

Testing of Relationships in an OSI Management Information Base

Brigitte Baer*
 University of Frankfurt
 Department of Computer Science
 P.O.Box 111932
 D-60054 Frankfurt/Main
 Germany
 baer@informatik.uni-frankfurt.de

Alexander Clemm* †
 Munich Network Management Team
 University of Munich
 Department of Computer Science
 Leopoldstr. 11b
 D-80802 Munich
 Germany
 clemm@informatik.uni-muenchen.de

Abstract

In open distributed environments such as in OSI network management, a procedure of conformance testing is essential for increasing the level of confidence that component implementations from different sources actually meet their specifications as a prerequisite for their ability to interact as intended. This applies not only to OSI communication protocols but also to open management information. In particular, this includes relationships between managed objects, an aspect which has been largely ignored so far but which deserves particular attention and which we therefore focus on in this paper. Using the OSI General Relationship Model as a basis, we discuss how respective conformance requirements can be identified which serve as a starting point for the development of test cases.

1 Introduction

Conformance testing addresses the problem of how to determine whether the behavior that an implementation exhibits conforms to the behavior defined in its specification. The issue of conformance testing is of particular importance in open environments where components from different sources and manufacturers have to interwork. Here, a procedure of conformance testing can be substantial in increasing the level of confidence that an implementation acts according to its specification and that it will be able to interact in an open environment with other components as expected.

The problem of conformance testing also applies to the OSI network management arena for which openness of implementations of many different vendors and their ability to interwork is required. Besides conformance of management protocol implementations (such as CMIP [14]), for which ordinary protocol conformance testing methodologies [15] apply conformance of management information to its specification is a key issue. This involves the testing of the *Management Information Base* (MIB) with its *Managed Objects* (MOs) that represent the underlying network resources to be managed. Conformance of a MIB is an assumption for the proper functioning of management applications which operate on MOs and directly depend on the correct implementation of these MOs.

First approaches for testing the conformance of MOs can be found in [7,9,12]. Those approaches all have in common that they look at MOs in isolation; they do not cover aspects that involve

*The authors' work has also been supported by IBM European Networking Center, Heidelberg, Germany.

†A.Clemm is now with Raynet GmbH, Munich, Germany.

combinations of MOs or the context of the MIB as a whole. However, MOs are not isolated from each other but maintain relationships reflecting the interworking and dependencies among the underlying network resources. The importance of relationships has been acknowledged by work on the ISO *General Relationship Model* (GRM) [18] and other activities [5,3,19]. The GRM is essentially an ‘attachment’ to the basic information model. It allows for an additional specification of those aspects of MOs that relate them to other MOs in order to document those aspects in a more formal manner and to add structure to models of management information as a whole. Although the GRM has some shortcomings [5], it provides an important supplement to the OSI information model and will be referred to in the further discussion.

Independent of the existence of the GRM, relationship aspects must be considered in conformance testing as they are in any case present in a MIB. This has already been recognized in [1] where a ‘relationship view’ has been introduced as an integral part of a conformance testing methodology for MOs. Formal specification of relationship aspects using the GRM makes the task of determining their conformance requirements and deriving according test cases easier than basing the task on informal MO behavior specifications only. The purpose of this paper is to investigate the subject of relationship conformance testing with respect to the GRM. This includes to examine the conformance requirements that can be derived from the aspects specified in the GRM and to address the problems associated with the development of test cases for relationships.

To set the stage, we will first summarize the basic concepts of the GRM in section 2. A general knowledge of OSI management and the OSI information model with its *Guidelines for the Definition of Managed Objects* (GDMO) [16,17] is assumed. Section 3 gives an overview over conformance testing concepts. In section 4, we use a classification scheme to systematically identify relationship conformance requirements that result from those relationship aspects that are formally specified in the GRM. These requirements form the basis for the derivation of abstract test cases for relationships. The according process is explained in section 5 by a relationship example dealing with an ATM cross connection. Some conclusions are offered in section 6.

2 The general relationship model

The aim of the GRM is to provide additional specification means for the definition of relationships in a formal manner. This concerns for instance MO attributes referring to other MOs or constraints concerning the joint behavior of MOs [19] in behavior specifications. The representation and management of relationships per se as part of a MIB are like before based on the well known basic OSI management concepts. Thus, the GRM is an attempt to eliminate shortcomings associated with the specification of relationships between MOs in the conventional plain OSI information model while leaving it in itself unaffected.

According to the GRM, relationships between MOs are modeled independently of MOs in terms of *Managed Relationships*. A MO bound in a relationship is known as a *participant*. Common characteristics of relationships are summarized in *Managed Relationship Classes* (MRCs) for which new templates are provided. MRCs can but do not have to be derived from one or more other MRCs.

