INTERNATIONAL TELECOMMUNICATION UNION

CCITT X.710

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

DATA COMMUNICATION NETWORKS: OPEN SYSTEMS INTERCONNECTION (OSI); MANAGEMENT

COMMON MANAGEMENT INFORMATION SERVICE DEFINITION FOR CCITT APPLICATIONS

Recommendation X.710

Geneva, 1991

FOREWORD

The CCITT (the International Telegraph and Telephone Consultative Committee) is a permanent organ of the International Telecommunication Union (ITU). CCITT 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 Plenary Assembly of CCITT which meets every four years, establishes the topics for study and approves Recommendations prepared by its Study Groups. The approval of Recommendations by the members of CCITT between Plenary Assemblies is covered by the procedure laid down in CCITT Resolution No. 2 (Melbourne, 1988).

Recommendation X.710 was prepared by Study Group VII and was approved under the Resolution No. 2 procedure on the 22nd of March 1991.

CCITT NOTE

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

© ITU 1991

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.

Recommendation X. 710

COMMON MANAGEMENT INFORMATION SERVICE DEFINITION FOR CCITT APPLICATIONS [1)]

CONTENTS

1 *Scope*

2 *References*

3 *Definitions*

4 *Symbols and abbreviations*

5 *Conventions*

6 *Service overview*

7 *Functional units*

8 *Service definition*

9 *Sequencing information*

Annexes – A CMISE-service-user behaviour

1 Scope

This Recommendation defines an Application Service Element (the Common Management Information Service Element), which may be used by an application process in a centralized or decentralized management environment to exchange information and commands for the purpose of systems management, as defined by the OSI Management Framework in ISO/IEC 7498-4 [1]. This Recommendation is positioned in the application layer of Recommendation X.200 [2] and is defined according to the model provided by ISO/IEC 9545 [3].

This Recommendation defines:

– a set of service primitives that constitute the application service element;

– the parameters that are passed in each service primitive;

– any necessary information for the semantic description of each service primitive.

This Recommendation does not define:

– the nature of any implementation intended to provide the defined service;

– the semantics associated with the information or commands that are exchanged by means of the service;

– the manner in which management is accomplished by the user of the service;

– the nature of any interactions which result in the use of the service.

No requirement is made for conformance to this Recommendation.

---

1) Recommendation X.710 and ISO/IEC 9595: *Information technology – Open Systems Interconnection – Common management information service definition* were developed in close collaboration and are technically identical.

2 References

[1] ISO/IEC 7498-4: *Information processing systems – Open systems interconnection – Basic reference model – Part 4: Management framework*, 1989.

[2] CCITT Recommendation *Reference model of open systems interconnection for CCITT applications,* Blue Book, Fascicule VIII.4, Rec. X.200, ITU, Geneva, 1988.

[3] ISO/IEC 9545: *Information processing systems – Open systems interconnection – Application layer structure (ALS),* 1989

[4] CCITT Recommendation *Association control service definition for CCITT applications,* Blue Book, Fascicle VIII.4, Rec. X.217, ITU, Geneva, 1988.

[5] CCITT Recommendation *Open systems interconnection (OSI) Layer service definition conventions,* Blue Book, Fascicule VIII.4, Rec. X.210, ITU, Geneva, 1988.

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 Recommendation X.200 [2]:

a) application-service-element;

b) open system;

c) systems-management.

3.2 *Management framework defintions*

This Recommendation makes use of the following terms defined in ISO/IEC 7498-4 [1]:

a) managed object;

b) management information;

c) systems management application-entiry.

3.3 *ACSE definitions*

This Recommendation makes use of the following terms defined in Recommendation X.217 [4];

a) application-association;

b) application context;

c) association;

d) association-initiator.

3.4 *Service conventions definitions*

This Recommendation makes use of the following terms defined in Recommendation X.210 [5]:

a) confirm (primitive);

b) confirmed-service;

c) indication (primitive);

d) non-confirmed-service;

e) request (primitive);

f) response (primitive).

3.5 *Additional definitions*

3.5.1 **attribute**

A property of a managed object. An attribute has a value.

3.5.2 **common Management Information Service Element**

The particular application-service-element defined in this Recommendation.

3.5.3 **common Management Information services**

The set of services provided by the Common Management Information Service Element.

3.5.4 **CMISE-service-provider**

An abstraction of the totality of those entities which provide CMISE services to peer CMISE-service-users.

3.5.5 **CMISE-service-user**

The part of an application process that makes use of the Common Management Information Service Element.

3.5.6 **functional unit**

The unit of service used for the negotiation of service options.

3.5.7 **invoking CMISE-service-user**

The CMISE-service-user that invokes a management operation or notification.

3.5.8 **performing CMISE-service-user**

The CMISE-service-user that performs a management operation or notification invoked by a peer CMISE-service-user.

3.5.9 **set-valued attribute**

An attribute whose value is a mathematical set of values of the same type. Values in the set cannot be repeated and no ordering is implied.

4 Symbols and abbreviations

ACSE Association Control Service Element

ASE Application Service Element

CMIS Common Management Information service

CMISE Common Management Information Service Element

Conf Confirm

Ind Indication

Req Request

Rsp Response

5 Conventions

This Recommendation defines services for CMIS following the descriptive conventions defined in Recommendation X.210 [5]. In § 8, the definition of each CMIS service includes a table that lists the parameters of its primitives. The definition of parameters in the Rsp/Conf column of a table apply only to the confirmed service. For a given primitive, the presence of each parameter is described by one of the following values:

M the parameter is mandatory;

(=) the value of the parameter is equal to the value of the parameter in the column to the left;

U the use of the parameter is a service-user-option;

– the parameter is not present in the interaction described by the primitive concerned;

C the parameter is conditional. The condition(s) are defined by the text which describes the parameter.

6 Service overview

Management information services are used by application processes in peer open systems, to exchange information and commands for the purpose of systems management.

There are two types of information transfer service:

– a management notification service,

– a management operation service.

The Common Management Information service provides additional structuring facilities. These enable:

– multiple responses to confirmed operations to be "linked" to the operation by the use of a linked identification parameter;

– operations to be performed on multiple managed objects, selected to satisfy some criteria and be subject to a "synchronizing" condition.

The CMISE services are listed in Table 1/X.710.

TABLE 1/X.710

CMISE services

Service	Type
M-CANCEL-GET	Confirmed
M-EVENT-REPORT	Confirmed/non-confirmed
M-GET	Confirmed
M-SET	Confirmed/non-confirmed
M-ACTION	Confirmed/non-confirmed
M-CREATE	Confirmed
M-DELETE	Confirmed

6.1 *Association services*

This Recommendation does not provide separate services for the establishment and release of application associations. The CMISE-service-user relies on the services of Recommendation X.217 [4] for the control of application-associations.

During the association establishment phase, various ASEs may exchange initialisation information to establish an association using ACSE. The application context specifies the rules required for coordinating the information belonging to different ASEs, embedded in ACSE user information service parameters. The application context, presentation and session requirements are conveyed using parameters of the A-ASSOCIATE service.

The A-RELEASE and A-ABORT services of Recommendation X.217 [4] are used for the termination of an association. These may be invoked by either of the CMISE-service-users.

6.2 *Management notification services*

The definition of the notification and the consequent behaviour of the communicating entities is dependent upon the specification of the managed object which generated the notification and is outside the scope of the Common Management Information service. However, certain notifications are used frequently within the scope of systems management and CMIS provides the following definition of the common service that may be used to convey management information applicable to the notification.

The M-EVENT-REPORT service is invoked by a CMISE-service-user to report an event about a managed object to a peer CMISE-service-user. The service may be requested in a confirmed or a non-confirmed mode. In the confirmed mode, a reply is expected.

6.3 *Management operation services*

The definition of the operation and the consequent behaviour of the communicating entities is dependent upon the specification of the managed object at which the operation is directed and is outside the scope of the Common Management Information Service. However, certain operations are used frequently within the scope of systems management and CMIS provides the following definitions of the common services that may be used to convey management information applicable to the operations.

