Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on July 25, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2008).
Abstract

With rapid development of next generation networks, it is expected that a separate effort to study data modeling languages in the interest of network management should be undertaken. Based on a good understanding of the requirements of data modeling in next generation network management domain, evaluation on management data modeling languages becomes an essential way for the purpose of standardization to replace proprietary data models in the near future. Our project aims to establish a framework for evaluation to measure the capabilities of management data modeling languages in meeting those requirements by a set of criteria, which are modeling approaches, interoperability, readability, conformance, data representation, extensibility and security considerations.
# Table of Contents

1. Introduction ................................................. 4
2. Proposed Evaluation Framework ................................. 5
   2.1. Modeling Approaches .................................... 5
   2.2. Interoperability ....................................... 5
       2.2.1. Protocol Independence ............................. 5
       2.2.2. Naming Independence ............................... 6
   2.3. Readability .............................................. 6
       2.3.1. Human Readability ................................. 6
       2.3.2. Machine Readability ............................... 6
   2.4. Data Representation ..................................... 7
       2.4.1. Diversity of Data Types .......................... 7
       2.4.2. Specification of Configuration Data, State Data
               and Statistics Data .................................. 7
   2.5. Conformance .............................................. 7
       2.5.1. Backward Compatibility ............................ 7
       2.5.2. Versioning .......................................... 8
       2.5.3. Definition of Event Notification Messages ......... 8
       2.5.4. Definition of Error Messages ..................... 8
   2.6. Extensibility ............................................. 8
       2.6.1. Extensibility of Data Structures .................. 8
       2.6.2. Extensibility of Data Types ........................ 8
       2.6.3. Extensibility of Elements and Attributes ........... 9
   2.7. Security Considerations .................................. 9
       2.7.1. Granularity of Access Control ..................... 9
       2.7.2. Lock Mechanism ..................................... 9
3. Validation .................................................. 11
4. Conclusions .................................................. 16
5. Security Considerations ....................................... 17
6. IANA Considerations .......................................... 18
7. References .................................................... 19
   7.1. Normative References .................................... 19
   7.2. Informative References .................................. 20
Authors’ Addresses ............................................. 21
Intellectual Property and Copyright Statements .................. 22
1. Introduction

The definition of Information Model (IM) and Data Model (DM) should be seriously considered for network management solutions. IMs always model Managed Objects (MOs) at a conceptual level and are protocol-neutral, while DMs are defined at a concrete level, implementing in different ways and are protocol-specific. As for each network management model, a data modeling language is quite necessary for the description of the managed resources.

Obviously, the work on evaluating data modeling languages for the sake of next generation network management is comparatively indispensable. However, the fact is that a reused evaluation framework for management data modeling languages is still greatly lacking. The aim of our project is then to establish an evaluation framework to measure the capabilities of management data modeling languages in adapting to the requirements of ever-evolving network management and apply it to examine existing languages for validation.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].
2. Proposed Evaluation Framework

Nowadays data modeling is under hot research, but it is still a preparatory period for corresponding research on network management. It becomes necessary to put forward a universal evaluation framework for data modeling languages to level their capabilities in satisfying the requirements of future network management. And our proposed evaluation framework is based on a set of criteria, which are modeling approaches, interoperability, readability, conformance, data representation, extensibility and security considerations.

2.1. Modeling Approaches

Four main modeling approaches should be considered, including data-oriented one, command-oriented one, object-oriented/object-based one and document-oriented one. The data-oriented approach models all management aspects through data objects, and at least two operations ("get" and "set") should be defined. The command-oriented approach defines a large number of management operations, specifying not the details but the commands to get/set selected information. The object-oriented/object-based approach combines the data-oriented approach and the command-oriented approach in view of integration. The document-oriented approach represents state information, statistics information and configuration information of a device as a structured document.

