INTERNATIONAL TELECOMMUNICATION UNION

ITU-T X.754

(02/2000)

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

SERIES X: DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

OSI management – Management functions and ODMA functions

Enhanced event control function

ITU-T Recommendation X.754

(Formerly CCITT Recommendation)

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS AND OPEN SYSTEM COMMUNICATIONS

PUBLIC DATA NETWORKS	
Services and facilities	X.1–X.19
Interfaces	X.20–X.49
Transmission, signalling and switching	X.50–X.89
Network aspects	X.90–X.149
Maintenance	X.150–X.179
Administrative arrangements	X.180–X.199
OPEN SYSTEMS INTERCONNECTION	
Model and notation	X.200–X.209
Service definitions	X.210–X.219
Connection-mode protocol specifications	X.220–X.229
Connectionless-mode protocol specifications	X.230–X.239
PICS proformas	X.240–X.259
Protocol Identification	X.260–X.269
Security Protocols	X.270–X.279
Layer Managed Objects	X.280–X.289
Conformance testing	X.290–X.299
INTERWORKING BETWEEN NETWORKS	
General	X.300–X.349
Satellite data transmission systems	X.350–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	
Networking	X.600–X.629
Efficiency	X.630–X.639
Quality of service	X.640–X.649
Naming, Addressing and Registration	X.650–X.679
Abstract Syntax Notation One (ASN.1)	X.680–X.699
OSI MANAGEMENT	
Systems Management framework and architecture	X.700–X.709
Management Communication Service and Protocol	X.710–X.719
Structure of Management Information	X.720–X.729
Management functions and ODMA functions	**X.730–X.799**
SECURITY	X.800–X.849
OSI APPLICATIONS	
Commitment, Concurrency and Recovery	X.850–X.859
Transaction processing	X.860–X.879
Remote operations	X.880–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation X.754

Enhanced event control function

Summary

This Recommendation presents enhancement event control functions to handle the unavailability of the logging systems.

Source

ITU-T Recommendation X.754 was prepared by ITU-T Study Group 4 (1997-2000) and approved under the WTSC Resolution 1 procedure on 4 February 2000.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. 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 turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSC Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

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

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

ITU-T Recommendation X.754

Enhanced event control function

1 Scope

This function is an enhancement to the functionality specified in ITU-T X.734 | ISO/IEC 10164-5 and provides the capability for configuring managed systems to be able to retain events in case the connection to the managing system is lost. This function models the retention of event reports and the forwarding of these retained event reports by an aggregate of support managed objects called a **disseminator complex**. The disseminator complex consists of one **disseminator log** and at least one **disseminator queue** managed object instance. Event reports, which cannot be forwarded, due to communications failure conditions may be retained for subsequent transmission. The model for the enhanced event control function is built upon the model for logging defined in ITU-T X.735 | ISO/IEC 10164-6 and the event report management model defined in ITU-T X.734 | ISO/IEC 10164-5. However, the mechanisms and managed objects used to implement this functionality though derived from the above Recommendations are different from those specified there. This model will support forwarding of event reports under both normal (real-time) and failure conditions and may, therefore, be implemented as a sole mechanism as an alternative to the mechanisms specified in ITU-T X.734. However, this function is not a replacement for the functionality specified in ITU-T X.734 and systems not requiring storage of events during failure conditions need only support the simpler functionality specified in ITU-T X.734.

This Recommendation:

– establishes user requirements for the enhanced event control function;

– establishes models that relate the services provided by the function to user requirements;

– defines the services provided by the function;

– specifies the protocol that is necessary in order to provide the services;

– defines the relationship between the services and SMI operations and notifications;

– defines relationships with other systems management functions;

– specifies conformance requirements.

This Recommendation does not:

– define the nature of any implementation intended to provide the enhanced event control function;

– specify the manner in which management is accomplished by the user of the enhanced event control function;

– define the nature of any interactions which result in the use of the enhanced event control function;

– specify the services necessary for the establishment, normal and abnormal release of a management association;

– specify the authorization requirements for the use of the enhanced event control function or for any associated activity;

– define the managed objects related to the management of particular protocol machines.

2 Normative references

The following ITU-T Recommendations and other references 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 and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regulary published.

2.1 Identical Recommendations | International Standards

– ITU-T X.200 (1994) | ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model.*

– ITU-T X.210 (1993) | ISO/IEC 10731:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: Conventions for the definition of OSI services.*

– ITU-T X.680 (1997) | ISO/IEC 8824-1:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*

– ITU-T X.681 (1997) | ISO/IEC 8824-2:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification.*

– ITU-T X.682 (1997) | ISO/IEC 8824-3:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification.*

– ITU-T X.683 (1997) | ISO/IEC 8824-4:1998, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications.*

– ITU-T X.690 (1997) | ISO/IEC 8825-1:1998, *Information technology – ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).*

– ITU-T X.691 (1997) | ISO/IEC 8825-2:1998, *Information technology – ASN.1 encoding rules – Specification of Packed Encoding Rules (PER).*

– ITU-T X.701 (1997) | ISO/IEC 10040:1998, *Information technology – Open Systems Interconnection – Systems management overview.*

– ITU-T X.710 (1997) | ISO/IEC 9595:1998, *Information technology – Open Systems Interconnection – Common management information service.*

– ITU-T X.721 (1992) | ISO/IEC 10165-2:1992, *Information technology – Open Systems Interconnection – Structure of management information: Definition of management information.*

– ITU-T X.730 (1992) | ISO/IEC 10164-1:1993, *Information technology – Open Systems Interconnection – Systems Management: Object management function.*

– ITU-T X.731 (1992) | ISO/IEC 10164-2:1993, *Information technology – Open Systems Interconnection – Systems Management: State management function.*

– ITU-T X.735 (1992) | ISO/IEC 10164-6:1993, *Information technology – Open Systems Interconnection – Systems Management: Log control function.*

– ITU-T X.739 (1993) | ISO/IEC 10164-11:1994, *Information technology – Open Systems Interconnection – Systems Management: Metric objects and attributes.*

2.2 Paired Recommendations | International Standards equivalent in technical content

– ITU-T X.290 (1995), *OSI conformance testing methodology and framework for protocol Recommendations for ITU-T applications – General concepts.*

ISO/IEC 9646-1:1991, *Information technology – Open Systems Interconnection – Conformance testing methodology and framework – Part 1: General concepts.*

– ITU-T X.700 (1992), *Management framework for Open Systems Interconnection (OSI) for CCITT applications.*

ISO/IEC 7498-4:1989, *Information processing systems – Open Systems Interconnection – Basic Reference Model – Part 4: Management framework.*

3 Definitions

For the purposes of this Recommendation, the following definitions apply.

3.1 Basic reference model definitions

This Recommendation makes use of the following terms defined in ITU-T X.200 | ISO/IEC 7498-1:

a) open system;

b) systems management.

3.2 Service convention definitions

This Recommendation makes use of the following term defined in ITU-T X.210 | ISO/IEC 10731:

– primitive.

3.3 Management framework definitions

This Recommendation makes use of the following terms as defined in ITU-T X.700 and ISO/IEC 7498-4:

a) management information;

b) managed object;

c) systems-management-application-entity.

3.4 Systems management overview definitions

This Recommendation makes use of the following terms defined in ITU-T X.701 | ISO/IEC 10040:

a) agent role;

b) dependent conformance;

c) general conformance;

d) management support object;

e) manager role;

f) notification;

g) systems management functional unit;

h) systems management operation.

3.5 Common management information service definitions

This Recommendation makes use of the following terms defined in ITU-T X. 710 | ISO/IEC 9595:

a) attribute;

b) common management information services;

c) common management information service element.

3.6 OSI conformance testing definitions

This Recommendation makes use of the following term defined in ITU-T X.290 and ISO/IEC 9646-1:

– system conformance statement.

3.7 Definitions from ITU-T X.734

a) Discriminator;

b) event forwarding discriminator;

c) event report management function.

3.8 Definitions from ITU-T X.735

a) Log;

b) log record.

