The OSI Managed-object Model

Colin Ashford

Bell-Northern Research
P.O. Box 3511, Station C,
Ottawa, Canada,
K1Y 4H7

Abstract. The challenge facing the International Organization for Standardization (ISO) in the early eighties, in developing Open Systems Interconnection (OSI) protocol standards for network management, was to ensure that such protocols should, on the one hand, be standardised but, on the other, be capable of managing a myriad of resource types. ISO met the challenge by developing a single internationally-standardised carriage protocol (CMIP), and tools to produce information models that would reflect the resources being managed. Such an approach makes it possible for the same carriage protocol to carry management messages for many different types of resources. In developing its information modelling tools and services, ISO has adopted an object-oriented approach: the resources to be managed are modelled as managed objects or aggregates of managed objects. The managed-object model is similar to popular object-oriented programming-language models but it includes a number of features that reflect the special requirements of network management. These requirements include: asynchronous operation, active resources, a distributed environment, compatibility, and feature optionality. Fulfilling these requirements lead to the inclusion of concepts such as event-notification, multiple object-selection, packages, and allomorphism. The next generation of network-management standards will need to address the demands of large, multi-protocol, mutable networks. How these requirements might affect the evolution of the managed-object model and services is considered.

1 Introduction

Over the last half-dozen years, the International Organization for Standardization (ISO) has progressed a number of Open Systems Interconnection (OSI) network-management standards to international standards status. These standards define an architecture, protocols, information modelling and notation tools, and functional models "...which enable managers to plan, organise, supervise, control, and account for the use of interconnection services..." [ISO89a]. In developing these standards, ISO took an object-oriented approach to information modelling and, as a consequence, developed an object model for network management—the OSI managed-object model. Recently there has been interest in the object model both from the standpoint of systems implementation [NMF93] and application integration [HoTh93].

The purpose of this paper is: to introduce the important aspects of the OSI managed-object model and associated services; to present some of the design requirements and indicate how they were met; and to consider how the model and services might evolve to meet the needs of the next generation of network-management systems.

The layout of the paper is as follows: §2 gives some background concerning OSI network management and the OSI Reference Model; §3 presents the OSI management architecture and services; §4 considers the evolution of the model and services; and §5 offers some conclusions.

2 Background

In this section we try to answer the questions: "What is OSI network management?" and "How should network-management protocols be designed in the context of OSI?"

2.1 What is OSI Network Management?

Figure 1(a) gives a pictorial representation of a typical network management environment: *managing* systems (stylised as work stations) communicate with *managed* systems (stylised as equipment cabinets) across a *management network.* Figure 1(b) shows a single instance of communication between a managing and a managed system.

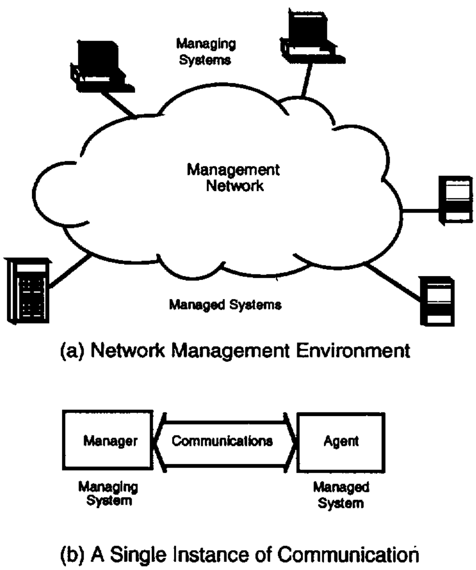

Figure 1 OSI Network Management

The purpose of the OSI network-management standards is to standardise the management information that flows across the interface between the managing and managed systems. To develop the actual messages that will flow across the interface, the model shown in Figure 1(b) needs to be further refined (by, for example, the addition of managed objects). It should be noted that this refinement simply helps in the specification of the messages; it in no way constrains implementors in their choice of development techniques. In particular, implementors are not required to use an object-oriented approach to systems development (although they might well chose to do so). Implementors are free to implement real systems as they choose—their implementations only have to conform to the specification of the "bits on the wire".

ISO/IEC 7498-4 [ISO89a] forms the basis for the standardisation of OSI network-management. It categorises the requirements for management in five functional areas: fault management, configuration management, performance management, security management, and accounting management. See Figure 2 for more details.

The question is: "How can these broad requirements be translated into messages between systems?" To answer the question, we will have to take a short detour to review the basic elements of the OSI Reference Model.

