Network Working Group
Request for Comments: 3444
Category: Informational

A. Pras
University of Twente
J. Schoenwaelder
University of Osnabrueck

January 2003

On the Difference between Information Models and Data Models

Status of this Memo

Copyright Notice

Abstract

There has been ongoing confusion about the differences between Information Models and Data Models for defining managed objects in network management. This document explains the differences between these terms by analyzing how existing network management model specifications (from the IETF and other bodies such as the International Telecommunication Union (ITU) or the Distributed Management Task Force (DMTF)) fit into the universe of Information Models and Data Models.

This memo documents the main results of the 8th workshop of the Network Management Research Group (NMRG) of the Internet Research Task Force (IRTF) hosted by the University of Texas at Austin.

Table of Contents

1. Introduction

Currently multiple languages exist to define managed objects. Examples of such languages are the Structure of Management Information (SMI) [1], the Structure of Policy Provisioning Information (SPPI) [2] and, within the DMTF, the Managed Object Format (MOF) [3]. Despite the fact that multiple languages exist, a number of people still believe that none of these languages really suits all needs.

There have been many discussions to understand the advantages and disadvantages, as well as the main differences, between various languages. For instance, the IETF organized a BoF on "Network Information Modeling" (NIM) at its 48th meeting (Pittsburgh, August 2000). During these discussions, it turned out that people had a different understanding of the main terms, which caused confusion and long arguments. In particular, the meaning of the terms "Information Model" (IM) and "Data Model" (DM) turned out to be controversial.

In an attempt to address this issue, the IRTF Network Management Research Group (NMRG) dedicated its 8th workshop (Austin, December 2000) to harmonizing the terminology used in information and data modeling. Attendees included experts from the IETF, DMTF and ITU, as well as academics who do research in this field (see the Acknowledgments section for a list of participants). The main outcome of this successful workshop -- a better understanding of the terms "Information Model" and "Data Model" -- is presented in this document.

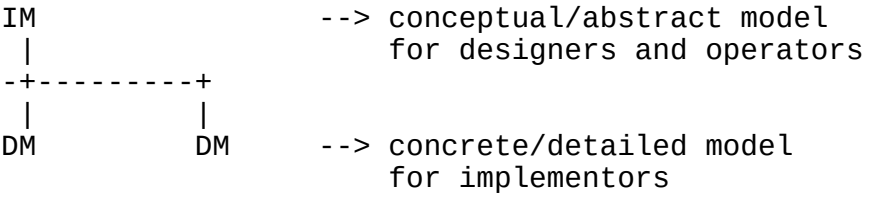
Short definitions of these terms can also be found elsewhere (e.g., in RFC 3198 [8]). Compared to most other documents, this one explains the rationale behind the proposed definitions and provides examples.

2. Overview

One of the key observations made at the NMRG workshop was that IMs and DMs are different because they serve different purposes.

The main purpose of an IM is to model managed objects at a conceptual level, independent of any specific implementations or protocols used to transport the data. The degree of specificity (or detail) of the abstractions defined in the IM depends on the modeling needs of its designers. In order to make the overall design as clear as possible, an IM should hide all protocol and implementation details. Another important characteristic of an IM is that it defines relationships between managed objects.

DMs, conversely, are defined at a lower level of abstraction and include many details. They are intended for implementors and include protocol-specific constructs.



The relationship between an IM and DM is shown in the Figure above. Since conceptual models can be implemented in different ways, multiple DMs can be derived from a single IM.

Although IMs and DMs serve different purposes, it is not always possible to precisely define what kind of details should be expressed in an IM and which ones belong in a DM. There is a gray area where IMs and DMs overlap -- just like there are gray areas between the models produced during the analysis, high-level design and low-level design phases in object-oriented software engineering. In some cases, it is very difficult to determine whether an abstraction belongs to an IM or a DM.

3. Information Models

IMs are primarily useful for designers to describe the managed environment, for operators to understand the modeled objects, and for implementors as a guide to the functionality that must be described and coded in the DMs. The terms "conceptual models" and "abstract models", which are often used in the literature, relate to IMs. IMs can be implemented in different ways and mapped on different protocols. They are protocol neutral.

