Proc. VLDB 2001 Workshop on Databases in Telecommunications (DBTel 2001),
Rome, Italy, September 10, 2001

Toward Universal Information Models in Enterprise Management

Jean-Philippe Martin-Flatin

AT&T Labs Research, 180 Park Avenue, Bldg. 103, Rm. A001,
Florham Park, NJ 07932, USA
jp.martin-flatin@ieee.org

Abstract. The DMTF's recent work on management information modeling in the IP world has highlighted that a number of problems are still unsolved in this important area of enterprise management. In this paper, we identify five: finding the right level of abstraction, building on past experience, devising good models, finding a good trade-off between quality and timeliness of new models, and attracting the best experts in the field in standardization efforts. We propose to alleviate them by splitting information modeling into two phases that involve different people with different skills. In the first phase, designers and experts in a given technology (be it a router, a service, a policy, etc.) capture the core issues for managing it in a Universal Information Model (UIM) that is independent of any management architecture. At this stage, low-level engineering details are ignored. In the second phase, code-oriented engineers instantiate the UIM into a data model specific to a management architecture (e.g., an SNMP MIB or a CIM schema). These people are specialists of SNMP or WBEM, but are not necessarily experts in the technology being managed.
Keywords: Information Model, Data Model, Internet, Management.

1 Introduction

In the Internet Protocol (IP) world, since the early 1990s, network management has been dominated by the Internet Engineering Task Force (IETF)'s management architecture, named after its communication protocol: the Simple Network Management Protocol (SNMP) [12]. In the meantime, systems and application management have mostly relied on proprietary solutions. Service and policy-based management are still in their infancy: standards are still being defined, and are thus not yet widely supported by deployed equipment. To date, integrated enterprise management is still wishful thinking.

This situation may change in the near future. The Distributed Management Task Force (DMTF) is currently working on a new management architecture: Web-Based Enterprise Management (WBEM). This alternative to SNMP management encompasses the entire realm of enterprise management: network element management, network management, systems management, service management, application management, policy-based management, etc. The main strengths of WBEM over SNMP are its object-oriented information model—the Common Information Model (CIM) [2]—, the large scope of management areas that it attempts to model (as opposed to SNMP's focus on network element management), and its interest in low-level, machine-oriented management abstractions as well as high-level, people-oriented abstractions—unlike SNMP, which is characterized by instrumentation Management Information Bases (MIBs). These qualities, plus others that are not all of technical nature (e.g., the fact that it is backed by most of the major vendors in the IT and networking industries), make WBEM a serious contender of SNMP for this decade.

So far, most of the DMTF's work has focused on information modeling—namely, the definition of the CIM Core and Common Models [2]. This sudden rash of activity in an area that had remained fairly quiet for several years has unveiled a number of problems for management-application designers and domain-specific modelers. We identified five. Interestingly enough, none of them are specific to WBEM or CIM.

First, finding the right level of abstraction for an information model is not an easy task. It is quite difficult to devise a model that is neither cluttered with low-level engineering details, nor overly generic and abstract.

Second, over the years, the open management community has not built a reputation of being immune to the *reinvent the wheel* antipattern—a pervasive disease in software engineering. In the recent past, the DMTF has produced intense efforts in information modeling; but some of its work was a waste of time, for it was duplicating previous efforts by the IETF in the same field. Similarly, throughout the 1990s, the two main standardization efforts in management—Open Systems Interconnection (OSI) management and SNMP—have, to a large extent, followed parallel tracks with few constructive interactions.

Third, a number of information models are not good enough. Some mistakes are minor, but others may be time bombs. For deployment reasons, correcting a bogus or incomplete model takes a lot of time—so much so that in practice, the market is generally stuck for years with whatever information models have been standardized, be they good or bad. Standards bodies should therefore pay better attention to devise good models in the first place. The main causes of this problem are delineated in the next two problems.

Fourth, in the design of management information models, "fast is not beautiful", so to say. Some information models leave a lot to be desired because they were moved too quickly through the standardization process, in order for vendors to swiftly put their software products on the market.

Fifth, researchers, especially from academia, have had little impact on the standardization of management information models in the IP world. Instead, the IETF and DMTF models have traditionally been devised by the engineers in charge of designing and coding management applications. (There are only few exceptions to this rule.) More generally, standards bodies have often proved unable to attract the best technology experts in the field. This has contributed to the flaws and limitations exhibited by some information models.