Future management data modeling language should implement an integration of various modeling approaches, a possible scenario of which is a data-oriented view for monitoring, a command-oriented view for operations and a document-oriented view for configuration. Note that, the very language should avoid implementing the same function with simple combination of these approaches.

2.2. Interoperability

2.2.1. Protocol Independence

Protocol independence means that the language defines management data supporting any protocol instead of belonging to some specific protocol. In other words, the DM defined by this language can be implemented on any platform that installs different protocols.

In order to integrate with existing network management technologies, DMs should be defined by protocol-neutral modeling languages that can be mapped on different underlying protocols.
2.2.2. Naming Independence

Naming independence is a desired mechanism provided by the language that specifies how name collisions are handled, and thus uniquely identifies attributes, groups of attributes, and events.

Since a management data modeling language is required to be protocol-independent, and protocols typically use different approaches to name instances, it has to support multiple instance naming systems. Being naming independence, the language needs to think about the relationships between DMs. More efforts should then be made to ensure implementation of the language not being interfered by problems of different objects from multiple modules with the same name.

2.3. Readability

2.3.1. Human Readability

Human readability is the capability by which administrators can directly read and understand representations including input and output (requirements, responses, error messages, etc).

Only if administrators conveniently read and understand meanings of the DM, can they efficiently write and use it. This also does favor to the interoperation between DMs and administrators. It is then desirable that all DMs used for a network management solution are well formed according to the data modeling language.

2.3.2. Machine Readability

Machine readability refers to the feature that description of the relevant DM can be understood by computers, thus related applications can be quickly developed. Its implementation largely depends on semantic expressiveness, and its speed also has very close relation with the parse-ability of machines. Note that, each data modeling language has a different level of semantic expressiveness, which includes several facets like concepts, relations and behaviors, and it is not easy to measure semantic expressiveness. Nowadays, this problem can be temporally reduced to a problem of integrating different management data modeling languages.

Future network management protocol aims in enabling the system to automate its management process. From this point of view, semantic expressiveness is quite essential for better machine readability. For example, the behavior defined by data modeling languages should be well understood, so that the automation requirements towards network management can be promoted and become much more promising.
2.4. Data Representation

2.4.1. Diversity of Data Types

Diversity of data types implies that data types should be diverse enough so that the modeling language can support various data. Hence, data with a suitable type can be clearly described and understood for users.

More structured data types are needed to make DMs much simpler to design and implement in the field of network management. It is said to be better that data types defined by a management modeling language should be as various as possible and emphasis should be placed on creating application-level ones especially for the configuration.

2.4.2. Specification of Configuration Data, State Data and Statistics Data

Configuration data is the set of read-write data, while both state data and statistic data are read-only, only different in the scope of practical use. The DM specified for a network device should identify what is configuration data, what is state data and what is statistic data without the trouble to separate container elements.

When a device is performing configuration operations, a number of problems would arise if state data and statistic data were included in configuration data, and in order to account for these issues, future network management protocol should recognize the difference among state data, statistic data and configuration data, and provides operations for each. Thus for the data modeling language it then becomes necessary to make a clear distinction between (a) configuration data and (b) state data and statistic data.

2.5. Conformance

2.5.1. Backward Compatibility

Backward compatibility illuminates that new versions of the data modeling language can be used to define the relevant DM for the purpose of network management as the older one used to do.

This capability is quite important, for the reason that it eliminates the need to start over when a new data modeling language is used. As for network management, it means that new versions of the data modeling language that define management content can be rolled out in a way that does not break existing supporters.
2.5.2. Versioning

Versioning explains that each version of the data modeling language is complete, thus easy to control.

This capability promotes the maintenance of backwards compatibility and does not need to change to the new language if it is also backwards compatible.

2.5.3. Definition of Event Notification Messages

Definition of event notification messages needs to be ensured by the data modeling language to allow a single definition of notification content to be sent either asynchronously or synchronously.

Network management protocols are desired to support asynchronous notifications, and as for a future data modeling language, not only notification messages but also types of the events should be clearly identified.

