



INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Q.822

(04/94)

**SPECIFICATIONS OF
SIGNALLING SYSTEM No.7**

**STAGE 1, STAGE 2 AND STAGE 3
DESCRIPTION FOR THE Q3 INTERFACE –
PERFORMANCE MANAGEMENT**

ITU-T Recommendation Q.822

(Previously "CCITT Recommendation")

FOREWORD

The ITU-T (Telecommunication Standardization Sector) is a permanent organ of the International Telecommunication Union (ITU). The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1 (Helsinki, March 1-12, 1993).

ITU-T Recommendation Q.822 was prepared by ITU-T Study Group 11 (1993-1996) and was approved under the WTSC Resolution No. 1 procedure on the 7th of April 1994.

NOTE

In this Recommendation, the expression “Administration” is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

© ITU 1994

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

CONTENTS

	<i>Page</i>
1 Scope, purpose, and field of application	1
1.1 Scope	1
1.2 Purpose	1
1.3 Field of application	1
2 References	1
2.1 Recommendations referred to in this Recommendation	1
3 Terms and definitions, abbreviations	2
3.1 Terms and definitions	2
3.2 Abbreviations	3
4 Performance management	4
4.1 Performance management functions	4
4.1.1 Performance management data collection functions	4
4.1.2 Performance management data storage functions	4
4.1.3 Performance management thresholding functions	5
4.1.4 Performance management data reporting functions	5
4.2 Performance management model	5
4.3 Performance management information	7
4.3.1 Managed object classes	7
4.3.1.1 Current data	9
4.3.1.2 History data	10
4.3.1.3 Threshold data	10
4.3.2 Package definitions	11
4.3.2.1 Counter threshold list package	11
4.3.2.2 Filter suppression package	11
4.3.2.3 Gauge threshold list package	11
4.3.2.4 History data measurement list package	12
4.3.2.5 History data suspect interval flag package	12
4.3.2.6 History retention package	12
4.3.2.7 Maximum number of suppressed intervals package	12
4.3.2.8 Measurement list package	13
4.3.2.9 Number of suppressed intervals package	13
4.3.2.10 Object delete notification package	13
4.3.2.11 Observed managed object package	13
4.3.2.12 Scheduled performance monitoring report package	14
4.3.2.13 Threshold package	14
4.3.2.14 Threshold verification package	15
4.3.2.15 Zero suppression package	16
4.3.3 Attribute definitions	16
4.3.3.1 Counter threshold attribute list	16
4.3.3.2 Elapsed time	16
4.3.3.3 Gauge threshold attribute list	16
4.3.3.4 History data Id	17
4.3.3.5 History retention	17

	<i>Page</i>
4.3.3.6	Maximum suppressed intervals..... 17
4.3.3.7	Measurement list..... 18
4.3.3.8	Monitored entity types 18
4.3.3.9	Number of suppressed intervals 18
4.3.3.10	Observed object class..... 18
4.3.3.11	Report all attributes 18
4.3.3.12	Period end time 19
4.3.3.13	Suppress additional thresholds..... 19
4.3.3.14	Suspect interval flag..... 19
4.3.3.15	Threshold data Id 20
4.3.3.16	Threshold data instance..... 20
4.3.4	Name binding definitions..... 20
4.3.5	Abstract syntax productions..... 21
4.3.6	Support objects from other CCITT Recommendations..... 22
4.3.6.1	Support objects from the summarization function 22
4.3.6.2	Support objects from definition of management information..... 22
4.3.7	Performance management control..... 22
4.4	Performance management service definition..... 23
4.5	Functional unit definitions..... 23
5	Protocol..... 23
5.1	Abstract syntax 23
5.1.1	Managed objects 23
5.1.1.1	Referenced managed objects..... 23
5.1.1.2	Defined managed objects 23
5.1.2	Attributes..... 23
5.1.2.1	Attributes imported from the metric objects and attributes function 23
5.1.2.2	Attributes imported from the summarization function..... 24
5.1.2.3	Attributes imported from definition of management information..... 24
5.1.2.4	Attributes defined in this Recommendation..... 24
5.1.3	Notifications..... 24
5.1.3.1	Referenced notifications 24
5.2	Negotiation of functional units 25
6	Relationship with other Recommendations..... 25
7	Conformance 25
7.1	General conformance class requirements 25
7.1.1	Static conformance..... 25
7.1.2	Dynamic conformance 26
7.2	Dependent conformance class requirements..... 26
7.2.1	Static conformance..... 26
7.2.2	Dynamic conformance 26
7.3	Conformance to support managed object definitions 26
Annex A	Performance monitoring parameters 27
A.1	Abbreviations..... 27
A.2	Performance monitoring parameters attribute templates 28
A.2.1	Alarm indication signal second (AISS)..... 28
A.2.2	Alarm indication signal second far end (AISSFE)..... 28
A.2.3	Controlled slip second (CSS)..... 29
A.2.4	Controlled slip second far end (CSSFE) 29
A.2.5	Code Violation (CV)..... 29
A.2.6	Code violation, CP-bit parity (CVCP) 29
A.2.7	Code violation, CP-bit parity far end (CVCPFE) 29
A.2.8	Code violation, cyclic redundancy check (CVCRC) 29
A.2.9	Code violation, cyclic redundancy check far end (CVCRCFE)..... 30
A.2.10	Code violation far end (CVFE)..... 30
A.2.11	Code violation, P-bit (CVP)..... 30
A.2.12	Errored second (ES) 30

A.2.13	Errored second, type A (ESA)	30
A.2.14	Errored second, type A, CP-bit (ESACP)	31
A.2.15	Errored second, type A, CP-bit far end (ESACPF)	31
A.2.16	Errored second, type A, CRC (ESACRC)	31
A.2.17	Errored second, type A, CRC far end (ESACRCF).....	31
A.2.18	Errored second, type A, far end (ESAFE)	31
A.2.19	Errored second, type A, P-bit (ESAP)	32
A.2.20	Errored second, type B (ESB).....	32
A.2.21	Errored second, type B, CP-bit (ESBCP)	32
A.2.22	Errored second, type B, CP-bit far end (ESBCPF).....	32
A.2.23	Errored second, type B, CRC (ESBCRC)	32
A.2.24	Errored second, type B, CRC far end (ESBCRCF)	33
A.2.25	Errored second, type B, far end (ESBF)	33
A.2.26	Errored second, type B, P-bit (ESBP).....	33
A.2.27	Errored second, CP-bit parity (ESCP)	33
A.2.28	Errored second, CP-bit parity far end (ESCPF).....	33
A.2.29	Errored second, cyclic redundancy check (ESCRC).....	34
A.2.30	Errored second, cyclic redundancy check far end (ESCRCF).....	34
A.2.31	Errored second far end (ESFE)	34
A.2.32	Errored second, P-bit (ESP)	34
A.2.33	Loss of signal second (LOSS).....	34
A.2.34	Protection switching count (PSC)	35
A.2.35	Protection switching duration (PSD)	35
A.2.36	SEF/AIS severely errored framing alarm indication signal second (SAS)	35
A.2.37	SEF/AIS second, far end (SASF)	35
A.2.38	Severely errored second (SES)	35
A.2.39	Severely errored second, CP-bit (SESCP)	36
A.2.40	Severely errored second, CP-bit far end (SESCPF)	36
A.2.41	Severely errored second, CRC (SESCRC).....	36
A.2.42	Severely errored second, CRC far end (SESCRCF).....	36
A.2.43	Severely errored second far end (SESFE).....	36
A.2.44	Severely errored second-path (SESP)	37
A.2.45	Unavailable second (UAS)	37
A.2.46	Unavailable second far end (UASF).....	37
Annex B – Performance management information model overview.....		37
B.1	Introduction	37
B.2	Overview of the performance management information model	37
B.2.1	Collection of current PM data	38
B.2.2	History records	38
B.2.3	Scheduled report	39
B.2.4	Setting and reporting of PM thresholds	39
B.2.4.1	Thresholds applying to a single managed object	39
B.2.4.2	Thresholds applying to multiple managed objects	40
B.2.5	Historical retention of PM data	42
B.2.6	Scheduled summary or statistical reports.....	42
Annex C – PM model alternative		44
C.1	Performance management model.....	44
C.2	Performance management information.....	44
C.2.1	Difference scanner	44
C.2.2	Performance data component.....	45
C.3	Attribute definitions.....	46
C.3.1	Difference report list	46
C.4	Name binding definitions.....	46
C.5	Guidelines for the use of the model	46

ABSTRACT

This Recommendation provides a Stage 1, Stage 2 and Stage 3 Description for the Q3 Interface in a Telecommunication Management Network. Its focus is the parameter collection and storage, and the thresholding aspects of Performance Management. Included in this Description are specifications of the functions, management information, services, functional units, and protocols related to Performance Management. Significant re-use of OSI system Management specifications is described in the X.700-Series Recommendations.

This Recommendation may be applicable to other TMN-related interfaces.

KEYWORDS

Abstract syntax, attribute, function, functional unit, object class, parameter, performance management, performance monitoring, protocol, service.

STAGE 1, STAGE 2 AND STAGE 3 DESCRIPTION FOR THE Q3 INTERFACE – PERFORMANCE MANAGEMENT

(Geneva, 1994)

1 Scope, purpose, and field of application

1.1 Scope

This Recommendation provides a Stage 1, Stage 2 and Stage 3 Description (see Recommendation Q.68) for the parameter collection and thresholding aspects of Performance Management.

This Recommendation is part of a series of Recommendations that specify the Q3 interface (defined in Recommendation M.3010) requirements for communication:

- between an Operations System (OS) and a Network Element (NE);
- between an OS and a Mediation Device (MD);
- between an OS and a Q Adaptor (QA); and
- between Operations Systems in a Telecommunication Management Network (TMN).

1.2 Purpose

Current telecommunications networks are populated by a large and increasing number of OSs and NEs supplied by different vendors. Both the number and variety of networks and services have grown, creating a diversity of management needs. This growth has resulted in the proliferation of unique communication interfaces between OSs and NEs. The telecommunications industry stands to benefit from the standardization of these interfaces, designed to achieve interoperability between a broad range of OSs and NE/QAs using MDs where appropriate, and between OSs.

The primary purpose of this document is to provide a set of application messages and associated support objects for the parameter collection and thresholding aspects of Performance Management. Because of the desirability of providing common TMN solutions, these messages and support objects are expected to be applicable to other TMN or TMN-related interfaces.

1.3 Field of application

This Recommendation defines an information model based on the specification of Managed Object classes. These Managed Object classes are supported through the interactive services specified in ITU-T Recommendation Q.812.

2 References

The following Recommendations contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations are subject to revision, and parties to agreements based on this Recommendation are encouraged to investigate the possibility of applying the most recent edition of the Recommendations listed below.

2.1 Recommendations referred to in this Recommendation

- CCITT Recommendation M.3010, *Principles for a telecommunication management network (TMN)*.
- CCITT Recommendation M.3020, *TMN interface specification methodology*.

- CCITT Recommendation M.3100, *Generic network information model*.
- CCITT Recommendation M.3200, *TMN management services: overview*.
- ITU-T Recommendation Q.68, *Overview of methodology for developing management services*.
- ITU-T Recommendation Q.812, *Upper layer protocol profiles for the Q3 interface*.
- ITU-T Recommendation Q.821, *Stage 2 and Stage 3 description for the Q3 interface – Alarm surveillance*.
- CCITT Recommendation X.208, *Specification of Abstract Syntax Notation One (ASN.1)*.
- CCITT Recommendation X.209, *Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1)*.
- CCITT Recommendation X.216, *Presentation service definition for open systems interconnection for CCITT applications*.
- CCITT Recommendation X.701, *Systems management overview*.
- CCITT Recommendation X.710, *Management information service definition – Common management information service definition*.
- CCITT Recommendation X.711, *Management information protocol specification – Common management information protocol*.
- CCITT Recommendation X.721, *Definition of management information*.
- CCITT Recommendation X.722, *Guidelines for the definition of managed objects*.
- CCITT Recommendation X.730, *Object management function*.
- CCITT Recommendation X.731, *State management function*.
- CCITT Recommendation X.733, *Alarm reporting function*.
- CCITT Recommendation X.734, *Event report management function*.
- CCITT Recommendation X.735, *Log control function*.
- ITU-T Recommendation X.738, *Summarization function*.
- ITU-T Recommendation X.739, *Metric Objects and Attributes*.

3 Terms and definitions, abbreviations

3.1 Terms and definitions

This Recommendation makes use of the following terms defined in CCITT Recommendation M.3010:

- a) performance management;
- b) telecommunications management network (TMN).

This Recommendation makes use of the following terms defined in CCITT Recommendation M.3020:

TMN management function.

This Recommendation makes use of the following terms defined in CCITT Recommendation X.216:

application context.

This Recommendation makes use of the following terms defined in CCITT Recommendation X.700:

- a) managed object class;
- b) object instance.

This Recommendation makes use of the following terms defined in CCITT Recommendation X.701:

- a) agent;
- b) dependent conformance;
- c) general conformance;
- d) managed object class;
- e) managed system;
- f) management information;
- g) management support object;
- h) manager;
- i) managing system;
- j) notification.

This Recommendation makes use of the following terms defined in CCITT Recommendation X.710:
attribute.

This Recommendation makes use of the following terms defined in CCITT Recommendation X.720:

- a) inheritance;
- b) conditional package;
- c) name binding;
- d) package;
- e) relative distinguished name.

This Recommendation makes use of the following terms defined in CCITT Recommendation X.721:
alarm record (object class).

This Recommendation makes use of the following terms defined in CCITT Recommendation X.722:
template.

This Recommendation makes use of the following terms defined in CCITT Recommendation X.733:
Quality of service alarm (notification).

This Recommendation makes use of the following terms defined in CCITT Recommendation X.735:

- a) Log (managed object class);
- b) Log record (managed object class).

This Recommendation makes use of the following terms defined in ITU-T Recommendation X.738:

- a) Scan Report (notification);
- b) Statistical Report (notification).