Fault Management	The detection, isolation, and correction of abnormal network operation.
Configuration Management	The configuration of the network or components to meet changing user needs or alleviate congestion.
Performance Management	The monitoring and tuning of the performance of the network.
Security Management	The management of facilities required to support network security.
Accounting Management	The determination and allocation of costs for the use of network services.

Figure 2 OSI Management Functional Areas

2.2 A Crash Course on the OSI Reference Model

The OSI Reference Model [ISO84], see Figure 3, is a layered communications model in which the (n)-layer of the model provides services to the (n+1)-layer by means of (n)-layer protocols. For example the Network Layer of a particular implementation might provide routing services to the Transport Layer by means of X.25 packets. Figure 4 illustrates how this is done: each (n)-layer Protocol Data Unit (PDU) contains a slot, normally labelled "User Data", in which the (n+1)-layer PDU is carried; the (n+1)-layer PDU has a slot in which the (n+2)-layer PDU is carried, and so on. (N)-layer protocol design, then, is a question of providing (n)-layer services by means of (n)-layer PDUs with "holes" in them for (n+1)-layer PDUs. Except, of course, at the Application Layer where there is no (n+1)-layer.

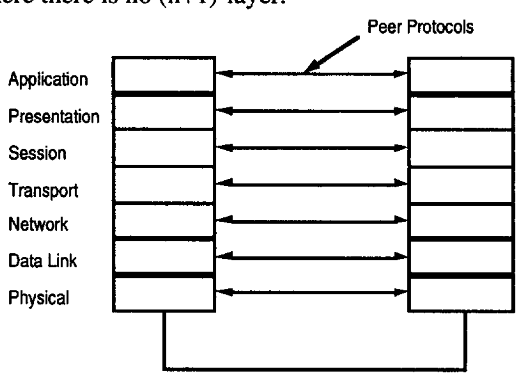

Figure 3 OSI Reference Model

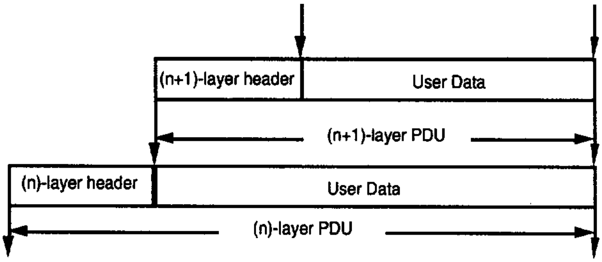

Figure 4 Protocol Data Units

The Application Layer is modelled as a "tool-box" of service elements such as Remote Operations Service Element [ISO89b; ISO89c] or Distributed Transaction Processing Service Element [ISO92d; ISO92e; ISO92f]. Service elements are combined into various *application contexts* to carry the application's messages.

2.3 Application-layer Message Development

It was recognised at a fairly early stage that, for OSI network-management, the application functionality was essentially open-ended—it would not be possible for one group to develop a set of messages that would accommodate all resource types. An open-ended message design was required. The design would have to cleanly separate the generic aspects (such as carriage protocol and access control) from the resource-specific aspects (such as message parameters and code points). Two design approaches were considered—a *functional approach* and a *model approach.*

The **functional approach** consists of designing generic messages to reflect the five functional areas and of refining these messages on a resource-by-resource basis. Such refinements would take the form of optional parameters or extensions to parameter lists.

The **model approach** consists of designing a set of messages to reflect a model of the resources to be managed and of refining the model on a resource-by-resource basis.

The latter approach offered the best method of separating the generic from the resource-specific and the best method of allowing multiple standards-groups to develop management message-sets independently—essentially by modelling the resources to be managed in terms of managed objects. The only question left was: "Which modelling paradigm?" A number of paradigms were considered including data-store, relational, and object-oriented. The object-oriented notions of encapsulation, object state, and inheritance were attractive and resonated well with evolving ideas of network-management modelling. Thus an object-oriented paradigm was adopted although the full ramifications of the choice were not apparent at the time!

3 OSI Management Architecture

The OSI management architecture consists of the managed-object model and a number of supporting object services and features.

3.1 OSI Managed-object Model

The managed-object model is similar to those found in many popular programming languages but it has a number of features that reflect the special requirements of network management.