3.9 Additional definitions

3.9.1 disseminator complex: An aggregation of managed objects, when taken together provide the capability of storing and queuing event reports for multiple destinations. For example to support the delivery of event reports to systems which may be temporarily unavailable or when the momentary real-time event load exceeds the channel capacity.

3.9.2 disseminator log: A log that temporarily stores event reports to be forwarded by disseminator queue objects when the disseminator queue objects cannot forward event reports in real-time.

3.9.3 disseminator queue: A managed object that specifies a list of stored event reports and that is able to forward these stored event reports to a specified destination.

4 Abbreviations

This Recommendation uses the following abbreviations:

ASN.1	Abstract Syntax Notation One
CMIS	Common management information service
CMISE	Common management information service element
EFD	Event forwarding discriminator
ERF	Event reporting function
Id	Identifier
MAPDU	Management application protocol data unit
PDU	Protocol data unit
SMAE	Systems management application entity
SMFU	Systems management functional unit
SMI	Structure of management information

5 Conventions

This Recommendation defines services for the event report management function following the descriptive conventions defined in ITU-T X.210 | ISO/IEC 10731.

6 Requirements

The requirements to be satisfied are:

a) the definition of a flexible event report control service which will allow systems to select which event reports are to be sent to particular managing systems;

b) the specification of the destinations (e.g. the identities of managing systems) to which event reports are to be sent;

c) the specification of a mechanism to control the forwarding of event reports, for example, by suspending and resuming their forwarding;

d) the ability for an external managing system to modify the conditions used in the reporting of events;

e) the ability to designate a backup location to which event reports can be sent if the primary location is not available, i.e. when communication with the primary system cannot be established;

f) the ability to retain information about events for some period of time;

g) the ability to automatically forward the retained information when the destinations become available;

h) the ability to explicitly manage the retention characteristics;

i) the ability to specify the delivery and removal policy for the event reports;

j) the ability to hold information about available and unavailable destinations;

k) the ability to retrieve event records that have been retained while a destination is unavailable.

6.1 Use cases

6.1.1 Event report administration

Requirements met by this use case diagram: a, b, c, d, e, h, i. See Figure 1.

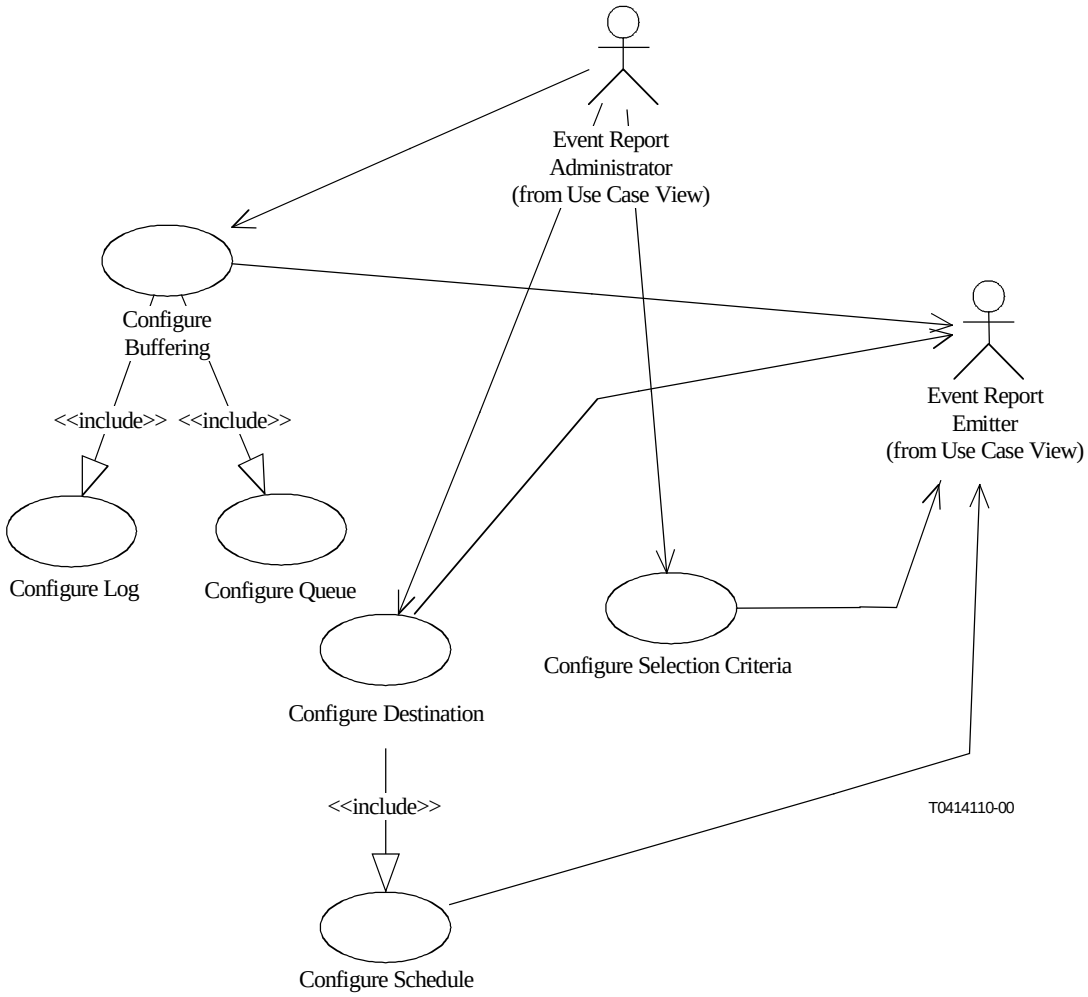

Figure 1/X.754 – Use case diagram for event report administration

6.1.2 Deliver event reports

Requirements met by this use case diagram: f, g, j, k. See Figure 2.

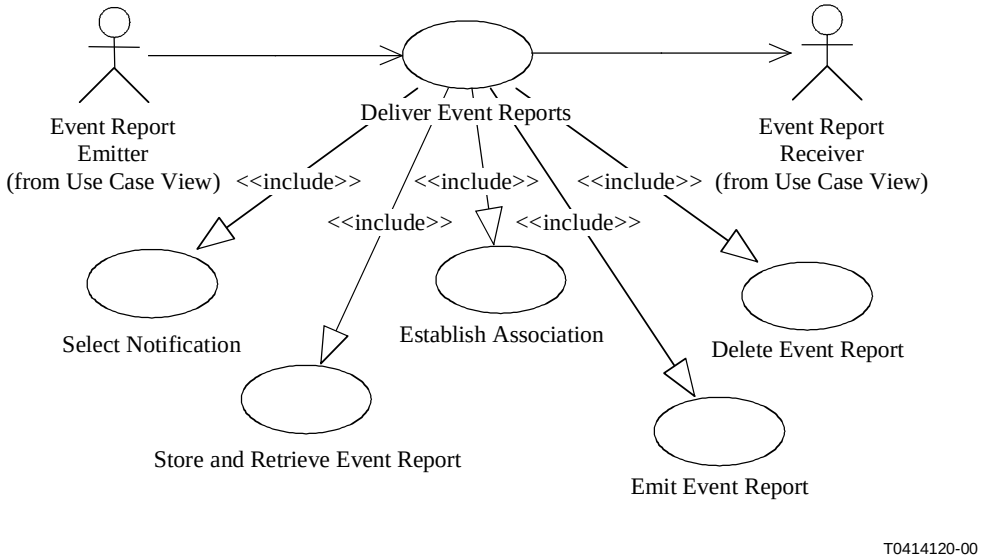

Figure 2/X.754 – Use case diagram for deliver event report

7 Analysis and design

7.1 Enhanced event control model

7.1.1 Overview of Objects

This function models the retention of event reports and the forwarding of these retained event reports by an aggregate of zero or more support managed objects called a **disseminator complex**. The disseminator complex consists of one **disseminator log** and at least one **disseminator queue** managed object instance. Event reports, which cannot be forwarded, because of communications failure conditions, may be retained for subsequent transmission. The model for the enhanced event control function is built upon the model for logging defined in ITU-T X.735 | ISO/IEC 10164-6 and the event report management model defined in ITU-T X.734 | ISO/IEC 10164-5. However, the mechanisms and managed objects used to implement this functionality though derived from the above Recommendations, are different from those specified there. This model will support forwarding of event reports under both normal (real-time) and failure conditions and may, therefore, be implemented as a sole mechanism as an alternative to the mechanisms specified in ITU-T X.734. Alternatively, systems not requiring storage of events during failure conditions need only support the mechanism specified in ITU-T X.734.