MRCs allow to specify certain constraints among participants. For this purpose, *roles* are used to model the properties of various related participants in a relationship. To play a given role, a MO may be required to possess a certain set of characteristics, specified in terms of a MO class (MOC) that any participant in that role will have to be compatible with. A *role cardinality* is used to specify how many MOs may participate in a given role in any one relationship. Also, roles can be specified to be ‘dynamic’ if MOs are allowed to enter and/or leave a relationship

without affecting its existence, as opposed to static roles where MOs remain participants in a relationship for its entire life span. In addition, in a behavior part any other aspects can be defined in natural language text for which no formal specification means are provided.

MRCs are defined independently of the representation of the relationship in a MIB. A so-called *role binding* template is provided which can be used to specify how a certain relationship is represented as part of management information. For this purpose, for each role the class(es) of MOs that can participate in the relationship in that role are specified and whether that includes subclasses. Relationship instances can be represented as part of management information in the following ways:

- **Name bindings:** A relationship is represented by naming, i.e., in a given relationship the participants in one role (subordinates) are contained in a participant (superior) of another role. The role binding identifies one or more name bindings that represent the relationship.
- **Attributes:** A relationship is represented by relationship attributes which participating MOs in a given role have to support. Their values identify related participants in other roles.
- **MOs:** The relationship is represented by dedicated MOs of a certain class. As a result, a relationship is explicitly represented in a MIB in terms of an instance of a relationship MOC called *relationship object*. All relationship MOCs have to be derived from the standardized MOC *relationshipObjectSuperClass*.
- **MO operations:** A relationship is implicitly represented by means of systems management operations. The behavior description in the role binding has to define the meaning of these operations when applied to participants of the relationship.

Role bindings also specify the effects of abstract *relationship operations* and their mapping to systems management operations. Relationship operations include e.g. operations to establish and terminate relationships, to bind and unbind MOs to/from a relationship, and to retrieve information about relationships. One or more mappings are allowed for the same operation. A behavior clause is used to define the semantics of each operation. The abstract relationship operations are not to be confused with relationship services in the sense of a 'relationship management function'; all they do is state in which way certain management operations that operate on MO aspects are to be interpreted from a relationship perspective.

In addition, a role binding allows to specify the effects associated with the dynamic departure of a participant in a relationship: whether it may not depart unless other roles have no participants, whether related MOs in other roles are to be deleted as a consequence, or whether the related MOs are released from the relationship. Access to certain attributes or actions can be prohibited. A behavior part describes any other impacts imposed as a consequence of the role binding. Several role bindings can be defined for a single MRC, reflecting different ways that the same kind of relationship is represented for different MO classes.

3 Conformance testing concepts

The purpose of conformance testing is to increase the probability that different OSI (protocol) implementations are able to interwork. In the *Conformance Testing Methodology and Framework* ([15]), conformance testing is defined to be the assessment process for determining whether the externally visible behavior of an OSI implementation conforms to the behavior required by its specification. A real system is said to exhibit conformance if it complies with the *conformance requirements*, e.g. certain capabilities or allowable behaviors, defined in the corresponding OSI standard in its communication with other real systems.

In order to harmonize the process of testing and certification for OSI implementations, the framework provides a methodology for specifying conformance test suites and defines procedures to be followed by implementation providers and test houses. A standardized test notation, called *Tree and Tabular Combined Notation* (TTCN), is proposed for the development of abstract test suites. TTCN aims at providing a common language in which test cases for various implementations can be expressed on an abstract level. Abstract test cases specify a series of actions (*test events*) that are needed to test a specific conformance requirement. The entirety of all test cases for a certain protocol specification forms the test suite. The use of standardized test suites and common procedures for testing the conformance of OSI implementations leads to comparability and acceptance of test results.

Although devoted to OSI protocols, the test case development and conformance assessment process described in the framework can also be applied to other OSI implementations, especially to MOs. A MO is said to exhibit conformance if it complies with the conformance requirements of its corresponding specification. Testing a MO for conformance requires the externally visible behavior of MOs to be observed by applying operations and analyzing their effects.

In [2], an architecture suitable for MO conformance testing is described. A test system in the role of a manager is responsible for executing test cases based on sending and receiving CMIS [13] requests to an agent in which the MOs to be tested are embedded (see Figure 1). If possible, resource specific test requests may be used to drive the resources in order to observe the reactions of MOs to real effects. A positive test verdict is only assigned if the responses received comply with the expected responses defined in the test cases. The test results are summarized in a test report. Conformance of agents and CMIS is presupposed because these components can be dealt with separately from MO testing [1]. Basing test events on standardized CMIS service primitives allows for the use of TTCN for the definition as well as the standardization of abstract test cases for MOs.