3.1.1 Managed objects

A managed object is defined as "...an abstraction of real resource..." [ISO89a] (e.g. a modem or a user service) and, as such, it should support the requirements of:

- attributes (e.g. usage statistics, configuration parameters);
- operations (e.g. reset and start test);
- asynchronous events (e.g. over temperature and loss of power);
- behaviour (resources are intelligent and have state); and
- implementation hiding (as far as possible, irrelevant differences in implementation should be hidden).

To meet these requirements, ISO/IEC 10165-1 [ISO92a] defines the managed-object model and associated services. Managed objects are characterised by:

- the *operation invocations* they accept;
- the *event notifications* they emit;
- the *attributes* (state) they make available; and
- the *behaviour* they exhibit.

Managed objects support three generic operation-invocations:

set request a change in the value of one or more attribute values; and

get request the retrieval of one or more attribute values; and

delete request the deletion of an object.

In addition managed objects support a parameterised operation-invocation *action* and a parameterised event-notification *notify*. These operation-invocations and event-notifications allow the object model to support an open-ended set of operations and events.

As shown in Figure 5, object invocations and notifications are modelled as acting directly on the surface of the managed objects. As we shall show later in this section, these object invocations and notifications will be carried between managing and managed systems by means of a carriage protocol.

3.1.2 Managed-object Structuring

The need to structure resources within a managed systems was considered important both for the naming and the selection of resources. The structure of the *Management Information Base (MIB)* reflects this need. The MIB can be regarded as a number of 1:n containment relationships that form a tree of managed objects. The containment relationships define the hierarchical naming-structure of managed objects and may also specify required behaviour of members of the relationship. The MIB also provides the object structuring on which multiple-object selection is based.

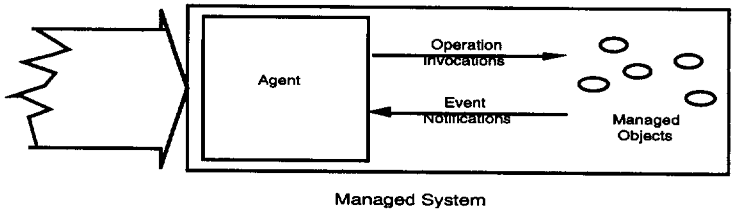

Figure 5 Invocations and Notifications

3.1.3 Managed Relationships

Relationships between resources, such as that of back-up, switching, connecting, and existence, can be as important as the resources themselves from the point of view of network management. It was regarded as important to be able to discover and to manage such relationships, but, by using the existing carriage protocol.

A managed relationship is defined as "...a semantic binding between one or more managed objects which affects the behaviour of the bound objects" [ISO93a]. A managed relationship is characterised by:

- the semantics of the relationship (how the bound objects affect one another);
- the roles of the managed objects bound in the relationship; and
- the reification of the relationship.

In order to manage relationships using the existing carriage protocol, relationships may be reified in terms of managed objects, attribute pointers, or MIB containment.

3.2 Managed-object Services

The managed-object model is supported by an infrastructure of services and tools.

3.2.1 Managed-object Life Cycle

In order to reflect real resources, the managed-object model needs to support the concept of multiple instances of a class. Thus all managed objects are instances of a class and all objects have unambiguous names. Managed objects can be created remotely by conceptually applying a set of instantiation rules to a managed-object specification. The rules control the initial values of attributes and the naming of the created object instance.

An important concept in OSI protocol specifications is that of optionality. The features of protocol specifications are often subsetted into functional units: to be conformant, implementations must support at least a kernel functional unit and, in particular circumstances, may be required to support additional functional units. The OSI managed-object model supports a similar concept—*packages*. Managed-objects are defined in terms of packages of characteristics (operations, events, attributes, and behaviour) that are included on either a mandatory or conditional basis. Real resources are often designed with a large number of optional features and the mechanism of packages helps limit the combinatorial explosion that would occur if sub-classing alone were used to model the feature combinations.

3.2.2 Managed-object Specification

Protocol standards are the first (and most important) step towards interoperability and it is crucial that the specifications be both clear and unambiguous. The formal (in the sense of unambiguous) specification of the syntax of the characteristics of managed objects is accomplished by the use of a template language [ISO92b] and a data-typing language, ASN.1 [ISO90].

ISO has not agreed on a mechanism for the formal specification of the behaviour of managed objects; at the moment natural language is used. A number of attempts at using various formal description techniques for the specification of the behaviour of managed objects have been reported [SiMa91; ISO92g]; ISO is currently reviewing this issue.

In order to permit re-use, class specifications are arranged in a specialisation/generalisation hierarchy linked by multiple inheritance.

