurementJob` is stopped when `IRPManager` stops the job.

Figure A.2.3: 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.

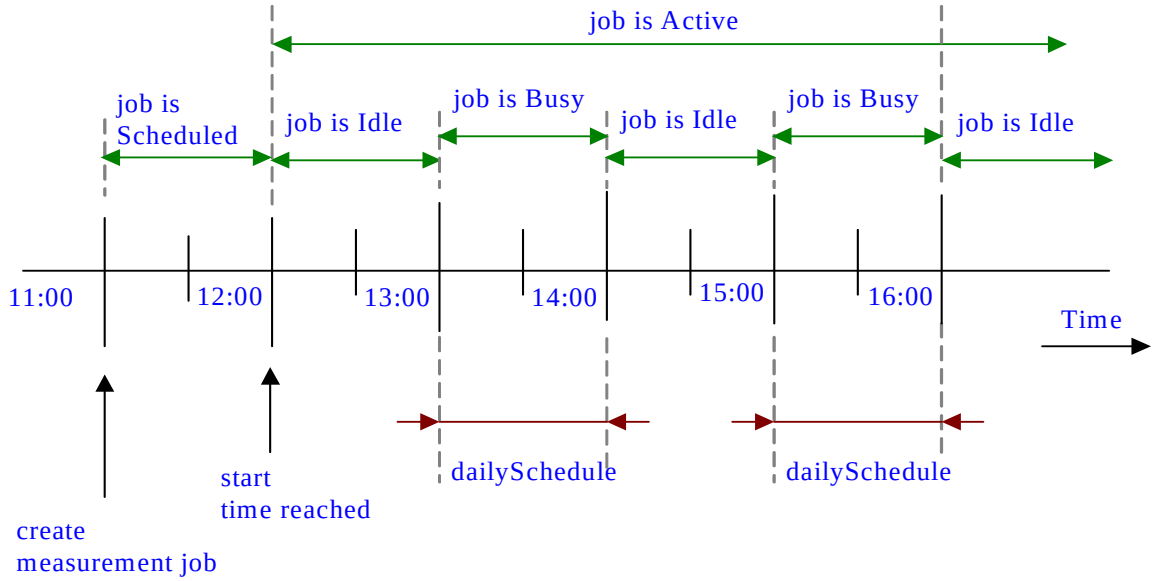


At 12:00: the `MeasurementJob` becomes active when start time is reached. Since there is no schedule for the job, it starts monitoring measurement types immediately.
At 14:00: the `MeasurementJob` is stopped automatically when stop time is reached.

Figure A.2.4: 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.



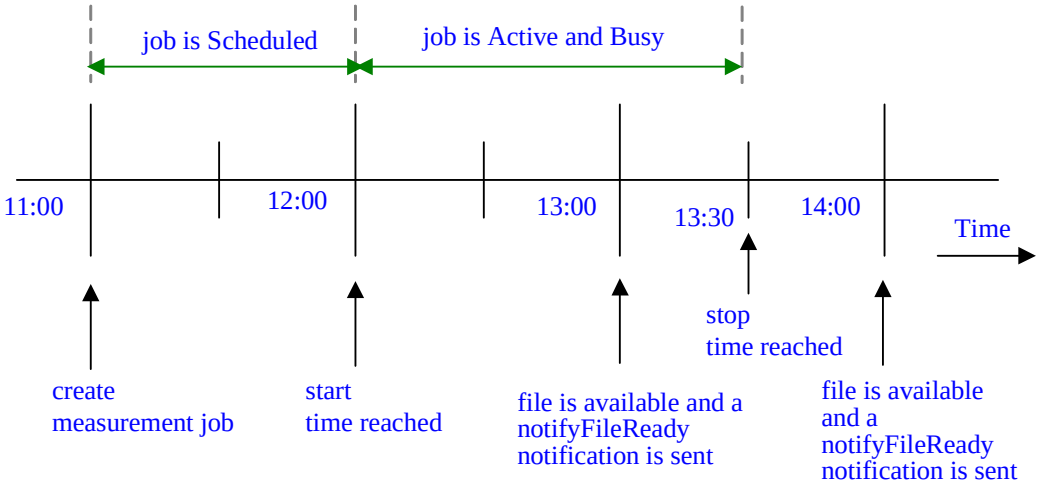
At 12:00: the `MeasurementJob` becomes active when start time has been reached. Since there is a schedule for the job, the job monitors measurement types only during the specified schedule. During other times, the job is idle.