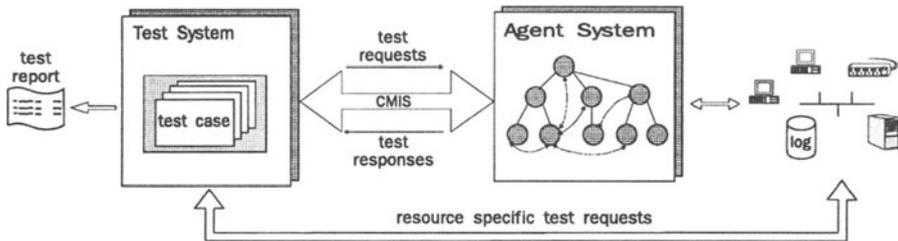


Figure 1: MO test architecture.

In order to structure the test case development process for MOs, a distinction is made between three different views. This concept requires to focus on MOs in isolation, to address the interactions between related MOs, and to take into account the consistency of a MO with its underlying resource. The MO conformance testing concepts can not be introduced in length within this paper. For further details it is referred to [2].

4 Analysis of relationship aspects

4.1 Specification requirements for relationships

In the context of the OSI information model, specification and conformance testing are related in the following sense (see Figure 2):

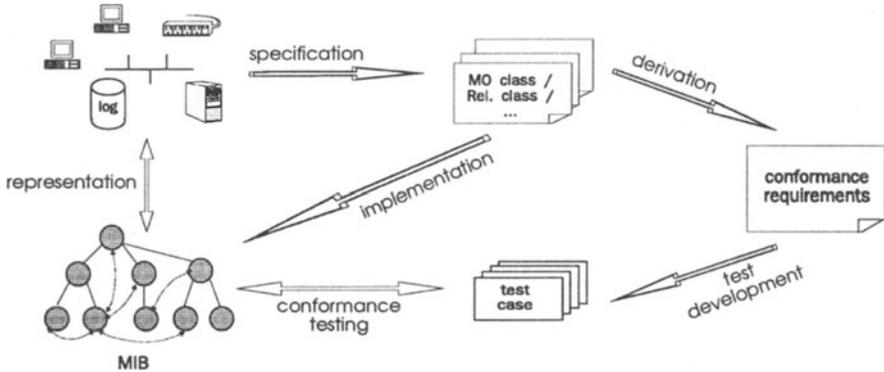


Figure 2: Relation between specification and conformance testing.

- Specification looks at aspects of the managed resources and represents them by means of the information model using dedicated specification tools.
- Conformance testing looks at specified aspects and checks whether the behavior exhibited by the management information conforms to the behavior defined in the specification.

Accordingly, the very same aspects that are relevant for specification are also relevant for conformance testing. A classification of the various aspects being involved in MO relationships has been presented in [5] as a basis for the evaluation and derivation of MO relationship specification means. This same classification can serve as the basis for the derivation of conformance requirements. Aspects of relationships can be grouped along the following perspectives:

- **Structure:** This perspective covers aspects of relationships that are concerned with describing them as a part of management information, i.e., the way they provide associations between the MOs they relate and the rules according to which they add structure to the MIB as a whole. This includes e.g. aspects such as properties of relationship participants (i.e. roles), for instance prerequisites that a MO has to fulfill in order to be allowed to participate in a relationship in that role.

With respect to the GRM, this perspective covers also aspects concerning the instantiation of relationships. This is because the modeler is not only responsible for the specification of abstract relationship properties but also for the representation of those relationships as part of the MIB. Aspects such as role cardinalities stating how many MOs may participate in a role in any one relationship instance or constraints imposed on the leaving and joining of relationship instances by MOs have to be considered as well. (A relationship approach with a different philosophy [4] keeps instantiation aspects transparent to modeler and application and instead hides them in an information layer in order to provide better ‘data independence’ – here such aspects do not apply.)

- **Effects:** This perspective is concerned with effects of relationships on participating MOs, as relationships often imply that operation of one MO affects the other. For instance, if a

MO participates in a relationship, it may no longer be deleted because of that relationship. Another example is that a MO is deleted as a side effect of the deletion of another MO it is related to. It also includes possible dependencies of MO attributes on other participants of a relationship, e.g., of an operational state attribute of a MO that is functionally dependent on another MO.

- **Management:** This perspective covers aspects of relationships that relate to their need to be managed and accessed as part of management information; for instance, whether a relationship is subject to manipulation by management operation.
- **Object Orientation:** Those aspects deal with the embedding of relationships into the (object oriented) OSI information model; for instance aspects related to inheritance.

[5] also mentions a fifth perspective, ‘network management context’ that deals with particular management application requirements for dealing with relationships. This, however, is of no importance with respect to the GRM as it applies less to the OSI information than to the OSI functional model.