3.2.3 Carriage Protocol

The carriage protocol, CMIP [ISO91a] supports the messaging needs of the managed and managing systems and the need for access control, multiple and conditional object-selection, and asynchronous operation.

CMIP is an object-based, asynchronous, request/reply carriage protocol between managing and managed systems. It provides two types of service:

1. the carriage of event notifications asynchronously emitted by managed objects; and
2. the carriage of operation invocations directed towards managed objects.

The event-notification PDU provides fields for identifying:

- the managed object associated with the event;
- the event type;
- the time of the event; and
- an extensible parameter field to carry event-specific or object-specific information.

The operation-invocation PDU provides fields for:

- identifying the target object or objects;
- identifying the operation;
- enabling access to objects;
- correlating the reply with the invocation; and
- an extensible parameter field to carry object-specific or operation-specific information.

The identification of the target managed-object can be based on either:

- its name;
- its position in the Management Information Base;
- a predicate referring to its attributes; or
- a combination of all three.

These features allow a managing system to affect a number of objects with a single operation-invocation (termed *scoping*) and to select objects for operation invocations depending on their dynamic state (termed *filtering*).

3.3 Summary of OSI Network Management Architecture

The complete OSI network management architecture is shown in Figure 6. The *manager*, in the managing system, communicates with an *agent*, in the managed system by means of CMIP. The agent notionally:

- imposes access constraints;
- performs object selection for in-coming operation invocations; and
- provides dissemination for out-going event notifications.

Managed objects are arranged in a singly-rooted tree that serves as a basis for naming and multiple-object selection.

We emphasise again that the internal structuring of the managed system is purely illustrative and does not constrain actual implementations.

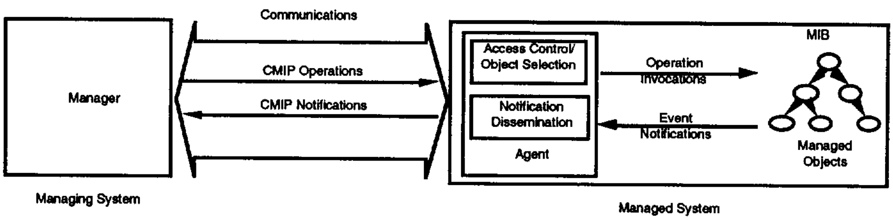

Figure 6 OSI Network Management Architecture

3.4 Managed-object Compatibility

Class inheritance is viewed as a mechanism that can improve interoperability between managed and managing systems in the face of asynchronous system-updates and vendor specialisation. A managing system may not have full understanding of certain specialised object classes but could, to some useful degree, manage instances of such classes as instances of more generalised classes of which it does have full understanding. Such interoperability can be achieved by the managing system essentially ignoring information it receives but does not understand—although ignoring information is potentially dangerous. Interoperability can also be achieved by the managed system through object substitutability [WeZd88] or object and service mutation.

It has been recognised that the managed-object subclassing hierarchy is not a subtyping hierarchy [Ame87] and thus instances of subclasses cannot be safely substituted for (and subsequently managed as) instances of their superclasses. This unfortunate state of affairs is because the inheritance rules [ISO92a] permit:

- invocation and result signatures to be extended;
- attribute ranges to be changed; and
- new notification-types to be added.

ISO/IEC 10165–1 [ISO92a] therefore specifies additional compatibility constraints on the inheritance rules to improve substitutability. It also imposes additional requirements on objects and object services to dynamically alter their behaviour to further improve substitutability. The combination of compatibility constraints and the requirements for dynamic alteration of object and service behaviour to improve substitutability is termed **allomorphism**.

4 Next-generation Network Management Standards

The current generation of OSI network-management standards is based on a model of a single managing-system communicating with a single managed-system using a single carriage protocol. Although these standards are an important and necessary first step to practical network-management solutions, the model and services must be evolved to meet the challenges of large, multi-protocol, mutable networks. In this section we give our views on the demands to be met by the next generation of network management standards and how the managed-object model and associated services may need to evolved.

4.1 Requirements

The next generation of network management standards must, we believe, reflect the following broad trends in network management:

1. **distributed** management—the managed and managing functions will be distributed throughout the management network;
2. **inter-networking**—multi management-protocols are a fact of life and have to be accommodated; and
3. **interoperability evolution**—management networks will be required to dynamically adapt to new object classes and new versions of existing object classes.