At 13:00 and 15:00: the `MeasurementJob` becomes busy when the daily schedule has been reached.

Figure A.2.5: 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 is reached. 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 `notifyFileReady` notification to subscripted `IRPManager`.
At 13:30: the `MeasurementJob` is automatically stopped when stop time is reached.
At 14:00: `PMIRP` will emit the last `notifyFileReady` notification.

Figure A.2.6: Scenario 6

Annex B (normative): Threshold related performance alarms Triggering Events

This annex defines the Triggering Events for the various performance alarms. The Triggering Events normally are defined by each notification type, such as `notifyChangedAlarm`, under the clause "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, 3 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 `IRPAgent` that supports `notifyChangedAlarm`. Clause B.2 defines those for `IRPAgent` that does not support `notifyChangedAlarm`.

Clause B.3 provides samples of triggering events.

B.1 IRPAgent supporting notifyChangedAlarm

Table B.1

	To State-0	To State-1	To State-2	To State-3	To State-4
From State-0	no emission	1 notifyNewAlarm	1 notifyNewAlarm	1 notifyNewAlarm	1 notifyNewAlarm
From State-1	1 notifyClearedAlarm	no emission	1 notifyChangedAlarm	1 notifyChangedAlarm	1 notifyChangedAlarm
From State-2	1 notifyClearedAlarm	1 notifyChangedAlarm	no emission	1 notifyChangedAlarm	1 notifyChangedAlarm
From State-3	1 notifyClearedAlarm	1 notifyChangedAlarm	1 notifyChangedAlarm	no emission	1 notifyChangedAlarm
From State-4	1 notifyClearedAlarm	1 notifyChangedAlarm	1 notifyChangedAlarm	1 notifyChangedAlarm	no emission

The horizontal '0', '1', '2', '3' and '4' are the to-states indicating the current measurementType value with respect to the different threshold levels. Threshold levels are associated with severity level (thresholdSeverity). Higher threshold level associates with higher severity level (thresholdSeverity).

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", '3' and '4' are the from-states indicating the last-read measurementType value with respect to the threshold levels.

A threshold has a value and may have a hysteresis. A threshold with hysteresis has a threshold-high value and a threshold-low value that are different from the threshold value. A threshold without 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 (a) rising across or (b) reach 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 (a) falling across or (b) reach 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 performance alarm emission when from-state transits to to-state. The `notifyNewAlarm` and `notifyChangedAlarm` shall carry the severity level (perceivedSeverity) associated with the to-state (thresholdSeverity).

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

(See 6.3.8.1 for clarification of the alternatives (a) and (b))

B.2 IRPAgent not supporting notifyChangedAlarm

Table B.2

	To State-0	To State-1	To State-2	To State-3	To State-4
From State-0	no emission	1 notifyNewAlarm	1 notifyNewAlarm	1 notifyNewAlarm	1 notifyNewAlarm
From State-1	1 notifyClearedAlarm	no emission	1 notifyClearedAlarm 1 notifyNewAlarm	1 notifyClearedAlarm 1 notifyNewAlarm	1 notifyClearedAlarm 1 notifyNewAlarm
From State-2	1 notifyClearedAlarm	1 notifyClearedAlarm 1 notifyNewAlarm	no emission	1 notifyClearedAlarm 1 notifyNewAlarm	1 notifyClearedAlarm 1 notifyNewAlarm
From State-3	1 notifyClearedAlarm	1 notifyClearedAlarm 1 notifyNewAlarm	1 notifyClearedAlarm 1 notifyNewAlarm	no emission	1 notifyClearedAlarm 1 notifyNewAlarm
To State-4	1 notifyClearedAlarm	1 notifyClearedAlarm 1 notifyNewAlarm	1 notifyClearedAlarm 1 notifyNewAlarm	1 notifyClearedAlarm 1 notifyNewAlarm	no emission

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

B.3 Examples

B.3.1 Example 1

Figure B.3.1 illustrates multi-level thresholdSeverity thresholding behaviour. The horizontal axis indicates the time intervals specified by the `Monitor.monitorGranularityPeriod` (monitorGP). The vertical axis indicates the monitored measurementType values. The measurementType of this example is of type gauge. The `Monitor.direction` is set to "Increasing".