4.2 Generic conformance requirements

Test objectives for abstract test cases are aligned with conformance requirements of a certain specification. Conformance requirements have to be determined before starting to develop test cases. As proposed in [15], conformance requirements should be part of the conformance clause of a standard. Looking at OSI information modeling standards, explicit conformance statements are still missing today. Therefore, these have to be added as extensions to the standard documents. In the meantime, efforts have been started to define so called *Managed Object Conformance Statement* (MOCS) proformas as extensions to standardized MOCs and *Managed Relationship Conformance Statements* (MRCS) proformas for MRCs. Such proformas focus on static MO/relationship capabilities, such as the support of packages or relationship operations in an implementation. However, these proformas do not cover the complete set of conformance requirements of a MO or a relationship. For instance, requirements resulting from the behavior part of a specification are outside the scope of these documents.

The specification requirements introduced in the previous chapter are used as a starting point for the derivation of conformance requirements. This is because aspects relevant for specification also lead to aspects that are subject to testing. Correct specification is presupposed in this discussion as ensuring the consistency of a specification is not subject to conformance testing. In the following, we investigate which generic conformance requirements result from the various relationship perspectives with respect to the specification means of the GRM, independent of the particular representation of a relationship in the MIB:

Structure:

- Requirements concerning relationship participants:
 - In order for a MO to participate in a given relationship role, its characteristics must be compatible with the characteristics for that role, i.e., the MO class referenced in the MRC.
- Requirements concerning relationship and relationship instance:
 - ◊ The required role cardinality must not be violated.
 - ◊ If roles are static, participants are not allowed to enter or leave an established relationship instance.
 - ◊ MOs must not be related with each other if there is no role binding that would allow instances of their classes to be related in the respective roles in that particular class of relationship.

As a consequence, any operation that would violate these constraints must be rejected.

There are other common requirements resulting from relationship aspects that are not part of the formal specification but can be expressed in relationship behavior clauses. We want to name a few to give an impression of what further requirements relationships can imply:

- ◊ A MO may only be allowed to enter or leave a certain relationship role if the state of the MO (i.e. certain attribute values) corresponds to the state required in the specification.
- ◊ In order to fulfill a certain role in a given relationship, a MO can be required to fulfill some role in another relationship. A MO can also be prohibited from participating in instances of different MRCs simultaneously.
- ◊ A MO may be allowed to enter or leave a given relationship only if other MOs enter or leave the relationship simultaneously.

Effects: (on participants)

- An attribute of a relationship participant must not be altered if specified in the respective role binding as 'restricted'. Operations attempting to manipulate such attributes must be rejected.
- Actions of relationship participants must not be performed (and accordingly have to be rejected) if specified in the respective role binding as 'restricted'.
- A participant of a relationship must not be deleted if the respective role binding specifies for the respective role 'only-if-none-in-roles' and other MOs are in the specified roles.
- When deleting a relationship participant, related MOs in other roles must be deleted if specified in a 'deletes-all-in-roles' clause in the respective role binding.
- When deleting a relationship participant, related MOs in other roles must no longer participate in the according relationship instance if specified in a 'releases-all-in-roles' clause in the respective role binding.

Again, further requirements can result from relationship aspects expressed in relationship behavior clauses, e.g., any dependencies between attribute values of related MOs.

Management: Relationship management solely occurs as an indirect effect of management of MOs. The role binding defines the mapping of abstract relationship operations to systems management operations. The conformance requirements associated with this perspective refer to the correctness of systems management operations when applied to relationship instances. In particular, this concerns preconditions and postconditions associated with a relationship operation as specified in the behavior clause of the corresponding operations mapping.

Object Orientation: A MRC derived from other MRCs inherits their characteristics. With the kind of strict inheritance defined for the GRM, conformance requirements of relationship superclasses apply to relationship subclasses. Conformance requirements resulting from inherited features are grouped along and added to the perspectives explained previously.

The representation of a relationship determines to which extent relationship information is explicitly available in a MIB and how it can be monitored/controlled by management applications or a test system, respectively. Therefore, the representation independent conformance requirements explained above translate into representation dependent conformance requirements for the respective relationship representations. For instance, a conformance requirement related to a bounded role cardinality by a number n can translate to the conformance requirement that e.g. the set-valued attribute representing that relationship must not contain more than n members.

It should be noted that there is a different kind of relationship information available in a MIB when using different representations for the same relationship. The representation by means of a relationship object is the most powerful alternative. It provides information about the relationship class, its name, and the role binding in use while other representations do not.

Furthermore, the representation by MOs and attributes do have in common that it is possible to directly identify participants in roles. This information is only implicitly available when using name bindings and can hardly be obtained when representing relationships by management operations. Management operations therefore represent the weakest alternative for expressing relationship information in a MIB. An important consequence is that the conformance requirements can differ for the same kind of relationship for different representations of the relationship.

