

A Methodology for Conformance Testing of Managed Objects

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

Andreas Mann
 IBM European Networking Center
 Vangerowstr. 18
 D-69115 Heidelberg
 Germany
 amann@vnet.ibm.com

Abstract

The trend towards distributed applications and heterogeneous networks increases the complexity of systems and network management. The standardization bodies have defined a framework for the specification and open exchange of management information. Conformance, one of the key requirements for the interworking of open management systems, was hardly discussed. Few groups have realized the importance of defining conformance not only for management protocols but also for management information. This paper presents a methodology for management information testing and explains differences with respect to communication protocol testing. It is shown how conformance requirements can be derived and how tests can be performed. An outlook points out the adaptation of an existing TTCN based test tool for the definition and execution of test cases for management information.

Keywords: OSI Management, Conformance Testing, Managed Objects

1 Introduction

As a result of an increasing number of interconnected systems, efficient network and systems management products are necessary to guarantee the availability and performance of services. The task of managing large, heterogeneous networks has gained more importance during the last years. International standardization bodies have recognized the need for standardized network management concepts. For instance, ISO has published a set of standards for the management of OSI networks. According to its object oriented information model, manageable resources are modelled as *Managed Objects* (MOs). MOs are instances of *Managed Object Classes* (MOCs) defining formally the visible properties of particular types of resources using a standardized notation (*Guidelines for the Definition of Managed Objects*, GDMO [16]). ISO and other organizations have defined a set of MOCs suitable for managing OSI, or non-OSI, heterogeneous networks. One problem that has not been kept pace with is the question of MOC implementations conforming to their GDMO specifications.

So far, ISO has standardized conformance testing methods in the area of OSI communication protocols [12]. Besides basic definitions and a number of possible test methods, the framework lists the requirements to be followed by implementation providers and test houses. A standardized test notation, called the *Tree and Tabular Combined Notation* (TTCN), is defined for the development of abstract test cases.

*The author's work is supported by IBM European Networking Center.

However, conformance testing is not only of interest in the area of OSI communication protocols but also for network management. Management applications are mainly based on the observation and manipulation of MOs. Therefore, conformance of MOs is an important issue for the correctness of network management applications. This means, it is desirable to determine whether a given MOC implementation behaves according to its GDMO specification. In contrast to protocol conformance testing, testing of MOs is more like testing a service, i.e., information services about resources through a standardized interface. The specific characteristics of MOs in contrast to protocols have to be considered and to be adapted when developing a conformance testing methodology for MOs.

Several organizations [2,3,8] are already working on the subject of conformance testing for MOs. However, their methodologies lack a complete analysis of the meaning of MO conformance as well as MO conformance requirements [1]. The problem of MOs linked to other MOs or its resource is not covered. In this paper, we introduce a conformance testing methodology that takes into account all requirements that apply for MOs. The meaning of MO conformance is investigated. Following the introduction of an architecture suitable for MO testing, we identify conformance requirements for MOs and explain how these can be tested. Our methodology follows the guidelines of [12]. We propose TTCN to be used for the development of abstract test cases. This provides the basis to harmonize the process of testing and certification for MOs. A prototype implementation is described that allows for the definition of TTCN test cases for MOs and their automatic execution.

2 OSI network management

Two systems involved in management take one of two roles (*manager* or *agent* role) at a given point of time. Briefly, one speaks of manager and agent (see figure 1). The manager uses information from agents to perform management tasks. An agent provides information to various managers through a standardized interface. Agents handle the communication and give access to *Managed Objects* (MOs) [16] which represent physical and logical resources like modems, transport connections, software modules etc. For communicating with each other, manager and agent use the *Common Management Information Service* (CMIS [10]) and the *Common Management Information Protocol* (CMIP [11]) based on ROSE using ACSE. CMIS offers different primitives which can be mapped to corresponding MO management operations. [16] defines the set of management operations that can be performed at the *MO boundary*, the management interface of a MO.