6.3.1 The M-GET service is invoked by a CMISE-service-user to request the retrieval of management information from a peer CMISE-service-user. The service may only be requested in a confirmed mode, and a reply is expected.

6.3.2 The M-SET service is invoked by a CMISE-service-user to request the modification of management information by a peer CMISE-service-user. The service may be requested in a confirmed or a non-confirmed mode. In the confirmed mode, a reply is expected.

6.3.3 The M-ACTION service is invoked by a CMISE-service-user to request a peer CMISE-service-user to perform an action. The service may be requested in a confirmed or a non-confirmed mode. In the confirmed mode, a reply is expected.

6.3.4 The M-CREATE service is invoked by a CMISE-service-user to request a peer CMISE-service-user to create an instance of a managed object. The service may only be requested in the confirmed mode, and a reply is expected.

6.3.5 The M-DELETE service is invoked by a CMISE-service-user to request a peer CMISE-service-user to delete an instance of a managed object. The service may only be requested in the confirmed mode, and a reply is expected.

6.3.6 The M-CANCEL-GET service is invoked by a CMISE-service-user to request a peer CMISE-service-user to cancel a previously requested and currently outstanding invocation of the M-GETservice. The service may only be requested in the confirmed mode and a reply is expected.

6.4 *Management information tree*

Management information may be viewed as a collection of managed objects, each of which has attributes, and may have defined events and actions. Names of instances of managed objects are arranged hierarchically in a management information tree.

It is conceivable that there may be dynamic changes to the management information tree and that this knowledge may not be instantaneously available to other open systems.

6.5 *Managed object selection*

Managed object selection involves two phases: scoping and filtering.

Scoping entails the identification of the managed object(s) to which a filter is to be applied.

Filtering entails the application of a set of tests to each member of the set of previously scoped managed objects to extract a subset.

The subset of scoped managed objects that satisfy the filter is selected for the operation.

Note – If no filter is specified, then the set of scoped managed objects is selected for the operation.

6.5.1 *Scoping*

The base managed object is defined as the root of the subtree of the management information tree from which the search is to commence. Four specifications of scoping level are defined:

a) the base object alone;

b) the n-th level subordinates of the base object;

c) the base object and all of its subordinates down to and including the n-th level;

d) the base object and all of its subordinates (whole subtree).

When the scope parameter is not specified, the scoped managed object is the object specified by the base object instance parameter.

Note – The base object is defined to be level zero.

6.5.2 *Filtering*

A filter is a set of one or more assertions about the presence or values of attributes in a scoped managed object. If the filter involves more than one assertion, the assertions are grouped together using logical operators. If the filter test succeeds for a given managed object, then that managed object is selected for performance of the operation.

6.5.3 *Synchronization*

A synchronization parameter is provided to allow the CMISE-service-user to indicate the manner in which operations are to be synchronized across managed object instances when multiple managed objects have been selected by the scope and filter mechanism. The CMISE-service-user may request one of two types of synchronization: atomic or best effort. Since the order in which object instances are selected by the filter is a local matter, synchronization based on order is not meaningful.

Note – CMIS does not provide a parameter to indicate synchronization across attributes within a managed object. This will be specified as part of the managed object behaviour and may vary for different attribute combinations and operations.

7 Functional units

The general service capabilities are designated as functional units, where functional units correspond to the support of service primitives or parameters.

7.1 *Kernel functional unit*

All of the CMISE services listed in Table 1/X.710, except for the M-CANCEL-GET service, are included in the kernel functional unit. For the services in the kernel functional unit, the linked identification parameter shall not be used unless the multiple reply functional unit is agreed for use on the association between the CMISE-service-users. The scope and synchronization parameters shall not be used unless the multiple object selection functional unit has been agreed. The filter parameter shall not be used unless the filter functional unit has been agreed.

7.2 *Additional functional units*

7.2.1 *Multiple object selection functional unit*

This functional unit makes available the use of the scope and synchronization parameters in the services in the kernel functional unit. These parameters are not present in the M-EVENT-REPORT and M-CREATE services.

If the multiple object selection functional unit is proposed, then the multiple reply functional unit shall also be proposed.

7.2.2 *Filter functional unit*

This functional unit makes available the use of the filter parameter in the services in the kernel functional unit. The filter parameter is not present in the M-EVENT-REPORT and M-CREATE services.

7.2.3 *Multiple reply functional unit*

This functional unit makes available the use of the linked identification parameter in the services in the kernel functional unit. The linked identification parameter is not present in the M-EVENT-REPORT and M-CREATE services.

Multiple replies to a single management operation may only occur if the invoking CMISE-service-user selects multiple managed objects or requests an M-ACTION operation for a single managed object in which the action is defined to produce multiple responses.

Note – The use of the multiple reply functional unit may result in a large amount of data to be returned. Currently, CMIS provides only the M-CANCEL-GET service for controlling the flow of data. Additional mechanisms for controlling data flow or for controlling an operation are for further study.

7.2.4 *Extended service functional unit*

This functional unit makes available presentation layer services in addition to the P-DATA service.

7.2.5 *Cancel get functional unit*

This functional unit makes available the use of the M-CANCEL-GET service.

8 Service definition

The CMISE services are listed in Table 1/X.710.

Parameters returned as part of the confirm primitive may occur as the result of a successful operation (these are described as "included in the success confirmation"), or as the result of an error condition (these are described as "included in the failure confirmation").

Some operations may report an error code. In the event of multiple errors with one of the errors being a security violation, the error code "access denied" shall be returned.

8.1 *Association services*

8.1.1 *Association establishment*

The A-ASSOCIATE service of Recommendation X.217 [4] is invoked by a CMISE-service-user to establish an association with a peer CMISE-service-user. Association establishment is the first phase of any instance of management information service activity.

Table 2/X.710 lists the parameters that are defined by this Recommendation to be the CMIS specific part of the user information parameter of the A-ASSOCIATE service. This information is specified by the association-initiator and exchanged when establishing an association. Exchange of this initialisation information is required prior to using management operation and notification services.

TABLE 2/X.710

A-ASSOCIATE user information

Parameter name	Req/Ind	Rsp/Conf
Functional units	U	U
Access control	U	U
User information	U	U

8.1.1.1 *Functional units*

When supplied by the initiating CMISE-service-user, this parameter specifies the set of additional functional units that the initiating CMISE-service-user is proposing for use on the association. When returned by the responding CMISE-service-user, this parameter specifies the set of additional functional units that the responding CMISE-service-user is proposing for use on the association.

When this parameter is not supplied, it is assumed that no additional functional unit is proposed.

Any additional functional unit that has been proposed by both CMISE-service users is agreed to be available for use on the association.

If the extended service functional unit is successfully negotiated, then presentation layer services other than the P-DATA service are available for use. Details of those other presentation services, and how they are used, are described in the definitions of the application context in use on the association.

8.1.1.2 *Access control*

This parameter is information of unspecified form to be used as input to access control functions in establishing default access privileges for all exchanges on the association. If a subsequent service request specifies an access control parameter, then the access privileges for that and only that invocation of the service are determined from that parameter.

8.1.1.3 *User information*

The initiating CMISE-service-user and/or the responding CMISE-service-user may optionally include user information on the request and/or response primitive respectively. The meaning of this parameter is application context specific.

8.1.2 *Association release*

The A-RELEASE service of Recommendation X.217 [4] is invoked by a CMISE-service-user to request the orderly termination of an association between peer application entities. This Recommendation does not specify any use of the parameters of the A-RELEASE service.

The A-ABORT service is invoked by a CMISE-service-user to request the abrupt termination of the association between peer application entities.

Table 3/X.710 lists the parameters that are defined by this Recommendation to be the CMIS specific part of the user information parameter of the A-ABORT service.

TABLE 3/X.710

A-ABORT user information

CMIS Parameter	A-ABORT Req/Ind
Abort source	M
User information	U

8.1.2.1 *Abort source*

The abort source parameter indicates the initiating source of the abort. It takes one of the following symbolic values:

– CMISE-service-provider,

– CMISE-service-user.