In this article, we propose to alleviate these five problems by splitting information-modeling efforts into two phases, which involve different people with different skills and different goals. In the first phase, information modelers, versed into abstraction and architectural aspects, produce a Universal Information Model (UIM) that is independent of any management architecture. These experts in a specific technology—e.g., in IP routers, differentiated services, Quality-of-Service (QoS) provisioning, Service-Level Agreements (SLAs), or IP telephony—capture the core issues for managing this technology, and ignore low-level engineering details. In the second phase, engineers versed into programming instantiate the UIM into a data model that is specific to a management architecture—e.g., an SNMP MIB specified in SMI (Structure of Management Information [8]), or a CIM schema expressed in MOF (Managed Object Format [3]). These people are specialists of a management architecture (e.g., SNMP or WBEM), but are not necessarily experts in the technology being managed.

We believe that two-tier information modeling should not be specific to the IP world, and would equally improve the quality of management information models outside the IETF and DMTF realms (e.g., for fixed and cellular telephone networks relying on OSI management). But for the sake of clarity, the scope of this article is limited to enterprise management in the IP world.

The remainder of this paper is organized as follows. First, we detail the problem statement. Next, we shed some light on our vision of two-tier information modeling. We then present a research agenda for UIMs and investigate related work. Finally, we summarize this article and present directions for future work.

2 Five Problems in Management Information Modeling

Let us now dive into the problems outlined in the introduction and show their interdependencies. Readers who are not familiar with the way management information is currently modeled by the IETF and DMTF are referred to [6]. In particular, this technical report summarizes the languages used to express data models (MOF for WBEM, SMI for SNMP, and SPPI for policies), the communication protocols used to transfer messages (SNMP, HTTP for WBEM, and COPS for policies), and the languages used for representing and encoding management data in these messages (BER for SNMP and XML for WBEM).

Problem 1: Between overly generic and cluttered models

Finding the right level of abstraction for management information models is a difficult exercise. There are many options. At one end of the spectrum, some theoreticians produce overly complex models. By trying to be too generic, they define extremely abstract concepts that are not very useful to the market. An example of this is the Object Management Group (OMG)'s four-tier meta-model architecture [9]: It started as a much needed solution to a real problem, but it is so complex that few people really understand how to use it efficiently in practice.

At the other end of the spectrum, we find developers that are only concerned with programming. They clutter information models with so many details that the big picture of the models is blurred, and sometimes even hidden. This approach is epitomized by SNMP MIBs: their design often looks cryptic to information modelers. We claim that one cannot understand the convoluted design of some MIBs without a full understanding of the limitations of SMIv2, the language used to specify them. For instance, the definition of multi-dimensional tables in a MIB is severely constrained by the notion of *conceptual tables* in SMIv2 [11] and the lack of support for nested tables in this language. To give another example, SMIv2 does not support the remote invocation of actions (*à la* OSI) or methods (*à la* Java RMI), but relies instead on an ugly kind of programming by side effect, whereby setting an integer to different values allows a manager to trigger different actions on an agent [5]. In short, information models in the SNMP world are often weirdly structured—at least to the eyes of non-SNMP information modelers—because of the limitations in the language used to express them.

Note that cluttered models are not specific to SNMP. In WBEM, the language used to express information models (MOF) is considerably richer than SMIv2, but this does not prevent CIM models from being cluttered with low-level programming details. An example of this was a recent discussion of the DMTF Events WG on whether a new property (i.e., an attribute in standard object-oriented parlance) called `AdditionalText` should be added to CIM events, as some organizations felt that the `Description` property, currently used to describe events, was insufficient in some cases [10]. Such details are very relevant to application developers, and completely irrelevant to information modelers. Another example is the Unified Modeling Language (UML [9]) diagrams that the DMTF uses to depict its information models. Although these diagrams can be useful to get the big picture of a model, they are often very detailed and fairly difficult to understand for a person with no or little knowledge of CIM. In particular, experienced information modelers with an SNMP or OSI background sometimes find it difficult to map their knowledge onto the DMTF's information models, especially when a well-established terminology is changed (see Problem 2).

Problem 2: The *reinvent the wheel* antipattern

In 1998, the DMTF moved from desktop management to enterprise management. Since then, its information-modeling activities have been thriving. By and large, this work has been, and is still, performed independent of the IETF's. Few people belong to both DMTF and IETF WGs. As a result, cross-pollination between these two communities is rare, and the know-how accumulated by the IETF in the 1990s is often ignored by the DMTF[1]. We believe that this is a typical occurrence of the *reinvent the wheel* antipattern [1].