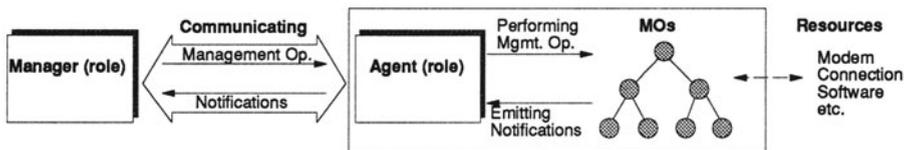


Figure 1: Systems management interactions

The characteristics to be defined for MOs consist of four parts: **Attributes** represent the state of the MO and its underlying resource containing values like names, state variables, counters, routing tables. **Actions** are operations that can be performed on MOs, e.g., rebooting a workstation or starting some tests. **Notifications** are messages sent asynchronously by MOs to one or more managers, e.g., to inform them about exceeded threshold values or serious problems of the underlying resource. **Behavior** typically specifies (by natural language text) the link

between the resource and the MO (how the resource's status is reflected in attribute values), the effects of links to other MOs, the conditions when notifications are emitted by a MO, etc. The *Guidelines for the Definition of Managed Objects* (GDMO [16]) provide a set of templates linked with ASN.1 modules [9] for specifying the characteristics of MOCs. The templates restrict value ranges of attributes or define initial values for attributes. Further, they specify the operations allowed at the MO boundary, e.g, whether attributes can be read or changed. One MOC may *inherit* the properties of another MOC.

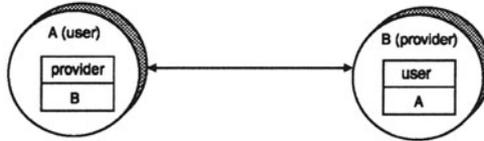


Figure 2: Service relationship

MOs are not isolated objects but maintain **relationships** to other MOs. [13] defines the way relationships between MOs are specified and handled. So called *role attributes* represent relationships between MOs, different attributes representing different relationships. Within a given relationship, each MO has its role. E.g., the attributes 'provider' and 'user' indicate different roles in the service provider versus service user relationship (see figure 2). These attributes are pointers to another MO or to several MOs. Typically, a pointer is realized as the name of a MO.

3 The methodology

In [1] a concept has been proposed grouping all aspects of interest for MO testing into different views. Several views have been identified taking into account all interactions of a MO which influence the behavior of a MO in its environment. Behavior does not only mean the behavior parts of GDMO specifications but all other effects visible at a MO boundary as a result of an operation or an internal resource event.

The following views are of importance in the context of this paper:

management view: correctness of MO operations as well as the availability of the specified GDMO properties, e.g., changeability of attributes through management operations

relationship view: consistency of relationships between MOs, e.g., state changes in one MO may require state changes in related MOs

resource view: the information contained in a MO has to be consistent with the resource, e.g., state correspondence between MO and resource

Testing a MO for conformance to its GDMO specification requires the externally visible behavior of a MO to be observed at its boundary by applying operations and analyzing the results. Grouping MO aspects into views suggests to partition the task of testing according to the views.

3.1 Test architecture

It is necessary to identify *points of control and observation* (PCOs) for controlling the interactions of a MO with management applications, its resource, and other MOs in its environment. Unfortunately, MO interface descriptions are not standardized but are specific for every agent system. The only MO interactions which are standardized in an informal way are the management operations permitted at a MO boundary. As a consequence, when developing standardized

(abstract) conformance tests, a *MO under test* (MOUT) can only be observed and manipulated based on standardized CMIS service primitives. There is no direct access to the MOUT. A test system in the role of a manager will be used for sending and receiving CMIS service primitives to/from the agent system in which a MOUT is embedded. This approach allows a MOUT to be tested in its real environment, i.e., interworking with related MOs and its underlying resource.

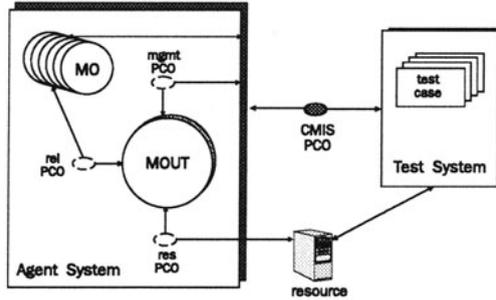


Figure 3: Test architecture

As can be seen in figure 3, a CMIS PCO is used for the exchange of CMIS service primitives between a test system and the agent system in which a MOUT is embedded. To emphasize the concept of structuring MO testing into views, 'virtual (not accessible) PCOs' for the different views are included in the agent system of the test architecture. Because there is no possibility to observe reactions of a MOUT to internal resource events directly, a resource dependent part has to be added to a test system allowing to read and initiate state changes in the resource. This additional component requires the test developer to exactly specify the interface to the resource as being used by the test system. It would be the MOUT provider's responsibility to provide this component. This is very similar to the coordinated method as defined in [12]. However, in contrast to protocol conformance testing, there is no standardized interface to the resource. Instead, the interface has to be defined in tests explicitly.

