An Evaluation Framework for Data Modeling Languages in Network Management Domain

Hui Xu, *Student Member*, *IEEE*, Debao Xiao, Yanan Chang and Limiao Chen

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. This paper establishes 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. Usability of the proposed framework is validated by its application to compare existing management data modeling languages and the result shows that, SMIng is the language with best implementation of most criteria, while SMI and MOF/CIM are near SMIng capabilities.**

Index Terms—**Data modeling language, evaluation framework, network management, next generation network.**

I. Introduction

Since network management research and standardization started in late 1980s, several solutions have been proposed, from protocol-based ones (mainly including CMIP and SNMP), to policy-based ones, and to distributed ones with the use of COBRA and Web technologies. Meanwhile, with the evolution of the Internet, complexity of computer networks has greatly increased, when more and more network resources need to be effectively managed. Hence, new techniques have been employed by either various commercial network management solutions or research communities, such as XML-based network management (possibly with the use of Web services technologies), cooperative network management, improved Root Cause Analysis (RCA) techniques, semantic network management, active network management, predictive network management, autonomic network management with intelligent self-managing devices, and so on [1].

So 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.

To the best of our knowledge, few studies have been done on describing the capabilities of data modeling languages in network management domain. Reference [2] assembles a list of data modeling languages available for corresponding network management technologies that are used or under active development, but the evaluation is still general, only identified as strong, neutral and weak points without a detailed classification for characteristics. Reference [3] [4] regard management data modeling languages as a term of lightweight ontology because they just define information of the management domain without definition of the axioms or constraints present in heavyweight ontology, which makes them difficult to reason, and compare them in terms of their semantic expressiveness, the evaluation on which is limited to only one facet. Reference [5] provides some of the lessons learned from the SMIng project, which need to be considered by designers of future data modeling languages for network management protocols, calling for further summarization.

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 this paper 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 remainder of this paper is organized as follows. Section II proposes a common evaluation framework and in order to validate it, Section III gives a brief presentation to data modeling languages available in the field of network management. Consequently in Section IV, the proposed framework is applied to summarize the characteristics of possible languages through comparison and its usability is

Manuscript received November 23, 2007. This work was supported in part by the Scientific and Technological Planning Project of Wuhan City, P. R. China under Grant No. 200710421130 and the Key Scientific and Technological Project of Hubei Province, P.R. China under Grant No. 2004AA103A01.

Hui Xu is with the Institute of Computer Network and Communication, Huazhong Normal University, Wuhan, 430079, P. R. China (phone: 86-027-61368682; e-mail: xuhui_1004@hotmail.com).

Debao Xiao is with the Institute of Computer Network and Communication, Huazhong Normal University, Wuhan, 430079, P. R. China (e-mail: dbxiao@mail.ccnu.edu.cn).

Yanan Chang is with the Institute of Computer Network and Communication, Huazhong Normal University, Wuhan, 430079, P. R. China (e-mail: cyn_23@mails.ccnu.edu.cn).

Limiao Chen is with the Institute of Computer Network and Communication, Huazhong Normal University, Wuhan, 430079, P. R. China (e-mail: chenlimiao@mails.ccnu.edu.cn).

also demonstrated. Section V concludes this paper.

II. 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.

A. *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.

B. *Interoperability*

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) *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.

C. *Readability*

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) *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.

D. *Data Representation*

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) *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.

E. Conformance

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) 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.

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.

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.

F. Extensibility

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) 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.

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.

G. Security Considerations

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) 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.

III. Language Presentation

In order to validate the proposed evaluation framework, for one thing, the languages being measured, which 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, will be presented in chronological order with a brief introduction.

A. Guidelines for the Definition of Managed Objects

Specified in ITU-T X.722/ISO/IEC 10165, GDMO is a structured description language that provides a way to define classes of objects for demonstration of their attributes, behaviors and ancestry, adopting Abstract Syntax Notation One (ASN.1) as its format. GDMO uses an object-oriented paradigm, always describing the managed resources either concrete or abstract in the Internet, mainly for the definition of constructs and behaviors of MOs for the sake of TMN-based systems and CMIP services. Moreover, it adds some other features to allow a better reuse of the defined management data, which at the same time adds a lot of complexity to the language.

During the past years, GDMO has been widely used in the area of TMN management, but some drawbacks still exist. For example, its power of expression is not strong enough, especially when being used to describe the feature of behaviors. Additionally, instead of formalized definitions, it uses the natural languages, such as the BEHAVIOUR template, which is open to any definition, hence not so accurate and always causing ambiguity.