8.1.2.2 *User information*

The initiating source of the abort may include user information. The meaning of this parameter is application context specific.

8.2 *Management notification service*

The M-EVENT REPORT service is used by a CMISE-service-user to report an event to a peer CMISE-service-user. It is defined as a confirmed and a non-confirmed service.

8.2.1 *M-EVENT-REPORT parameters*

Table 4/X.710 lists the parameters for this service.

TABLE 4/X.710

M-EVENT-REPORT parameters

Parameter name	Req/Ind	Rsp/Conf
Invoke identifier	M	(M=)
Mode	M	–
Managed object class	M	U
Managed object instance	M	U
Event type	M	C(=)
Event time	U	–
Event information	U	–
Current time	–	U
Event reply	–	C
Errors	–	C

8.2.1.1 *Invoke identifier*

This parameter specifies the identifier assigned to the notification. It can be used to distinguish this notification from other notifications or operations that the CMISE-service-provider may have in progress.

8.2.1.2 *Mode*

This parameter specifies the mode requested for the operation. The possible values are:

– confirmed,

– non-confirmed.

8.2.1.3 *Managed object class*

This parameter specifies the class of the managed object in which the event occurred. It may be included in any confirmation.

8.2.1.4 *Managed object instance*

This parameter specifies the instance of the managed object in which the event occurred. It may be included in any confirmation.

8.2.1.5 *Event type*

This parameter specifies the type of event being reported. It may be included in the success confirmation and shall be included if the event reply parameter is included.

8.2.1.6 *Event time*

This parameter contains the time of generation of the event.

8.2.1.7 *Event information*

This parameter contains information that the invoking CMISE-service-user is able to supply about the event.

8.2.1.8 *Current time*

This parameter contains the time at which the response was generated. It may be included in the success confirmation.

8.2.1.9 *Event reply*

This parameter contains the reply to the event report. It may be included in the success confirmation.

8.2.1.10 *Errors*

This parameter contains the error notification for the operation. It shall be included in the failure confirmation. The following errors may occur:

- duplicate invocation: the invoke identifier specified was allocated to another notification or operation;
- invalid argument value: the event information value specified was out of range or otherwise inappropriate;
- mistyped argument: one of the parameters supplied has not been agreed for use on the association between the CMISE-service-users;
- no such argument: the event information specified was not recognized;
- no such event type: the event type specified was not recognized;

– no such object class: the class of the specified managed object was not recognized;

– no such object instance: the instance of the specified managed object was not recognized;

– processing failure: a general failure in processing the notification was encountered;

– resource limitation: the notification was not processed due to resource limitation;

– unrecognized operation: the operation is not one of those agreed between the CMISE-service-users.

8.2.2 *M-EVENT-REPORT procedures*

8.2.2.1 The invoking CMISE-service-user reports an event to a performing CMISE-service-user by issuing an M-EVENT-REPORT request primitive to the CMISE-service-provider.

8.2.2.2 If the CMISE-service-provider accepts the request, then it issues an M-EVENT-REPORT indication primitive to the performing CMISE-service-user. Otherwise, the CMISE-service-provider rejects the request and the following procedures do not apply.

8.2.2.3 In the confirmed mode, the performing CMISE-service-user reports acceptance or rejection of the M-EVENT-REPORT request primitive by issuing an M-EVENT-REPORT response primitive to the CMISE-service-provider.

8.2.2.4 In the confirmed mode, the CMISE-service-provider issues an M-EVENT-REPORT confirm primitive to the invoking CMISE-service-user.

8.3 *Management operation services*

8.3.1 *M-GET service*

The M-GET service is used by a CMISE-service-user to retrieve attribute values from a peer CMISE-service-user. It is defined as a confirmed service. This service may be cancelled by an invocation of the M-CANCEL-GET service.

8.3.1.1 *M-GET parameters*

Table 5/X.710 lists the parameters for this service.

TABLE 5/X.710

M-GET parameters

Parameter name	Req/Ind	Rsp/Conf
Invoke identifier	M	M
Linked identifier	–	C
Base object class	M	–
Base object instance	M	–
Scope	U	–
Filter	U	–
Acces control	U	–
Synchronization	U	–
Attribute identifier list	U	–
Managed object class	–	C
Managed object instance	–	C
Current time	–	U
Attribute list	–	C
Errors	–	C

8.3.1.1.1 *Invoke identifier*

This parameter specifies the identifier assigned to the operation. It can be used to distinguish this operation from other notifications or operations that the CMISE-service-provider may have in progress.

Each response shall have a unique invoke identifier; the final response shall have an invoke identifier equal to that of the invoke identifier provided in the indication primitive.

8.3.1.1.2 *Linked identifier*

If multiple replies are to be sent for this operation, then this parameter specifies the identification that is provided by the performing CMISE-service-user when those replies are returned. The linked identifier shall have the same value as that of the invoke identifier provided in the indication primitive.

8.3.1.1.3 *Base object class*

This parameter specifies the class of the managed object that is to be used as the starting point for the selection of managed objects to which the filter (when supplied) is to be applied.

8.3.1.1.4 *Base object instance*

This parameter specifies the instance of the managed object that is to be used as the starting point for the selection of managed objects to which the filter (when supplied) is to be applied.

8.3.1.1.5 *Scope*

This parameter indicates the subtree, rooted at the base managed object, which is to be searched. The levels of search that may be performed are

– the base object alone;

– the n-th level subordinates of the base object;

– the base object and all of its subordinates down to and including the n-th level;

– the base object and all of its subordinates.

The default scope is the base object alone.

8.3.1.1.6 *Filter*

This parameter specifies the set of assertions that defines the filter test to be applied to the scoped managed object(s).Multiple assertions may be grouped using the logical operators AND, OR and NOT. Each assertion may be a test for equality, ordering, presence, or set comparison. Assertions about the value of an attribute are evaluated according to the matching rules associated with the attribute syntax. If an attribute value assertion is present in the filter and that attribute is not present in the scoped managed object, then the result of the test for that attribute value assertion shall be evaluated as FALSE. The managed object(s) for which the filter test evaluates to TRUE is (are) selected for the application of the operation. If the filter is not specified, all of the managed objects included by the scope are selected.

8.3.1.1.7 *Access control*

This parameter is information of unspecified form to be used as input to access control functions.

8.3.1.1.8 *Synchronization*

This parameter indicates how the invoking CMISE-service-user wants the M-GET operations synchronized across the selected object instances. Two ways of synchronizing a series of operations are defined:

– atomic: all managed objects selected for the operation are checked to ascertain if they are able to successfully perform the operation. If one or more is not able to successfully perform the operation, then none perform it, otherwise all perform it;

– best effort: all managed objects selected for the operation are requested to perform it.

If this parameter is not supplied, best effort synchronization is performed. If the base managed object alone is selected for the operation, this parameter (if present) is ignored.

8.3.1.1.9 *Attribute identifier list*

This parameter contains a set of attribute identifiers for which the attribute values are to be returned by the performing CMISE-service-user. If this parameter is omitted, all attribute identifiers are assumed. The definitions of the attributes are found in the specification of the managed object class.

8.3.1.1.10 *Managed object class*

If the base object alone is specified, then this parameter is optional; otherwise it shall specify the class of the managed object whose attribute values are returned. It may be included in any confirmation.

8.3.1.1.11 *Managed object instance*

If the base object alone is specified, then this parameter is optional; otherwise it shall specify the instance of the managed object whose attribute values are returned. It may be included in any confirmation.

8.3.1.1.12 *Current time*

This parameter contains the time at which the response was generated. It may be included in the success confirmation.

8.3.1.1.13 *Attribute list*

This parameter contains the set of attribute identifiers and values that are returned by the performing CMISE-service-user. It shall be included in the success confirmation.

8.3.1.1.14 *Errors*

This parameter contains the error notification for the operation. It shall be included in the failure confirmation. The following errors may occur:

- access denied: the requested operation was not performed for reasons pertinent to the security of the open system;
- class instance conflict: the specified managed object instance is not a member of the specified class;
- complexity limitation: the requested operation was not performed because a parameter was too complex;
- duplicate invocation: the invoke identifier specified was allocated to another notification or operation;
- get list error: one or more attribute values were not read for one of the following reasons:
  - access denied: the requested operation was not performed for reasons pertinent to the security of the open system;
  - no such attribute: the identifier for the specified attribute or attribute group was not recognized.