The architecture explained above can be used for every MOUT or agent system, respectively. It allows for the reuse or even standardization of abstract test cases for MOs. Nevertheless, it remains to be proven that the test architecture is suitable for covering all MO view requirements. Other methodologies [2,3,8] developed for MO conformance testing have also based their test method on the exchange of CMIS service primitives. But their methodologies are only aligned with the testing of management view aspects. They have not considered the testing of relationships or proven the adequacy of their approach for relationship testing yet.

3.2 Developing test cases for MOs

Test objectives for abstract test cases are aligned with conformance requirements of a certain specification. As a consequence, conformance requirements have to be determined before starting to develop test cases. Separating aspects of interest for conformance testing into views implies that conformance requirements and the development of abstract test cases can be partitioned according to the views identified [1]. As proposed in [12], conformance requirements should be part of the conformance clause of a standard. Looking at OSI information modelling standards, explicit conformance statements are still missing. 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, e.g. [15]. A MOCS proforma is similar to the concept of a *Protocol Implementation Conformance Statement* (PICS) proforma and focuses on static MO capabilities. However, MOCS proformas do not cover the complete set of conformance requirements of a MO, e.g., requirements resulting from the behavior part of a GDMO specification are outside the scope of MOCS proformas.

For each view, we analyze the conformance requirements and show how these can be tested by sending and receiving CMIS service primitives. Using a standardized protocol interface based on ASN.1 as a test interface implies that TTCN is suitable for defining test cases. Because the focus of our methodology is only on the testing of MOs, several assumptions are made. E.g., consistency of GDMO specifications, the correctness of the CMIS services as well as the conformance of the agent is required. In addition, the underlying resource is assumed to behave correctly.

3.2.1 The management view

Conformance requirements for a MO result from the GDMO templates of the corresponding MOC. Requirements for the management view refer to specified MO operations and other GDMO properties including inherited features. The OSI information modelling standards define the syntax and the semantics of the GDMO templates. It is possible to determine generic conformance requirements for every GDMO template on an abstract level. In [6] a catalogue containing generic conformance requirements for GDMO templates has been established. This catalogue can serve as a basis for determining the requirements that apply for a certain MO. First, requirements are derived from the catalogue for the GDMO templates used in the specification. In a second step, specific requirements have to be added, e.g., initial values or explicit value ranges for attributes, etc. This method has been exercised for the standardized *log* MOC ([6,14]).

The standardized management operations support observation and manipulation of MOs. Test cases are needed covering the functionality of the MO operations and other GDMO properties. For every MO operation to be tested, a corresponding CMIS request can be invoked. The correctness of an applied MO operation can be determined by reading the MO status and comparing it with the status expected as defined in the GDMO specification. The MO operations are also used for testing other properties. E.g., for testing notifications, a MO operation is used to set the status of a MO requiring the emission of a certain notification. Because notifications are forwarded to the test system, a test case can provide for the receipt of the expected notification. Comparing MO testing to object-oriented program testing approaches, e.g. [4,17,18], the testing of management view aspects complies with unit testing of objects. Because objects, and therefore also MOs, can be interpreted as state machines, state based techniques can be applied for deriving test cases for MOs. However, these have to be combined with methods for inheritance testing [4,17].

3.2.2 Testing relationships

Relationships are represented either explicitly by using role attributes in a MOC or implicitly by referring to other MOs in the (natural language) behavior part of a GDMO specification. In contrast to the management view, no dedicated operations are provided in the OSI information modelling standards for the management of relationships. In the behavior templates only informal statements are possible. Nevertheless, relationships have a severe influence on MOs and have to be considered while testing the conformance of MOs.

There are mainly two categories of relationship requirements. On one hand, there are require-

ments that apply to the participants in a relationship. On the other hand, requirements concern the relationship itself. For instance, constraints for and effects of the establishment and termination of relationships can be defined in a GDMO specification. The requirements have to be guaranteed for preserving the consistency of relationships between MOs. Because a detailed list of conformance requirements for the relationship view is beyond the scope of this paper, some representative examples are given:

Requirements for participants:

1. constraints for MO operations, e.g., the value of a writable attribute in a MO can not be changed by management operation if the MO is bound in a certain relationship
2. a request to change the state of one MO may also require state changes in related MOs

Requirements for the establishment or termination of relationships:

3. a MO can only enter/release a certain relationship if the MO is in a certain state
4. the specified number of MOs (upper or lower limit) which can participate in a relationship in a certain role has to be preserved
5. the successful establishment or termination of a relationship may require attribute value changes or the emission of notifications in related MOs

Conceptually, MOs are responsible for the consistency of relationship information and its effects on other MOs. Because there is no standardized interface for relationship interactions, these can not be observed directly. Other mechanisms have to be used to infer the correct behavior of related MOs. In contrast to protocols, conformance of a MO can not be tested in isolation but only in the context of related MOs. We show how test cases for the conformance requirements mentioned above can be based on the CMIS interface for sending and receiving test events (transaction problem not taken into account):

1. A relationship can be established by issuing a CMIS request to set the role attributes of the corresponding MOs. In a second step, a request is sent to change the value of the attribute. For correct behavior, this request has to be rejected if the attribute operation is restricted.
2. First, a CMIS test event has to be sent to a MO requiring a related MO to change its state. To verify this requirement, a CMIS request has to be applied to read the state of the related MO. The response provides the information whether the state conforms to the expected state defined in the GDMO specification.
3. Before establishing/terminating a relationship, a snapshot from the current state of the MO can be taken. The response to the establishment/termination of a relationship has to be evaluated dependent on the state of the MO.
4. CMIS requests for adding or removing participants of a certain relationship can be sent in order to violate lower or upper limits of allowed relationship participants. Depending on rejection or acceptance of such requests it can be decided whether the constraints are satisfied.
5. Based on the GDMO specification, the test system knows to which state a MO should change or what kind of notification has to be emitted as a result of the establishment or termination of a relationship. By applying CMIS requests to the corresponding attributes, the test system is able to read the values and to compare them with the required ones. Because all notifications are forwarded to the test system, a test case can contain test events for the receipt of the required notifications.

Relationship testing can be compared to the task of integration testing in object-oriented program testing. Integration testing is realized as the combination of units (objects) that have been tested in isolation. It is stated that objects are not adequately tested until they are also tested via integration testing [17]. However, integration testing strategies are an open issue in

object-oriented program testing. One approach is to establish a call graph for the messages exchanged between objects. Testing then starts with objects that do not call other objects. It is continued using only previously validated objects. The application of these ideas for deriving a relationship test strategy is for further study.

3.2.3 Resource aspects

In order to ensure consistency of a MO and its resource, some sort of (non-standardized) information exchange between MO and resource is needed. On one hand, attribute values in a MO must be updated as a result of state changes in the resource. On the other hand, management operations influencing the state of a MO may require state changes in the resource. The OSI information modelling standards do not provide any concepts or services for the exchange of information between MO and resource. Only informal instructions concerning the resource can be included in behavior template of a GDMO specification. To verify these instructions, resource view requirements have to be considered while testing the conformance of a MO. Even if requirements for real effects can be determined, this does not necessarily imply testability of these requirements. The importance as well as the difficulty of dealing with real effects has also been seen by [5].

Some example requirements are:

1. the correctness of attribute values that refer to a resource state, e.g., attribute values that have to comply with resource register values
2. constraints can be specified in order to determine the circumstances under which the values of attributes have to change as a reaction to an internal resource event, e.g., a counter attribute is only incremented if a certain resource event has occurred and other attributes have certain values
3. changes in the resource result in effects on its MO, e.g., attribute value changes, emission of notifications, relationship changes, etc.
4. management operations on a MO may require a state change in the resource, e.g., replacing an attribute value in a MO which corresponds to a resource state may cause the resource to reset

Test cases verifying the consistency of MO and its resource (assuming the correct behavior of the resource) can be based on the manipulation of one of them and evaluating the visible results. For the MO, the CMIS interface is sufficient for both: issuing test events which require the MO and the resource to change its state as well as receiving test events that return information about state changes in the MO initiated by the resource. This is because resource changes can only result in effects concerning the management or the relationship view of a MO. The suitability of the CMIS interface for observing visible effects at the management or the relationship view has been explained in previous chapters. Test events addressing the resource remain resource specific.

4 Conclusion

A conformance testing methodology for MOs has been presented, taking into account different aspects of GDMO specifications (views). An architecture has been described for the exchange of test information between a test system and a MO based on the standardized CMIS service primitives. The architecture is suitable for every MO or agent system, respectively. Conformance requirements have been identified for the different MO views. We have explained how the requirements mentioned can be tested using the CMIS interface for accessing MOs. TTCN is

sufficient for the definition of test cases for MOs. This allows for the standardization of test cases for MOs and provides the basis to harmonize the process of testing and certification. In contrast to protocols, a MO can only be tested in the context of an agent system, related MOs, and its underlying resource.