5 Test case development for relationships

5.1 The ATM cross connection relationship

Exemplary, we have extracted relationship information from an object catalogue for the management of an *Asynchronous Transfer Mode* (ATM) cross connection [8]. For the relationship information expressed in the MOCs of the catalogue, explicit MRCs and role bindings have been defined using the specification tools of the GRM. These relationship specifications are used as a starting point for the development of abstract test cases. The first step in this process is to determine the conformance requirements which have to be derived from the MRC and the role binding specifications. This task is guided by the relationship perspectives explained in the previous chapter. The conformance requirements then provide the basis for the second step, the development of abstract test cases for relationships. This proceeding will be explained for a specific example.

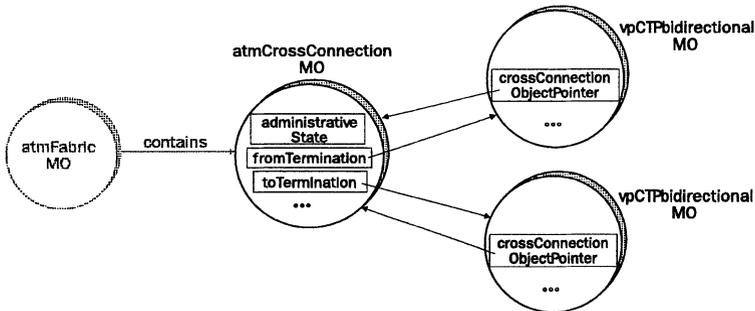


Figure 3: MOs involved in the establishment of an ATM cross connection relationship.

In [8], a **vpCTPbidirectional** MOC is defined to model a virtual path termination point where a virtual path link connection is originated and terminated. An **atmCrossConnection** MOC is specified to represent a relationship between two instances of **vpCTPbidirectional**. On instantiation of an **atmCrossConnection** MO, a virtual path link connection is established between two **vpCTPbidirectional** MOs. The values of two attributes (**toTermination** and **fromTermination**) of the **atmCrossConnection** MO refer to the cross connected MOs. In addition, the cross connected **vpCTPbidirectional** MOs provide an attribute (**crossConnectionObjectPointer**) pointing back to the **atmCrossConnection** MO. The deletion of the **atmCrossConnection** MO terminates the cross connection and the pointers to the **atmCrossConnection** MO have to be deleted in both participants. An instance of the MOC **atmFabric** is responsible for managing the establishment and release of all ATM cross connections for an ATM cross connected network element. For instance, if the establishment of a new ATM cross connection is requested the **atmFabric** MO creates a new **atmCrossConnection**

```

crossConnection RELATIONSHIP CLASS
    BEHAVIOR ...;
    ROLE toTerminationPoint ROLE CARDINALITY (1..1) REGISTERED AS ...;
    ROLE fromTerminationPoint ROLE CARDINALITY (1..1) REGISTERED AS ...;
REGISTERED AS ...;

crossConnectionRepresentation ROLE BINDING
    RELATIONSHIP CLASS crossConnection;
    BEHAVIOR ...;
    RELATIONSHIP OBJECT atmCrossConnection
    ROLE toTerminationPoint
        RELATED CLASSES connectionTerminationPointBidirectional AND SUBCLASSES
            RELATED BY RELATIONSHIP OBJECT USING ATTRIBUTE toTermination;
    ROLE fromTerminationPoint
        RELATED CLASSES connectionTerminationPointBidirectional AND SUBCLASSES
            RELATED BY RELATIONSHIP OBJECT USING ATTRIBUTE fromTermination;
    OPERATIONS MAPPING
        ESTABLISH MAPS TO OPERATION ACTION atmConnect OF atmFabric
            WITH BEHAVIOR ...;
        TERMINATE MAPS TO OPERATION ACTION disconnect OF atmFabric
            WITH BEHAVIOR ...;
    ...
REGISTERED AS ...;

```

Figure 4: Cross connection relationship specification.

MO which is contained in the `atmFabric` MO. Figure 3 shows the MOs that are involved in the establishment of an ATM cross connection relationship. For further details of the MOCs introduced it is referred to [8].

The MOCs explained above lead to the specification of a `crossConnection` relationship class depicted in Figure 4. There, two roles for the `crossConnection` relationship class are defined, `toTerminationPoint` and `fromTerminationPoint`. In both roles only one participant is allowed to take part in a `crossConnection` relationship. Although not using the specification tools of the GRM, the specifier(s) of the object catalogue have decided to represent an ATM cross connection by an explicit relationship object. This results in the representation by relationship object `atmCrossConnection` in the role binding for the `crossConnection` relationship class (see Figure 4). The ‘related classes’ constructs for both roles prescribe that instances of the MOC `connectionTerminationPointBidirectional` or any subclasses may participate in the relationship. As `vpCTPbidirectional` is an indirect subclass of `connectionTerminationPointBidirectional`, instances of `vpCTPbidirectional` are allowed to participate in both roles in the relationship.