ISO is committed to an evolutionary approach to the further development of OSI network-management standards enabling the maximum degree of compatibility with existing implementations. A key tool in this regard is the ability to enhance protocol services by combining, rather than altering, existing protocols. Protocol standards are combined in new application contexts to provide enhanced capabilities. This permits a "mix-and-match" approach to standards development.

4.2 Distributed Management Architecture

The current OSI network-management architecture draws a clear distinction between the manag*ing* and manag*ed* functionality and where each resides. In practice, such distinctions are not so clear and will become more blurred as networks increase in size.

Management functionality will increasingly be distributed amongst peer and subordinate management systems on a functional, geographical, or operational basis. Similarly managed objects may be aggregated or fragmented over multiple managed-systems and managed relationships may span multiple managed-systems. In some cases the direct management of individual managed objects may be impractical and it will be necessary to provide standardised mechanisms for abstracting the characteristics of a number of managed objects into a higher-level surrogate object. For similar reasons it will be necessary to standardise mechanisms to define and manage *domains* of objects that are subject to the same *policy*. In fact the whole notion of distinct managing and managed systems will become blurred as intermediate systems provide run-time abstraction, delegation, and domain functions. The OSI managed-object model and services will need to be evolved to support such changes.

In order to obtain knowledge of the features of managed systems, the current standards require managed systems to be interrogated individually; as networks increase in size this will become impractical to carry out on a routine basis. Private caching of such knowledge can help, but this could be provided publicly in systems such as traders [ISO93c] and name servers [ISO91b]. It is likely that access to these services will be available by means of new application-contexts rather that by the enhancement of existing protocols.

There will be an increasing need for transactions involving multiple managed-systems and exhibiting so-called ACID properties (Atomicity, Consistency, Isolation, and Durability). ISO is currently developing an application context to combine Distributed Transaction Processing with CMIP [ISO93b].

4.3 Inter-network Management

Multiple management-protocols, difficult though they are for implementors to deal with, are a fact of life. Protocol mapping schemes have been proposed [NMF92] but such schemes require on-going enhancement to support evolving object-class libraries. Other proposal include adapter objects [OMG91], multiple interfaces to objects [ISO92c], and common object models [HoTh93]. A promising approach seems to be *federation* [Gor92], where management domains agree to federate by means of high-level inter-domain interfaces.

4.4 Interoperability Evolution

In the long term, we believe that management systems will have to accommodate new classes of resource and relationships dynamically. Thus formal notations describing the syntax and semantics of managed objects, aggregates of managed objects, and managed relationships will be required. Such descriptions will allow valued-added applications such as load-balancing, dynamic routing, and surveillance to automatically accommodate new resource types. For similar reasons we believe that managed object versioning and compatibility rules will need to evolve.

5 Conclusions

The managed-object model adopted by ISO to help in the development of network-management protocol standards nicely separates the generic from the resource-specific and permits a decentralised, but disciplined, approach to the development of resource-specific management messages. The model supports features, such as event notification and scoping, to meet requirements peculiar to network management. Object substitutability is complicated by the particular rules governing inheritance, however, a useful degree of interoperability is provided by means of allomorphism. The carriage protocol, CMIP, reflects network-management concerns of security, asynchronous operation, and bandwidth conservation.

The next generation of network-management standards will need to reflect the requirements of managing large, multi-protocol, mutable networks. The managed-object model and its related services will need to evolve to meet these requirements.

References

[Ame87] Inheritance and Subtyping in a Parallel Object-Oriented Language. In *Proceedings of European Conference on Object-Oriented Programming* (Oslo, Norway, August 1988). LNCS 322, Springer-Verlag, New York, 1988.

[Gor92] Goranson, H.T. Services in the SIRIUS-BETA Inter-Integration Domain. In *Enterprise Integration Modelling, Proceedings of the First International Conference.* Ed. Petrie, C.J. The MIT Press, Cambridge, Mass, 1992.

[HoTh93] Hodges, R. and Thompson, C. *Workshop Report, Workshop on Application Integration Architectures.* Texas Instruments, Dallas, Texas. 1993.

[ISO84] International Organization for Standardization and International Electrotechnical Commission. *Information Technology — Open Systems Interconnection — Basic Reference Model.* International Standard ISO/IEC 7498 : 1984.

[ISO89a] International Organization for Standardization and International Electrotechnical Commission. *Information technology — Open Systems Interconnection — Basic Reference Model — Part 4: Management Framework.* International Standard ISO/IEC 7498–4 : 1989.