This Recommendation makes use of the following terms defined in ITU-T Recommendation X.739:
Scanner (managed object class).

3.2 Abbreviations

ASN.1	Abstract Syntax Notation One
CCITT	International Telephone and Telegraph Consultative Committee
GDMO	Guidelines for the definition of managed objects

IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
MAPDU	Management Application Protocol Data Unit
MD	Mediation Device
NE	Network Element
OS	Operations System
OSI	Open Systems Interconnection
PM	Performance Management
PMN	Performance Monitoring
RDN	Relative Distinguished Name
QA	Q Adaptor
TMN	Telecommunications Management Network

4 Performance management

4.1 Performance management functions

The following clauses describe the TMN Management functions for Performance Management that are being considered in this Recommendation. These TMN Management functions are defined in accordance with the TMN interface specification methodology in Recommendation M.3020.

4.1.1 Performance management data collection functions

PM data collection refers to the ability for the NE to collect the various PM data relating to a single monitored entity in that NE. The following specific functions are associated with the collection activity:

- a) *Assign PM data collection interval* – The TMN manager instructs the NE about the duration of the PM data collection interval for a given entity or set of entities.
- b) *Suspend/Resume PM data collection* – The TMN manager instructs the NE to suspend/resume the performance monitoring data collection activity for a given monitored entity or set of monitored entities.
- c) *Reset PM data* – The TMN manager instructs the NE to reset the performance monitoring counters for a given monitored entity or set of monitored entities.
- d) *Schedule PM data collection* – The TMN manager instructs the NE to schedule the data collection activity for a given monitored entity or set of monitored entities to take place within specified time periods.

4.1.2 Performance management data storage functions

PM data storage refers to the optional capability for the NE to store historical PM data on each monitored entity for a prescribed time duration. The NE can also store summarized or statistical data derived from various monitored entities. The following specific functions are associated with the storage activity:

- a) *Assign PM history duration* – The TMN manager instructs the NE to establish the duration during which to maintain a specific record of PM historical data.

- b) *Screen PM data storage* – The TMN manager instructs the NE to originate historical data based on some screening criteria (e.g. suppress “all-zero” data).
- c) *Remove PM history data* – The TMN manager instructs the NE to remove specific historical PM data.

4.1.3 Performance management thresholding functions

PM thresholding refers to the ability for the NE to inform the TMN manager of any threshold crossing. It also provides the TMN manager with the means for establishing thresholding criteria. The following specific functions are associated with the thresholding activity:

- a) *Assign PM threshold* – The TMN manager instructs the NE to establish the thresholding criteria for PM data of a given monitored entity or set of monitored entities.
- b) *Report PM threshold violation* – The NE informs the TMN manager of a PM parameter threshold violation having occurred in a specific monitored entity.

4.1.4 Performance management data reporting functions

PM data reporting refers to the optional capability for the NE to report PM data on a scheduled basis, or as a result of a spontaneous request from the TMN manager. A report may contain data from a given monitored entity, or it can contain summarized data or data derived statistically from a set of monitored entities. The following specific functions are associated with the reporting activity:

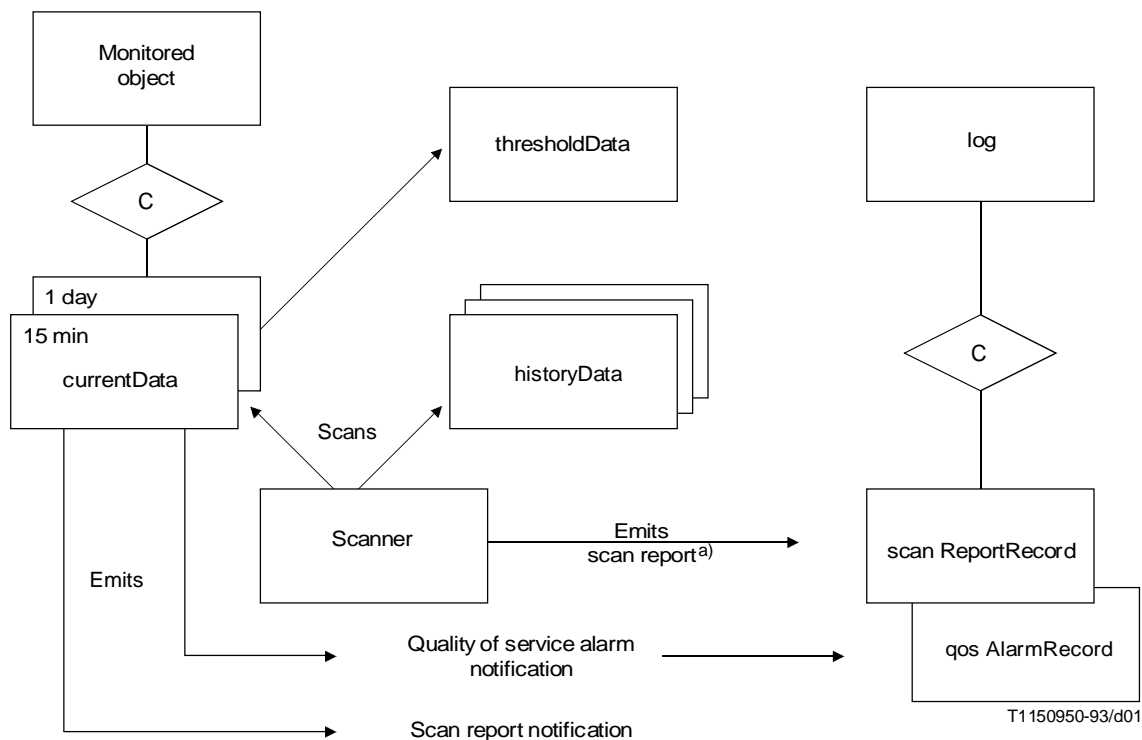
- a) *Request PM data* – The TMN manager issues a spontaneous request to the NE for current or historical PM data information on a given monitored entity or set of monitored entities.
- b) *Report PM data* – The NE issues a scheduled report to the TMN manager with current or historical PM data information from a given monitored entity or set of monitored entities.
- c) *Allow/Inhibit PM data reports* – The TMN manager allows/inhibits the emission of scheduled reports in the NE.
- d) *Screen PM data reports* – The TMN manager instructs the NE to issue PM data reports to the TMN manager based on some screening criteria (e.g. suppress “all-zero” data).

4.2 Performance management model

This subclause is intended to provide a conceptual framework for understanding how the various managed object classes are used to provide the performance monitoring services described in this Recommendation. A more detailed description of the model and examples of different possible configurations are provided in Annex B.

The object model for Performance Management is shown in Figure 1. Current PM data is collected for a Monitored Object by a *currentData* object class or its subclasses. Instances of the *currentData* object class or its subclasses are contained by the monitored object. At the end of each performance interval, the duration of which is determined by the *granularityPeriod* attribute, a summary report (*scanReport*) may be issued and a *historyData* object may be created to record the performance measurements for that interval. Thresholds may be established by use of the *thresholdData* object. When a threshold is violated by a performance measurement an alarm is emitted by the *currentData* object and logged as required. Performance measurements can be aggregated or statistically summarized by use of *Scanner* objects as defined in Recommendation X.738.

NOTE 1 – Existing system management Standards and Recommendations allow the PM attributes to be included within the monitored managed object (e.g. OSI Network and Transport layer objects in ISO/IEC 10733 and ISO/IEC 10737). The intention of this Recommendation is not to invalidate the continued use of such models. However, the model provided in this Recommendation encourages the specification of application-specific performance attributes within a separate managed object.



- a) The scan reports may include: an aggregation of measurements over time for a single monitored object, or across a number of monitored objects; an aggregation of measurements from a single interval across a number of monitored objects; or statistics for a single or multiple intervals across a single or multiple monitored objects.

FIGURE 1/Q.822

Objet model

The specific objects in the model are:

- *Monitored Object* – This object is the managed object for which the performance measurements are being collected. It represents the resource being measured (e.g. Trail Termination Point).
- *currentData Object* – This object contains the measurements for the resource being monitored for a specified time interval (e.g. 15 min.). In most cases the instantiated managed object will be an instance of a subclass of currentData. This subclass will have performance measurement attributes appropriate to the resource represented by the class of the monitored object (e.g. SDH performance measurements). At the end of each interval the currentData object may emit a scanReport notification which may result in a

corresponding Event Report being sent to a managing system (it is not mandatory that the discriminator construct in the Log be configured such that this notification is logged). Also, at the end of each interval a historyData object may be created containing the same attributes as the currentData object with values of the performance measurements at the end of the interval.

The currentData object may contain a pointer to a thresholdData object. If any of the thresholds (defined in the referenced thresholdData object) are violated a Quality of Service (QOS) alarm notification is emitted by the currentData object. The resulting alarm record may be logged.

The generic currentData object class shall not be used for technology specific interfaces where standardized technology specific subclasses of currentData exist.

- *historyData Object* – This object will contain a copy of the performance management and other selected attributes that are present in the currentData object at the end of the current interval (e.g. 15 min.). A new instance of this object class is created at the end of each interval.
- *thresholdData Object* – This object contains a set of threshold values which correspond to a set of measurements defined for one or more classes of currentData. The thresholdData object is referenced from the currentData object by a pointer. If any of the thresholds specified in the thresholdData object are violated by the measurements in the referencing currentData object, the currentData object immediately issues a Quality of Service alarm notification.
- *Scanners* – Any of the scanner objects which are defined in Recommendation X.738 may be used to scan the contents of either the currentData or historyData objects. These scanners may be used to aggregate sets of measurements from a number of currentData objects representing a number of different Monitored Objects and/or a number of historyData objects for one or more monitored entities. These scanner objects may simply aggregate the measurements into a scanReport notification for bulk transfer to a managing system, or they may be used to perform statistics on the measurements (e.g. mean, variance, etc.) for inclusion in a scanReport which can be sent to the managing system or stored in the log.

Scanners used to aggregate measurements include: simpleScanner and dynamicSimpleScanner. Those used to perform statistics include: meanScanner, meanVarianceScanner and minMaxScanner.

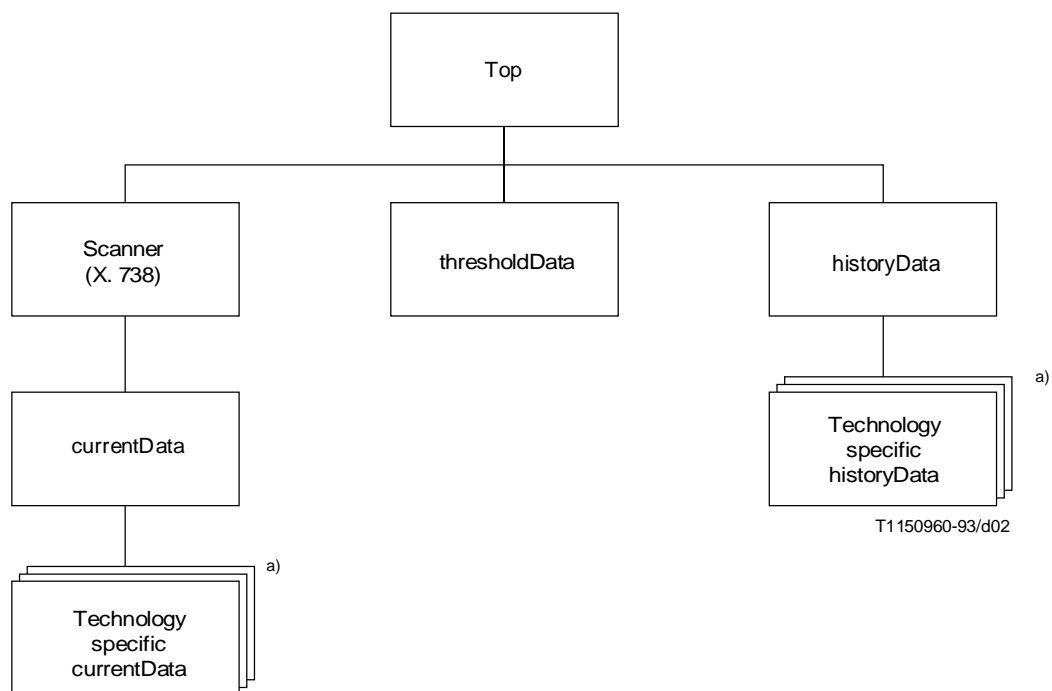
NOTE 2 – The historyData object provides more flexible access to performance measurements than the scanReport, since the measurements are held in individual attributes rather than a single complex attribute. The use of historyData also provides a closer association of the contained information with the monitored object that does the scanReport. In the generic log there is no mechanism to restrict log records in a similar way to historyData (which may be implicitly deleted after a number of intervals).

4.3 Performance management information

This Recommendation provides generic definitions of managed objects and attributes associated with Performance Management.

4.3.1 Managed object classes

The following support object classes (or their subclasses) support the Performance Management functions specified in this Recommendation. Definitions are provided in GDMO template form, as specified in CCITT Recommendation X.722. The inheritance hierarchy for these support objects is shown in Figure 2 below and the naming hierarchy is shown in Figure 3.



a) These objects will be defined in other Recommendations.