An important characteristic of IMs is that they can (and generally should) specify relationships between objects. Organizations may use the contents of an IM to delimit the functionality that can be included in a DM.

IMs can be defined in an informal way, using natural languages such as English. An example of such an IM is provided by RFC 3290 [9], which describes a conceptual model of a Diffserv Router and specifies the relationships between the components of such a router that need to be managed. Within the IETF, however, it is exceptional that an IM be explicitly described, and even more that the IM and DM be specified in separate RFCs. In such cases, the document specifying the IM is usually an Informational RFC whereas the document defining the DM usually follows the Standards Track [4]. In general, most of

the RFCs that define an SNMP Management Information Base (MIB) module also include some kind of informal description explaining parts of the model behind that MIB module. Such a model can be considered as a document of an IM. An example of this is RFC 2863, which defines "The Interfaces Group MIB" [10]. But most MIB modules published to date include only a rudimentary and incomplete description of the underlying IM.

Alternatively, IMs can be defined using a formal language or a semi-formal structured language. One of the possibilities to formally specify IMs is to use class diagrams of the Unified Modeling Language (UML). Although such diagrams are still rarely used within the IETF, several other organizations routinely use them for defining IMs, including the DMTF, the ITU-T SG 4, 3GPP SA5, the TeleManagement Forum, and the ATM Forum. An important advantage of UML class diagrams is that they represent objects and the relationships between them in a standard graphical way. Because of this graphical representation, designers and operators may find it easier to understand the underlying management model. Although there are other techniques to graphically represent objects and relationships (e.g., Entity-Relationship (ER) diagrams), UML presents the advantage of being widely adopted in the industry and taught in universities. Also, many tools for editing UML diagrams are now available. UML is standardized by the Object Management Group (OMG) [5].

In general, it seems advisable to use object-oriented techniques to describe an IM. In particular, the notions of abstraction and encapsulation, as well as the possibility that object definitions include methods, are considered to be important.

4. Data Models

Compared to IMs, DMs define managed objects at a lower level of abstraction. They include implementation- and protocol-specific details, e.g. rules that explain how to map managed objects onto lower-level protocol constructs.

Most of the management models standardized to date are DMs. Examples include:

- Management Information Base (MIB) modules defined within the IETF. The language (syntax) used to define these DMs is called the "Structure of Management Information" (SMI) [1] and is derived from ASN.1 [6].

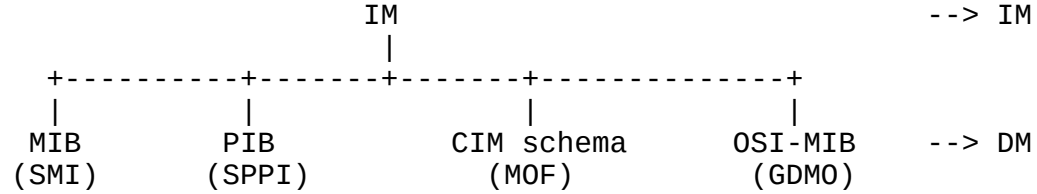
- o Policy Information Base (PIB) modules, developed within the IETF. Their syntax is defined by the "Structure of Policy Provisioning Information" (SPPI) [2], which is close to SMI and is also derived from ASN.1 [6].

- o Management Information Base (MIB) modules, originally defined by the ISO and currently maintained and enhanced by the ITU-T. The syntax of these DMs is specified in the "Guidelines for the Definition of Managed Objects" (GDMO) [7]. GDMO MIB modules make use of object-oriented principles.

- o CIM Schemas, developed within the DMTF. The DMTF publishes them in two forms: graphical and textual. The graphical forms use UML diagrams and are not normative (because not all details can be represented graphically). They can be downloaded from the DMTF Web site in PDF and Visio formats. (Visio is a tool to draw UML class diagrams.) The textual forms are normative and written in a language called the "Managed Object Format" (MOF) [3]. CIM Schemas are object-oriented.