A particular managed system may support zero or more disseminator complexes. The number of disseminator complexes to be supported depends on:

1) whether the system is designed to support manageable configuration of system event storage capabilities;

2) the degree of flexibility is to be provided.

For instance, the support of multiple disseminator complexes would allow the configuration of different storage and retention characteristics for different event types.

Figure 3 is a schematic representation of the enhanced event control model.

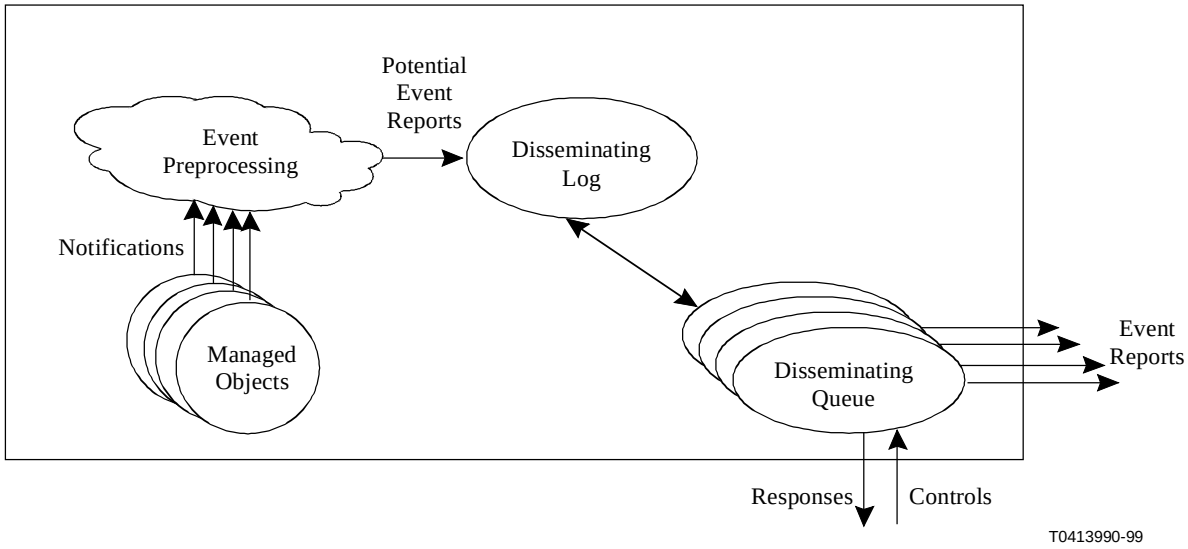

Figure 3/X.754 – Enhanced event control model

The disseminator log is a subclass of the log and has the same attributes, operations and behaviour with the additions described below.

Both the disseminator log and disseminator queue have discriminator constructs and time schedules. The relationship between these parameters needs to be controlled by the manager in order to get meaningful behaviour.

- In general, the active time intervals of the queue objects should be sub-intervals of the active time-intervals of the disseminator log. This is because events that occur outside the active time-intervals of the log will obviously not be seen by the disseminator queue.
- With respect to the discriminator construct settings, since the disseminator queue discriminator construct only receives those events that are passed by the disseminator log discriminator construct.

NOTE – It should be noted that it is not meaningful to specify that the discriminator construct in the disseminator queue select events that have already been filtered out by the disseminator log discriminator construct. In other words, the criteria specified in the disseminator log become a precondition for any events that will be seen by its disseminator queue objects.

The schedule attributes of the log and queue objects determine when events will be processed by these objects. Events occurring outside these time intervals are ignored, (i.e. they are not stored for later transmission). When a destination is unavailable during the time of occurrence of the event, the actual transmission of the event report may lie outside the scheduled interval in which the event occurred (i.e. the event report may be forwarded the next time the queue is on-duty).

If the disseminator queue has started to transmit events that had been queued and the schedule expires, then no new events are added to the queue but the events currently in the queue are transmitted in their entirety.

7.1.2 Overview of object behaviour

The following paragraphs describe the expected behaviour of the disseminator complex objects.

NOTE 1 – It needs to be noted that this Recommendation is not trying to specify detailed implementation of the actual information interchange between disseminator log and disseminator queue objects. It is the overall behaviour of disseminator complexes that is intended to be consistent.

NOTE 2 – This Recommendation does not constrain the physical implementation of disseminator logs and disseminator queues. For example, a disseminator log may consist of a combination of RAM and disk storage, and events may first be cached in memory and only if an indication is received that the events cannot be handled in real time will the events actually be written to disk storage.

The disseminator log will log all events that occur during its scheduled time period that meet the criteria specified in its discriminator construct. The disseminator log is associated with one or more disseminator queue objects. Whenever a new event is logged by the disseminator log, the disseminator log passes the record Id and associated event report to the disseminator queue objects. Each disseminator queue object then determines if the event report satisfies its own discriminator construct.

a) If the criteria are satisfied and the disseminator queue is on-duty then:
   1) If a destination system is available the event report is dispatched in the usual manner and the log is no longer required to retain this record, for this queue. This implies that if a disseminator queue object specifies a primary and a list of back-up destinations, the event report is considered to have been successfully delivered if any of the listed destinations is available.
   2) If a destination is unavailable, the log is required to retain this event record and the queue retains the record Id.

b) If the criteria are not satisfied or the disseminator queue is inactive the log is informed that the record is no longer required by this queue.

When all contained disseminator queue objects no longer have a need for a particular event report record, the record is autonomously deleted from the disseminator log.

There are two counts used to describe the behaviour:

– queue count: the total number of queues that will access a given disseminator log;

– user count: for each event record there is a counter of the number of queues for a given disseminator log that require a given event record to be stored in the log.

The disseminator log object will continually keep track, (user count), of how many disseminator queue objects it has, to permit, determining how many "event record no longer needed" indications are required before a given event can be deleted. When that event record is no longer needed by any of the queues associated with that disseminator log, the event record is automatically deleted. When a new disseminator queue is added, the queue count needs to be incremented by one for the events that are generated after creation of that disseminator queue. When a disseminator queue is deleted, the queue count needs to be decremented by one for the events that are generated after deletion of that disseminator queue. Prior to the disseminator queue object's deletion, it will indicate to the log which events are no longer needed.

NOTE 3 – The two counts (user count and queue count) discussed in this clause are not visible attributes of the event record. However, if a manager wants to delete event reports, the log can be queried by an action to determine how many destinations may still require this event report.

NOTE 4 – When all disseminator queues are delivering events in real-time, the lifetime of event records in the disseminator log will be ephemeral and the record Id can continuously be reused as long as no events are stored in the log; otherwise, record Ids are expected to monotonically increase with time. Thus, when the system is working in real time mode, it is meaningless for the manager to read the event records stored (i.e. those passing through the log) since these will disappear almost immediately. The retrieval is only meaningful if the system cannot deliver events and is storing these event reports for later transmission.

When a manager deletes event reports from a disseminator log, the log causes the corresponding record Ids to be deleted from the disseminator queue objects.

When the disseminator log with halt behaviour becomes full and no longer accepts new event reports, it shall increment the discard counter for every event report that it discards. When records

are subsequently deleted from the halting log, or transmission resumes, the log will generate an event containing the number of records discarded while it was halted and the discard count shall reset to zero. This notification is sent via an attribute-value change notification with the new value set to zero and the old value set to the discard count prior to its being reset.