The attribute values that could be read are returned:

- invalid filter: the filter parameter contains an invalid assertion or an unrecognized logical operator;
- invalid scope: the value of the scope parameter is invalid;
- mistyped argument: one of the parameters supplied has not been agreed for use on the association between the CMISE-service-users;
- no such object class: the class of the specified managed object was not recognized;
- no such object instance: the instance of the specified managed object was not recognized;

– operation cancelled: the operation was cancelled by an M-CANCEL-GET operation, and no further attribute values will be returned by this invocation of the M-GET service;

– processing failure: a general failure in processing the operation was encountered;

– resource limitation: the operation was not performed due to resource limitation;

– synchronization not supported: the type of synchronization specified is not supported;

– unrecognized operation: the operation is not one of those agreed between the CMISE-service-users.

8.3.1.2 *M-GET procedures*

8.3.1.2.1 The invoking CMISE-service-user requests a performing CMISE-service-user to retrieve attribute value(s) by issuing an M-GET request primitive to the CMISE-service-provider.

8.3.1.2.2 If the CMISE-service-provider accepts the request, then it issues an M-GET indication primitive to the performing CMISE-service-user. Otherwise, the CMISE-service-provider rejects the request and the following procedures do not apply.

8.3.1.2.3 If the operation cannot be performed, then the performing CMISE-service-user rejects the M-GET request by issuing an M-GET response primitive with the appropriate error code. In this case, the following procedures do not apply.

8.3.1.2.4 If only one response is to be generated, then procedures § 8.3.1.2.5, 8.3.1.2.6 and 8.3.1.2.7 shall be ignored.

8.3.1.2.5 The performing CMISE-service-user retrieves therequested attribute value(s) and generates a response which includes results and/or error notifications. The linked identifier shall be present in the service primitive, with its value to be set equal to the invoke identifier of the indication primitive, and the managed object class and instance shall be present.

8.3.1.2.6 The CMISE-service-provider issues an M-GET confirm primitive to the invoking CMISE-service-user which shall include the linked identifier and managed object class and instance.

8.3.1.2.7 Procedures § 8.3.1.2.5 and 8.3.1.2.6 shall be repeated until all replies have been completed.

8.3.1.2.8 The completion of the response is indicated by the performing CMISE-service-user issuing an M-GET response primitive which shall not contain the linked identifier, and, if there were linked responses generated by procedures § 8.3.1.2.5 and 8.3.1.2.6, which shall not contain the attribute list.

8.3.1.2.9 The CMISE-service-provider issues an M-GET confirm primitive to the invoking CMISE-service-user, completing the M-GET operation.

8.3.1.3 *M-CANCEL-GET service*

The M-CANCEL-GET service is used by an invoking CMISE-service-user to request the cancellation of a previously requested and currently outstanding invocation of the M-GET service by a peer CMISE-service-user. It is defined as a confirmed service.

8.3.1.3.1 *M-CANCEL-GET parameters*

Table 6/X.710 lists the parameters for this service.

TABLE 6/X.710

M-CANCEL-GET paramaters

Parameter name	Req/Ind	Rsp/Conf
Invoke identifier	M	M(=)
Get invoke identifier	M	–
Errors	–	C

8.3.1.3.1.1 *Invoke identifier*

This parameter specifies the identifier assigned to the operation. It can be used to distinguish this operation from other notifications or operations that the CMISE-service-provider may have in progress.

8.3.1.3.1.2 *Get invoke identifier*

This parameter specifies the identifier assigned to the previously requested and currently outstanding M-GET operation.

8.3.1.3.1.3 *Errors*

This parameter contains the error notification for the operation. It shall be included in the failure confirmation. The following errors may occur:

– duplicate invocation: the invoke identifier specified was already allocated to another notification or operation;

– mistyped operation: the get invoke identifier parameter did not refer to an M- GET operation;

– no such invoke identifier: the get invoke identifier parameter was not recognized;

– processing failure: a general failure in processing the operation was encountered;

– resource limitation: the operation was not performed due to resource limitation;

– unrecognized operation: the operation is not one of those agreed between the CMISE-service-users.

8.3.1.3.2 *M-CANCEL-GET procedures*

8.3.1.3.2.1 The invoking CMISE-service-user requests a performing CMISE-service-user to cancel a previously requested and currently outstanding M-GET operation by issuing an M-CANCEL-GET request primitive to the CMISE-service-provider.

8.3.1.3.2.2 If the CMISE-service-provider accepts the request, then it issues an M-CANCEL-GET indication primitive to the performing CMISE-service-user. Otherwise, the CMISE-service-provider rejects the request and the following procedures do not apply.

8.3.1.3.2.3 If the operation cannot be performed, then the performing CMISE-service-user rejects the M-CANCEL-GET request by issuing an M-CANCEL-GET response primitive with the appropriate error code. In this case, the following procedures do not apply.

8.3.1.3.2.4 The performing CMISE-service-user cancels the outstanding M-GET operation and issues an M-GET response primitive which shall contain the operation cancelled error and which shall not contain the managed object class and managed object instance parameters, and an M-CANCEL-GET response primitive to the CMISE-service-provider.

8.3.1.3.2.5 If there are any M-GET response primitives issued by the performing CMISE-service-user after the invoking CMISE-service-user issues the M-CANCEL-GET request primitive, but before the performing CMISE-service-user receives the M-CANCEL-GET indication primitive, then the invoking CMISE-service-user shall receive M-GET confirm primitives for those M-GET response primitives.

8.3.1.3.2.6 The CMISE-service-provider issues an M-GET confirm primitive to the invoking CMISE-service-user which shall include the operation cancelled error indication, completing the M-GET operation, and an M-CANCEL-GET confirm primitive to the invoking CMISE-service-user, completing the M-CANCEL-GET operation.

8.3.2 *M-SET service*

The M-SET service is used by an invoking CMISE-service-user to request the modification of attribute values by a peer CMISE-service-user. It is defined as a confirmed and a non-confirmed service.

8.3.2.1 M-SET parameters

Table 7/X.710 lists the parameters for this service.

TABLE 7/X.710

M-SET parameters

Parameter name	Req/Ind	Rsp/Conf
Invoke identifier	M	M
Linked identifier	–	C
Mode	M	–
Base object class	M	–
Base object instance	M	–
Scope	U	–
Filter	U	–
Access control	U	–
Synchronization	U	–
Managed object class	–	C
Managed object instance	–	C
Modification list	M	–
Attribute list	–	U
Current time	–	U
Errors	–	C

8.3.2.1.1 *Invoke identifier*

This parameter specifies the identifier assigned to the operation. It can be used to distinguish this operation from other notifications or operations that the CMISE-service-provider may have in progress.

Each response shall have a unique invoke identifier; the final response shall have an invoke identifier equal to that of the invoke identifier provided in the indication primitive.

8.3.2.1.2 *Linked identifier*

If multiple replies are to be sent for this operation, then this parameter specifies the identification that is provided by the performing CMISE-service-user when those replies are returned. The linked identifier shall have the same value as that of the invoke identifier provided in the indication primitive.

8.3.2.1.3 *Mode*

This parameter specifies the mode requested for the operation. The possible values are

– confirmed,

– non-confirmed.

8.3.2.1.4 *Base object class*

This parameter specifies the class of the managed object that is to be used as the starting point for the selection of managed objects to which the filter (when supplied) is to be applied.

8.3.2.1.5 *Base object instance*

This parameter specifies the instance of the managed object that is to be used as the starting point for the selection of managed objects to which the filter (when supplied) is to be applied.

8.3.2.1.6 *Scope*

This parameter indicates the subtree, rooted at the base managed object, which is to be searched. The levels of search that may be performed are:

– the base object alone;

– the n-th level subordinates of the base object;

– the base object and all of its subordinates down to and including the n-th level;

– the base object and all of its subordinates.