2.5.4. Definition of Error Messages

Definition of error messages indicates that error messages generated by network management applications should be identified by the data modeling language.

Error messages, which are created by applications as a result of performing network management operations against the related DM, need to be included in the modeling language.

2.6. Extensibility

2.6.1. Extensibility of Data Structures

Extensibility of data structures shows that the data modeling language has the capability to add new data structures with no need to affect available ones, and thus expresses the relations among data effectively and operates on data effortlessly.

With increase of isomerization and complexity of the Internet, there is a great need for the data modeling language to have this ability for the practice of network management.

2.6.2. Extensibility of Data Types

Extensibility of data types reveals that data types defined by the modeling language can be extended easily, so that the language can support various kinds of management data both simply and clearly.
Considering application of future protocols to manage heterogeneous networks, especially for the use of configuration management, more and more new data types should be added to satisfy different presentation needs.

2.6.3. Extensibility of Elements and Attributes

Extensibility of elements and attributes means that types of element nodes and attributes defined by the data modeling language shouldn’t be too fixed to extend. When there is a need to add new types of elements or new attributes for existing elements, the operation of "creation" should be done properly conveniently.

Objects of great variety need to be managed in next generation network management, which means the demand of adding object types. Hence, the data modeling language should have this capability, in order that everyone can manage the objects both simply and effectively.

2.7. Security Considerations

2.7.1. Granularity of Access Control

Granularity of access control refers to the precision of accessing data from the relevant DM. There are mainly two levels of granularity, which are coarse one and fine one. Using coarse granularity of access control, a bulk of data can be retrieved and edited from the DM, such as getting the whole data from MIB. And fine granularity refers to a detailed operation to a small part of data, such as elements.

Both coarse granularity and fine granularity have their advantages and disadvantages. For example, implementation of coarse granularity is simple, while reusability is very poor. Hence, the tradeoff between coarse granularity and fine granularity becomes quite necessary for data modeling especially when merging and mapping information across multiple systems or data stores, since granularity may not match in the process of mapping.

2.7.2. Lock Mechanism

Lock mechanism cannot be ignored by management data modeling languages in order to guarantee security of the configuration.

As to some devices, it is quite hard to determine which parameters are administratively configured and which are obtained via mechanisms such as routing protocols. Taking configuration management into consideration, an implementation should figure out how users lock an
entire configuration database, even if users do not have "write" access to the entire database. Furthermore, it is also of great importance to a partial lock of a configuration data store. Although it's not clear how serious this problem is, the solution is now an open issue.
3. Validation

The languages being measured are Guidelines for the Definition of Managed Objects (GDMO) for CMIP, Structure of Management Information (SMI) with its different versions for SNMP, Management Information Format (MIF) for DMI, Managed Object Format/Common Information Model (MOF/CIM) for WBEM, and Structure of Management Information, next generation (SMIng) for both SNMP and COPS-PR, all of which are available nowadays in the field of network management.

First, Table 1 shows which modeling approach each language adopts in the interest of network management. As is indicated in Table 1, object-based approach is especially distinguished from object-oriented approach, since the former one is an incomplete version of the latter one.

<table>
<thead>
<tr>
<th></th>
<th>GDMO</th>
<th>SMI</th>
<th>MIF</th>
<th>MOF/CIM</th>
<th>SMIng</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data-Oriented Approach</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Command-Oriented Approach</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object-Based Approach</td>
<td></td>
<td></td>
<td>#</td>
<td></td>
<td>#</td>
</tr>
<tr>
<td>Object-Oriented Approach</td>
<td>#</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Document-Oriented Approach</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Modeling approach adopted by each language

Second, Table 2 illustrates the comparison result in terms of the evaluation criteria listed in Section 2 except for modeling approaches, the comparison on which has been presented in Table 1. Note that, our measurement is classified as the following four levels.

a. A minus sign (-) means that the language does not have such a capability.

b. An asterisk sign (*) denotes that the language is weak in this capability.

c. A plus sign (+) is used when the language is good at this capability.
d. Two plus sign (++) is placed when the language completely possesses this capability.