B. Structure of Management Information

Originally developed from the similar concept in OSI network management, SMI (including two versions, SMIv1 and SMIv2) defines organization, composing and identifier used in the framework of SNMP. The objects defined by SMI are presented as scalar variables or tables, making it rather simple to construct, and data types are much more than GDMO, for SMI has three kinds of types, including simple ones, structured ones and application ones.

Unfortunately, SMI also has some drawbacks, which hinder the application of SNMP to manage future networks. The root lies in the fact that SMI uses a data-oriented approach to model all management aspects. First of all, SMI is insufficient to represent hierarchical network configuration data, which is one of the main reasons for SNMP being used mostly in monitoring for fault and performance but hardly used for configuration management. Second, SMI is such a conceptually simple language that it is usually quite difficult to be used for modeling complex management operations.

C. Management Information Format

As a component of DMI, MIF was first proposed in 1993, which is a format used to describe a hardware or software component for the definition of desktop-related information. As a text file, MIF consists of one or more groups with attributes for the description of each component and tables as well, which are somewhat similar to SMI, or even simpler, since table keys are always internal to corresponding tables, and associations can not be defined in this way.

Note that, DMI-to-SNMP translation is always necessary, for its objective is to promote the integration of SNMP-based and DMI-based solutions. And the most important question in this translation is the mapping from MIF to SMI.

D. Managed Object Format/Common Information Model

Firstly proposed by DMTF in 1997, CIM is the core part of WBEM, which is developed to solve the problem of heterogeneous management environment. CIM provides a data modeling environment in the form of object-oriented design diagrams and a language-neutral description of the model known as the MOF, which is also object-oriented and much more powerful than MIF. Based on Interface Definition Language (IDL), MOF contains a set of intrinsic data types. With the goals of better readability, its syntax provides a way to describe object-oriented class and instance definitions in textual form. Moreover, its format is quite simple and easy to edit, so that it can be comparatively readable by both humans and machines.

Some other characteristics need to be seriously considered, which are (a) MOF only defines information of types and descriptions, hence it is quite difficult to extend, and (b) CIM specification describes the mappings from MOF to other data modeling languages in network management domain, but the syntactic and semantic conformance in the mapping process remains difficult to achieve.

E. Structure of Management Information, next generation

Proposed by IRTF, the SMIng project started in 1999, aiming to address some drawbacks of SMIv2 and create a new kind of management data modeling language to integrate SMI and Structure of Policy Provisioning Information (SPPI), avoiding the use of ASN.1. Applying an object-based approach to model MOs, SMIng has more advantages in expressiveness compared to SMI, such as better capability in defining data types, improved table definition, consideration of some operations, definition of attribute groups and event groups. Additionally, it is also possible to define extensions, which specify new elements by providing the syntax with which they should comply. However, due to disagreement of both the SMIng syntax and the relationship between SMIng and SMI, SMIng didn't finally become a standard data modeling language for network management.

Although it was expected that SMIng be an integrated data modeling language that could be adapted for different network management protocols and thus closer to an information modeling language, the complexity of design increased when moving towards being a protocol-neutral language, much more than designers used to think of. Additionally, the commonality of SMIng is only limited to two protocols, which are SNMP and COPS-PR. Furthermore, since the restrictive difference of various protocols is quite great, the mapping method adopted by SMIng fails to provide a good treatment with this problem.

IV. Validation

In order to justify the validity of the proposed evaluation framework, we apply it to compare five typical management data modeling languages, which have been introduced in Section III. And the characteristics of these languages will then be summarized through comparison based on given criteria for evaluation. During this process, usability of the proposed framework will also be illuminated.

A. Comparison

Using the common evaluation framework, the comparison of data modeling languages available in network management domain is performed in two steps.

First, Table I shows which modeling approach each language adopts in network management domain.

Note that, in Table I, we especially distinguish object-based approach from object-oriented approach, since the former one is an incomplete version of the latter one.

TABLE I. MODELING APPROACH ADOPTED BY EACH LANGUAGE

Modeling Approaches	Data Modeling Languages				
	GDMO	*SMI*	*MIF*	*MOF/ CIM*	*SMIng*
Data-Oriented		√			
Command-Oriented					
Object-Based			√		√
Object-Oriented	√			√	
Document-Oriented					

Second, Table II demonstrates the comparison result in terms of the criteria for evaluation except for modeling approaches, the comparison in which has been presented in Table I. And our measurement is classified as the following four levels.

• A minus sign (–) means that the language does not have such a capability

• An asterisk sign (*) denotes that the language is weak in the capability

• A plus sign (+) is used when the language is good at the capability

• Two plus sign (++) is placed when the language completely possesses the capability

As is indicated in Table II, some facts in terms of criteria can be gained as follows.