FIGURE 2/Q.822
Object inheritance

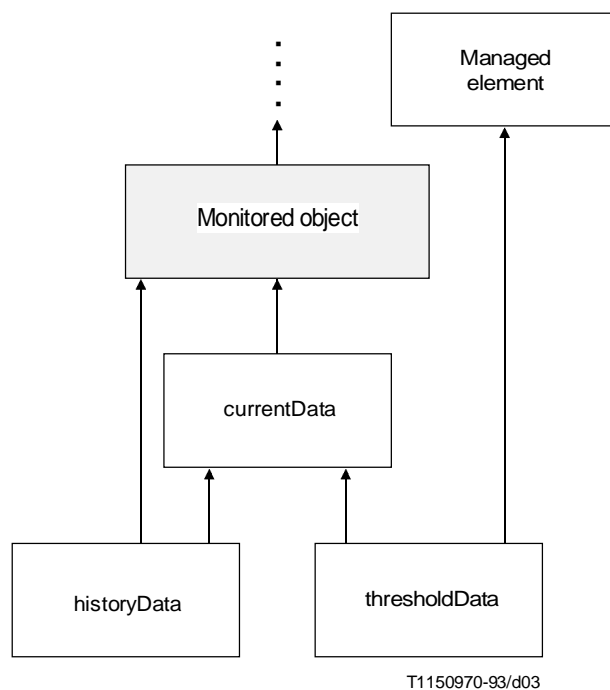


FIGURE 3/Q.822
Naming hierarchy

4.3.1.1 Current data

currentData MANAGED OBJECT CLASS

DERIVED FROM "Recommendation X.739:1993": scanner;

CHARACTERIZED BY currentDataPkg PACKAGE

BEHAVIOUR currentDataBehaviour BEHAVIOUR

DEFINED AS

"The currentData object class is a class of managed support objects that contain the current performance data. This object class is a particular type of scanner (see ITU-T Recommendation X.739) that scans its own attributes.

*-- In this Recommendation an instance of current data is assumed to be contained in the managed
-- object being monitored (such as the trailTerminationPointSink managed object defined in Recommenda-
-- tion M.3100) and has performance data pertaining to the containing managed object.*

Performance parameters are modelled as attributes. These attributes either appear explicitly in the definition of subclasses of currentData or the measurementListPkg is used for this purpose if the instantiated object class is currentData.

Typically, the performance attributes may be derived from the **counter** attribute or may be derived from the **gauge** attribute. Both the counter and gauge attributes are defined in CCITT Recommendation X.721. All attributes with a 'Count' type syntax (e.g. errored seconds) shall be locally cleared (reset) at the end of the granularity period following the scanning function. To provide continuous counters, the granularityPeriod can be set to an arbitrarily large value using the days CHOICE.

If the administrativeState is locked or the operationalState is disabled or the currentData object is scheduled off-duty, the values of the counters are undefined and historyData objects, if supported, are not created at the end of the granularityPeriod."

::

ATTRIBUTES

suspectIntervalFlag

REPLACE-WITH-DEFAULT

DEFAULT VALUE Q822-PM-ASN1Module.defaultCurrentDataSuspectIntervalFlag

GET,

elapsedTime

GET;

::

CONDITIONAL PACKAGES

filterSuppressionPkg

PRESENT IF "an instance supports it and the zeroSuppressionPkg is not present",

historyRetentionPkg

PRESENT IF "historyData objects are to be created at the end of an interval.",

maxSuppressedIntervalsPkg

PRESENT IF "an instance supports it and at least one of zeroSuppressionPkg or filterSuppressionPkg is present.",

measurementListPkg

PRESENT IF "an instance supports it or the object class is currentData",

numSuppressedIntervalsPkg

PRESENT IF "suppression counts are required and the filterSuppressionPkg or the zeroSuppressionPkg is present.",

observedManagedObjectPkg

PRESENT IF "an instance supports it.",

scheduledPMReportPkg

PRESENT IF "scheduled notifications are to be emitted.",

thresholdPkg

PRESENT IF "a Quality of Service Alarm Notification is to be emitted for threshold crossing.",

zeroSuppressionPkg

PRESENT IF "an instance supports it and the filterSuppressionPkg is not present.";

REGISTERED AS {q822ObjectClass 1};

4.3.1.2 History data

historyData MANAGED OBJECT CLASS

DERIVED FROM "Recommendation X.721:1992": top;

CHARACTERIZED BY historyDataPkg PACKAGE

BEHAVIOUR historyDataBehaviour BEHAVIOUR

DEFINED AS

Measurement attributes in the historyData object are an exact copy of the attributes in the corresponding currentData object at the end of the interval.

The time at the end of the interval is indicated by the value of the attribute periodEndTime.

::

ATTRIBUTES

historyDataId

GET,

periodEndTime

GET,

"Recommendation X.739:1993": granularityPeriod

GET;;;

CONDITIONAL PACKAGES

historyDataMeasurementListPkg

PRESENT IF "the measurementListPkg is present in the corresponding currentData instance",

historyDataSuspectIntervalFlagPkg

PRESENT IF "an instance supports it.",

numSuppressedIntervalsPkg

PRESENT IF "this package is present in the corresponding currentData instance.",

objectDeleteNotificationPkg

PRESENT IF "an instance supports it.",

observedManagedObjectPkg

PRESENT IF "this package is present in the corresponding currentData instance.";

REGISTERED AS {q822ObjectClass 2};

4.3.1.3 Threshold data

thresholdData MANAGED OBJECT CLASS

DERIVED FROM "Recommendation X.721:1992": top;

CHARACTERIZED BY thresholdDataPkg PACKAGE

BEHAVIOUR thresholdDataBehaviour BEHAVIOUR

DEFINED AS

"The thresholdData object class is a class of managed support objects that contains the values of the threshold settings for the PM parameters. At least one of the counterThresholdListPkg or the gaugeThresholdListPkg must be instantiated."


```

;;
ATTRIBUTES
    thresholdDataId
    GET;

;;
CONDITIONAL PACKAGES

counterThresholdListPkg
    PRESENT IF "an instance supports it and the gaugeThresholdListPkg is not present.",

gaugeThresholdListPkg
    PRESENT IF "an instance supports it and the counterThresholdListPkg is not present",

thresholdVerificationPkg
    PRESENT IF "an instance supports it.",

"Recommendation M.3100:1992": attributeValueChangeNotificationPackage
    PRESENT IF "an instance supports it .",

"Recommendation M.3100:1992": createDeleteNotificationsPackage
    PRESENT IF "an instance supports it .";

REGISTERED AS {q822ObjectClass 3};

```

4.3.2 Package definitions

4.3.2.1 Counter threshold list package

```

counterThresholdListPkg PACKAGE
    ATTRIBUTES
        counterThresholdAttributeList
    GET-REPLACE
    ADD-REMOVE;
REGISTERED AS {q822Package 1};

```

4.3.2.2 Filter suppression package

```

filterSuppressionPkg PACKAGE
    BEHAVIOUR filterSuppressionPkgBehaviour BEHAVIOUR
    DEFINED AS

```

"The discriminatorConstruct attribute is used to suppress the creation of historyData objects. The behaviour of the discriminatorConstruct is defined in 8.1.1.2 CCITT Rec. X.734. If the discriminatorConstruct evaluates to FALSE for the filtering criteria on the attributes specified in the discriminatorConstruct, no historyData objects are created. An empty discriminator construct will evaluate to TRUE, thus allowing the creation of historyData objects. Any attribute of the object which includes this package may be used in defining the discriminator construct."

```

;;
    ATTRIBUTES
        "Recommendation X.721:1992": discriminatorConstruct
    DEFAULT VALUE Attribute-ASN1Module.defaultDiscriminatorConstruct
    GET-REPLACE;
REGISTERED AS {q822Package 2};

```

4.3.2.3 Gauge threshold list package

```

gaugeThresholdListPkg PACKAGE
    ATTRIBUTES
        gaugeThresholdAttributeList
    GET-REPLACE
    ADD-REMOVE;
REGISTERED AS {q822Package 3};

```

4.3.2.4 History data measurement list package

historyDataMeasurementListPkg PACKAGE

ATTRIBUTES

measurementList

GET;

REGISTERED AS {q822Package 4};

4.3.2.5 History data suspect interval flag package

historyDataSuspectIntervalFlagPkg PACKAGE

BEHAVIOUR historyDataSuspectIntervalFlagPkgBehaviour BEHAVIOUR

DEFINED AS

"The historyDataSuspectIntervalFlagPkg shall be instantiated in the historyData object only if there are one or more suspected performance measurements in the corresponding accumulation interval."

::

ATTRIBUTES

suspectIntervalFlag

GET;

REGISTERED AS {q822Package 5};

4.3.2.6 History retention package

historyRetentionPkg PACKAGE

BEHAVIOUR historyRetentionPkgBehaviour BEHAVIOUR

DEFINED AS

"At the end of each interval a historyData object instance is created if the historyData creation has not been suppressed (i.e. by zero or filter suppression). The class of the historyData object will correspond to the class of the creating currentData object: an instance of the historyData object will be instantiated if there is an instance of the currentData object. The behaviour in subclasses of currentData should specify the subclass of historyData that may be instantiated. The values of the historyData measurement attributes are a copy of the values of the corresponding attributes of the currentData measurement attributes at the end of the interval.

Once the new historyData object is created, it will be retained in the NE for at least the duration equivalent to the number of intervals specified in the historyRetention attribute."

::

ATTRIBUTES

historyRetention

GET-REPLACE;

REGISTERED AS {q822Package 6};

4.3.2.7 Maximum number of suppressed intervals package

maxSuppressedIntervalsPkg PACKAGE

ATTRIBUTES

maxSuppressedIntervals

GET-REPLACE;

REGISTERED AS {q822Package 7};

4.3.2.8 Measurement list package

measurementListPkg PACKAGE

BEHAVIOUR measurementListPkgBehaviour BEHAVIOUR

DEFINED AS

"The measurementList attribute contains performance management measurements (each measurement is in the form of an attribute identifier and its value) for a specific monitored object. Typically, the performance attributes may be derived from the **counter** attribute or may be derived from the **gauge** attribute. Both the counter and gauge attributes are defined in CCITT Recommendation X.721. All attributes with a 'Count' type syntax (e.g. ES) shall be locally cleared (reset) at the end of the granularity period following the scanning function.

The measurement list shall not contain parameters which have already been defined as an attribute of that currentData object subclass.

The measurementList is only used when technology specific (currentData subclass) is not available. Typical, but not restrictive use of this package is as a home for not yet standardized performance parameters. This package should not be used as an escape from standardizing technology specific currentData subclasses.

NOTE – Measurements in the measurement list cannot be individually reset to zero."

;;

ATTRIBUTES

measurementList

GET-REPLACE;

REGISTERED AS {q822Package 8};

4.3.2.9 Number of suppressed intervals package

numSuppressedIntervalsPkg PACKAGE

ATTRIBUTES

numSuppressedIntervals

GET;

REGISTERED AS {q822Package 9};

4.3.2.10 Object delete notification package

objectDeleteNotificationPkg PACKAGE

BEHAVIOUR objectDeleteNotificationBehaviour BEHAVIOUR

DEFINED AS

"An objectDeletion notification shall not be emitted when a historyData object is deleted as a result of the expiry of the historyRetention duration."

;;

NOTIFICATIONS

"Recommendation X.721:1992": objectDeletion;

REGISTERED AS {q822Package 10};

4.3.2.11 Observed managed object package

observedManagedObjectPkg PACKAGE

ATTRIBUTES

observedObjectClass

GET,

"Recommendation X.739:1993": observedObjectInstance

GET;

REGISTERED AS {q822Package 11};

4.3.2.12 Scheduled performance monitoring report package

scheduledPMReportPkg PACKAGE

BEHAVIOUR scheduledPMReportPkgBehaviour BEHAVIOUR

DEFINED AS

"Instances of the currentData object emit the scanReport notification containing the performance monitoring data, at the end of each performance interval, as indicated by the granularityPeriod attribute.

Each performance monitoring attribute is included in the scanReport if its identifier is listed in the scanAttributeIdList or the numericAttributeIdArray. In the latter case, measurements must be included in the scanReport in the same order as specified in the numericAttributeIdArray attribute and valueOnly (in the NumericMeasure choice type) should be the only choice used to report such measurements.

Performance monitoring attributes (parameters) in the measurementList attribute are included in the scanReport as if they were individually scanned attributes of a scanned object. That is, the measurementList is scanned as if each measurement in the list were a separate attribute of the currentData object. The measurementList is not included in the scanReport as a single attribute.

Specific measurement attributes of subclasses of currentData may be included in the scanReport notification if their identifier is listed in either the scanAttributeIdList or the numericAttributeIdArray.

In order to guarantee their inclusion in an event report, the attribute ids for the following attributes of currentData (if instantiated in currentData) must appear in the onceReportAttributeIdList attribute:

- The granularityPeriod attribute.
- The suspectIntervalFlag attribute.
- The numSuppressedIntervals attribute, if present.
- The observedObjectClass and observedObjectInstance attributes, if present."

;;

ATTRIBUTES

"Recommendation X.738:1993": scanAttributeIdList

GET-REPLACE

ADD-REMOVE,

"Recommendation X.738:1993": numericAttributeIdArray

GET-REPLACE

"Recommendation X.738:1993": onceReportAttributeIdList

GET-REPLACE

ADD-REMOVE;

NOTIFICATIONS

"Recommendation X.738:1993": scanReport;

REGISTERED AS {q822Package 12};

4.3.2.13 Threshold package

thresholdPkg PACKAGE

BEHAVIOUR thresholdPkgBehaviour BEHAVIOUR

DEFINED AS

"This package is used to report threshold violations on performance data. The thresholdDataInstance attribute is a pointer to a thresholdData object that contains threshold limits for performance parameters. Whenever the value of a PM parameter violates its threshold setting, a qualityofServiceAlarm notification is emitted. The attribute violating the threshold has to be reported using the thresholdInfo field of the alarm.

New thresholds resulting from modifying the thresholdDataInstance attribute or from changing a threshold value in the referenced thresholdData object, should take effect immediately. If an alarm condition exists previous to the occurrence of a threshold value change (i.e. an old threshold had been violated), and the new threshold value is outside of the range of the old threshold value (e.g. in the case of an increasing counter, the new threshold value is greater than the old threshold value), and the current value of the measurement is within the allowable range of the new threshold value, then a QOS Alarm notification is emitted with a severity of 'clear'. If the new threshold value is set within the range of the old threshold value, such that the new threshold is violated, a QOS Alarm notification is emitted if an alarm condition is not already outstanding.

An alarm report which contains a Perceived Severity parameter with a value of 'cleared' and a Correlated Notifications parameter shall only indicate the clearing of those alarms whose Notification Identifiers are included in the set of Correlated Notifications. An alarm report which contains a Perceived Severity parameter with a value of 'cleared', but no Correlated Notifications parameter, shall indicate the clearing of alarms based on the value of the Alarm Type, Probable Cause, and Specific Problems parameters."