NOTE 5 – In this case, the inclusion of the old value must be supported in the sender side implementation of the attributeValueChange notification, in order to contain the old value upon the reset behaviour

7.1.3 Disseminator log

The disseminator log contains the following additional attributes:

1) **Aging Period**: This attribute is part of a conditional package and defines the length of time for which an event record will be retained in the log. The aging period is measured relative to the time the record is inserted into the log. When an event record is deleted from the log, the disseminator log indicates to the disseminator queue objects that it no longer has the event record with that particular record Id available. A value of zero (0) indicates that event reports will only be delivered in real-time. A value of minus one (–1) indicates that no aging will occur as a part of this queue; i.e. events will be retained in the queue until they are either deleted by management action, overwritten if the queue wraps successfully transmitted by the queue. The default value is minus 1. Absence of the package also indicates that no aging will occur as part of this log, i.e. events will be retained in the log until they are either deleted by management action, overwritten if the log wraps or no longer needed by any of the queues. In general, the value of this attribute should be set to be equal to, or greater than, the greatest aging period of any of its contained queues.
2) **Discard Count**: This attribute counts the number of event records that were discarded due to a log full condition. The value is set to zero if the log is a wrapping log.

7.1.4 Disseminator queue

The disseminator queue is a managed object class derived from event forwarding discriminator, and has the same attributes, operations and behaviour with the additions described below. The disseminator queue allows management to configure policies for queuing and forwarding event reports which are not immediately deliverable to the specified destination.

The following constraints apply to the attributes and packages inherited from event forwarding discriminator:

1) The destination address attribute is permitted to only have a single destination attribute.

The following new packages and attributes are defined:

1) **Record List**: This attribute contains a list of the record Ids in the disseminator log that are still required by the disseminator queue object for transmission.
2) **Queue Discipline**: This attribute specifies whether, upon availability of the remote destination, the most recent or oldest events are sent first.
3) **Maximum Queue Size**: This attribute specifies the maximum size of the queue in terms of the number of records (i.e. event reports).
4) **Current Queue Size**: This attribute specifies the number of records that are currently waiting in the queue for transmission to a specified destination.
5) **Queue Full Action**: This attribute determines what happens when the maximum size of the queue has been reached. The value is either "wrap" or "halt", the semantics are similar to the log full action attribute defined in DMI. If the value is wrap, the oldest event record Id in the queue is removed when a new event record Id is to be inserted, else if the value is halt the new event record Id is not added to the queue. If the queue is a halting queue, then in response to each new record Id passed to the queue by the disseminator log it shall indicate to the disseminator log that it has no need for that record, until the queue enters a state where

it can accept additional events. If the queue is a wrapping queue the queue shall indicate that the event record corresponding to the deleted event record Id is no longer required.

6) **Queue Alarm Threshold**: Specifies the percentage of the maximum queue length at which a threshold alarm will be sent to a management system.

NOTE 1 – This threshold is only meaningful if the alarms will be processed by a different disseminator complex; otherwise, this alarm would be queued behind the existing alarms.

7) **Aging Period**: This attribute defines the length of time for which an event record Id will be retained in the queue. The aging period is measured relative to the time the record is inserted into the queue. When an event record Id is deleted from the queue, the disseminator queue indicates to the disseminator log that it no longer has any need for the associated event record. A value of zero (0) indicates that event reports will only be delivered in real-time. A value of minus one (–1) indicates that no aging will occur as a part of this queue; i.e. events will be retained in the queue until they are either deleted by management action, overwritten if the queue wraps successfully transmitted by the queue. The default value is minus 1. In general this time should be set to be less than the aging period specified in the disseminator log, if this time is larger than the corresponding time specified in the log it will have no effect as the event records will have been removed prior to expiry of the queues aging period

NOTE 2 – It may also be possible to define two queue objects that send events to the same destination but differ in the events they select and the aging applied to these events. This would allow different aging policies to be applied to different event types, e.g. attributeValueChange and securityAlarms could be aged over different time periods.

The disseminator queue object specifies a destination to which event reports are to be sent and manages the queue of event reports that have been retained in the disseminator log. The destination may include a list of back-up addresses. If this is the case the disseminator queue will deliver events to the destination that is currently active and will only require event retention by the disseminator log if the primary and all back-up destinations are unavailable. In case of a system with back-up addresses event transmission will resume as soon as one of the destinations becomes available. When retrieving the event record from the disseminator log, the disseminator queue object does not re-discriminate the event report (this assures that any events that occurred prior to a change in the disseminator queue's discriminator construct values are not lost), deletes the record Id and logging time attribute from the event record and transmits the remaining information as if it was an event report that emanated from the managed object. This design decision assures that if a manager changes either the schedule or discrimination criteria in a disseminator log or queue, it can only affect events that have occurred after the manager has changed the reporting characteristics.

When a disseminator queue object is deleted, it shall indicate to the disseminator log that it no longer has any need for the event records identified by the record Ids in its record list attribute.

If the mode indicator indicates that the event reports are to be sent in an unconfirmed mode, the disseminator queue object will indicate to the disseminator log immediately upon transmission of the event report that it no longer has any need for the record associated with the particular record. How the disseminator queue obtains this information is beyond the scope of this Recommendation. For unconfirmed notifications, the status of the association will be used to determine successful transmission. This means, if there is an association established, the notification will be sent and assumed to have been received by the manager. If there is no association, the event record will be queued.

If the mode indicator indicates that the event reports are to be sent in confirmed mode and an acknowledgement has been received from the remote end, the disseminator queue object will indicate to the disseminator log that corresponding event record is no longer needed. How the disseminator queue obtains this information is beyond the scope of this Recommendation.

7.2 Class diagrams

Figure 4 is a simplified class diagram, showing inheritance relationships between the object classes defined in this Recommendation.

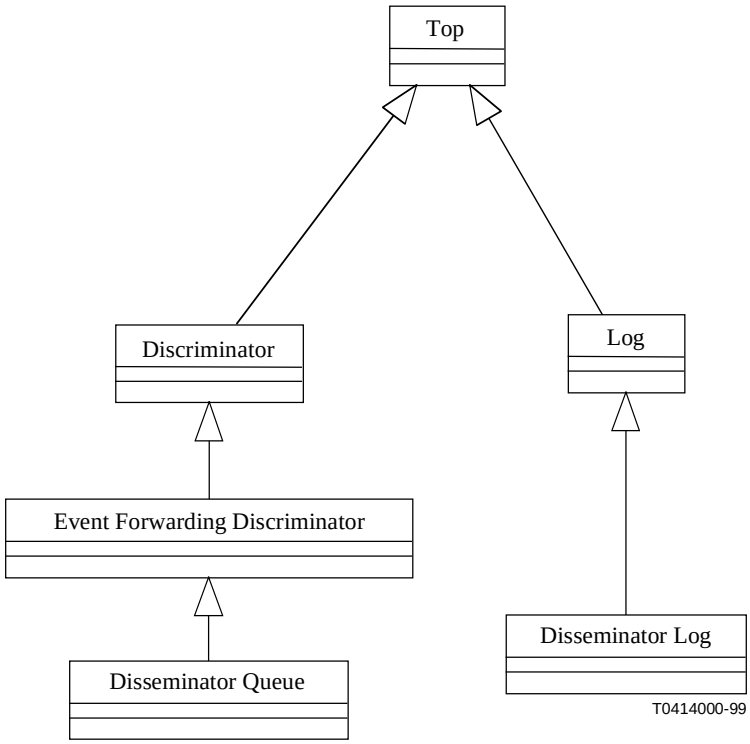

Figure 4/X.754 – Object class inheritance relationships

Figure 5 shows the containment relationships for the object classes defined in this Recommendation. The actor labelled as Event Report Emitter in this diagram represents the managed system from which the event reports are emitted. The actor labelled as Event Report Receiver represents managing systems which have the role of being a destination for receiving event reports. As shown in Figure 5, the receiver may have its own log to store events that it receives.