(1) Almost all the current data modeling languages are weak in features related to data representation, extensibility and security considerations, which reveals that these three capabilities are quite important when modeling management data and need to be taken into consideration fairly earlier than others.

(2) Only a few facets of both interoperability and conformance are involved in some particular languages, one example of which is SMIng, only focusing on half of these facets. As for next generation network management, this level is still far from the aim.

(3) All these languages attach some importance to readability, especially for machine readability, on which they adopt different methods to promote their semantic expressiveness.

It can then be concluded from Table II that, SMIng does best implementation of most criteria, and SMI and MOF/CIM follow SMIng in the capabilities of data modeling.

TABLE II. COMPARISON RESLUT

Criteria	Data Modeling Languages				
	GDMO	*SMI*	*MIF*	*MOF/ CIM*	*SMIng*
Interoperability					
Protocol Independence	–	–	+	–	*
Naming Independence	–	–	–	–	–
Readability					
Human Readability	*	*	*	*	+
Machine Readability	+	*	*	+	+
Data Representation					
Diversity of Data Types	–	*	–	–	–
Specification of Configuration Data, State Data and Statistics Data	–	–	–	–	–
Conformance					
Backward Compatibility	–	+	*	*	–
Versioning	–	–	–	+	*
Definition of Event Notification Messages	–	++	–	–	++
Definition of Error Messages	–	–	–	–	–
Extensibility					
Extensibility of Data Structures	–	–	–	–	–
Extensibility of Data Types	–	+	–	+	+
Extensibility of Elements and Attributes	–	–	–	–	+
Security Considerations					
Granularity of Access Control	–	+	–	–	+
Lock Mechanism	–	–	–	–	–

B. *Summary*

1) *For this comparison*

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.

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

(2) 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.

(3) 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.

(4) 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.

(5) 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.

(6) Their extensibility is quite deficient, which can not satisfy application needs of network management solutions.

(7) 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.

2) For the proposed framework

The comparison result presented in TABLE I and TABLE II can be utilized to demonstrate the usability of our proposed evaluation framework. As stated above, the study in Reference [3] [4] can be induced to "machine readability" as a facet of one evaluation criteria in the framework. Moreover, the study in Reference [5] will be taken as an example for further argumentation, as illustrated in Fig. 1.

As is demonstrated in Fig. 1, current considerations for future data modeling language in the scope of network management learned from the SMIng project can be embodied by our proposed evaluation framework in the form of criteria or just one or more facets of them.

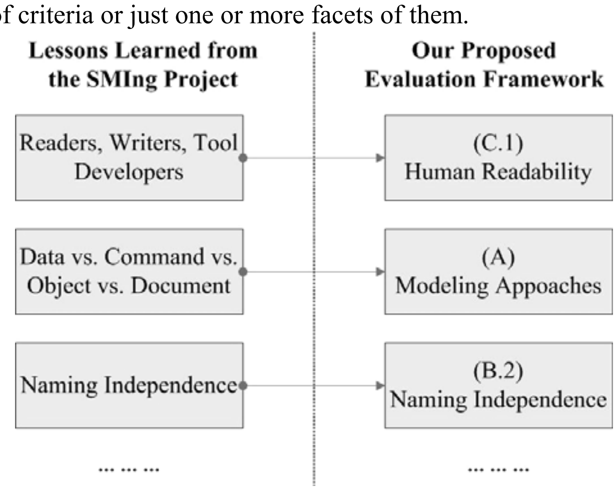

Figure 1. An example for argumentating the usability of our proposed evaluation framework

V. CONCLUSION

This paper establishes an evaluation framework for management data modeling languages and validates the proposed framework 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.

ACKNOWLEDGMENT

The authors would like to thank Wuhan BeiDa JadeBird NetSoft Inc., P. R. China, for its technical supports that benefited this paper.

REFERENCES

[1] A. Gupta, "Network management: current trends and future perspectives", *Journal of Network and Systems Management*, vol. 14, December 2006, pp. 483-491.

[2] J. Schoenwaelder, "Overview of the 2002 IAB network management workshop", RFC 3535, May 2003.

[3] J. E. López de Vergara, V. A. Villagrá, J. I. Asensio and J. Berrocal, "Ontologies: giving semantics to network management models", *IEEE Network*, vol. 17, May-June 2003, pp. 15-21.

[4] J. E. López de Vergara, V. A. Villagrá and J. Berrocal, "Applying the Web Ontology Language to management information definitions", *IEEE Communications Magazine*, vol. 42, July 2004, pp. 68-74.

[5] J. Schoenwaelder, "Protocol independent network management data modeling languages -- lessons learned from the SMIng project", draft-schoenw-sming-lessons-01.txt, September 2007, work in progress.