The default scope is the base object alone.

8.3.2.1.7 *Filter*

This parameter specifies the set of assertions that defines the filter test to be applied to the scoped managed object(s). Multiple assertions may be grouped using the logical operators AND, OR and NOT. Each assertion may be a test for equality, ordering, presence, or set comparison. Assertions about the value of an attribute are evaluated according to the matching rules associated with the attribute syntax. If an attribute value assertion is present in the filter and that attribute is not present in the scoped managed object, then the result of the test for that attribute value assertion shall be evaluated as FALSE. The managed object(s) for which the filter test evaluates to TRUE are selected for the application of the operation. If the filter is not specified, all of the managed objects included by the scope are selected.

8.3.2.1.8 *Access control*

This parameter is information of unspecified form to be used as input to access control functions.

8.3.2.1.9 *Synchronization*

This parameter indicates how the invoking CMISE-service-user wants the M-SET operations synchronized across the selected object instances. Two ways of synchronizing a series of operations are defined:

– atomic: all managed objects selected for the operation are checked to ascertain if they are able to successfully perform the operation. If one or more is not able to successfully perform the operation, then none perform it, otherwise all perform it;

– best effort: all managed objects selected for the operation are requested to perform it.

If this parameter is not supplied, best effort synchronization is performed. If the base managed object alone is selected for the operation, this parameter (if present) is ignored.

8.3.2.1.10 *Managed object class*

If the base object alone is specified, then this parameter is optional; otherwise it shall specify the class of the managed object whose attribute values were modified. It may be included in any confirmation.

8.3.2.1.11 *Managed object instance*

If the base object alone is specified, then this parameter is optional; otherwise it shall specify the instance of the managed object whose attribute values were modified. It may be included in any confirmation.

8.3.2.1.12 *Modification list*

This parameter contains a set of attribute modification specifications, each of which contains:

a) attribute identifier: the identifier of the attribute or attribute group whose value(s) is/are to be modified. An attribute group identifier shall only be specified when the set to default modify operator is specified;

b) attribute value: the value(s) to be used in the modification of the attribute. The use of this parameter is defined by the modify operator. This parameter is optional when the set to default modify operator is specified and if supplied, shall be ignored;

c) modify operator: the way in which the attribute value(s) (if supplied) is/are to be applied to the attribute. The possible operators are:

   – replace: the attribute value(s) specified shall be used to replace the current value(s) of the attribute;

   – add values: the attribute value(s) specified shall be added to the current value(s) of the attribute. This operator shall only be applied to a set-valued attribute and shall perform a set union (in the mathematical sense) between the current value(s) of the attribute and the attribute value(s) specified. Value(s) specified in the attribute value parameter which are already in the current value(s) of the attribute shall not cause an error to be returned;

   – remove values: the attribute value(s) specified shall be removed from the current value(s) of the attribute. This operator shall only be applied to a set-valued attribute and shall perform a set difference (in the mathematical sense) between the current value(s) of the attribute and the attribute value(s) specified. Value(s) specified in the attribute value parameter which are not in the current value(s) of the attribute shall not cause an error to be returned;

   – set to default: when this operator is applied to a single-valued attribute, the value of the attribute shall be set to its default value. When this operator is applied to a set-valued attribute, the value(s) of the attribute shall be set to their default value(s) and only as many values as defined by the default shall be assigned. When this operator is applied to an attribute group, each member of the attribute group shall be set to its default value(s). If there is no default value defined, the "invalid operation" error shall be returned.

The modify operator is optional, and if it is not specified, the replace operator shall be assumed.

8.3.2.1.13 *Attribute list*

This parameter contains a set of attributes (one for each of the attribute identifiers specified in the modification list), each with its value(s) following the modification. It may be included in the success confirmation.

8.3.2.1.14 *Current time*

This parameter contains the time at which the response was generated. It may be included in the success confirmation.

8.3.2.1.15 *Errors*

This parameter contains the error notification for the operation. It shall be included in the failure confirmation. The following errors may occur:

- access denied: the requested operation was not performed for reasons pertinent to the security of the open system;
- class instance conflict: the specified managed object instance is not a member of the specified class;
- complexity limitation: the requested operation was not performed because a parameter was too complex;
- duplicate invocation: the invoke identifier specified was allocated to another notification or operation;
- invalid filter: the filter parameter contains an invalid assertion or an unrecognized logical operator;
- invalid scope: the value of the scope parameter is invalid;
- mistyped argument: one of the parameters supplied has not been agreed for use on the association between the CMISE-service-users.
- no such object class: the class of the specified managed object was not recognized;
- no such object instance: the instance of the specified managed object was not recognized;
- processing failure: a general failure in processing the operation was encountered;
- resource limitation: the operation was not performed due to resource limitation;
- set list error: one or more attribute values were not modified for one of the following reasons:

  a) access denied: the requested operation was not performed for reasons pertinent to the security of the open system;

  b) invalid attribute value: the attribute value specified was out of range or otherwise inappropriate;

  c) invalid operator: the modify operator specified is not recognized;

  d) invalid operation: the modify operator specified may not be performed on the specified attribute;

  e) no such attribute: the identifier for the specified attribute was not recognized.

The attribute values that could be modified, were modified:

- synchronization not supported: the type of synchronization specified is not supported;
- unrecognized operation: the operation is not one of those agreed between the CMISE-service-users.

8.3.2.2 *M-SET procedures*

8.3.2.2.1 The invoking CMISE-service-user requests a performing CMISE-service-user to modify attribute value(s) by issuing an M-SET request primitive to the CMISE-service-provider.

8.3.2.2.2 If the CMISE-service-provider accepts the request, then it issues an M-SET indication primitive to the performing CMISE-service-user. Otherwise, the CMISE-service-provider rejects the request and the following procedures do not apply.

8.3.2.2.3 In the non-confirmed mode the following procedures shall be ignored.

8.3.2.2.4 If the operation cannot be performed, then the performing CMISE-service-user rejects the M-SET request by issuing an M-SET response primitive with the appropriate error code. In this case, the following procedures do not apply.

8.3.2.2.5 If only one response is to be generated, then procedures § 8.3.2.2.6, 8.3.2.2.7 and 8.3.2.2.8 shall be ignored.

8.3.2.2.6 The performing CMISE-service-user modifies the requested attribute value(s) and generates a response which includes results and/or error notifications. The linked identifier shall be present in the service primitive, with its value to be set equal to the invoke identifier of the indication primitive, and the managed object class and instance shall be present.

8.3.2.2.7 The CMISE-service-provider issues an M-SET confirm primitive to the invoking CMISE-service-user which shall include the linked identifier and managed object class and instance.

8.3.2.2.8 Procedures § 8.3.2.2.6 and 8.3.2.2.7 shall be repeated until all replies have been completed.

8.3.2.2.9 The completion of the response is indicated by the performing CMISE-service-user issuing an M-SET response primitive which shall not contain the linked identifier, and, if there were linked responses generated by procedures § 8.3.2.2.6 and 8.3.2.2.7, which shall not contain the attribute list.

8.3.2.2.10 The CMISE-service-provider issues an M-SET confirm primitive to the invoking CMISE-service-user, completing the M-SET operation.

8.3.3 *M-ACTION service*

The M-ACTION service is used by a CMISE-service-user to request a peer CMISE-service-user to perform an action on managed object(s). It is defined as a confirmed and a non-confirmed service.

8.3.3.1 *M-ACTION parameters*

Table 8/X.710 lists the parameters for this service.

TABLE 8/X.710

M-ACTION parameters

Parameter name	Req/Ind	Rsp/Conf
Invoke identifier	M	M
Linked identifier	–	C
Mode	M	–
Base object class	M	–
Base object instance	M	–
Scope	U	–
Filter	U	–
Managed object class	–	C
Managed object instance	–	C
Access control	U	–
Synchronization	U	–
Action type	M	C(=)
Action information	U	–
Current time	–	U
Action reply	–	C
Errors	–	C

8.3.3.1.1 *Invoke identifier*

This parameter specifies the identifier assigned to the operation. It can be used to distinguish this operation from other notifications or operations that the CMISE-service-provider may have in progress.