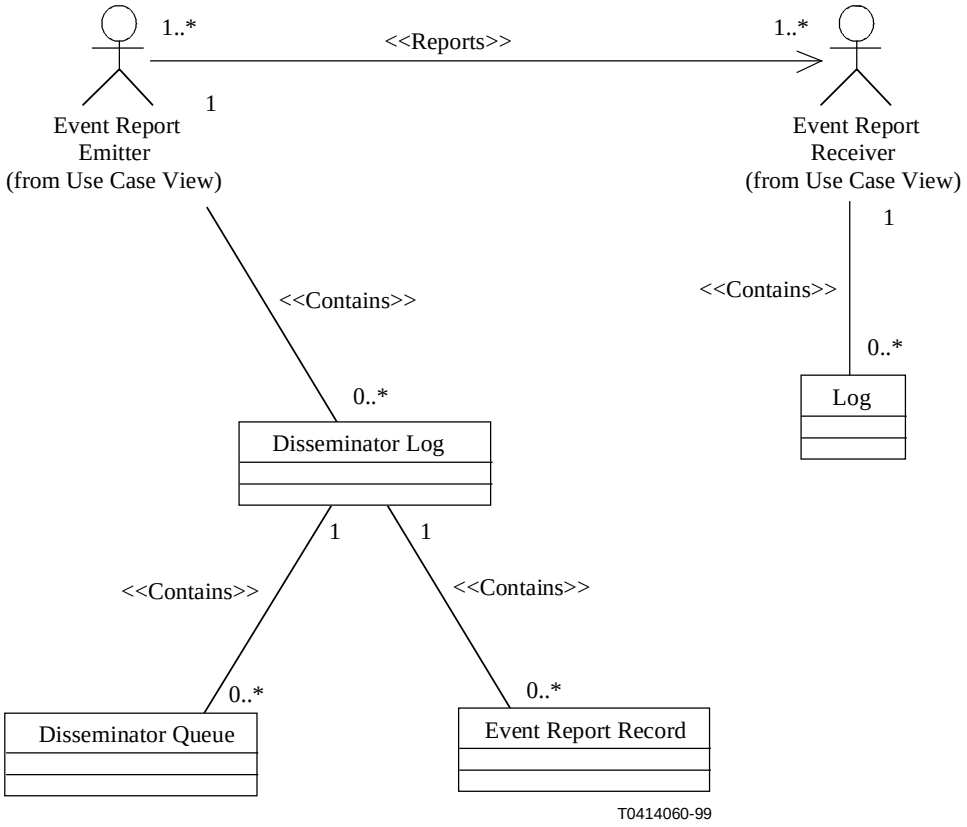

Figure 5/X.754 – System and class containment relationships

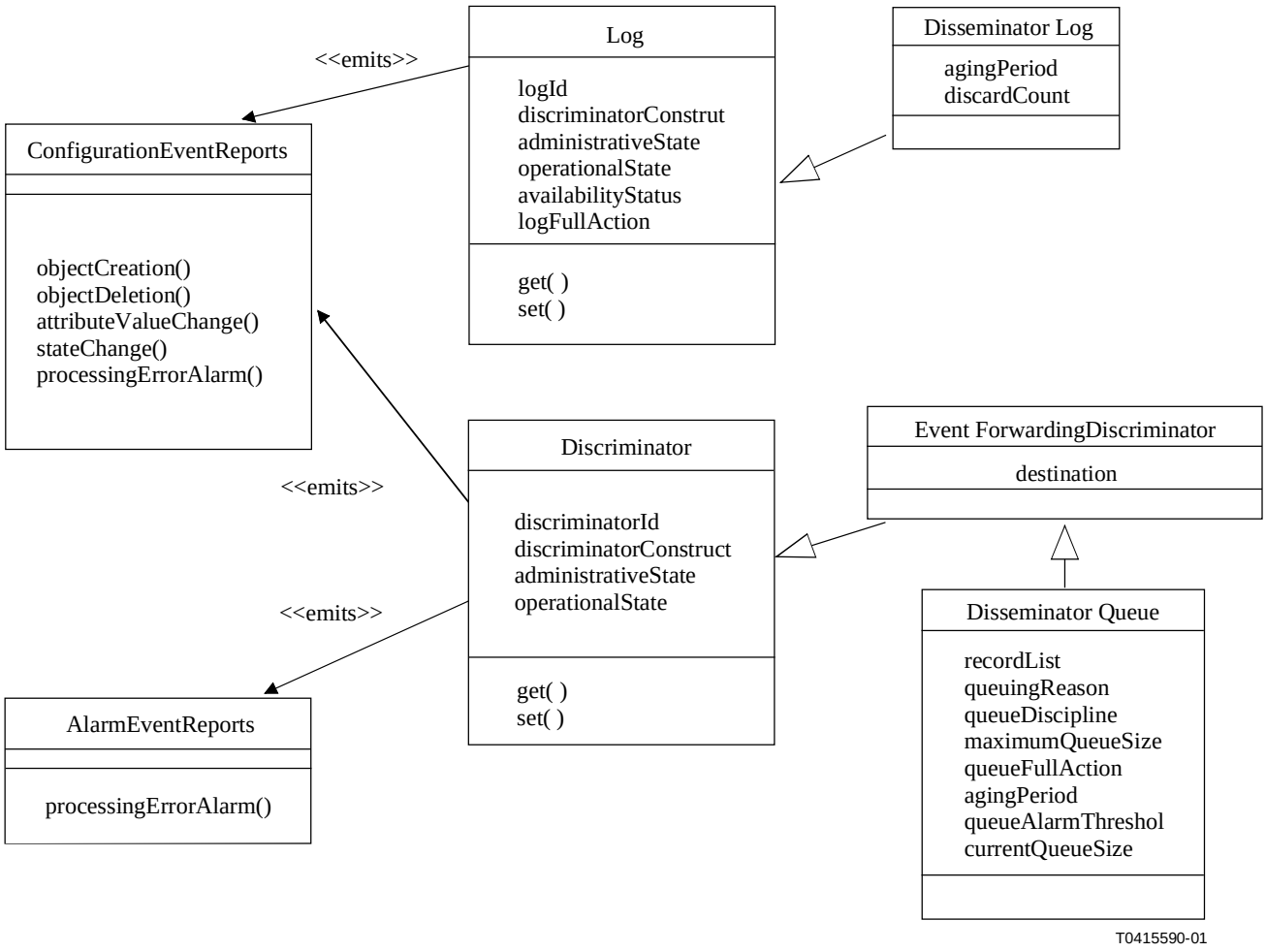

Figure 6/X.754 – Class diagram with attributes and notifications

7.3 Sequence diagrams

7.3.1 Real-time event emission in unconfirmed mode

See Figure 7.