5.2 Derivation of conformance requirements

The conformance requirements for the `crossConnection` relationship are derived from the specification depicted in Figure 4 and are grouped along the identified relationship perspectives. To our experience, it is easier to derive conformance requirements from formal relationship specifications than from informal relationship specifications only. As the resulting conformance requirements for the `crossConnection` relationship can not be introduced in length within this

paper, only excerpts are listed below:

Structure:

- The role cardinality (1..1) must not be violated for either the `toTerminationPoint` role nor the `fromTerminationPoint` role. I.e., the value of the `toTermination` attribute and the value of the `fromTermination` attribute in an `atmCrossConnection` MO have to refer to a single participant.
- Participants can not enter or leave an established `crossConnection` (because it is a static relationship).
- In order for a MO to participate in a `crossConnection` relationship in the `toTerminationPoint` or `fromTerminationPoint` role, the MOC of the potential participant must be `connectionTerminationPointBidirectional` or a specialization of this MOC.

Effects: (on participants)

- The value of the `crossConnectionObjectPointer` attribute of a participant in the `crossConnection` relationship has to be the name of the corresponding `atmCrossConnection` MO.
- On deletion of a MO participating in the `crossConnection` relationship, the corresponding `atmCrossConnection` MO has to be deleted (behavior requirement). As a result, the related MO in the other role is released from the relationship.
- If the value of the administrative state of the `atmCrossConnection` MO is 'locked' no traffic can pass through cross connected MOs participating in this relationship (behavior requirement).

Management:

- On establishment of a new `crossConnection` relationship, i.e. requesting the action `atmConnect`, an instance of the MOC `atmCrossConnection` has to be created and a participant in each role has to be bound. The value of the `toTermination` attribute has to be the name of the participant in the `toTerminationPoint` role and the value of the `fromTermination` attribute has to be the name of the participant in the `fromTerminationPoint` role.
- On termination of a `crossConnection` relationship, i.e. requesting the action `disconnect`, the corresponding `atmCrossConnection` MO has to be deleted.

Apart from deriving conformance requirements for each relationship separately (*intra relationship requirements*), there may be effects specified for the participant in one relationship that will also cause effects on participants in other relationships. This kind of conformance requirements, which we call *inter relationship requirements*, can only occur if relationship specifications allow that MOs can participate in different relationships simultaneously. Suppose a *dependency* relationship which requires that a participant in a *parent* role can only be deleted if all MOs in the *dependent* role are deleted as well. One of the dependent MOs however also participates in a `crossConnection` relationship in role `toTerminationPoint` for which the condition 'releases-all-in-roles fromTerminationPoint' has been specified. As a result of requesting the deletion of the MO in the *parent* role, not only the MOs in the *dependent* role have to be deleted but the MO participating in the `crossConnection` relationship in role `fromTerminationPoint` has to leave its relationship. The results presented in [21] remind us that having tested each relationship in isolation does not necessarily imply that this is also sufficient for testing the composition of relationships if these relationships are interdependent. Therefore, inter relationship requirements have to be taken into account in conformance testing as well. When comparing the testing of related MOs with testing clusters of objects in object oriented programs, this conclusion is also acknowledged by work on object oriented program testing approaches. In [20], it is stated that

special attention has to be paid to classes of which instances can be bound to more than one cluster.

5.3 Development of abstract test cases

Testing of related MOs is based on the observation and manipulation of MOs making use of systems management operations only. This requires access to all MOs involved in the relationship to be tested. Each conformance requirement identified has to be addressed in one or more test cases. Abstract test cases for relationships heavily depend on the mapping information contained in role bindings. In particular, this applies to test events for requesting relationship operations and test events for observing the reactions in related MOs that have to be mapped to corresponding systems management operations. Figure 5 shows a simplified example test case defined in TTCN focusing on the requirement that a MO can only participate in a `crossConnection` relationship if the MOC of the potential participant corresponds to `connectionTerminationPointBidirectional` or a specialization of this class.

Test Case Dynamic Behavior				
Test Case Name : crossConnection_establish_with_invalid_participant				
Group :				
Purpose : verify that it is not possible to bind a participant in a cross connection relationship if the class of the participant does not correspond to connectionTerminationPointBidirectional or any subclass				
Default :				
Comments :				
Nr	Label	Behavior Description	Constraints Ref	Verdict
1		+preamble		
2		!MActionRequest START Timer	atmConnectReq	
3	L1	?MEventReportIndication		
4		GOTO L1		
5		?MActionConfirm CANCEL Timer	atmConnectCnf	(PASS)
6		+postamble		
7		?OTHERWISE CANCEL Timer		(FAIL)
8		+postamble		
9		?TIMEOUT		(INCONC)
10		+postamble		

Figure 5: Example TTCN test case for the `crossConnection` relationship.