Methodologies developed by other organizations do not address the problem of relationship testing. They are mainly aligned with the testing of management view aspects [1]. Conformance requirements as well as test coverage strategies are hardly discussed. We have recognized the importance of MO relationships and resource aspects. This led to the inclusion of the relationship and the resource view when defining the meaning of MO conformance. In the future, we will address the development of test strategies covering the requirements identified for the different views. Combining state-based techniques with methods for inheritance and integration/cluster testing seems to be a promising starting-point.

Currently, we are developing a prototype test system for MOs. It is based on an existing protocol conformance testing product (*Automated Protocol Test System/2*, APTS/2 [7]). APTS/2 includes a TTCN editor for the specification of test cases as well as an execution system supporting automatic test case selection and its execution. The definition of MOCS and MOXIT (*Managed Object eXtra Information for Testing*, similar to the concept of *protocol implementation extra information for testing*) and its use during execution is also supported. Test reports are generated. A protocol driver is being implemented as an extension to APTS/2 for exchanging CMIS service primitives between test system and MOs or agent systems, respectively. In contrast to some other, commercially available test systems like OSI-TEST, SEMA's Object Tester, XRTLE, TOG, an advantage of our prototype is the availability of a TTCN editor. It aids in producing abstract test cases by checking the syntax and static semantics of the entered TTCN test cases. The TTCN test cases developed for the standardized log MOC [6,14] will be used as an application for the automatic execution of MO tests.

References

- [1] B.Bär, *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] 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.
- [3] EWOS PT-16, *Framework for conformance and testing of network management profiles*, Report 1 of EWOS/EG NM/PT-16, June 1992.
- [4] M.J.Harrold, J.D.McGregor, K.J.Fitzpatrick, *Incremental Testing of Object-Oriented Class Structures*, 14th Int. Conference on Software Engineering, ACM, May 1992.
- [5] M.E. van der Haven, M.G.L.Kockelmans, E.J.Slotboom, *Real Effects Testing of OSI Applications*, 4th Int. Workshop on Protocol Test Systems, Leidschendam, the Netherlands, October 1991.
- [6] C.Henkels, *Definition und Durchführung von Konformitätstests für Managed Objects*, Diploma Theses (in German), University of Frankfurt, computer science department, January 1994.
- [7] IBM, *Automated Protocol Test System/2 User's Guide*, SV40-0373-00, June 1993.
- [8] ISO, *Final Answer to Q1/63.1 (Meaning of Conformance to managed objects)*, ISO/IEC JTC 1/SC 21 N 6194, May 1991.
- [9] ISO, *Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)*, ISO Int. Standard 8824, 1990.

- [10] ISO, *Information Processing Systems - Open Systems Interconnection - Common Management Information Service Definition*, ISO Int. Standard 9595, second edition, 1991.
- [11] ISO, *Information Processing Systems - Open Systems Interconnection - Common Management Information Protocol*, ISO Int. Standard 9596, second edition, 1991.
- [12] ISO, *Information Processing Systems - Open Systems Interconnection - Conformance Testing Methodology and Framework*, ISO Int. Standard 9646, 1991/92.
- [13] ISO, *Information Technology - Open Systems Interconnection - Systems Management - Attributes for Representing Relationships*, ISO Int. Standard 10164-3, January 1992.
- [14] ISO, *Information Technology - Open Systems Interconnection - Systems Management - Log Control Function*, Int. Standard 10164-6, January 1992.
- [15] ISO, *Information Technology - Open Systems Interconnection - Systems Management - Log Control Function - Amendment 1: MOCS/PICS Proforma*, Proposed Draft Amendment ISO/IEC 10164-6/PDAM 1, N 7574, May 1993.
- [16] ISO, *Information Technology - Open Systems Interconnection - Management Information Services - Structure of Management Information*, ISO Int. Standard 10165, January 1992.
- [17] C.D.Turner, D.J.Robson, *State-Based Testing and Inheritance*, Technical Report TR-1/93, Computer Science Division, University of Durham, Durham, England, April 1993.
- [18] C.D.Turner, D.J.Robson, *The Testing of Object-Oriented Programs*, Technical Report TR-13/93, Computer Science Division, University of Durham, England, February 1993.