;;

ATTRIBUTES

reportAllAttributes

DEFAULT VALUE Q822-PM-ASN1Module.defaultReportAllAttributes

GET-REPLACE,

suppressAdditionalThresholds

DEFAULT VALUE Q822-PM-ASN1Module.defaultSuppressAdditionalThresholds

GET-REPLACE,

thresholdDataInstance

GET-REPLACE

ADD-REMOVE

NOTIFICATIONS

"Recommendation X.721:1992": qualityofServiceAlarm

"Recommendation Q.821:1992": logRecordIdParameter

"Recommendation Q.821:1992": correlatedRecordNameParameter

"Recommendation Q.821:1992": suspectObjectListParameter;

REGISTERED AS {q822Package 13};

4.3.2.14 Threshold verification package

thresholdVerificationPkg PACKAGE

BEHAVIOUR thresholdVerificationPkgBehaviour **BEHAVIOUR**

DEFINED AS

"This package allows for the agent to validate that the threshold settings defined in a created thresholdData object are reasonable. Since the attributes to which a threshold may be applied and the permissible threshold values are dependent on the type of entity being monitored and the granularity period, the agent must be aware of this information. This package provides the monitoredEntityTypes and granularityPeriod attributes needed to verify that the threshold values set in the counterthresholdAttributeList or gaugethresholdAttributeList are reasonable. If the thresholdData object receives a request to set the values in the counterthresholdAttributeList or gaugethresholdAttributeList it may reject the M-SET service request by returning an Error reply of setListError if the values are not appropriate for the monitored entity type and interval combination. In some cases the actual values set in the counterthresholdAttributeList or gaugethresholdAttributeList may be different from the requested values for the threshold. In this case, the next lowest threshold value supported by the managed system is used. This value is returned in the Reply to a confirmed M-SET service request.

The values used for the granularityPeriod attribute and monitoredEntityTypes attribute shall be consistent with the actual values of granularityPeriod used in the system and with the class of actual monitored objects."

```

;;
ATTRIBUTES
"Recommendation X.739:1993": granularityPeriod
GET,
monitoredEntityTypes
GET;
REGISTERED AS {q822Package 14};

```

4.3.2.15 Zero suppression package

```

zeroSuppressionPkg PACKAGE
    BEHAVIOUR zeroSuppressionPkgBehaviour BEHAVIOUR
    DEFINED AS

```

"When this package is present and an interval terminates with 'all-zeros' performance measurements, there is no scanReport notification issued and no historyData object is created."

```

;;
REGISTERED AS {q822Package 15};

```

4.3.3 Attribute definitions

4.3.3.1 Counter threshold attribute list

```

counterThresholdAttributeList ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Q822-PM-ASN1Module.CounterThresholdAttributeList;
    MATCHES FOR EQUALITY;
    BEHAVIOUR counterThresholdAttributeListBehaviour BEHAVIOUR
    DEFINED AS

```

"This attribute contains a set of threshold settings for performance attributes of the counter type (e.g. errored seconds). Each threshold setting consists of the attribute identifier, the threshold value and (optionally) the severity of the threshold-exceeded event."

```

;;
REGISTERED AS {q822Attribute 1};

```

4.3.3.2 Elapsed time

```

elapsedTime ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Q822-PM-ASN1Module.ElapsedTime;
    MATCHES FOR EQUALITY;
    BEHAVIOUR elapsedTimeBehaviour BEHAVIOUR
    DEFINED AS

```

"This attribute represents the difference between the current time and the start of the present summary interval."

```

;;
REGISTERED AS {q822Attribute 2};

```

4.3.3.3 Gauge threshold attribute list

```

gaugeThresholdAttributeList ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Q822-PM-ASN1Module.GaugeThresholdAttributeList;
    MATCHES FOR EQUALITY;
    BEHAVIOUR gaugeThresholdAttributeListBehaviour BEHAVIOUR
    DEFINED AS

```

"This attribute contains a set of threshold settings for performance attributes of the gauge type. Each threshold setting consists of the attribute identifier, the threshold value and (optionally) the severity of the threshold-exceeded event."

```

;;
REGISTERED AS {q822Attribute 3};

```

4.3.3.4 History data Id

historyDataId ATTRIBUTE

WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NameType;
MATCHES FOR EQUALITY;
BEHAVIOUR historyDataIdBehaviour **BEHAVIOUR**
DEFINED AS

"This attribute is used as the RDN attribute for instances of the historyData object class."

;;

REGISTERED AS {q822Attribute 4};

4.3.3.5 History retention

historyRetention ATTRIBUTE

WITH ATTRIBUTE SYNTAX Q822-PM-ASN1Module.HistoryRetention;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR historyRetentionBehaviour **BEHAVIOUR**
DEFINED AS

"This attribute specifies the minimum number of intervals that the historyData object instance (just being created) must be preserved."

;;

REGISTERED AS {q822Attribute 5};

4.3.3.6 Maximum suppressed intervals

maxSuppressedIntervals ATTRIBUTE

WITH ATTRIBUTE SYNTAX Q822-PM-ASN1Module.MaxSuppressedIntervals;
MATCHES FOR EQUALITY;
BEHAVIOUR maxSuppressedIntervalsBehaviour **BEHAVIOUR**
DEFINED AS

"In conjunction with record compression, the maxSuppressedIntervals attribute limits the maximum number of suppressed intervals that will be collected without creating an instance of the historyData object.

For example, consider an instance of (a subClass of) currentData with maxSuppressedIntervals set to 32, and the interval set to 15 minutes. For record compression, it means that after 32 consecutive suppressed (e.g. all-zero) intervals (8 hours) at least one historyData record (with all zero PM Parameters) will be generated with a count of 32. This ensures that at least one historyData record per maxSuppressedIntervals will be created.

If the numSuppressedIntervalsPkg is present, the value of this attribute cannot exceed the maximum possible value of the numSuppressedIntervals attribute (this maximum value is a local matter). If a request to modify the value of this attribute exceeds this maximum values then a CMIP error of invalid Attribute Value is returned."

;;

REGISTERED AS {q822Attribute 6};

4.3.3.7 Measurement list

measurementList ATTRIBUTE

WITH ATTRIBUTE SYNTAX Q822-PM-ASN1Module.AttributeList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;
BEHAVIOUR measurementListBehaviour BEHAVIOUR
DEFINED AS

"This attribute contains a set of performance parameter measurements for a monitored object. Each measurement is represented by the attribute identifier (e.g. errored seconds) and its value. Measurements may only be included in this list if they do not already exist as explicit measurement attributes of an instance of a subclass of currentData or of an instance of a subclass of historyData."

;;
REGISTERED AS {q822Attribute 7};

4.3.3.8 Monitored entity types

monitoredEntityTypes ATTRIBUTE

WITH ATTRIBUTE SYNTAX Q822-PM-ASN1Module.MonitoredEntityTypes;
MATCHES FOR EQUALITY;
BEHAVIOUR monitoredEntityTypesBehaviour BEHAVIOUR
DEFINED AS

"This attribute specifies the monitored entities to which the specified thresholds apply."

;;
REGISTERED AS {q822Attribute 8};

4.3.3.9 Number of suppressed intervals

numSuppressedIntervals ATTRIBUTE

WITH ATTRIBUTE SYNTAX Q822-PM-ASN1Module.NumSuppressedIntervals;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR numSuppressedIntervalsBehaviour BEHAVIOUR
DEFINED AS

"This attribute, if present, is used to count the number of consecutive intervals for which suppression (i.e. non-emission of a scanReport notification and non-creation of a historyData instance) has occurred. This attribute reflects performance measurements up to, but not including, the current interval. This attribute gets incremented at the end of an interval if suppression has occurred, otherwise, the attribute is reset."

;;
REGISTERED AS {q822Attribute 9};

4.3.3.10 Observed object class

observedObjectClass ATTRIBUTE

WITH ATTRIBUTE SYNTAX Q822-PM-ASN1Module.ObjectClass;
MATCHES FOR EQUALITY;
BEHAVIOUR observedObjectClassBehaviour BEHAVIOUR
DEFINED AS

"This attribute identifies the managed object class of the managed object that is being observed for the purpose of Performance Management."

;;
REGISTERED AS {q822Attribute 10};

4.3.3.11 Report all attributes

reportAllAttributes ATTRIBUTE

WITH ATTRIBUTE SYNTAX Q822-PM-ASN1Module.ReportAllAttributes;
MATCHES FOR EQUALITY;
BEHAVIOUR reportAllAttributesBehaviour BEHAVIOUR
DEFINED AS

"This attribute is used to indicate whether all of the measurement attributes in a currentData instance are returned in the Monitored attributes field of the Quality of Service alarm. If the value of this attribute is True, all attribute values are returned. If the value of this attribute is FALSE (default), the Quality of Service alarm does not contain the monitored attributes field."

;;

REGISTERED AS {q822Attribute 11};

4.3.3.12 Period end time

periodEndTime ATTRIBUTE

WITH ATTRIBUTE SYNTAX Q822-PM-ASN1Module.PeriodEndTime;

MATCHES FOR EQUALITY;

BEHAVIOUR periodEndTimeBehaviour BEHAVIOUR

DEFINED AS

"This attribute records the time at the end of the interval."

;;

REGISTERED AS {q822Attribute 12};

4.3.3.13 Suppress additional thresholds

suppressAdditionalThresholds ATTRIBUTE

WITH ATTRIBUTE SYNTAX Q822-PM-ASN1Module.SuppressAdditionalThresholds;

MATCHES FOR EQUALITY;

BEHAVIOUR suppressAdditionalThresholdsBehaviour BEHAVIOUR

DEFINED AS

"This attribute is used to indicate whether additional threshold crossing notifications will be emitted by the currentData object. The value of true for this attribute indicates that no further threshold crossing notifications will be emitted until the end of the interval."

;;

REGISTERED AS {q822Attribute 13};

4.3.3.14 Suspect interval flag

suspectIntervalFlag ATTRIBUTE

WITH ATTRIBUTE SYNTAX Q822-PM-ASN1Module.SuspectIntervalFlag;

MATCHES FOR EQUALITY ;

BEHAVIOUR suspectIntervalFlagBehaviour BEHAVIOUR

DEFINED AS

"This attribute is used to indicate that the performance data for the current period may not be reliable. Some reasons for this to occur are:

- Suspect data were detected by the actual resource doing data collection.
- Transition of the administrativeState attribute to/from the 'lock' state.
- Transition of the operationalState to/from the 'disabled' state.
- Scheduler setting that inhibits the collection function.
- The performance counters were reset during the interval.
- The currentData (or subclass) object instance was created during the monitoring period."

;;

REGISTERED AS {q822Attribute 14};

4.3.3.15 Threshold data Id

thresholdDataId ATTRIBUTE

WITH ATTRIBUTE SYNTAX ASN1DefinedTypesModule.NameType;
MATCHES FOR EQUALITY;
BEHAVIOUR thresholdDataIdBehaviour **BEHAVIOUR**
DEFINED AS

"This attribute is used as the RDN attribute for instances of the thresholdData object class."

;;

REGISTERED AS {q822Attribute 15};

4.3.3.16 Threshold data instance

thresholdDataInstance ATTRIBUTE

WITH ATTRIBUTE SYNTAX Q822-PM-ASN1Module.ThresholdDataInstance;
MATCHES FOR EQUALITY;
BEHAVIOUR thresholdDataInstanceBehaviour **BEHAVIOUR**
DEFINED AS

"This attribute identifies an instance of the thresholdData object class which contains the threshold settings used by the currentData instance."

;;

REGISTERED AS {q822Attribute 16};

4.3.4 Name binding definitions

No name bindings for the currentData object class are provided in this Recommendation because of the variety of monitored objects for which the performance management data can be collected. Users of this Recommendation are expected to specify name bindings for currentData or its subclasses to make them instantiable. The rules governing the automatic creation of instances of managed objects are outside the scope of this Recommendation.

historyData-currentData NAME BINDING

SUBORDINATE OBJECT CLASS historyData **AND SUBCLASSES;**
NAMED BY SUPERIOR OBJECT CLASS currentData **AND SUBCLASSES;**
WITH ATTRIBUTE historyDataId;
DELETE
DELETES-CONTAINED-OBJECTS;

REGISTERED AS {q822NameBinding 1};

NOTE – In a given application, the naming of historyData objects shall be consistent for all instances of historyData (i.e. all shall be named from the monitored entity or all shall be named from the currentData or its subclasses).

thresholdData-managedElement NAME BINDING

SUBORDINATE OBJECT CLASS thresholdData **AND SUBCLASSES;**
NAMED BY SUPERIOR OBJECT CLASS
"Recommendation M.3100:1992": managedElement **AND SUBCLASSES;**
WITH ATTRIBUTE thresholdDataId;
CREATE
WITH-REFERENCE-OBJECT,
WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE
ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS {q822NameBinding 2};

4.3.5 Abstract syntax productions

Q822-PM-ASN1Module {ccitt(0) recommendation(0) q(17) q822(822) asn1Module(2)
q822ASN1Module(0)}

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- EXPORTS everything

IMPORTS

AttributeId, ObjectInstance, ObjectClass,

FROM

CMIP-1 {joint-iso-ccitt ms(9) cmip(1) version1(1) protocol(3)}

AttributeList

FROM Attribute-ASN1Module {joint-iso-ccitt ms(9) smi(3) part2(2) asn1Module(2) 1}

SeverityIndicatingThreshold, SeverityIndicatingGaugeThreshold

FROM

MetricModule {joint-iso-ccitt ms(9) function(2) part11(11) asn1Module(2) 0}

TimeInterval

FROM

Q821-ASN1Module {ccitt(0) recommendation(0) q(17) q821(821) ans1Module(2)
q821ASN1Module(0)};

q822InformationModel OBJECT IDENTIFIER ::=

{ccitt(0) recommendation(0) q(17) q822(822) informationModel(0)}

q822ObjectClass OBJECT IDENTIFIER ::=

{q822InformationModel managedObjectClass(3)}

q822Package OBJECT IDENTIFIER ::=

{q822InformationModel package(4)}

q822Attribute OBJECT IDENTIFIER ::=

{q822InformationModel managedObjectClass(7)}

q822NameBinding OBJECT IDENTIFIER ::=

{q822InformationModel nameBinding(6)}

defaultCurrentDataSuspectIntervalFlag

SuspectIntervalFlag ::= FALSE

defaultReportAllAttributes

ReportAllAttributes ::= FALSE

defaultSuppressAdditionalThresholds

SuppressAdditionalThresholds ::= FALSE

CounterThresholdAttributeList ::= SET OF CounterThresholdSetting

CounterThresholdSetting ::= SEQUENCE {

attributeId	AttributeId,
severityIndicatingThreshold	SeverityIndicatingThreshold

}

ElapsedTime ::= CHOICE {

elapsedTime	TimeInterval,
notAvailable	NULL

}

GaugeThresholdAttributeList ::= SET OF GaugeThresholdSetting

GaugeThresholdSetting ::= SEQUENCE {

attributeId	AttributeId,
severityIndicatingGaugeThreshold	SeverityIndicatingGaugeThreshold

```

}

HistoryRetention ::= INTEGER

MaxSuppressedIntervals ::= INTEGER

MonitoredEntityTypes ::= SET OF ObjectClass

NumSuppressedIntervals ::= INTEGER

PeriodEndTime ::= GeneralizedTime

ReportAllAttributes ::= BOOLEAN

SuppressAdditionalThresholds ::= BOOLEAN

SuspectIntervalFlag ::= BOOLEAN

ThresholdDataInstance ::= SET OF ObjectInstance

END

```

4.3.6 Support objects from other CCITT Recommendations

The following object classes defined in Recommendations X.738, Summarization Function, X.734, Event Report Management Function and X.735, the Log Management Function, can be used to support the Performance Management functions specified in this Recommendation.