The TTCN test case consists of a header containing overview information like a test case name, the test purpose etc. and a body for the test case behavior. The body is partitioned into different columns. In a *Behavior Description* column, test events to be sent to the system under test and its possible responses are defined. Send events are indicated by a `!`. A `?` is used to denote receive events. A so-called *preamble* describes a sequence of test events needed to drive the system under test into a state from which the test body will start. The so-called *postamble* sets the system back to a stable end state after the test body has been executed. An entry in the *Constraints Ref* column refers to a specification of the data values (parameters) to be transmitted in a send event or expected as part of a received event. In the *Verdict* column, a verdict for the received test event is given.

In our example test case in Figure 5, a `MActionRequest` is sent to an agent which is responsible for invoking an action on an instance specified in the corresponding constraint `atmConnectReq`

(see behavior line 2). According to this constraint, the action `atmConnect` has to be called on an instance of the MOC `atmFabric` requesting a new cross connection to be established between two MOs. Due to space restrictions, the actual constraints can not be depicted. In this example, we assume that one of the participants specified in the constraint `atmConnectReq` does not match the required class for its role. Different receive events have to be distinguished as a result of the `MActionRequest`. As MOs can issue notifications asynchronously, event reports can be received. As the purpose of the test case does not focus on notifications, these are ignored in a loop until any other event is received (see behavior line 3 and 4). If a `MActionConfirm` event occurs and the data received complies with the data specified in the constraint `atmConnectCnf`, the test case verdict `PASS` is assigned. In this example, the error message 'mismatchingInstance' is expected stating that an incorrect participant given in the request has led to the rejection of the action. In the case that a `MActionConfirm` with invalid data values or any other event is received (see behavior line 7), the test case verdict is `Fail`. In order to take into account that no response is sent from the agent, a timer is started whenever sending a new test event (see behavior line 2). A `TIMEOUT` event is generated by the test system indicating that no events have been received within the timer interval. According to [15], timeout events lead to the test case verdict `INCONCLUSIVE` (see behavior line 9).

When defining relationships between resources, the correctness of the resulting conformance requirements have to be verified during the relationship testing process. However, under certain circumstances there can be conformance requirements which do not necessarily have to be addressed in the testing process. This is the case if a relationship conformance requirement only focuses on physical relationships between resources, or in the terms of [6] on *descriptive* aspects of relationships. Suppose the following example: A dependency relationship between two MOs has been modeled that represents a functional dependency of their underlying resources. A requirement for this relationship could be that if the operational state of one resource changes to 'disabled' this has also to be the case for the dependent resource. Assuming the proper functioning of the resources, the state values of the corresponding MOs will have to change to 'disabled' as well. If the MOs participating in the dependency relationship behave really as images of their underlying resources (this should be the case having tested the MOs in isolation), there is no need to test such kind of conformance requirements.

The overall goal is to develop abstract test cases which 'cover' the intra and inter relationship requirements identified for each relationship in an object catalogue. The abstract test cases developed for the conformance requirements are used for testing the relationships in a whole MIB. Clearly, a test case can only address aspects that have explicitly been defined in a specification. If there exists a relationship between resources that is not specified in the model, the influences of this relationship can not be included in the testing process. The test suite for an object catalogue (including MOC, name binding, MRC, and role binding definitions) comprises the set of all abstract test cases developed for testing MOs in isolation combined with the abstract test cases developed for relationships. The difficulties of dealing with resources in the testing process have already been discussed in [2].

6 Conclusion

In this paper, we have discussed the subject of conformance testing in OSI network management with respect to relationships occurring between MOs in a MIB. Despite its high relevance, relationship conformance testing has been ignored so far, possibly because dedicated concepts for the treatment of relationships have for a long time been missing in OSI management. We have classified generic conformance requirements according to the perspectives put forward in

[5] for the specification of relationships, which refer to the same aspects that have to be checked during a procedure of conformance testing. We have explained how from a formal relationship specification appropriate conformance requirements can be derived. The resulting conformance requirements form the starting point for the development of abstract test cases for relationships. This process has been carried out for an example relationship derived from the object catalogue for an ATM cross connection.