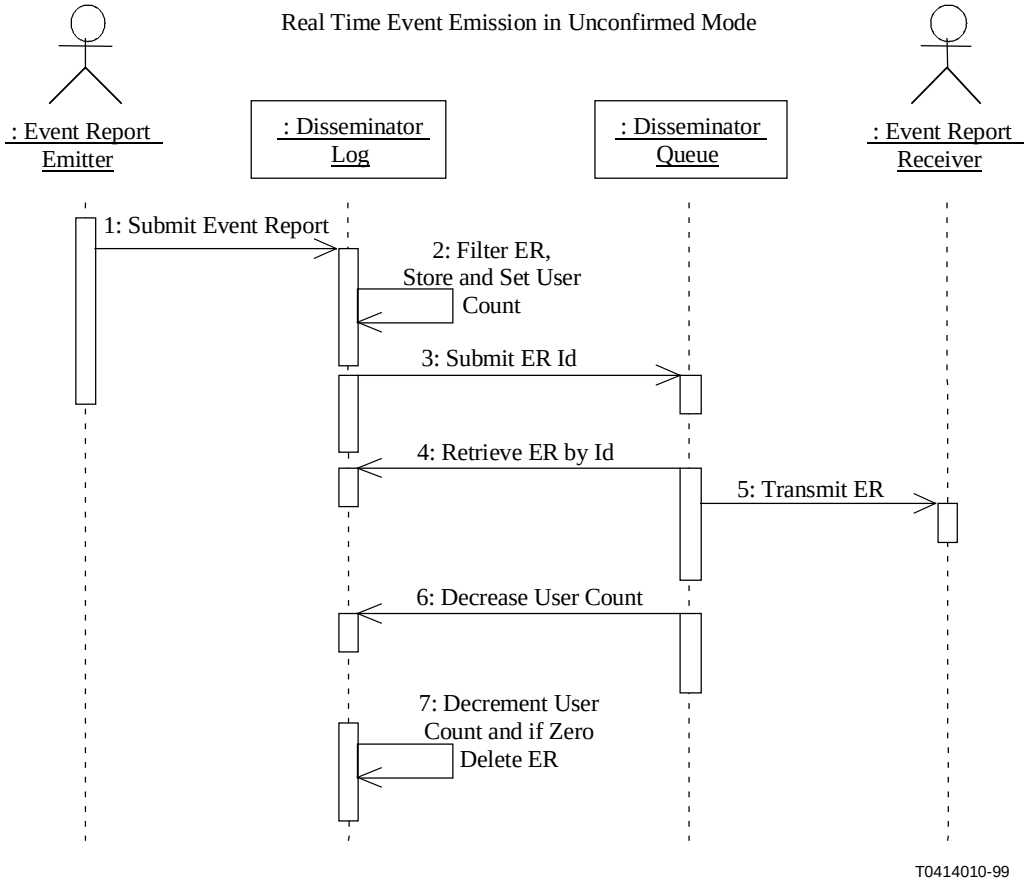

Figure 7/X.754 – Real-time event emission in unconfirmed mode

7.3.2 Real-time event emission in confirmed mode

See Figure 8.

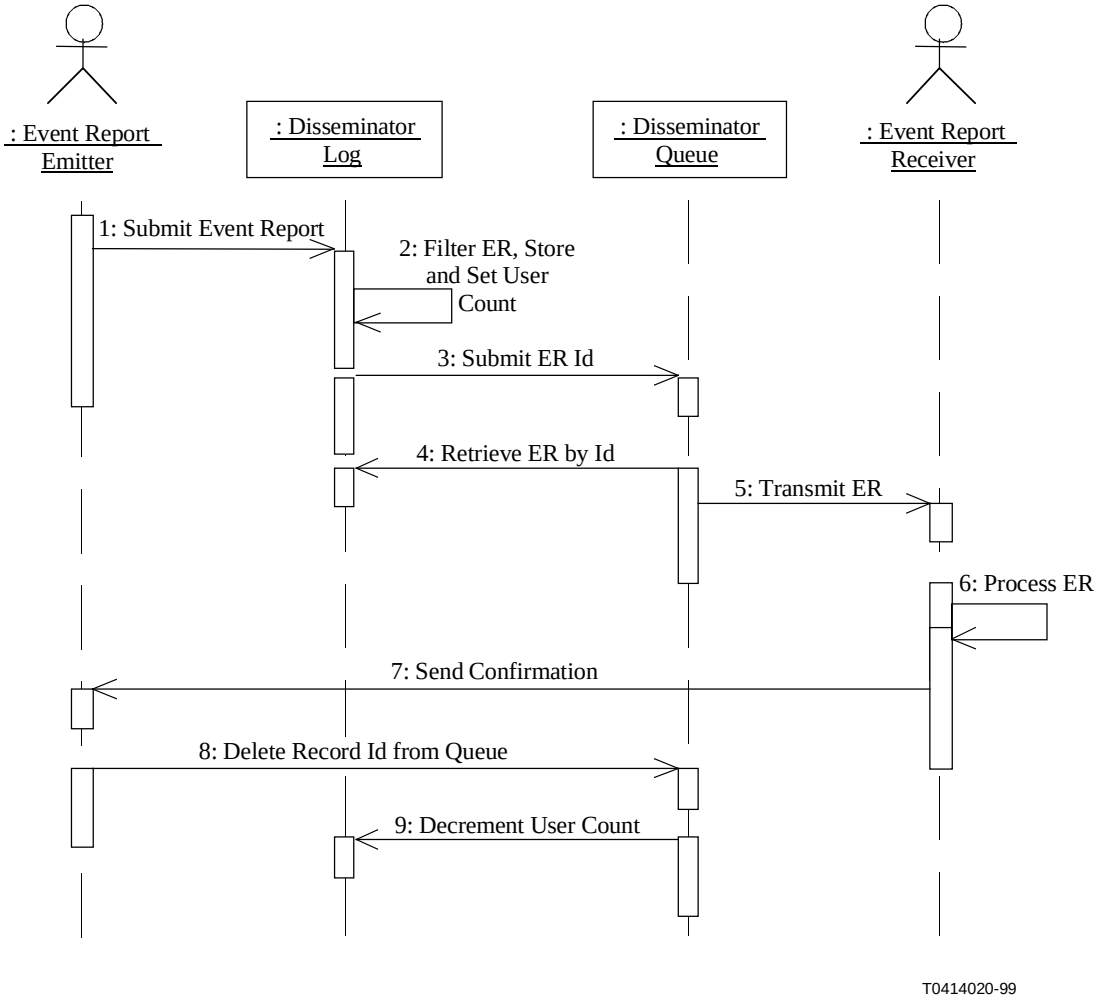

Figure 8/X.754 – Real-time event emission in confirmed mode

7.3.3 Delayed event emission in unconfirmed mode

See Figure 9.

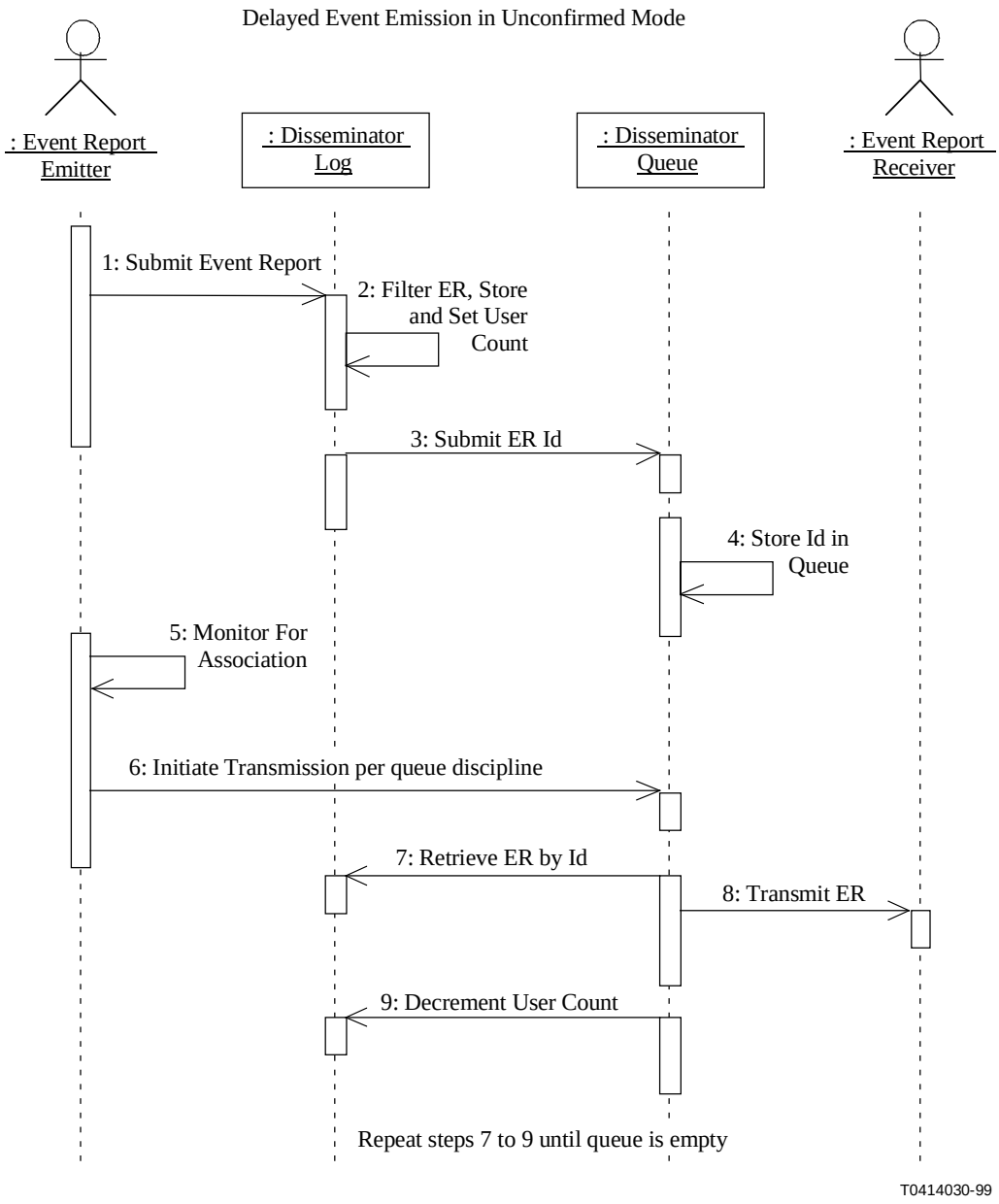

Figure 9/X.754 – Delayed event emission in unconfirmed mode

7.3.4 Event in queue time-out

See Figure 10.