4.3.6.1 Support objects from the summarization function

The following object classes defined in ITU-T Recommendation X.738 can be used to support the Performance Management functions specified in this Recommendation:

- dynamicScanner;
- meanScanner;
- meanVarianceScanner;
- minMaxScanner;
- simplecanner;
- scanReportRecord.

4.3.6.2 Support objects from definition of management information

The following support object classes (or their subclasses), defined in CCITT Recommendation X.721, can be used to support the Performance Management functions specified in this Recommendation:

- discriminator;
- eventForwardingDiscriminator;
- eventLogRecord;
- log;
- logRecord.

4.3.7 Performance management control

The Performance Management Control object may be required to accomodate the following requirement which has been specified:

On a per NE basis provide an alarm if the minimum designed duration is about to be violated with retention of the last n records for each object instance.

The definition of this object is for further study.

4.4 Performance management service definition

This Recommendation does not define any new services.

4.5 Functional unit definitions

The following functional units defined in CCITT Recommendation X.730 may be negotiated for the purpose of managing the objects defined in this Recommendation:

- control;
- monitor;
- objectEvents.

The following functional units defined in CCITT Recommendation X.731 may be negotiated for the purpose of managing the objects defined in this Recommendation:

- stateChangeReporting.

The following functional units defined in CCITT Recommendation X.733 may be negotiated for the purpose of managing the objects defined in this Recommendation:

- alarmReporting.

The following functional units defined in CCITT Recommendation X.738 may be negotiated for the purpose of managing the objects defined in this Recommendation:

- summarization event reporting functional unit.

5 Protocol

5.1 Abstract syntax

5.1.1 Managed objects

5.1.1.1 Referenced managed objects

This Recommendation references the following support managed objects for which the abstract syntax is specified in ITU-T Recommendation X.739:

scanner.

This Recommendation references the following support managed objects for which the abstract syntax is specified in CCITT Recommendation X.721:

top.

5.1.1.2 Defined managed objects

This Recommendation defines the following managed objects:

- a) currentData;
- b) historyData;
- c) thresholdData.

5.1.2 Attributes

5.1.2.1 Attributes imported from the metric objects and attributes function

This Recommendation references the following management attributes defined in ITU-T Recommendation X.739:

granularityPeriod.

5.1.2.2 Attributes imported from the summarization function

This Recommendation references the following management attributes defined in ITU-T Recommendation X.738:

- a) numericAttributeIdArray;
- b) onceReportAttributeIdList;
- c) scanAttributeIdList.

5.1.2.3 Attributes imported from definition of management information

This Recommendation references the following management attributes, the abstract syntax of which is specified in CCITT Recommendation X.721:

discriminatorConstruct.

5.1.2.4 Attributes defined in this Recommendation

This Recommendation defines the following management attributes, the abstract syntax of which is specified in 4.3.3:

- a) counterThresholdAttributeList;
- b) elapsedTime;
- c) gaugeThresholdAttributeList;
- d) historyDataId;
- e) historyRetention;
- f) maxSuppressedIntervals;
- g) measurementList;
- h) monitoredEntityTypes;
- i) numSuppressedIntervals;
- j) observedObjectClass;
- k) reportAllAttributes;
- l) periodEndTime;
- m) suppressAdditionalThresholds;
- n) suspectIntervalFlag;
- o) thresholdDataId;
- p) thresholdDataInstance.

5.1.3 Notifications

5.1.3.1 Referenced notifications

This Recommendation references the following events defined in CCITT Recommendation X.730:

- a) objectCreation;
- b) objectDeletion;
- c) attributeValueChange.

This Recommendation references the following events defined in CCITT Recommendation X.731:

stateChange.

This Recommendation references the following events defined in CCITT Recommendation X.733 and used as described in 4.3.2.13:

qualityofServiceAlarm.

This Recommendation references the following events defined in ITU-T Recommendation X.738:

scanReport.

5.2 Negotiation of functional units

Within the Systems management application context, the mechanism for negotiating the functional units is described by CCITT Recommendation X.701.

NOTE – This subject requires further study.

6 Relationship with other Recommendations

This Recommendation uses definitions from CCITT Recommendation X.721.

This Recommendation uses services defined in CCITT Recommendation X.730 for the creation and deletion of managed objects, the retrieval and update of attributes, the notifications on object creation, object deletion and attribute value changes.

This Recommendation uses services defined in CCITT Recommendation X.731 for the notification of state changes.

This Recommendation uses services defined in ITU-T Recommendation X.733 for the reporting of alarms.

This Recommendation uses definitions and services from ITU-T Recommendation X.738.

The incorporation of the metric objects as defined in ITU-T Recommendation X.739 is for further study.

7 Conformance

There are two conformance classes: general conformance class and dependent conformance class. A system claiming to implement the elements of procedure for system management services referenced by this Recommendation shall comply with the requirements for either the general or the dependent conformance class as defined in the following clauses. The supplier of the implementation shall state the class to which the conformance is claimed.

NOTE – The use of the two terms “general conformance class” and “dependent conformance class”, is under review. However, this Recommendation continues to use these terms in order to be consistent with CCITT Rec. X.701 | ISO/IEC 10040 and other Recommendations and Standards under the general title Information Technology – Open Systems Interconnection – Systems Management. When a final answer to ISO/IEC JTC1 SC21 Question Q1/49.9 (on the long term solution to general and dependent conformance) has been approved, it is intended to clarify and/or correct this conformance clause together with the related clauses in those other Standards.

7.1 General conformance class requirements

A system claiming general conformance shall support this function for all managed object classes that import the management information defined in this Recommendation.

NOTE – This is applicable to all subclasses of the management support object classes defined in this Recommendation.

7.1.1 Static conformance

The system shall:

- a) support the role of manager or agent or both, with respect to the pmControl functional unit;
- b) for each supported package defined in 4.3, support the transfer syntax derived from the encoding rules specified in CCITT Recommendation X.209 and named {joint-iso-ccitt asn1(1) basicEncoding(1)}, for the purpose of generating and/or interpreting the MAPDUs defined by the abstract data types referenced in 5.1.2 and 5.1.3 for the role supported in a) above;
- c) when acting in the agent role, support at least one or more instances of the currentData managed object class or any of its subclasses;
- d) a system need not support sending all of the named types within ITU-T Recommendation X.739 MetricModule.TimePeriod CHOICE.

7.1.2 Dynamic conformance

The system shall, in the role(s) for which conformance is claimed, support the elements of procedure defined in:

- CCITT Recommendation X.730, for the PT-GET, PT-CREATE, PT-DELETE, PT-SET services.
- ITU-T Recommendation X.738, for the emission of scan reports, if the scheduledPMReportPkg is supported.
- CCITT Recommendation X.733, for the quality of service alarm reporting and used as described in 4.3.2.13, if the thresholdPkg is supported.
- CCITT Recommendation X.730 | ISO/IEC 10164-1 for the Object creation reporting and Object deletion reporting, if the create delete notifications package is supported.
- CCITT Recommendation X.730 | ISO/IEC 10164-1 for the Attribute value change reporting services, if the attribute value change notification package is supported.
- CCITT Recommendation X.731 | ISO/IEC 10164-2 for the State change reporting service, if the state change notification package is supported.

7.2 Dependent conformance class requirements

7.2.1 Static conformance

The system shall:

- a) for each supported package defined in 4.3, support the transfer syntax derived from the encoding rules specified in CCITT Recommendation X.209 and named {joint-iso-ccitt asn1(1) basicEncoding(1)}, for the purpose of generating and/or interpreting the MAPDUs defined by the abstract data types referenced in 5.1.2 and 5.1.3, as required by a referencing specification;
- b) support one or more instances of the PM object classes defined in this Recommendation or any of their subclasses, when acting in the agent role;
- c) a system need not support sending all of the named types within the MetricModule.TimePeriod CHOICE.

7.2.2 Dynamic conformance

The system claiming dependent conformance shall support the elements of procedure referenced by this Recommendation, as required by the referencing specification to implement the specified subset of functionality.

7.3 Conformance to support managed object definitions

The performance management objects supported by the open system shall comply with the specification in clause 4. For minimum conformance to the support managed objects defined in this Recommendation, at least one of the following packages: historyRetentionPkg, measurementListPkg, scheduledPMReportPkg or thresholdPkg of the currentData object

Annex A

Performance monitoring parameters

(This annex does not form an integral part of this Recommendation)

This Annex is provided as a guide and contains examples of how performance parameters could be defined as managed object attributes. Any specification that requires to use any of the examples provided here needs to formally include the attribute definition in that specification before it can be used.

A.1 Abbreviations

AISS	Alarm Indication Signal Second
AISSFE	Alarm Indication Signal Second Far End
AVA	Attribute Value Assertion
CSS	Controlled Slip Second
CSSFE	Controlled Slip Second Far End
CV	Code Violation
CVCP	Code Violation, CP-bit Parity
CVCPFE	Code Violation, CP-bit Parity Far End
CVCRC	Code Violation, Cyclic Redundancy Check
CVCRCFE	Code Violation, Cyclic Redundancy Check Far End
CVFE	Code Violation Far End
CVP	Code Violation, P-bit
ES	Errored Second
ESA	Errored Second, type A
ESACP	Errored Second, type A, CP-bit
ESACPF	Errored Second, type A, CP-bit Far End
ESACRC	Errored Second, type A, CRC
ESACRCFE	Errored Second, type A, CRC Far End
ESAFE	Errored Second, type A, Far End
ESAP	Errored Second, type A, P-bit
ESB	Errored Second, type B
ESBCP	Errored Second, type B, CP-bit
ESBCPF	Errored Second, type B, CP-bit Far End
ESBCRC	Errored Second, type B, CRC
ESBCRCFE	Errored Second, type B, CRC Far End
ESBFE	Errored Second, type B, Far End
ESBP	Errored Second, type B, P-bit

ESCP	Errored Second, CP-bit Parity
ESCPFE	Errored Second, CP-bit Parity Far End
ESCRC	Errored Second, Cyclic Redundancy Check
ESCRCFE	Errored Second, Cyclic Redundancy Check Far End
ESFE	Errored Second Far End
ESP	Errored Second, P-bit
LOSS	Loss of Signal Second
PSC	Protection Switching Count
PSD	Protection Switching Duration
SAS	SEF/AIS Severely Errored Framing Alarm Indication Signal Second
SASFE	SAS Far End
SES	Severely Errored Second
SESCP	Severely Errored Second, CP-bit
SESCPFE	Severely Errored Second, CP-bit Far End
SESCRC	Severely Errored Second, CRC
SESCRCFE	Severely Errored Second, CRC Far End
SESFE	Severely Errored Second Far End
SESP	Severely Errored Second-Path
UAS	Unavailable Second
UASFE	Unavailable Second Far End

A.2 Performance monitoring parameters attribute templates

A.2.1 Alarm indication signal second (AISS)

aAISS ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR aAISSBehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing one or more AIS defects.”

::
REGISTERED AS {xx};

A.2.2 Alarm indication signal second far end (AISSFE)

aAISSFE ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR aAISSFEBehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing one or more AISs which were received by the far end terminal.”

::
REGISTERED AS {xx};

A.2.3 Controlled slip second (CSS)

cSS ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR cSSBehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing one of more controlled slips.”

::
REGISTERED AS {xx};

A.2.4 Controlled slip second far end (CSSFE)

cSSFE ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR cSSFEBehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing one or more controlled slips which were received by the far end terminal.”

::
REGISTERED AS {xx};

A.2.5 Code Violation (CV)

cV ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR cVBehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing one or more code violations.”

::
REGISTERED AS {xx};

A.2.6 Code violation, CP-bit parity (CVCP)

cVCP ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR cVCPBehaviour BEHAVIOUR
DEFINED AS

“This attribute is counted when a CP-bit parity violation is detected in an M-frame.”

::
REGISTERED AS {xx};

A.2.7 Code violation, CP-bit parity far end (CVCPFE)

cVCPFE ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR cVCPFEBehaviour BEHAVIOUR
DEFINED AS

“This attribute is counted when a CP-bit parity violation is detected in an M-frame which was received by the far end terminal.”