The test case development process for the relationship specifications defined for the ATM cross connection MOCs is supported by a prototype test system for MIBs allowing for the definition of abstract test cases in TTCN and its automatic execution. The test system is based on an existing protocol conformance test tool (*Automated Protocol Test System/2* [11]) for which an extension has been implemented providing for the exchange of CMIS service primitives between test system and a management system [10]. The test system provides the platform for the practical application of our concepts with respect to management information testing. In particular, the test cases developed for the ATM cross connection MOCs will be applied to a prototype MIB which is being implemented as part of an European research project (RACE II PREPARE) dealing with the cooperative end to end service management across heterogeneous Integrated Broadband Communication Networks. Finally, it should be noted that the procedure of testing relationships introduced in this paper is not only of interest for conformance testing but can also aid in an integrated development/testing life cycle of MIB implementations.

Acknowledgements

We wish to thank our colleagues, the research staff directed by Prof. Geihs at the University of Frankfurt, the Munich Network Management Team of the Munich Universities directed by Prof. Hegering, and IBM ENC's system and network management department.

References

- [1] B.Baer, *A Conformance Testing Approach for Managed Objects*, 4th IFIP/IEEE Int. Workshop on Distributed Systems: Operations & Management, Long Branch, New Jersey, USA, October 1993.
- [2] B.Baer, A.Mann, *A Methodology for Conformance Testing of Managed Objects*, 14th Int. IFIP Symposium on Protocol Specification, Testing, and Verification, Vancouver, BC, Canada, June 1994.
- [3] S.Bapat, *Towards Richer Relationship Modeling Semantics*, IEEE Journal on Selected Areas in Communication Vol.11 No.9, December 1993.
- [4] A.Clemm, *Incorporating Relationships into OSI Management Information*, 2nd IEEE Network Management and Control Workshop, Tarrytown, NY, September 1993.
- [5] A.Clemm, *Modellierung und Handhabung von Beziehungen zwischen Managementobjekten im OSI-Netzmanagement*, Dissertation, University of Munich, June 1994.
- [6] A.Clemm, O.Festor, *Behaviour, Documentation, and Knowledge: An Approach for the Treatment of OSI-Behaviour*, 4th IFIP/IEEE Int. Workshop on Distributed Systems: Operations & Management, Long Branch, New Jersey, USA, October 1993.
- [7] CTS3-NM, *Methodology Report on Object Testing*, The Establishment of a European Community Testing Service for Network Management, Deliverable 3, Brussels, Directorate-General XIII-E4, April 1992.
- [8] ETSI, *B-ISDN Management Architecture and Management Information Model for the ATM crossconnect*, ETSI/NA5 WP BMA, April 1994.

- [9] EWOS PT-16, *Framework for conformance and testing of network management profiles*, Report 1 of EWOS/EG NM/PT-16, June 1992.
- [10] W.Herrnkind, *Design und Implementierung einer Erweiterung eines Konformitätstestwerkzeugs für den Einsatz in OSI-Netzmanagementsystemen*, Diploma Thesis (in German), University of Frankfurt, Department of Computer Science, January 1995.
- [11] IBM, *Automated Protocol Test System/2 User's Guide*, SV40-0373-00, June 1993.
- [12] ISO, *Final Answer to Q1/63.1 (Meaning of Conformance to managed objects)*, ISO/IEC JTC 1/SC 21 N 6194, May 1991.
- [13] ISO, *Information Processing Systems - Open Systems Interconnection - Common Management Information Service Definition*, ISO Int. Standard 9595, second edition, 1991.
- [14] ISO, *Information Processing Systems - Open Systems Interconnection - Common Management Information Protocol - Part1: Specification*, ISO Int. Standard 9596-1, second edition, 1991.
- [15] ISO, *Information Processing Systems - Open Systems Interconnection - Conformance Testing Methodology and Framework*, ISO Int. Standard 9646, 1991/92.
- [16] ISO, *Information Technology - Open Systems Interconnection - Management Information Services - Structure of Management Information - Part 1: Management Information Model*, ISO Int. Standard 10165-1, January 1992.
- [17] ISO, *Information Technology - Open Systems Interconnection - Management Information Services - Structure of Management Information - Part 4: Guidelines for the Definition of Managed Objects*, ISO Int. Standard 10165-4, January 1992.
- [18] ISO, *Information Technology - Open Systems Interconnection - Management Information Services - Structure of Management Information - Part 7: General Relationship Model*, ISO Draft Int. Standard 10165-7, March 1994.
- [19] H.Kilov, J.Ross, *Generic Concepts for Specifying Relationships*, IEEE/IFIP 1994 Network Operations and Management Symposium, Orlando, Florida, February 1994.
- [20] J.D.McGregor, T.D.Korson, *Integrated Object-Oriented Testing and Development Processes*, Communications of the ACM, Vol. 37 No. 9, September 1994.
- [21] E.J.Weyuker, *The Evaluation of Program-Based Software Test Data Adequacy Criteria*, Communications of the ACM, June 1988.