Each response shall have a unique invoke identifier; the final response shall have an invoke identifier equal to that of the invoke identifier provided in the indication primitive.

8.3.3.1.2 *Linked identifier*

If multiple replies are to be sent for this operation, then this parameter specifies the identification that is provided by the performing CMISE-service-user when those replies are returned. The linked identifier shall have the same value as that of the invoke identifier provided in the indication primitive.

8.3.3.1.3 *Mode*

This parameter specifies the mode requested for the operation. The possible values are:

– confirmed,

– non-confirmed.

8.3.3.1.4 *Base object class*

This parameter specifies the class of the managed object that is to be used as the starting point for the selection of managed objects to which the filter (when supplied) is to be applied.

8.3.3.1.5 *Base object instance*

This parameter specifies the instance of the managed object that is to be used as the starting point for the selection of managed objects to which the filter (when supplied) is to be applied.

8.3.3.1.6 *Scope*

This parameter indicates the subtree, rooted at the base managed object, which is to be searched. The levels of search that may be performed are:

– the base object alone;

– the n-th level subordinates of the base object;

– the base object and all of its subordinates down to and including the n-th level;

– the base object and all of its subordinates.

The default scope is the base object alone.

8.3.3.1.7 *Filter*

This parameter specifies the set of assertions that defines the filter test to be applied to the scoped managed object(s). Multiple assertions may be grouped using the logical operators AND, OR and NOT. Each assertion may be a test for equality, ordering, presence, or set comparison. Assertions about the value of an attribute are evaluated according to the matching rules associated with the attribute syntax. If an attribute value assertion is present in the filter and that attribute is not present in the scoped managed object, then the result of the test for that attribute value assertion shall be evaluated as FALSE. The managed object(s) for which the filter test evaluates to TRUE are selected for the application of the operation. If the filter is not specified, all of the managed objects included by the scope are selected.

8.3.3.1.8 *Managed object class*

If the base managed object alone is specified, then this parameter is optional; otherwise it shall specify the class of the managed object that performed the action. It may be included in any confirmation.

8.3.3.1.9 *Managed object instance*

If the base object alone is specified, then this parameter is optional; otherwise it shall specify the instance of the managed object that performed the action. It may be included in any confirmation.

8.3.3.1.10 *Access control*

This parameter is information of unspecified form to be used as input to access control functions.

8.3.3.1.11 *Synchronization*

This parameter indicates how the invoking CMISE-service-user wants the M-ACTION operations synchronized across the selected object instances. Two ways of synchronizing a series of operations are defined:

– Atomic: all managed objects selected for the operation are checked to ascertain if they are able to successfully perform the operation. If one or more is not able to successfully perform the operation, then none perform it, otherwise all perform it.

– Best effort: all managed objects selected for the operation are requested to perform it.

If this parameter is not supplied, best effort synchronization is performed. If the base managed object alone is selected for the operation, this parameter (if present) is ignored.

8.3.3.1.12 *Action type*

This parameter specifies a particular action that is to be performed. It may be included in the success confirmation and shall be included if the action reply parameter is included.

8.3.3.1.13 *Action information*

This parameter specifies extra information when necessary to further define the nature, variations, or operands of the action to be performed. The syntax and semantics of the parameter depend upon the action requested.

8.3.3.1.14 *Current time*

This parameter contains the time at which the response was generated. It may be included in the success confirmation.

8.3.3.1.15 *Action reply*

This parameter contains the reply to the action. It may be included in the success confirmation.

8.3.3.1.16 *Errors*

This parameter contains the error notification for the operation. It shall be included in the failure confirmation. The following errors may occur:

– access denied: the requested operation was not performed for reasons pertinent to the security of the open system;

– class instance conflict: the specified managed object instance is not a member of the specified class;

– complexity limitation: the requested operation was not performed because a parameter was too complex;

– duplicate invocation: the invoke identifier specified was allocated to another notification or operation;

– invalid argument value: the action information value specified was out of range or otherwise inappropriate;

– invalid filter: the filter parameter contains an invalid assertion or an unrecognized logical operator;

– invalid scope: the value of the scope parameter is invalid;

– mistyped argument: one of the parameters supplied has not been agreed for use on the association between the CMISE-service-users;

– no such action: the action type specified was not supported;

– no such argument: the action information specified was not supported;

– no such object class: the class of the specified managed object was not recognized;

– no such object instance: the instance of the specified managed object was not recognized;

– processing failure: a general failure in processing the operation was encountered;

– resource limitation: the operation was not performed due to resource limitation;

– synchronization not supported: the type of synchronization specified is not supported;

– unrecognized operation: the operation is not one of those agreed between the CMISE-service-users.

8.3.3.2 *M-ACTION procedures*

8.3.3.2.1 The invoking CMISE-service-user requests a performing CMISE-service-user to perform an action on a set of managed objects by issuing an M-ACTION request primitive to the CMISE-service-provider.

8.3.3.2.2 If the CMISE-service-provider accepts the request, then it issues an M-ACTION indication primitive to the performing CMISE-service-user. Otherwise, the CMISE-service-provider rejects the request and the following procedures do not apply.

8.3.3.2.3 In the non-confirmed mode the following procedures shall be ignored.

8.3.3.2.4 If the operation cannot be performed, then the performing CMISE-service-user rejects the M-ACTION request by issuing an M-ACTION response primitive with the appropriate error code. In this case, the following procedures do not apply.

8.3.3.2.5 If only one response is to be generated, then procedures § 8.3.3.2.6, 8.3.3.2.7 and 8.3.3.2.8 shall be ignored.

8.3.3.2.6 The performing CMISE-service-user applies the action to the managed object and generates a response which includes a result or an error notification. The linked identifier shall be present in the service primitive, with its value to be set equal to the invoke identifier of the indication primitive, and the managed object class and instance shall be present.

8.3.3.2.7 The CMISE-service-provider issues an M-ACTION confirm primitive to the invoking CMISE-service-user which shall include the linked identifier and managed object class and instance.

8.3.3.2.8 Procedures § 8.3.3.2.6 and 8.3.3.2.7 shall be repeated until all replies have been completed.

8.3.3.2.9 The completion of the response is indicated by the performing CMISE-service-user issuing an M-ACTION response primitive which shall not contain the linked identifier, and, if there were linked responses generated by procedures § 8.3.3.2.6 and 8.3.3.2.7, which shall not contain the action result.

8.3.3.2.10 The CMISE-service-provider issues an M-ACTION confirm primitive to the invoking CMISE-service-user, completing the M-ACTION operation.

8.3.4 *M-CREATE service*

The M-CREATE service is used by an invoking CMISE-service-user to request a peer CMISE-service-user to create a new managed object instance, complete with its identification and the values of its associated management information, and simultaneously to register its identification. It is defined as a confirmed service.

8.3.4.1 *M-CREATE parameters*

Table 9/X.710 lists the parameters for this service.

TABLE 9/X.710

M-CREATE parameters

Parameter name	Req/Ind	Rsp/Conf
Invoke identifier	M	M(=)
Managed object class	M	U
Managed object instance	U	C
Superior object instance	U	–
Access control	U	–
Reference object instance	U	–
Attribute list	U	C
Current time	–	U
Errors	–	C

8.3.4.1.1 *Invoke identifier*

This parameter specifies the identifier assigned to the operation. It can be used to distinguish this operation from other notifications or operations that the CMISE-service-provider may have in progress.

8.3.4.1.2 *Managed object class*

This parameter specifies the class of the new managed object instance which is to be created by the performing CMISE-service-user. The performing CMISE-service- user assigns to the new managed object instance, a set of attribute values as specified by the definition of its class. If the reference object instance field is not supplied by the invoking CMISE-service-user, those attributes whose values are not assigned in the attribute list field are assigned a set of default values as specified by the definition of the new managed object's class. This parameter may be included in any confirmation.