::
REGISTERED AS {xx};

A.2.8 Code violation, cyclic redundancy check (CVCRC)

cVCRC ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR cVCRCBehaviour BEHAVIOUR
DEFINED AS

“This attribute is counted when a CRC-9 violation is detected over 3 consecutive M-frames.”

;;
REGISTERED AS {xx};

A.2.9 Code violation, cyclic redundancy check far end (CVCRCFE)

cVCRCFE ATTRIBUTE
DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR cVCRCFEBehaviour BEHAVIOUR
DEFINED AS

“This attribute is counted when a CRC-9 violation is detected over 3 consecutive M-frames and received by the far end terminal.”

;;
REGISTERED AS {xx};

A.2.10 Code violation far end (CVFE)

cVFE ATTRIBUTE
DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR cVFEBehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing one or more code violations which were received by the far end terminal.”

;;
REGISTERED AS {xx};

A.2.11 Code violation, P-bit (CVP)

cVP ATTRIBUTE
DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR cVPBehaviour BEHAVIOUR
DEFINED AS

“This attribute is counted when a P-bit parity violation is detected in an M-frame.”

;;
REGISTERED AS {xx};

A.2.12 Errored second (ES)

eS ATTRIBUTE
DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSBehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing one or more digital transmission errors.”

;;
REGISTERED AS {xx};

A.2.13 Errored second, type A (ESA)

eSA ATTRIBUTE
DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSABehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing one and only one digital transmission error.”

;;
REGISTERED AS {xx};

A.2.14 Errored second, type A, CP-bit (ESACP)

eSACP ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSACPBbehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing one and only one CP-bit parity error.”

::
REGISTERED AS {xx};

A.2.15 Errored second, type A, CP-bit far end (ESACPFE)

eSACPFE ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSACPFEbehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing one and only one CP-bit parity error, which was received by the far end terminal.”

::
REGISTERED AS {xx};

A.2.16 Errored second, type A, CRC (ESACRC)

eSACRC ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSACRCbehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing one and only one CRC-9 error.”

::
REGISTERED AS {xx};

A.2.17 Errored second, type A, CRC far end (ESACRCFE)

eSACRCFE ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSACRCFEbehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing one and only one CRC-9 error and received by the far end terminal.”

::
REGISTERED AS {xx};

A.2.18 Errored second, type A, far end (ESAFE)

eSAFE ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSAFEbehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing one and only one digital transmission error which was received by the far end terminal.”

::
REGISTERED AS {xx};

A.2.19 Errored second, type A, P-bit (ESAP)

eSAP ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSAPBehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing one and only one P-bit parity error.”

::
REGISTERED AS {xx};

A.2.20 Errored second, type B (ESB)

eSB ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSBBehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing no less than two, and no more than X digital transmission errors.”

::
REGISTERED AS {xx};

A.2.21 Errored second, type B, CP-bit (ESBCP)

eSBCP ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSBCPBehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing no less than two, and no more than X CP-bit parity errors.”

::
REGISTERED AS {xx};

A.2.22 Errored second, type B, CP-bit far end (ESBCPFE)

eSBCPFE ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSBCPFEBehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing no less than two, and no more than X CP-bit parity errors, which were received by the far end terminal.”

::
REGISTERED AS {xx};

A.2.23 Errored second, type B, CRC (ESBCRC)

eSBCRC ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSBCRCBehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing no less than two, and no more than X CRC-9 errors.”

::
REGISTERED AS {xx};

A.2.24 Errored second, type B, CRC far end (ESBCRCFE)

eSBCRCFE ATTRIBUTE

**DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSBCRCFEBehaviour BEHAVIOUR
DEFINED AS**

“This attribute identifies a count of one second intervals containing no less than two, and no more than X CRC-9 errors and received by the far end terminal.”

**::
REGISTERED AS {xx};**

A.2.25 Errored second, type B, far end (ESBFE)

eSBFE ATTRIBUTE

**DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSBFEBehaviour BEHAVIOUR
DEFINED AS**

“This attribute identifies a count of one second intervals containing no less than two, and no more than X digital transmission errors which were received by the far end terminal.”

**::
REGISTERED AS {xx};**

A.2.26 Errored second, type B, P-bit (ESBP)

eSBP ATTRIBUTE

**DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSBPBehaviour BEHAVIOUR
DEFINED AS**

“This attribute identifies a count of one second intervals containing no less than two, and no more than X P-bit parity errors.”

**::
REGISTERED AS {xx};**

A.2.27 Errored second, CP-bit parity (ESCP)

eSCP ATTRIBUTE

**DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSCPBehaviour BEHAVIOUR
DEFINED AS**

“This attribute is a count of one second intervals containing the occurrence of one or more CVCP.”

**::
REGISTERED AS {xx};**

A.2.28 Errored second, CP-bit parity far end (ESCPFE)

eSCPFE ATTRIBUTE

**DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSCPFEBehaviour BEHAVIOUR
DEFINED AS**

“This attribute is a count of one second intervals containing the occurrence of one or more CVCP, which was received at the far-end.”

**::
REGISTERED AS {xx};**

A.2.29 Errored second, cyclic redundancy check (ESCRC)

eSCRC ATTRIBUTE

**DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSCRCBehaviour BEHAVIOUR
DEFINED AS**

“This attribute is a count of one second intervals containing the occurrence of one or more CVCRC.”

**::
REGISTERED AS {xx};**

A.2.30 Errored second, cyclic redundancy check far end (ESCRCFE)

eSCRCFE ATTRIBUTE

**DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSCRCFEBehaviour BEHAVIOUR
DEFINED AS**

“This attribute is a count of one second intervals containing the occurrence of one or more CVCRC and received by the far end terminal.”

**::
REGISTERED AS {xx};**

A.2.31 Errored second far end (ESFE)

eSFE ATTRIBUTE

**DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSFEBehaviour BEHAVIOUR
DEFINED AS**

“This attribute identifies a count of one second intervals containing one or more digital transmission errors which were received by the far end terminal.”

**::
REGISTERED AS {xx};**

A.2.32 Errored second, P-bit (ESP)

eSP ATTRIBUTE

**DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR eSPBehaviour BEHAVIOUR
DEFINED AS**

“This attribute is a count of one second intervals containing the occurrence of one or more CVP.”

**::
REGISTERED AS {xx};**

A.2.33 Loss of signal second (LOSS)

LOSS ATTRIBUTE

**DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR LOSSBehaviour BEHAVIOUR
DEFINED AS**

“This attribute identifies a count of one second intervals containing one or more loss of signal indicators.”

**::
REGISTERED AS {xx};**

A.2.34 Protection switching count (PSC)

pSC ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR pSCBehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of the number of times a monitored resource is switched to protection. Although this attribute is not currently assigned to any specific object class, this attribute could be included as a member of the Additional Attribute List.”

::
REGISTERED AS {xx};

A.2.35 Protection switching duration (PSD)

pSD ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR pSDBehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies the number of seconds that service was removed from the monitored resource. Although this attribute is not currently assigned to any specific object class, this attribute could be included as a member of the Additional Attribute List.”

::
REGISTERED AS {xx};

A.2.36 SEF/AIS severely errored framing alarm indication signal second (SAS)

sAS ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR sASBehaviour BEHAVIOUR
DEFINED AS

“This attribute identifies a count of one second intervals containing one or more Severely Errored Framing (SEF) events.”

::
REGISTERED AS {xx};

A.2.37 SEF/AIS second, far end (SASFE)

sASFE ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR sASFEBehaviour BEHAVIOUR
DEFINED AS

“This Attribute identifies a count of one second intervals containing one or more Severely Errored Framing (SEF) events which were received by the far end terminal.”

::
REGISTERED AS {xx};

A.2.38 Severely errored second (SES)

sES ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR sESBehaviour BEHAVIOUR
DEFINED AS

“This Attribute identifies a count of one second intervals containing more than X digital transmission errors or a Severely Errored Framing (SEF) event.”

::
REGISTERED AS {xx};

A.2.39 Severely errored second, CP-bit (SESCP)

sESCP ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR sESCPBehaviour BEHAVIOUR
DEFINED AS

“This Attribute identifies a count of one second intervals containing more than X CP-bit parity errors or a Severely Errored Framing (SEF) event..”

;;
REGISTERED AS {xx};

A.2.40 Severely errored second, CP-bit far end (SESCPFE)

sESCPFE ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR sESCPFEBehaviour BEHAVIOUR
DEFINED AS

“This Attribute identifies a count of one second intervals containing more than X CP-bit parity errors or a Severely Errored Framing (SEF) event, which were received by the far end terminal.”

;;
REGISTERED AS {xx};

A.2.41 Severely errored second, CRC (SESCRC)

sESCRC ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR sESCRCBehaviour BEHAVIOUR
DEFINED AS

“This Attribute identifies a count of one second intervals containing more than X CRC-9 errors or a Severely Errored Framing (SEF) event.”

;;
REGISTERED AS {xx};

A.2.42 Severely errored second, CRC far end (SESCRCFE)

sESCRCFE ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR sESCRCFEBehaviour BEHAVIOUR
DEFINED AS

“This Attribute identifies a count of one second intervals containing more than X CRC-9 errors or a Severely Errored Framing (SEF) event and received by the far end terminal.”

;;
REGISTERED AS {xx};

A.2.43 Severely errored second far end (SESFE)

sESFE ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR sESFEBehaviour BEHAVIOUR
DEFINED AS

“This Attribute identifies a count of one second intervals containing more than X digital transmission errors or a Severely Errored Framing (SEF) event received by the far end termination.”

REGISTERED AS {xx};

A.2.44 Severely errored second-path (SESP)

sESP ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR sESPBehaviour BEHAVIOUR
DEFINED AS

“This attribute is a count of one second intervals containing the occurrence of any of the following events: more than x CVs, one or more SEFs, one or more AISs. Candidate values of x are 447, 44 and 6 which correspond to BER threshold of 10^{-5} , 10^{-6} , and $1.5 \cdot 10^{-7}$ respectively. The default value of x is 44.”

::
REGISTERED AS {xx};

A.2.45 Unavailable second (UAS)

uAS ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR uASBehaviour BEHAVIOUR
DEFINED AS

“This Attribute identifies a count of one second intervals during which resource service was unavailable.”

::
REGISTERED AS {xx};

A.2.46 Unavailable second far end (UASFE)

uASFE ATTRIBUTE

DERIVED FROM "Recommendation X.721:1992": counter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR uASFEBehaviour BEHAVIOUR
DEFINED AS

“This Attribute identifies a count of one second intervals during which resource service was unavailable which were received by the far end termination.”

::
REGISTERED AS {xx};

Annex B

Performance management information model overview

(This annex does not form an integral part of this Recommendation)

B.1 Introduction

This annex provides a tutorial overview description of the Performance Management information model.

B.2 Overview of the performance management information model

This overview intended to provide a conceptual framework for understanding how the various managed object classes are used to provide the performance management services described in the body of this recommendation. This subclause is not intended to be a comprehensive description of the model. The actual managed object class and attribute definitions are found in the body of this Recommendation. The figures in this appendix are used to illustrate concepts only and do not incorporate many details present in the managed object class and attribute definitions found in the Performance Management Model.

The following list contains some of the capabilities provided by the PM information model:

- *Notification of threshold crossing violation* – A qualityOfServiceAlarm notification may be emitted by the currentData object.
- *History of threshold crossing data* – The qualityOfServiceAlarm notification may be stored in the log as an alarmRecord object for future retrievals.
- *History of Performance Management data* – A historyData that mirrors the performance data information in the currentData object may be created at the end of each performance interval.
- *Summary reporting of performance parameters* – A simpleScanner object may be used to scan multiple historyData or currentData objects to provide a summary report using a scanReport notification.
- *Statistical reporting of performance parameters* – A statistical scanner object may be used to scan multiple historyData or currentData objects to provide a statistical report using a statisticalReport notification.
- *Individual performance attributes retrieval capability* – The currentData class may be refined by specifying a technology specific subclass, e.g. V4 TTP, that contains technology specific performance attributes, e.g. ES.
- *Measurement List retrieval capability* – For an instantiated currentData object (no subclassing) the measurementList attribute is used for the collection of performance parameters. For an instantiated subclass of the currentData object the measurementList attribute may be used to collect additional performance parameters.
- *Notification of performance parameters* – A scanReport notification containing performance data may be emitted by the Currentdata object at the end of each performance interval.

B.2.1 Collection of current PM data

Current PM data is collected from a managed object by use of the managed object class currentData or its subclasses. Instances of the currentData object class or its subclasses are contained within the managed object being observed. Each currentData object class or its subclasses is self-contained in that it specifies data collection aspects such as the collection interval, and the types of measurement data to be collected. The measurement data found in currentData object class or its subclasses represents the current state of the counters and indicators of PM data. The values of the attributes may change over the duration of the collection interval and those attributes that represent counter type measurements are reset at the end of the interval. The collection of PM data from a managed object is depicted in Figure B.1.

B.2.2 History records

At the end of each collection interval, instances of historyData (or its subclasses) may be created that mirror the performance data information in a currentData instance (or its subclasses). The collection interval is specified in the attribute granularityPeriod inherited by the currentData class. The retention period (duration) of the created instance of historyData or its subclasses is denoted by the attribute historyRetention. When a new instance of historyData or a subclass is created, the oldest existing instance of the historyData object or a subclass may be deleted, provided that it has been retained for at least the duration prescribed.