[ISO89b] International Organization for Standardization and International Electrotechnical Commission. *Information Processing Systems — Text Communications — Remote Operations — Part 1: Model, Notation and Service Definition.* International Standard ISO/IEC 9072–1 : 1989.

[ISO89c] International Organization for Standardization and International Electrotechnical Commission. *Information Processing Systems — Text Communications — Remote Operations — Part 2: Protocol Specification.* International Standard ISO/IEC 9072–2 : 1989.

[ISO90] International Organization for Standardization and International Electrotechnical Commission. *Information Technology — Open Systems Interconnection — Specification of Abstract Syntax Notation One (ASN.1).* International Standard ISO/IEC 8824 : 1990.

[ISO91a] International Organization for Standardization and International Electrotechnical Commission. *Information Technology — Open Systems Interconnection — Common Management Information Protocol Specification.* International Standard ISO/IEC 9596 : 1991.

[ISO91b] International Organization for Standardization and International Electrotechnical Commission. *Information technology — Open Systems Interconnection — Management Information Services — The Directory.* International Standard ISO/IEC 9594 : 1991.

[ISO92a] International Organization for Standardization and International Electrotechnical Commission. *Information Technology — Open Systems Interconnection — Management Information Services — Structure of Management Information — Part 1: Management Information Model.* International Standard ISO/IEC 10165-1 : 1992.

[ISO92b] International Organization for Standardization and International Electrotechnical Commission. *Information Technology — Open Systems Interconnection — Management Information Services — Structure of Management Information — Part 4: Guidelines for the Definition of Managed Objects.* International Standard ISO/IEC 10165-4 : 1992.

[ISO92c] International Organization for Standardization and International Electrotechnical Commission. *Information Technology — Basic Reference Model for Open Distributed Processing — Part: 2 Descriptive Model.* Committee Draft ISO/IEC 10746–2. Dec. 1992.

[ISO92d] International Organization for Standardization and International Electrotechnical Commission. *Information technology — Open Systems Interconnection — Distributed transaction processing — OSI TP Model.* International Standard 10026–1 : 1992.

[ISO92e] International Organization for Standardization and International Electrotechnical Commission. *Information technology — Open Systems Interconnection — Distributed transaction processing — OSI TP Service.* International Standard 10026–2 : 1992.

[ISO92f] International Organization for Standardization and International Electrotechnical Commission. *Information technology — Open Systems Interconnection — Distributed transaction processing — OSI TP Protocol.* International Standard 10026–3 : 1992.

[ISO92g] International Organization for Standardization and International Electrotechnical Commission. Use of Z to Describe the Management Information Model. USA National Body Contribution, ISO/IEC JTC1/SC21/WG4 N1587. 1992.

[ISO93a] International Organization for Standardization and International Electrotechnical Commission. *Information Technology — Open Systems Interconnection — Management Information Services — Structure of Management Information — Part 2: General Relationship Model.* Committee Draft ISO/IEC 10165-7. Jan. 1993.

[ISO93b] International Organization for Standardization and International Electrotechnical Commission. *Information Technology — Open Systems Interconnection — Application Context for Systems Management with Transaction Processing.* Committee Draft ISO/IEC 11587, Jan. 1993.

[ISO93c] International Organization for Standardization and International Electrotechnical Commission. *Information Technology — Basic Reference Model for Open Distributed Processing — Part: 1 Overview and Guide to Use.* ISO/IEC JTC1/SC21/WG7 N755, Jan. 1993.

[NMF92] The Network Management Forum. *ISO/CCITT and Internet Management: Coexistence and Interworking Strategy*, Bernardsville, NJ, 1992.

[NMF93] Network Management Forum. *Comparison of the OMG and ISO/CCITT Object Models*, Bernardsville, NJ, 1993.

[OMG91] Object Management Group, *The Common Object Request Broker: Architecture and Specification*, Boulder, CO, 1991.

[SiMa91] Simon, L. D. and Marshall, L. S. Using VDM to Specify OSI Managed Objects. In *Proceedings of the IFIP TC6/WG6.1 Forth International Conference on Formal Description Techniques* (FORTE '91), Sydney, Australia, November 1991. Eds. G. A. Rose and K. R. Parker, Elsevier, 1991.

[WeZd88] Wegner, P. and Zdonik, S.B. Inheritance as an Incremental Modification Mechanism. In *Proceedings of European Conference on Object-Oriented Programming* (Oslo, Norway, August 1988). LNCS 322, Springer-Verlag, New York, 1988.