Further suppose that the measurementType is monitored for a (a) two-level-severity-threshold-crossings or (b) two-level-severity-threshold reaching. (See 6.3.8.1 for clarification of the alternatives (a) and (b))

The threshold for the lower-level-severity is set to 3 with a hysteresis low and high levels set to 2 and 4 respectively (i.e. hysteresis value = 1). The thresholdSeverity 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 (i.e. hysteresis value = 1). The thresholdSeverity level is Major.

The performance alarm triggering events are:

1. At monitorGP interval 3, trigger a `notifyNewAlarm` with perceivedSeverity (ps) = Minor.
2. At monitorGP interval 5, if the `IRPAgent` supports `notifyChangedAlarm`, trigger a `notifyChangedAlarm` with ps = Major; else a `notifyClearAlarm` and a `notifyNewAlarm` with ps = Major.
3. At monitorGP interval 8, if the `IRPAgent` supports `notifyChangedAlarm`, trigger a `notifyChangedAlarm` with ps = Minor; else a `notifyClearAlarm` and a `notifyNewAlarm` with ps = Minor.
4. At monitorGP interval 15, trigger a `notifyClearedAlarm` clearing the last emitted Minor alarm.

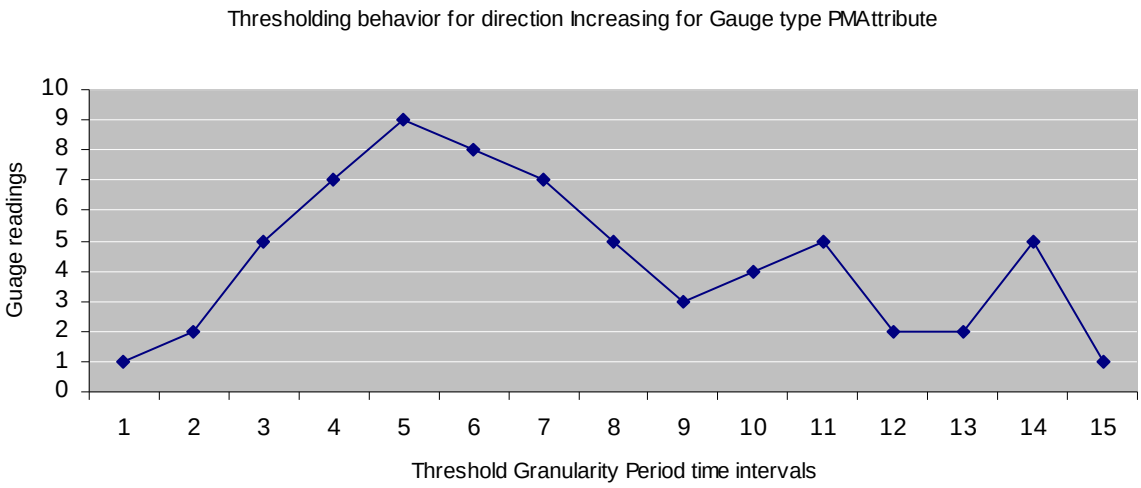


Figure B.3.1

B.3.2 Example 2

Figure B.3.2 illustrates another multi-level thresholdSeverity thresholding behaviour for `IRPAgent` supporting `notifyChangedAlarm`. 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 `Monitor.direction` is set to "Decreasing".

Further suppose that the measurementType is monitored for a two-level thresholdSeverity (a) threshold-crossing or (b) threshold reaching. (See 6.3.8.1 for clarification of the alternatives (a) and (b))

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

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

The performance alarm triggering events are:

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

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

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

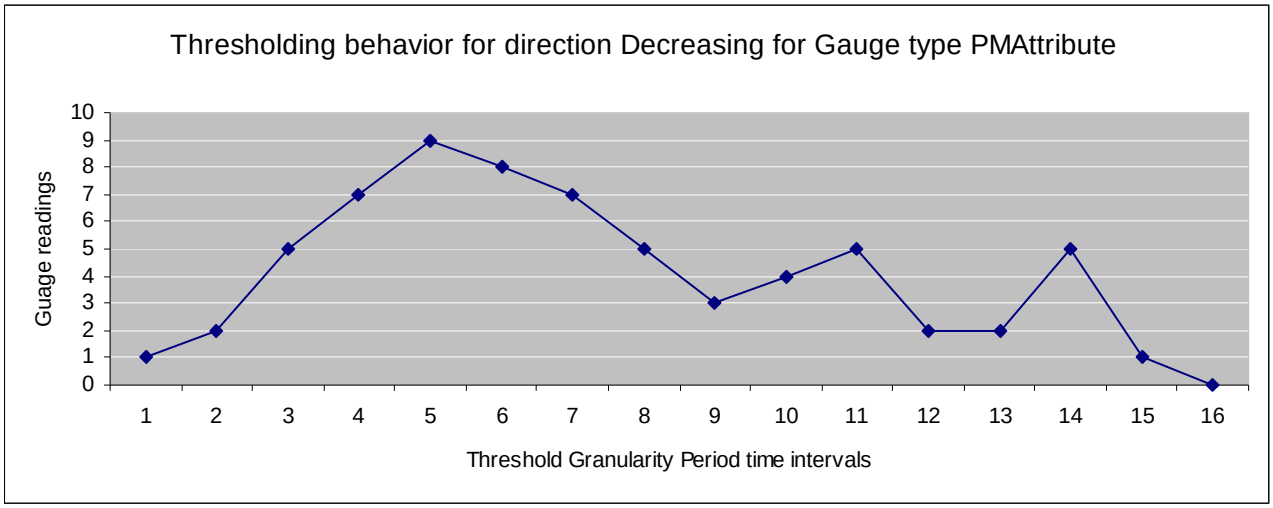


Figure B.3.2

B.3.3 Example 3

Figure B.3.3 illustrates a single-level thresholding behaviour. The horizontal axis indicates the time intervals specified by the monitorGP. The vertical axis indicates the monitored measurementType values. The measurementType of this example is of type gauge. The `Monitor.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 thresholdSeverity level is Critical.

The performance alarm triggering events are:

1. At monitorGP interval 3, trigger a `notifyNewAlarm` with ps = Critical.
2. At monitorGP interval 11, trigger a `notifyClearedAlarm` with ps = Cleared.