8.3.4.1.3 *Managed object instance*

This parameter specifies the instance of the managed object which is to be registered by the performing CMISE-service-user. If neither the managed object instance nor the superior object instance parameters is supplied by the invoking CMISE-service-user, then the performing CMISE-service-user assigns a value to this identification of instance. This parameter may be included in the success confirmation and shall be included if it is not supplied by the invoking CMISE-service-user.

8.3.4.1.4 *Superior object instance*

This parameter identifies the existing managed object instance which is to be the superior of the new managed object instance. If this parameter is supplied by the invoking CMISE-service-user, then the managed object instance parameter shall not be supplied.

8.3.4.1.5 *Access control*

This parameter is information of unspecified form to be used as input to access control functions.

8.3.4.1.6 *Reference object instance*

When this parameter is supplied by the invoking CMISE-service-user, it must specify an existing instance of a managed object, called the reference object, of the same class as the managed object to be created. Attribute values associated with the reference object instance become the default values for those not specified by the attribute list parameter.

8.3.4.1.7 *Attribute list*

When this parameter is supplied by the invoking CMISE-service-user, it contains a set of attribute identifiers and values that the performing CMISE-service-user is to assign to the new managed object instance. These values override the values for the corresponding attributes derived from either the reference object (if the reference object instance field is supplied) or the default value set specified in the definition of the managed object's class. When returned by the performing CMISE-service-user, this parameter contains the complete list of all attribute identifiers and values that were assigned to the new managed object instance. It may be included in the success confirmation.

8.3.4.1.8 *Current time*

This parameter contains the time at which the response was generated. It may be included in the success confirmation.

8.3.4.1.9 *Errors*

This parameter contains the error notification for the operation. It shall be included in the failure confirmation. The following errors may occur:

- access denied: the requested operation was not performed for reasons pertinent to the security of the open system;
- class instance conflict: the specified managed object instance may not be created as a member of the specified class;
- duplicate invocation: the invoke identifier specified was allocated to another notification or operation;
- duplicate managed object instance: the new managed object instance value supplied by the invoking CMISE-service-user was already registered for a managed object of the specified class;
- invalid attribute value: an attribute value specified was out of range or otherwise inappropriate;
- invalid object instance: the object instance name specified implied a violation of the naming rules;
- missing attribute value: a required attribute value was not supplied, and a default value was not available;
- mistyped argument: one of the parameters supplied has not been agreed for use on the association between the CMISE-service-users;
- no such attribute: an attribute specified was not recognized;
- no such object class: the class of the specified managed object was not recognized;
- no such object instance: the instance of the specified superior managed object was not recognized;
- no such reference object: the reference object instance parameter was not recognized;
- processing failure: a general failure in processing the operation was encountered;
- resource limitation: the operation was not performed due to resource limitation;
- unrecognized operation: the operation is not one of those agreed between the CMISE-service-users.

8.3.4.2 *M-CREATE procedures*

8.3.4.2.1 The invoking CMISE-service-user requests the creation and registration of a new managed object instance by issuing an M-CREATE request primitive to the CMISE-service-provider.

8.3.4.2.2 If the CMISE-service-provider accepts the request, then it issues an M-CREATE indication primitive to the performing CMISE-service-user. Otherwise, the CMISE-service-provider rejects the request and the following procedures do not apply.

8.3.4.2.3 The performing CMISE-service-user creates and registers the new managed object instance or rejects the M-CREATE request, and issues an M-CREATE response primitive to the CMISE-service-provider.

8.3.4.2.4 The CMISE-service-provider issues an M-CREATE confirm primitive to the invoking CMISE-service-user.

8.3.5 *M-DELETE service*

The M-DELETE service is used by an invoking CMISE-service-user to request a peer CMISE-service-user to delete managed object instance(s) and to deregister their identification. It is defined as a confirmed service.

8.3.5.1 *M-DELETE parameters*

Table 10/X.710 lists the parameters for this service.

TABLE 10/X.710

M-DELETE parameters

Parameter name	Req/Ind	Rsp/Conf
Invoke identifier	M	M
Linked identifier	–	C
Base object class	M	–
Base object instance	M	–
Scope	U	–
Filter	U	–
Access control	U	–
Synchronization	U	–
Managed object class	–	C
Managed object instance	–	C
Current time	–	U
Errors	–	C

8.3.5.1.1 *Invoke identifier*

This parameter specifies the identifier assigned to the operation. It can be used to distinguish this operation from other notifications or operations that the CMISE-service-provider may have in progress.

Each response shall have a unique invoke identifier; the final response shall have an invoke identifier equal to that of the invoke identifier provided in the indication primitive.

8.3.5.1.2 *Linked identifier*

If multiple replies are to be sent for this operation, then this parameter specifies the identification that is provided by the performing CMISE-service-user when those replies are returned. The linked identifier shall have the same value as that of the invoke identifier provided in the indication primitive.

8.3.5.1.3 *Base object class*

This parameter specifies the class of the managed object that is to be used as the starting point for the selection of managed objects to which the filter (when supplied) is to be applied.

8.3.5.1.4 *Base object instance*

This parameter specifies the instance of the managed object that is to be used as the starting point for the selection of managed objects to which the filter (when supplied) is to be applied.

8.3.5.1.5 *Scope*

This parameter indicates the subtree, rooted at the base managed object, which is to be searched. The levels of search that may be performed are:

– the base object alone;

– the n-th level subordinates of the base object;

– the base object and all of its subordinates down to and including the n-th level;

– the base object and all of its subordinates.

The default scope is the base object alone.

8.3.5.1.6 *Filter*

This parameter specifies the set of assertions that defines the filter test to be applied to the scoped managed object(s). Multiple assertions may be grouped using the logical operators AND, OR and NOT. Each assertion may be a test for equality, ordering, presence, or set comparison. Assertions about the value of an attribute are evaluated according to the matching rules associated with the attribute syntax. If an attribute value assertion is present in the filter and that attribute is not present in the scoped managed object, then the result of the test for that attribute value assertion shall be evaluated as FALSE. The managed object(s) for which the filter test evaluates to TRUE are selected for deletion. If the filter is not specified, all of the managed objects included by the scope are selected.

8.3.5.1.7 *Access control*

This parameter is information of unspecified form to be used as input to access control functions.

8.3.5.1.8 *Synchronization*

This parameter indicates how the invoking CMISE-service-user wants the M-DELETE operations synchronized across the selected object instances. Two ways of synchronizing a series of operations are defined:

– atomic: all managed objects selected for the operation are checked to ascertain if they are able to successfully perform the operation. If one or more is not able to successfully perform the operation, then none perform it, otherwise all perform it;

– best effort: all managed objects selected for the operation are requested to perform it.

If this parameter is not supplied, best effort synchronization is performed. If the base managed object alone is selected for the operation, this parameter (if present) is ignored.

8.3.5.1.9 *Managed object class*

If the base object alone is specified, then this parameter is optional; otherwise it shall specify the class of the managed object which was deleted. It may be included in any confirmation.

8.3.5.1.10 *Managed object instance*

If the base object alone is specified, then this parameter is optional; otherwise it shall specify the instance of the managed object which was deleted. It may be included in any confirmation.

8.3.5.1.11 *Current time*

This parameter contains the time at which the response was generated. It may be included in the success confirmation.

8.3.5.1.12 *Errors*

This parameter contains the error notification for the operation. It shall be included in the failure confirmation. The following errors may occur:

- access denied: the requested operation was not performed for reasons pertinent to the security of the open system;
- class instance conflict: the specified managed object instance is not a member of the specified class;
- complexity limitation: the requested operation was not performed because a parameter was too complex;
- duplicate invocation: the invoke identifier specified was allocated to another operation;
- invalid filter: the filter parameter contains an invalid assertion or an unrecognized logical operator;
- invalid scope: the value of the scope parameter is invalid;
- mistyped argument: one of the parameters supplied has not been agreed for use on the association between the CMISE-service-users;
- no such object class: the class of the specified managed object was not recognized;
- no such object instance: the instance of the specified managed object was not recognized;
- processing failure: a general failure in processing the operation was encountered;
- resource limitation: the operation was not performed due to resource limitation;
- synchronization not supported: the type of synchronization specified is not supported;
- unrecognized operation: the operation is not one of those agreed between the CMISE-service-users.