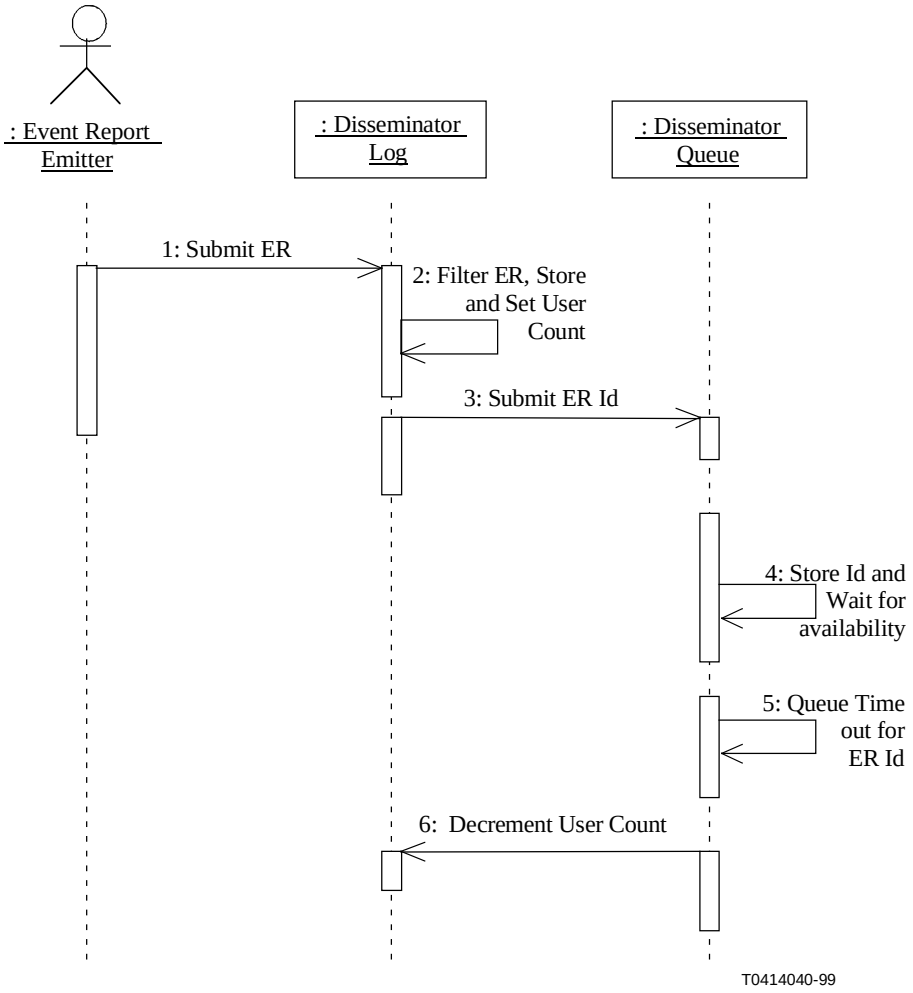

Figure 10/X.754 – Time-out of event in queue

7.3.5 Event in queue time-out for last user

See Figure 11.

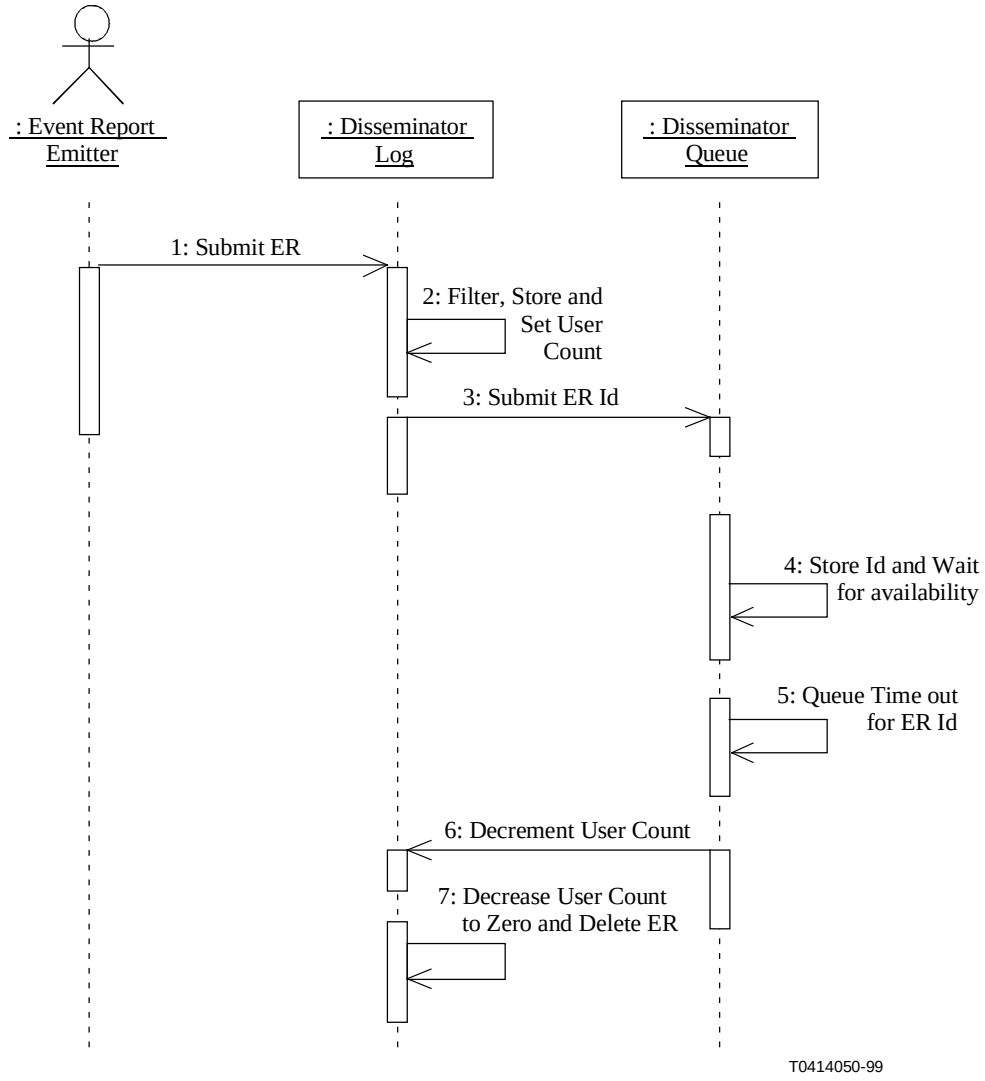

Figure 11/X.754 – Event in queue time-out for last user

8 Management information

8.1 Managed object classes

8.1.1 Disseminator log

The **disseminator log** managed object class is a subclass of log and provides for storage of event records for transmittal by disseminator queue objects.