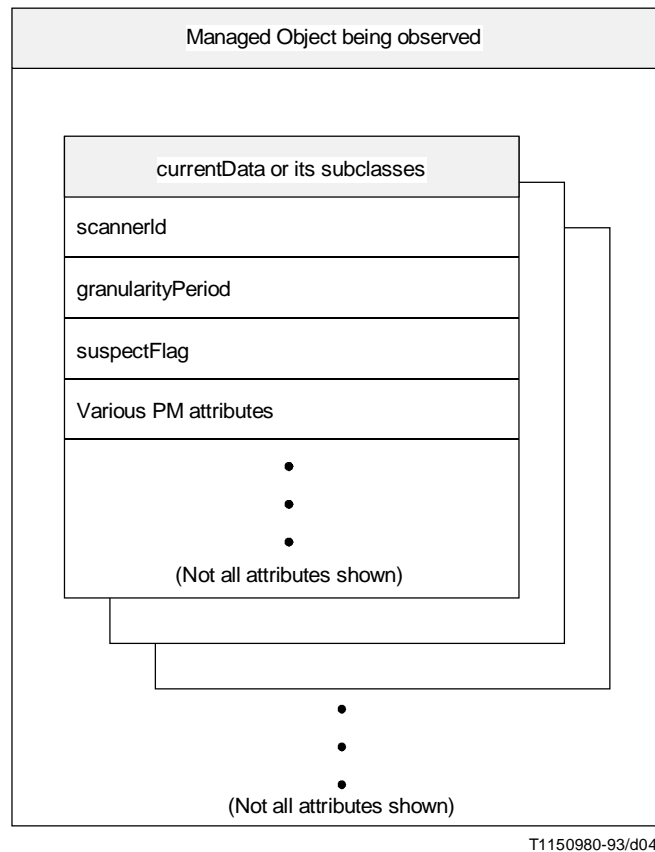


FIGURE B.1/Q.822
PM data collection from a managed object

B.2.3 Scheduled report

Instances of `currentData` or its subclasses may emit scheduled reports that contain performance data at the end of each collection interval. The collection interval is specified in the attribute `granularityPeriod` inherited by the `currentData` class. The performance data to be included in these reports are specified in the `scanAttributeIdList` and the `numericAttributeIdArray` attributes.

B.2.4 Setting and reporting of PM thresholds

Thresholds are established by use of the managed object class `thresholdData`. The `thresholdData` objects specify the thresholds being applied. Threshold values in a `thresholdData` object may apply to multiple `currentData` objects and `thresholdData` objects are pointed to by `currentData` objects. The `currentData` object or its subclasses indicates the collection interval used with the threshold. A `qualityOfServiceAlarm` notification is generated whenever a threshold is crossed. These notifications shall pass through an `eventForwardingDiscriminator` object to be reported and through the log object to be logged as an `alarmRecord` object.

B.2.4.1 Thresholds applying to a single managed object

A threshold may be specific to a single managed object. In this case, the `thresholdData` object is contained within the managed object being observed. A `currentData` object within the managed object points to a `thresholdData` object. This linkage is needed for consistency with the multiple managed object case. Figure B.2 illustrates thresholding for a single managed object instance.

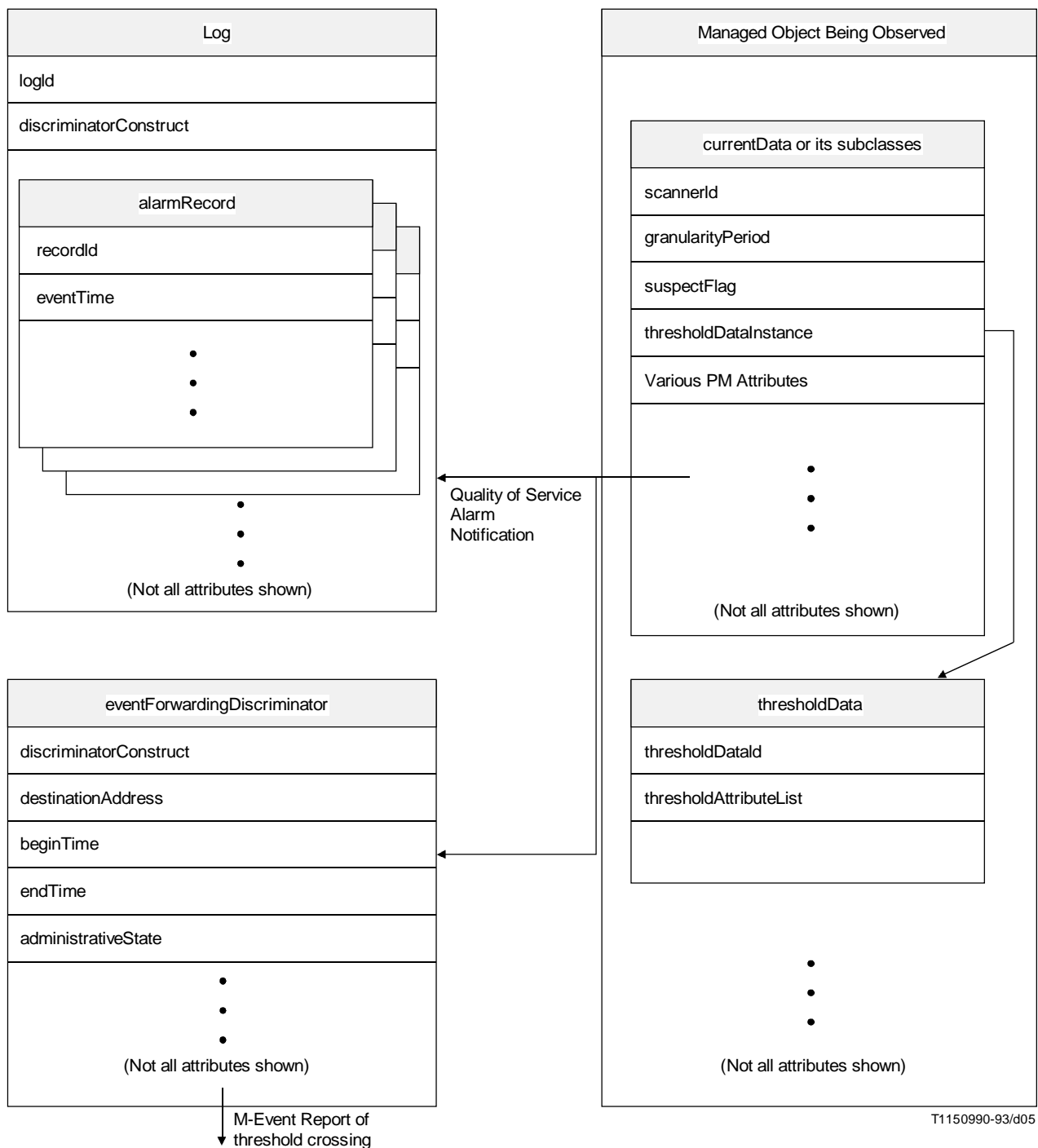
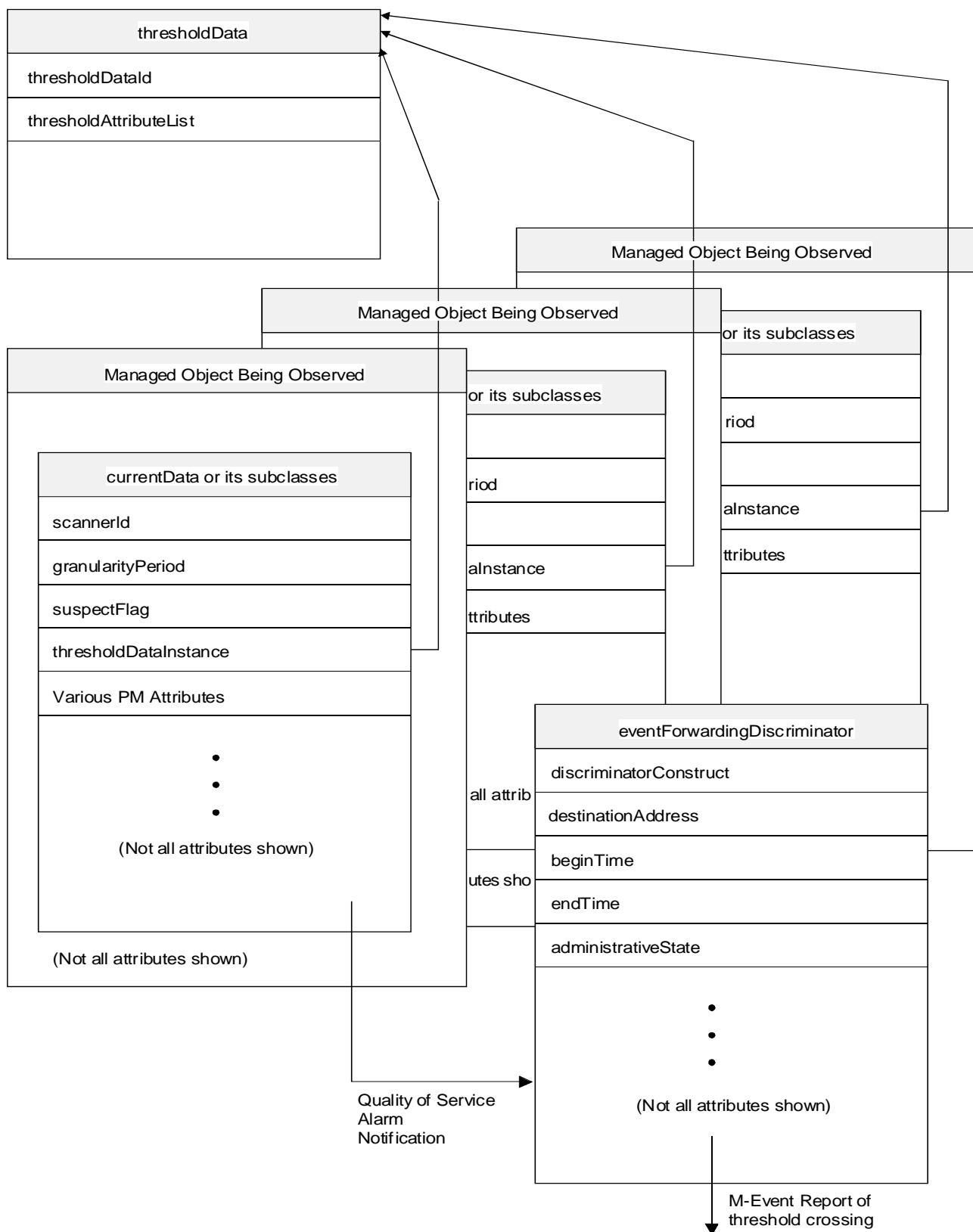


FIGURE B.2/Q.822
Threshold for a single managed object

B.2.4.2 Thresholds applying to multiple managed objects

It is sometimes desirable to have a threshold apply to a group of managed objects. In this case, the thresholdData object is external to the managed object being observed. It may be contained within the managedElement object. The thresholdData object is pointed to by all the currentData objects to which it applies. Figure B.3 illustrates a threshold applied to a group of managed objects. The historical retention of threshold crossing data is similar to the previous example and it is not shown in the Figure.



T115 1000-93/d06

FIGURE B.3/Q.822
Thresholds applied to a group of managed objects

B.2.5 Historical retention of PM data

The data found in the `currentData` object class or its subclasses consists of counters and measurements which are in the process of accumulating and changing during the interval. It is often useful to access the last completed interval. The data from the last completed interval, along with any data from preceding intervals, is termed historical data. A managed system may create an `historyData` object for each observed object at the end of each performance interval. Historical retention using `historyData` object is illustrated in Figure B.4.

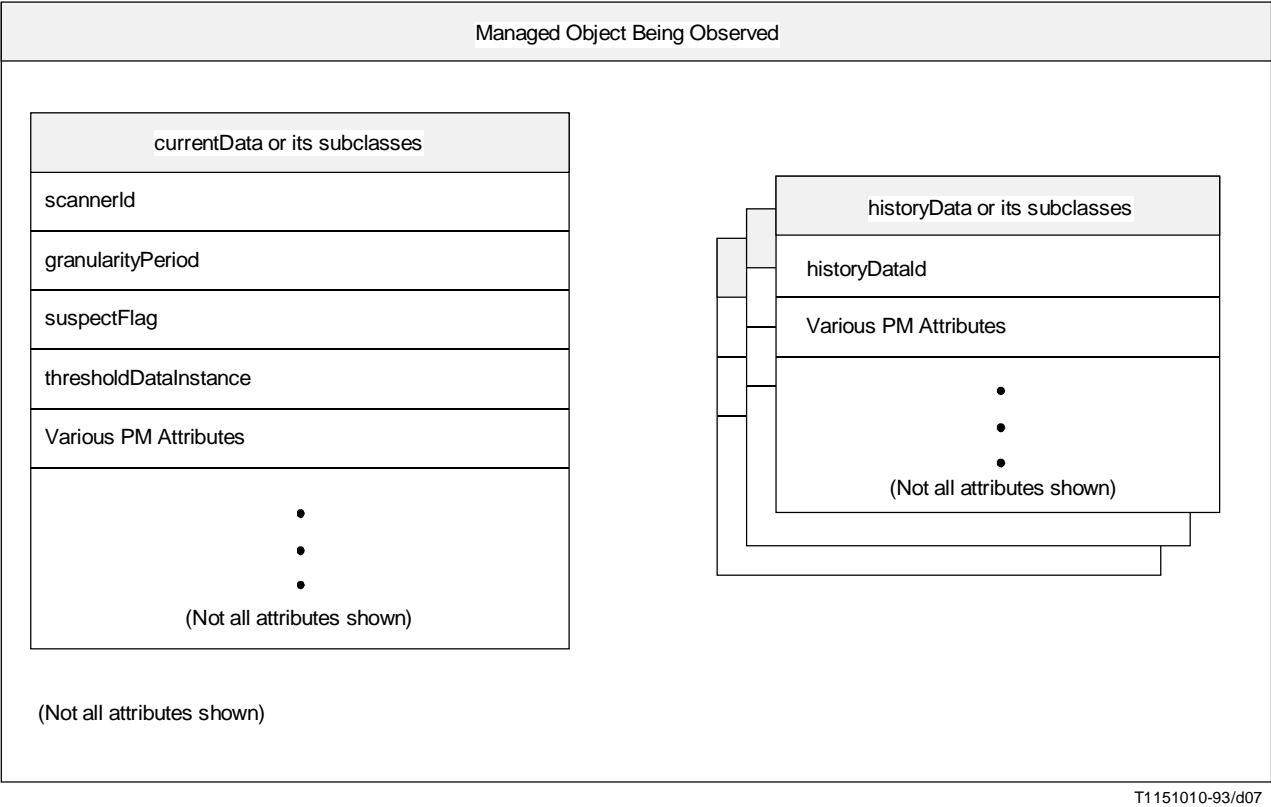


FIGURE B.4/Q.882
History retention of PM data

B.2.6 Scheduled summary or statistical reports

Data produced by instances of the `currentData` class or its subclasses may be compiled into summary or statistical reports. The data collected by these objects at the end of each collection interval is stored in `historyData` objects. The `simpleScanner` managed object scans the common attributes to be observed in each of the selected objects according to a specified schedule and produces an aggregated report. In addition, the statistical scanners perform a statistical calculation and report its results. The schedule for scanning and reporting may be specified as part of the scanner objects. The notifications generated by the scanner may be processed by an event forwarding discriminator for reporting to the managing system. The generation of PM summarized or statistical reports is depicted in Figure B.5.