8.3.5.2 *M-DELETE procedures*

8.3.5.2.1 The invoking CMISE-service-user requests a performing CMISE-service-user to delete managed object(s) by issuing anM-DELETE request primitive to the CMISE-service-provider.

8.3.5.2.2 If the CMISE-service-provider accepts the request, then it issues an M-DELETE indication primitive to the performing CMISE-service-user. Otherwise, the CMISE-service-provider rejects the request and the following procedures do not apply.

8.3.5.2.3 If the operation cannot be performed, then the performing CMISE-service-user rejects the M-DELETE request by issuing an M-DELETE response primitive with the appropriate error code. In this case, the following procedures do not apply.

8.3.5.2.4 If only one response is to be generated, then procedures § 8.3.5.2.5, 8.3.5.2.6 and 8.3.5.2.7 shall be ignored.

8.3.5.2.5 The performing CMISE-service-user deletes the specified managed object and generates a response. The linked identifier shall be present in the service primitive, with its value to be set equal to the invoke identifier of the indication primitive, and the managed object identifier shall be present.

8.3.5.2.6 The CMISE-service-provider issues an M-DELETE confirm primitive to the invoking CMISE-service-user which shall include the linked identifier and managed object identifier.

8.3.5.2.7 Procedures § 8.3.5.2.5 and 8.3.5.2.6 shall be repeated until all replies have been completed.

8.3.5.2.8 The completion of the response is indicated by the performing CMISE-service-user issuing an M-DELETE response primitive which shall not contain the linked identifier, and, if there were linked responses generated by procedures § 8.3.5.2.5 and 8.3.5.2.6, which shall not contain the delete result.

8.3.5.2.9 The CMISE-service-provider issues an M-DELETE confirm primitive to the invoking CMISE-service-user, completing the M-DELETE operation.

9 Sequencing information

9.1 *M-CREATE, M-DELETE services*

9.1.1 *Type of service*

These are confirmed services.

9.1.2 *Usage restrictions*

These services may only be invoked within the context of an established association.

9.1.3 *Disrupted service procedures*

These services do not disrupt any other service procedures.

9.1.4 *Disrupting service procedures*

These services are disrupted by the A-ABORT service procedure.

9.2 *M-EVENT-REPORT, M-SET, M-ACTION services*

9.2.1 *Type of service*

These are confirmed and non-confirmed services.

9.2.2 *Usage restrictions*

These services may only be invoked within the context of an established association.

9.2.3 *Disrupted service procedures*

These services do not disrupt any other service procedures.

9.2.4 *Disrupting service procedures*

These services are disrupted by the A-ABORT service procedure.

9.3 *M-GET service*

9.3.1 *Type of service*

This is a confirmed service.

9.3.2 *Usage restrictions*

This service may only be invoked within the context of anestablished association.

9.3.3 *Disrupted service procedures*

This service does not disrupt any other service procedure.

9.3.4 *Disrupting service procedures*

This service is disrupted by the M-CANCEL-GET and the A-ABORT service procedures.

9.4 *M-CANCEL-GET service*

9.4.1 *Type of service*

This is a confirmed service.

9.4.2 *Usage restrictions*

This service may only be invoked within the context of an established association.

9.4.3 *Disrupted service procedures*

This service disrupts the M-GET service procedures.

9.4.4 *Disrupting service procedures*

This service is disrupted by the A-ABORT service procedures.

ANNEX A

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

CMISE-SERVICE-USER BEHAVIOUR

A.1 *Background*

This annex describes the behaviour of the CMISE-service-user with respect to the processing of the management notifications and operations specified in this Recommendation.

A.2 *Event reporting*

An M-EVENT-REPORT notification is invoked by a CMISE-service-user whenever an event occurs that should be reported to a peer CMISE-service-user.

When the invoking CMISE-service-user has information associated with the occurrence of an event in a managed object, it supplies the identification of the managed object, the time that the event occurred, the type of event, and other event related information as the arguments to the M-EVENT-REPORT request primitive.

If the event report is invoked in the confirmed mode, a result or error primitive is issued by the performing CMISE-service-user to acknowledge its receipt to the invoking CMISE-service-user.

A.3 *Get management information*

When the performing CMISE-service-user receives an M-GET indication, it validates the semantics of the following parameters (no order implied):

– the base managed object;

– the optional access control information;

– the optional scope information;

– the optional filter information;

– the optional synchronization information;

– the list of attribute identifiers.

If any parameter is invalid, the operation terminates and the performing CMISE-service-user issues an error response. In the event of multiple errors being detected, with one of the errors being a security violation, the "access denied" error code shall be returned. If no errors are detected, the performing CMISE-service-user attempts to read the value(s) of the attribute(s) requested according to the filter, synchronization and security parameters, and returns the result(s) and/or error(s) as appropriate.

A.4 *Set management information*

When the performing CMISE-service-user receives an M-SET indication, it validates the semantics of the following parameters (no order implied):

– the base managed object;

– the optional access control information;

– the optional scope information;

– the optional filter information;

– the optional synchronization information;

– the list of attribute identifiers, optional values and optional modify operators.

If any parameter is invalid, the operation terminates and, in the confirmed mode, the performing CMISE-service-user issues an error response. In the event of multiple errors being detected, with one of the errors being a security violation, the "access denied" error code shall be returned. If no errors are detected, the performing CMISE-service-user attempts to modify the value(s) of the attribute(s) requested according to the filter, synchronization and security parameters, and, in the confirmed mode, returns the result(s) and/or error(s) as appropriate.

A.5 *Management action*

When the performing CMISE-service-user receives an M-ACTION indication, it validates the semantics of the following parameters (no order implied):

– the base managed object;

– the optional access control information;

– the optional scope information;

– the optional filter information;

– the optional synchronization information;

– the action type;

– the optional action information.

If any parameter is invalid, the operation terminates and, in the confirmed mode, the performing CMISE-service-user issues an error response. In the event of multiple errors being detected, with one of the errors being a security violation, the "access denied" error code shall be returned. If no errors are detected, the performing CMISE-service-user attempts to apply the action to the managed object(s) according to the filter, synchronization and security parameters, and, in the confirmed mode, returns the result(s) and/or error(s) as appropriate.

A.6 *Create a managed object*

When the performing CMISE-service-user receives an M-CREATE indication, it validates the semantics of the following parameters (no order implied):

- the managed object class and instance;
- the optional superior object instance;
- the optional access control information;
- the optional reference object;
- the list of attribute identifiers and values.

If any parameter is invalid, the operation terminates and the performing CMISE-service-user issues an error response. In the event of multiple errors being detected, with one of the errors being a security violation, the "access denied" error code shall be returned. If no errors are detected, the performing CMISE-service-user attempts to create the managed object, and returns the result and/or error as appropriate.

A.7 *Delete a managed object*

When the performing CMISE-service-user receives an M-DELETE indication, it validates the semantics of the following parameters (no order implied):

- the base managed object;
- the optional access control information;
- the optional scope information;
- the optional filter information;
- the optional synchronization information.

If any parameter is invalid, the operation terminates and the performing CMISE-service-user issues an error response. In the event of multiple errors being detected, with one of the errors being a security violation, the "access denied" error code shall be returned. If no errors are detected, the performing CMISE-service-user attempts to delete the managed object(s)according to the filter, synchronization and security parameters, and returns the result(s) and/or error(s) as appropriate.

A.8 *Cancel a get operation*

When the performing CMISE-service-user receives an M-CANCEL-GET indication, it validates the semantics of the get invoke identifier parameter. If the value of this parameter does not correspond to a currently outstanding M-GET operation, the operation terminates and the performing CMISE-service-user issues an M-CANCEL-GET error response. If no error is detected, the performing CMISE-service-user cancels the M-GET operation, returns an M-GET error response indicating that the operation has been cancelled, and returns an M-CANCEL-GET result indicating that the operation has been completed.