Because CIM Schemas support a graphical notation whereas IETF MIB modules do not, designers and operators may find it easier to understand CIM Schemas than IETF MIB modules. One could therefore argue that CIM Schemas are closer to IMs than IETF MIB modules.

The Figure below summarizes these examples. The languages that are used to define the DMs are shown between brackets.



To illustrate what details are included in a DM, let us consider the example of IETF MIB modules. As opposed to IMs, IETF MIB modules include details such as OID assignments and indexing structures. The relationships defined in the IM are implemented as OID pointers or realized through indexing relationships specified in INDEX clauses. Many other implementation-specific details are included, such as MAX-ACCESS and STATUS clauses and conformance statements.

A special kind of DM language is the SMIng language defined by the NMRG. This language was designed at a higher conceptual level than SMIv1/SMIv2 and SPPI. In fact, one of the intentions behind SMIng was to stop the proliferation of different DM languages within the IETF and to harmonize the various models. As a result, MIB and PIB

modules defined in SMIng can be mapped on different underlying protocols. There is a mapping on SNMP and another mapping on COPS-PR. SMIng is therefore more protocol neutral than other IETF approaches. It also supports some object-oriented principles and provides extension mechanisms that allow the addition of new features (e.g., the support for methods). New features can then be used when they are supported by the underlying protocols, without breaking SMIng implementations. Still, SMIng should be considered a DM. For instance, to express relationships between managed objects, techniques such as UML and ER diagrams still give better results because these diagrams are easier to understand.

Note that the IETF SMING Working Group took a different approach and decided not to use the SMIng language defined by the NMRG. Instead, the SMING Working Group is developing a third version of SMI (SMIv3) that is primarily targeted towards SNMP, and which incorporates only some of the ideas developed within the NMRG.

5. Security Considerations

The meaning of the terms Information Model and Data Model has no direct security impact on the Internet.

6. Acknowledgments

The authors would like to thank everyone who participated in the 8th NMRG workshop (in alphabetic order): Szabolcs Boros, Marcus Brunner, David Durham, Dave Harrington, Jean-Philippe Martin-Flatin, George Pavlou, Robert Parhonyi, David Perkins, David Sidor, Andrea Westerinen and Bert Wijnen.

7. Normative References

[1] McCloghrie, K., Perkins, D. and J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[2] McCloghrie, K., Fine, M., Seligson, J., Chan, K., Hahn, S., Sahita, R., Smith, A. and F. Reichmeyer, "Structure of Policy Provisioning Information (SPPI)", RFC 3159, August 2001.

[3] Distributed Management Task Force, "Common Information Model (CIM) Specification Version 2.2", DSP 0004, June 1999.

[4] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[5] Object Management Group, "Unified Modeling Language (UML), Version 1.4", formal/2001-09-67, September 2001.

[6] International Organization for Standardization, "Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)", International Standard 8824, December 1987.

[7] International Telecommunication Union, "Information technology - Open Systems Interconnection - Structure of Management Information: Guidelines for the Definition of Managed Objects", Recommendation X.722, 1992.

8. Informative References

[8] Westerinen, A., Schnizlein, J., Strassner, J., Scherling, M., Quinn, B., Herzog, S., Huynh, A., Carlson, M., Perry, J. and S. Waldbusser, "Terminology for Policy-Based Management", RFC 3198, November 2001.

[9] Bernet, Y., Blake, S., Grossman, D. and A. Smith, "An Informal Management Model for Diffserv Routers", RFC 3290, May 2002.

[10] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

9. Authors' Addresses

Aiko Pras
University of Twente
PO Box 217
7500 AE Enschede
The Netherlands

Phone: +31 53 4893778
EMail: pras@ctit.utwente.nl

Juergen Schoenwaelder
University of Osnabrueck
Albrechtstr. 28
49069 Osnabrueck
Germany

Phone: +49 541 969-2483
EMail: schoenw@informatik.uni-osnabrueck.de

10. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.