disseminatorLog MANAGED OBJECT CLASS
DERIVED FROM log;
CHARACTERIZED BY
disseminatorLogPackage PACKAGE
BEHAVIOUR
disseminatorLogBehaviour BEHAVIOUR

```
        DEFINED AS "This managed object class is used to retain event reports when any destination to which
event reports are to be sent by the contained destination queues becomes unavailable. Additional details are
defined in ITU-T X.734 | ISO/IEC 10164-5. Detailed behaviour is specified in section 7.1.3";;
        ATTRIBUTES
        discardCount                            GET;

        NOTIFICATIONS
                AttributeValueChangeNotification;
        -- This notification is used to send the discard count when the log begins to accept new events for
        -- a log that had halted.

        ACTIONS
                getDestinations      ACTION
                        BEHAVIOUR
                        getDestinationBehaviour   BEHAVIOUR
                        DEFINED BY "This action retrieves the list of destinations to which event reports
                corresponding to a particular record Id will be sent. Multiple records may be identified in the same
                request. If no record Id is specified then the action applies to all records currently in the log.;;
                MODE CONFIRMED;
                WITH INFORMATION SYNTAX EnhancedErASN1Productions.GetDestinationArgument;
                WITH REPLY SYNTAX EnhancedErASN1Productions.GetDestinationsResult;

CONDITIONAL PACKAGES
        agingPackage                PRESENT IF "supported by the implementation";

REGISTERED AS { eecObjectClass 1};
```

8.1.2 Disseminator queue

The **disseminator queue** managed object class is a subclass of event forwarding discriminator and provides for storage of event record Ids for transmittal of event reports by disseminator queue objects.

```
disseminatorQueue                       MANAGED OBJECT CLASS
DERIVED FROM                    eventForwardingDiscriminator;
CHARACTERIZED BY
disseminatorQueuePackage                PACKAGE
        BEHAVIOUR
        disseminatorLogBehaviour        BEHAVIOUR
        DEFINED AS "This managed object class is used to maintain a list of event report record Ids for a
        particular destination that is unavailable. Detailed behaviour is specified in section 7.1.4";;
        ATTRIBUTES
        recordIdList                                    GET-REPLACE,
        maximumQueueSize                                GET-REPLACE,
        currentQueueSize                                GET,
        queueAlarmThreshold                             GET-REPLACE,

        queueDiscipline                                 GET-REPLACE,
        queueFullAction                                 GET-REPLACE
        ,;;

        CONDITIONAL PACKAGES
        agingPackage                PRESENT IF "supported by the implementation";

REGISTERED AS {eecObjectClass 2};
```

8.2 Conditional packages

```
        agingPackage                PACKAGE
                BEHAVIOUR
                agingPackageBehaviour   BEHAVIOUR
```

DEFINED AS "This package is used to specify the period of time for which an object or event is to be aged. The aging period is measured relative to the time the record is inserted into the log. The action to be taken when the aging period expires needs to be specified in the object class using this package.";;

ATTRIBUTES
agingPeriod GET-REPALCE
DEFAULT VALUE -1;
REGISTERED AS {eecPackage 1};

8.3 Attributes

8.3.1 Aging period

The semantics of the **agingPeriod** attribute type are specified in the aging attribute in ITU-T X.734 | ISO/IEC 10164-5.

agingPeriod ATTRIBUTE
WITH ATTRIBUTE SYNTAX Attribute-ASN1Module.AgingPeriod;
MATCHES FOR EQUALITY;

REGISTERED AS { eecAttribute 1};

8.3.2 Queuing discipline

The semantics of the **queuingDiscipline** attribute type are specified in the queuing discipline attribute in ITU-T X.734 | ISO/IEC 10164-5.

queuingDiscipline ATTRIBUTE
WITH ATTRIBUTE SYNTAX Attribute-ASN1Module.QueuingDiscipline;
MATCHES FOR EQUALITY;

REGISTERED AS { eecAttribute 2};

8.3.3 Record ID list

The semantics of the **recordIdList** attribute type are specified in the record list attribute in ITU-T X.734 | ISO/IEC 10164-5.

recordIdList ATTRIBUTE
WITH ATTRIBUTE SYNTAX Attribute-ASN1Module.RecordIdList;
MATCHES FOR EQUALITY, SET-COMPARISON, SET-INTERSECTION;

REGISTERED AS { eecAttribute 3};

8.3.4 Current queue size

The semantics of the **currentQueueSize** attribute type are specified in the current queue size attribute in ITU-T X.734 | ISO/IEC 10164-5.

currentQueueSize ATTRIBUTE
WITH ATTRIBUTE SYNTAX Attribute-ASN1Module.QueueSize;
MATCHES FOR EQUALITY, ORDERING;

REGISTERED AS { eecAttribute 4};

8.3.5 Maximum queue size

The semantics of the **maximumQueueSize** attribute type are specified in the max queue size attribute in ITU-T X.734 | ISO/IEC 10164-5.

maximumQueueSize ATTRIBUTE
WITH ATTRIBUTE SYNTAX Attribute-ASN1Module.QueueSize;

MATCHES FOR EQUALITY, ORDERING;

REGISTERED AS { eecAttribute 5};

8.3.6 Queue alarm threshold

The semantics of the **queueAlarmThreshold** attribute type are specified in the queue alarm threshold attribute in ITU-T X.734 | ISO/IEC 10164-5.

queueAlarmThreshold ATTRIBUTE
WITH ATTRIBUTE SYNTAX Attribute-ASN1Module.CapacityAlarmThreshold;

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

REGISTERED AS {eecAttribute 6};

8.3.7 Queue full action

The semantics of the **queueFullAction** attribute type are specified in the queue full action attribute in ITU-T X.734 | ISO/IEC 10164-5.

queueFullAction ATTRIBUTE
WITH ATTRIBUTE SYNTAX Attribute-ASN1Module.LogFullAction
MATCHES FOR EQUALITY;

REGISTERED AS {eecAttribute 7};

9 ASN.1 module

ASN1DefinedTypesModule { itu-t(0) recommendation(0) x(24) eecmod(754) informationModel(0) asn1Modules(2) asn1DefinedTypesModule(0)}

DEFINITION IMPLICIT TAGS ::=

BEGIN

-- EXPORTS everything

IMPORTS

ObjectClass, ObjectInstance, Attribute FROM CMIP-1 { joint-iso-ccitt ms (9) cmip (1) modules (0) protocol (3) }
-- see ITU-T X.711
AdministrativeState, OperationalState, Management Extension, RecordId, Destination FROM Attribute-ASN1Module { joint-iso-ccitt ms(9) smi (3) part2 (2) asn1Module(1) 1}
-- see ITU-T X.721

-- OBJECT IDENTIFIERS
informationModel Objectidentifier::= { itu-t(0) recommendation(0) x(24) eecmod(754) informationModel(0}
eecManagedObjectClass OBJECT IDENTIFIER ::= {informationModel managedObjectClass(3)}
eecPackage OBJECT IDENTIFIER ::= {informationModel package(4)}
eecNameBinding OBJECT IDENTIFIER ::= {informationModel nameBinding(6)}
eecAttribute OBJECT IDENTIFIER ::= {informationModel attribute(7)}
eecAction OBJECT IDENTIFIER ::= {informationModel action(9)}
eecBehaviour OBJECT IDENTIFIER ::= {informationModel behaviour(11)}
eecNotification OBJECT IDENTIFIER ::= {informationModel notification(10)}

AgingPeriod ::= TimePeriod

SendToList::= SEQUENCE
{recordId RecordId,
DestinationList SET OF Destination}

```
GetDestinationArgument::= SEQUENCE OF RecordId

GetDestinationResult::= SEQUENCE OF SendToList

NonNegativeInteger ::= INTEGER (0..MAX)

QueuingDiscipline  ::= ENUMERATED
                { fifoTransmission (0),
                lifoTransmission (1)}

QueueSize ::= NonNegativeInteger -- contains a count of the number of records

RecordIdList          ::= SEQUENCE OF NonNegativeInteger

END
```

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks and open system communications**
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems

Geneva, 2001