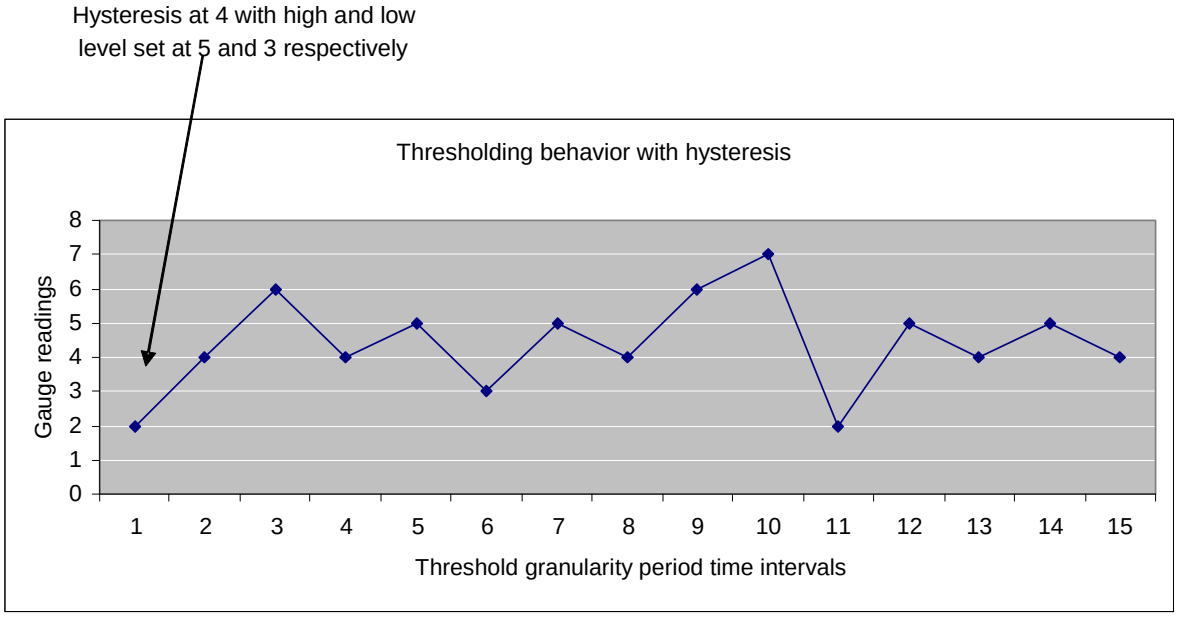


Figure B.3.3

Annex C (informative): Change history

Change history								
Date	**TSG #**	**TSG Doc.**	**CR**	**Rev**	**Subject/Comment**	**Cat**	**Old**	**New**
Jun 2003	SA_20	SP-030295	--	--	Submitted to TSG SA#20 for Information	--	1.0.0	
Dec 2003	SA_22	SP-030650	--	--	Submitted to TSG SA#22 for Approval	--	2.0.0	6.0.0
Jun 2004	SA_24	SP-040272	0001	--	Clarify and correct the specification of notifications of Monitor	F	6.0.0	6.1.0
Jun 2004	SA_24	SP-040272	0002	--	Add constraint that PM threshold hysteresis must be positive	F	6.0.0	6.1.0
Sep 2004	SA_25	SP-040558	0003	--	Add Measurement Job Overload Management function	B	6.1.0	6.2.0
Sep 2004	SA_25	SP-040557	0004	--	Align threshold alarm trigger to the definition in 32.411	F	6.1.0	6.2.0
Sep 2004	SA_25	SP-040556	0005	--	Extend the scope of ManagedEntity IOC to support collecting and monitoring measurement types related to vendor specific IOCs	B	6.1.0	6.2.0
Sep 2004	SA_25	SP-040556	0006	--	Add definition of post condition for operation suspendMeasurementJob	F	6.1.0	6.2.0
Dec 2004	SA_26	SP-040784	0007	--	Correct ambiguous precondition statement related to createThresholdMonitor operation	F	6.2.0	6.3.0
Dec 2004	SA_26	SP-040784	0008	--	Correct definition of ObjectClass and ObjectInstance in 'notifyMeasurementJobStatusChanged' and 'notifyThresholdMonitorStatusChanged'	F	6.2.0	6.3.0
Mar 2005	SA_27	SP-050041	0009	--	Remove the ambiguity that a PM IRP compliant system necessarily contains functionalities defined in Kernel CM IRP	F	6.3.0	6.4.0
Mar 2005	SA_27	SP-050041	0010	--	Apply the Generic System Context – Align with TS 32.150	F	6.3.0	6.4.0
Jun 2005	SA_28	SP-050293	0011	--	Correct the matching information of monitorId attribute	F	6.4.0	6.5.0
Mar 2006	SA_31	SP-060089	0012	--	Correct the misleading 'matching information' for object class and object instance in notifications	F	6.5.0	6.6.0
Jun 2006	SA_32	SP-060251	0013	--	Add in PM IRP IS the missing use of "empty list" to denote all known instances - Align with 32.401	F	6.6.0	6.7.0
Jun 2006	SA_32	SP-060251	0014	--	Correct parameter usage in notifyThresholdMonitorCreation	F	6.6.0	6.7.0
Sep 2006	SA_33	SP-060532	0015	--	Correct matching info of notificationType	F	6.7.0	6.8.0
Dec 2006	SA_34	SP-060710	0016	--	Add missing Notification Table for PMIRP	F	6.8.0	6.9.0
Dec 2006	SA_34	SP-060710	0017	--	Correction of the use of empty string in listMeasurementJob output parameter	F	6.8.0	6.9.0
Jun 2007	SA_36	--	--	--	Automatic upgrade to Rel-7 (no CR) at freeze of Rel-7. Deleted reference to CMIP SS, discontinued from R7 onwards.	--	6.3.0	7.0.0
Sep 2007	SA_37	SP-070613	0018	--	Align the definition of ManagedEntity in PM IRP with private objects classes defined in 32.404	F	7.0.0	7.1.0
Mar 2009	SA_43	SP-090207	0019	--	Include reference to SOAP Solution Set specification	D	7.1.0	8.0.0
Dec 2009	-	-	-	-	Update to Rel-9 version (MCC)	--	8.0.0	9.0.0
Jun 2010	SA_48	SP-100412	0020	-	Correct the information type of input parameter thresholdInfoList	F	9.0.0	9.1.0
Mar 2011	SA_51	SP-110095	0026	1	Add the missing exception in create measurement operation	F	9.1.0	10.0.0
2012-09	-	-	-	-	-	Update to Rel-11 version (MCC)	10.0.0	**11.0.0**
2012-12	SA_58	SP-120783	0027	1	Align usage of SupportIOC with repertoire and TS 32.152	F	11.0.0	**11.1.0**
2012-12	SA_58	SP-120783	0028	-	add reference to ITU-T Rec	F	11.0.0	**11.1.0**
2014-10	-	-	-	-	-	Update to Rel-12 version	11.1.0	**12.0.0**

						(MCC)		

History

Document history		
V12.0.0	October 2014	Publication