It can be concluded from Table 2 that, SM Ing is the language with best implementation of most criteria, while SMI and MOF/CIM are near SM Ing capabilities.
<table>
<thead>
<tr>
<th>Language</th>
<th>GDMO</th>
<th>SMI</th>
<th>MIF</th>
<th>MOF/CIM</th>
<th>SMIng</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interoperability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protocol Independence</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>*</td>
</tr>
<tr>
<td>Naming Independence</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Readability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Human Readability</td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>+</td>
</tr>
<tr>
<td>Machine Readability</td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Data Representation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diversity of Data Types</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Specification of Configuration Data, State Data and Statistics Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conformance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Backward Compatibility</td>
<td>-</td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>-</td>
</tr>
<tr>
<td>Versioning</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>*</td>
</tr>
<tr>
<td>Definition of Event Notification Messages</td>
<td>-</td>
<td>++</td>
<td>-</td>
<td>-</td>
<td>++</td>
</tr>
<tr>
<td>Definition of Error Messages</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Extensibility</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extensibility of Data Structures</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Extensibility of Data Types</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Extensibility of Elements and Attributes</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Security Considerations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 2: Summary of the compared characteristics

Undoubtedly, existing data modeling languages play an important role in traditional network management. Especially, SMI has commendably implemented performance management in SNMP-based network management. However, networks have become more and more complex and heterogeneous as well, so DMs based on these data modeling languages don’t seem to have enough ability to meet the requirements towards future network management. Using our proposed evaluation framework, the deficiencies of available languages have been clearly shown above, the main point of which is summarized as follows.

a. All of them only use a single modeling approach not integration demanded by data modeling.

b. Their interoperability is insufficient, though some efforts have been made. Traditional DMs are designed especially for certain protocols, always having close relation with specific operations of the protocols, and take their individual naming rules, way of expression, and so on, all of which lead to the poverty of universality in meeting the interoperable requirements of next generation network management solutions.

c. Their human readability is quite weak, while their machine readability is also fairly poor, which is far from the automatic aim of next generation network management.

d. Traditional DMs put emphasis on performance management, but take little consideration into configuration management. Future DMs should strengthen this point in order to satisfy a higher demand of configuration management, which has been clearly shown as one of the most important objectives of next generation network management.

e. As for conformance required by data modeling, backward compatibility and versioning are two features that are related but with a different focus. Additionally, few of the languages lay emphases on definition of event notification messages with exception of SMI and SMIng, which are in full procession of this capability. Furthermore, all these languages pay no attention to definition of error messages.
f. Their extensibility is quite deficient, which can not satisfy application needs of network management solutions.

g. In DMs specified by traditional modeling languages, mechanisms related to access control and lock is so simple that it cannot satisfy the demands of network security and adapt to complex network operations as well. Additionally, the level of network security requirements is being higher and higher with the popularity of next generation networks.
4. Conclusions

An evaluation framework for management data modeling languages has been established and validated by applying it to summarize the characteristics of existing languages through comparison. This framework is universal and can be reused to study data modeling in network management domain. Future work focuses on case study for application of this framework to next generation network management.
5. Security Considerations

None.
6. IANA Considerations

None.
7. References

7.1. Normative References


7.2. Informative References

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, 
"Introduction and Applicability Statements for Internet-
Authors’ Addresses

Debao Xiao
Institute of Computer Network and Communication
Huazhong Normal University
WuHan, HuBei 430079
P.R.China

Phone: +86 027 6786 6108
Email: dbxiao@mail.ccnu.edu.cn

Hui Xu
Institute of Computer Network and Communication
Huazhong Normal University
WuHan, HuBei 430079
P.R.China

Phone: +86 027 6136 8682
Email: xuhui_1004@hotmail.com