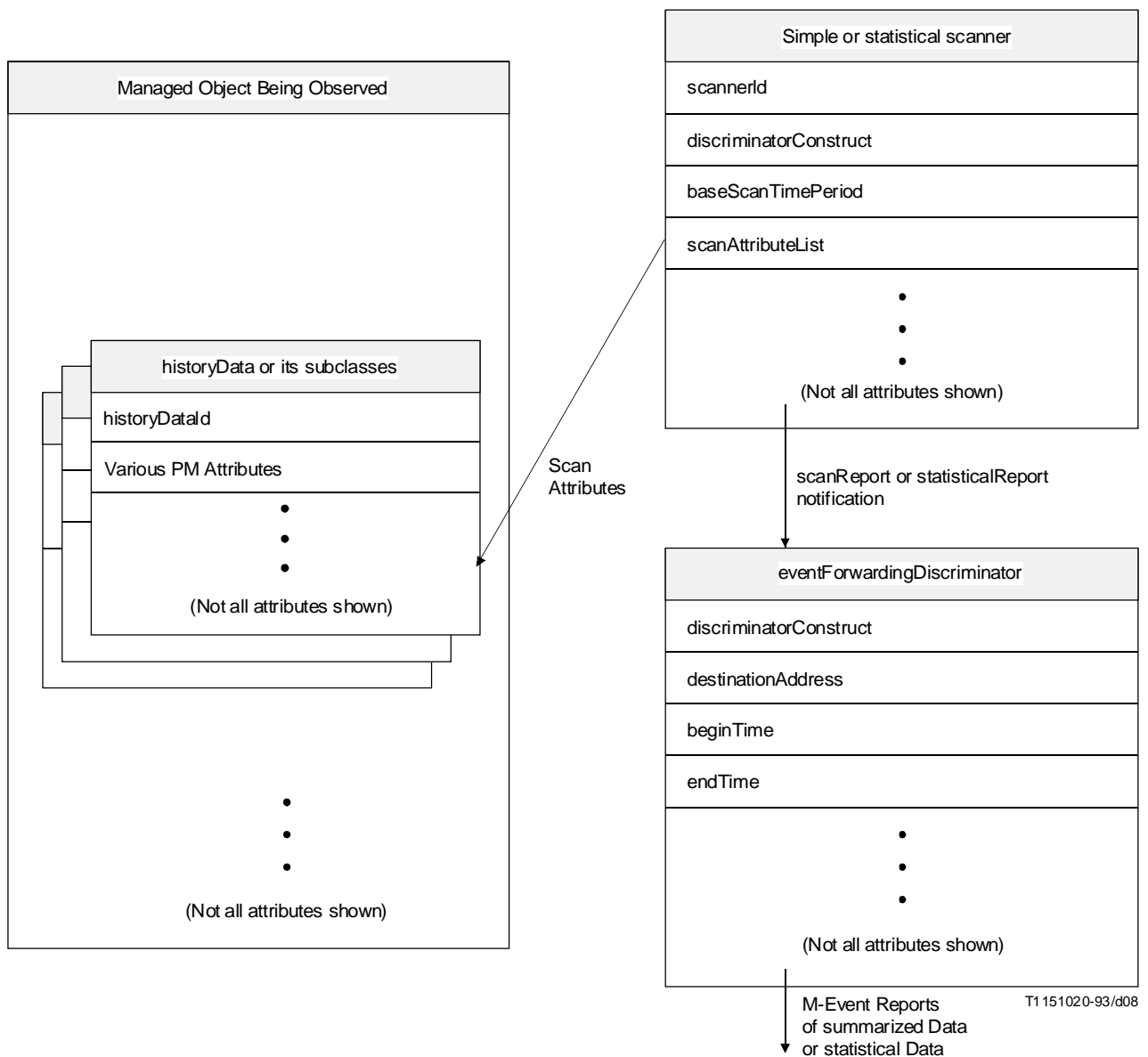


FIGURE B.5/Q.822
PM data summary or statistical reporting

Annex C

PM model alternative

(This annex does not form an integral part of this Recommendation)

An alternative method of providing performance management is described in this non-normative annex. This alternative requires the instantiation of fewer object instances.

The incorporation of this alternative into the standard is for further study.

C.1 Performance management model

The key part in this proposal is a new scanner, hereafter called Difference Scanner. For each scanned attribute the difference scanner can report the current observed attribute value or the difference between the current observed value and the previous interval value. The difference scanner can be used to scan current data and history data.

Because the difference scanner can scan non-resettable counters the scanned attributes can be defined in a Performance Data Component (PDC). This proposal does not require the use of `currentData` or `historyData` object classes. Furthermore the need for creation of current data and history data for different time intervals is eliminated because the time intervals are defined in the difference scanner.

The PDC object class is not intended to be instantiated. Technology specific subclasses can be instantiated or incorporated into technology specific managed objects via single or multiple inheritance. Instantiated subclasses will be contained in the managed objects in the same way as the current data (CD) object.

The PDC object class and the CD object class without any conditional packages in which the time interval is set to "infinity" are very similar.

The PDC object class uses the same mechanism for thresholding as the CD object class except that for each thresholded counter type attribute a corresponding resettable attribute must be defined.

This proposal introduces additional tools for performance monitoring. Guidelines are therefore proposed to aid the developers of technology specific standards, in choosing an alternative. It is proposed that the approach be decided on an MO class by MO class basis. For a given managed object class the options are:

- 1) This MO class only allows PM using the CD mechanism.
- 2) This MO class only allows PM using the PDC mechanism.
- 3) Each Administration may decide (for this MO class) which method to use (it is an MO option).

C.2 Performance management information

The major additions to Recommendation Q.822 for this proposal are the following support objects and associated management information.

C.2.1 Difference scanner

differenceScanner MANAGED OBJECT CLASS

DERIVED FROM "Recommendation X.738:1992": homogeneousScanner;
CHARACTERIZED BY differenceScannerPkg PACKAGE
BEHAVIOUR differenceScannerBehaviour BEHAVIOUR
DEFINED AS

"The differenceScanner is a class of managed support objects that makes observations of the same set of attributes across all managed object instances selected. For each attribute scanned, the value of that attribute may be reported or the difference between the current value and the value at the previous interval may be reported. Handling of counter overflow is a system internal matter.

The difference scanner generates notifications. The notification includes the observed values or calculated difference values. If the attribute is listed in the differenceReport attribute list then the difference between the current observed value and the previous interval value is reported. If the attribute is not present in the differenceReport attribute list then observed value of the attribute is reported. The difference value can only be computed for attributes with INTEGER or REAL syntaxes.

The observed values or computed difference values for attributes specified by the scan attribute identifier list are included in the notification paired with their attribute identifiers. Observed values or computed difference values for attributes specified by the numeric attribute identifier array are included in the notification without their attribute identifiers.

The difference scanner generates the Scan Report Notification.

If either the scan attribute identifier list or the numeric attribute identifier array is empty, then the corresponding parameter is not included in the summary report.

The values of the numeric attribute identifier array, the difference report list, the once report attribute Id list, and the time stamp report mode attributes may only be modified by the manager system if the administrative state attribute has a value of "locked". If the attribute value change notification package is present, modification of these attributes shall result in an attribute value change notification.

The attributes in the lists of attribute identifiers (scan attribute identifier list and numeric attribute identifier array) are scanned and their value or difference value is included in the report. At least one of these lists shall always be specified. When the selected managed object does not have the attribute in the scan identifier list, the attribute value is absent in the result. If any attribute in the numeric attribute identifier array is absent, a value of NULL shall be present.

If a difference operation is required on an attribute and the value at the previous interval is not known, a value of NULL shall be reported. If the difference scanner was disabled or inactive and the previous interval was not scanned, a difference value of NULL shall be returned.

If the once report attribute Id list package is present, then its associated reporting behaviour is exhibited.

A scan report notification with the name of the scanned objects and observed or computed values is emitted at the end of each scan."

;;

ATTRIBUTES

numericAttributeIdArray GET-REPLACE,
differenceReportList GET-REPLACE;
NOTIFICATIONS
scanReport;;;

CONDITIONAL PACKAGES

once ReportAttributeListPackage PRESENT IF "once report is supported";

REGISTERED AS {q822ObjectClass x};

C.2.2 Performance data component

performanceDataComponent MANAGED OBJECT CLASS

DERIVED FROM "Recommendation X.721:1992": top
CHARACTERIZED BY performanceDataPkg **PACKAGE**
BEHAVIOUR performanceDataBehaviour **BEHAVIOUR**
DEFINED AS

"The performanceDataComponent is a class of managed objects that contain the current performance data. This object class is not intended to be instantiated. Subclasses of this object class can be instantiated or incorporated into technology specific managed objects via single or multiple inheritance.

Typically, the performance attributes may use the syntax type of either the counter attribute (with 'Count' syntax type) or it should be derived from the gauge attribute. Both the counter and gauge attributes are defined in CCITT Recommendation X.721. Typically counters are monotonically increasing.

Thresholding cannot be used with monotonically increasing counters. Thresholding may only be applied to gauges or counters with built-in resetting intervals. Any internal resetting intervals must be specified explicitly in the behavior of the attribute as the counters cannot be reset externally.

;;

CONDITIONAL PACKAGES

thresholdPkg

PRESENT IF "a Quality of Service Alarm Notification is to be emitted for threshold crossing.";

REGISTERED AS {q822ObjectClass x};

C.3 Attribute definitions

C.3.1 Difference report list

differenceReportList ATTRIBUTE

WITH ATTRIBUTE SYNTAX DifferenceReportList;

MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;

BEHAVIOUR differenceReportListBehaviour BEHAVIOUR

DEFINED AS

“This attribute is a list of identifiers of attributes for which difference operations are to be performed. The identified attributes must be of either the integer or real type.”

;;

REGISTERED AS {q822Attribute x};

C.4 Name binding definitions

differenceScanner-managedElement NAME BINDING

SUBORDINATE OBJECT CLASS differenceScanner and SUBCLASSES;

NAMED BY SUPERIOR OBJECT CLASS

"Recommendation M.3100:1992": managedElement AND SUBCLASSES;

WITH ATTRIBUTE scannerId;

CREATE

WITH-REFERENCE-OBJECT

WITH-AUTOMATIC-INSTANCE-NAMING;

DELETE

ONLY-IF-NO-CONTAINED-OBJECTS;

REGISTERED AS {q822NameBinding x};

C.5 Guidelines for the use of the model

This subclause will give some guidance on how technology specific standards should use the model. It should also give information on relevant factors in selecting options (particularly between the two major alternatives).

This standard is generic. This means that in order to apply the model to a specific technology, it may be necessary to subclass the managed objects and add technology specific attributes. Different technologies will have different requirements for performance management. Different Administrations may have different philosophies for performance management. In particular, some technologies and Administrations may require extensive and sophisticated performance management capabilities. Other technologies and Administrations may require more rudimentary performance management capabilities.

In order to accommodate those technologies and administrations requiring sophisticated performance management requirements and those with simpler performance management requirements, two approaches have been provided.

The first approach involves collecting of performance data using the current data managed object (or its subclasses). This model provides considerable flexibility. It also requires three times the number of object instances (assuming history data is used). The second approach is based on the Performance Data Component (PCD) object class. Technology specific subclasses can be instantiated or incorporated in the managed Object by single or multiple inheritance. Using this approach entails the acceptance of some restrictions.

The second approach is only applicable to the monitoring of managed object classes fitting the following restrictions:

- 1) Current data and history data are not an explicit requirement. The MOD approach does not provide for structured retention of data (so that you can go back later and retrieve individual items). It does permit the logging of complete reports.
- 2) The organization standardizing the monitored managed object class is willing to subclass the managed object class to inherit the necessary PM attributes.
- 3) The monitored attributes are of the gauge or simple counter type. More complex counter types (mean values, etc.) or counters with special behavior (unavailable seconds) require the CD approach.
- 4) Scheduled pm notifications directly from the managed object are not required.
- 5) Only limited thresholding activity is required. Thresholding is allowed on gauges. If thresholds are used on event counters, then the thresholding interval must be built into the monitored object definition or attribute definition.
- 6) Zero suppression is not required.
- 7) Open pm attribute lists are not required (no measurementListPkg). All monitored attributes must be explicitly defined.
- 8) Statistical operations on the counters (mean, min-max, etc.) are not supported in the present proposal.
- 9) The counters cannot be reset or locked administratively. They are read only. They cannot be disabled independently of the monitored object.

If these conditions are acceptable to a given technology, then the second approach is viable.

In specifying the technology specific standard, three approaches can be taken for each monitored object:

- 1) The monitored object can only be monitored using current data. In this case only the appropriate subclass of current and history data would be defined.
- 2) The monitored object can only be monitored by instantiated or inherited subclass of the PDC object.
- 3) Both approaches are provided. The monitoring approach is determined at object class instantiation. Both the current data subclass and the PDC subclass must be defined and included in the managed object as mutually excluding conditional packages. This approach provides maximum flexibility to an Administration, but could provide a challenge for the managing OS. In order to make the situation manageable, a restriction should be added that all instances of a managed object class within a given NE should adopt the same approach.