[1] There are a few notable exceptions, e.g. in policy-based management. But the situation is even worse if we consider the lack of cross-pollination with other standards bodies that

Clearly, this is counterproductive, as the DMTF would be better off working on new issues, or issues ignored so far by the IETF. Worse, it also enables old problems to resurface, and requires new engineers to go through the same long and painful learning process that their SNMP elders went through a few years ago. A third problem is that the terminology keeps changing, as each new WG starts from scratch and redefines well-established concepts (e.g., see the new meaning given to the terms *event* and *notification*, and the intrusion of the concept of *indication* in the CIM 2.5 event model [4]). These terminological changes often cause confusion, especially when SNMP- and CIM-based management systems co-exist in the same Network Operations Center (NOC).

Problem 3: Some models are not good enough

The third problem is that some information models leave a lot to be desired. An example of this is the Address Translation Group, defined in MIB-I (RFC 1156) and deprecated shortly afterward in MIB-II (RFC 1213), because a major flaw had been discovered during the deployment phase (see RFC 1213, Section 3.6). Another example is the lack of per-interface access control lists in MIB-II, which has led many vendors to define their own in proprietary MIBs—or even worse, to demand that monitoring staff use the Command Line Interface (CLI) via `telnet`. A third example is the CIM event model, currently under revision, which promotes a confused idea of the severity of an event. The severity normally indicates whether an event is perceived to be critical, serious, or simply informative by the sending party. It gives a hint to the receiving party whether this event should be processed urgently. Indeed, the DMTF defines the severity levels `Information`, `Warning`, `Minor`, `Major`, `Critical`, and `Fatal`. But the WG recently accepted that a new severity level called `Clear` should enable a sending party to inform the receiving party that a certain problem has disappeared [10], which defeats the purpose of an event's severity.

This list of examples is by no means exhaustive, and a number of other SNMP MIBs and CIM schemas have well-known flaws or limitations. The two main causes that we identified for Problem 3 are detailed in Problems 4 and 5.

Problem 4: Fast design vs. good design

In our view, the main source of shortcomings in management information models is that the IETF and DMTF standardize these models too quickly. One reason for this is that the WGs who define information models are mostly driven by vendors, and in

---

have a respected expertise in management, e.g. the TeleManagement Forum (TMF), the International Telecommunications Union—Telecommunications Sector (ITU-T), the Object Management Group (OMG), and the International Organization for Standardization (ISO).

this business, vendors traditionally value speed over quality. Another reason is an overreaction to the slow pace of OSI management standardization in the early days of open management.

Vendor-driven WGs value speed over quality: In the IT and networking industries, vendors operate in very competitive markets, and they work hard to be the first to support a new technology. Being first is good for their image, their profits, and their stocks—sometimes irrespective of the quality of their implementation. Because a number of customers do not buy a technology unless it can be managed, most vendors release a new technology only once its management software is ready. Needless to say, management information modelers must meet drastic time schedules, squeezed between the time the technology is ready and the time it can be deployed. To meet these deadlines, they spend less time designing an information model than implementing it: they jump too quickly to engineering details. To make things worse, they also have an intense sense of competition with the other companies present in the WGs in charge of standardization. These are two strong incentives for designing information models very fast, and trying to rush them through the standardization process demanded by some of their customers.

The life span of a management information model is usually much larger than the time it takes a WG to design it; a typical ratio is 1:10. Consequently, saving one month on the design of important building blocks of an information model makes no sense in the long term. This small saving can make modelers overlook important aspects of the model and require, several years later, changes in the model and re-deployment of management software in the installed based—a major endeavor that can be immensely expensive for vendors and customers alike.

Overreaction to OSI management: In the late 1980s and early 1990s, the ISO and ITU-T standardized very slowly in the management field: standardization cycles took four years. To a certain degree, the way management information is currently modeled by the IETF is a reaction to the slow pace of OSI management information modeling in those days. The management community has been durably marked by this phenomenon, and the IETF and DMTF WGs are still afraid of engaging in never-ending discussions, seeking an elusive consensus. So they keep all design-related discussions short—even when the management issues are tough and demand long discussions. We refer to this remanence effect as *overreaction to OSI management*.

The "religious wars" between OSI management and SNMP are long past. WBEM has nothing to do with these wars. It would be useful for the management community to rid of this conditioned reflex, which might have contributed in the past to the success of SNMP-based management, but now causes more problems than it solves.

Problem 5: The best technology experts are rarely involved in standardization

Another important cause of Problem 3 is that the best technology experts in the field rarely participate in the WGs in charge of management standardization (and this goes beyond the sole IETF and DMTF). We see two reasons for that.

The big picture is blurred: Experts generally stay away from standardization efforts in management information modeling because low-level details are of no interest to them. They are more interested in defining a sound backbone and getting the main classes and relationships right. To an information modeler, it can be frustrating to have his/her model severely limited by a modest information meta-model, or constrained by the language used to express the information model—this language being supposedly "demanded by the market". In this respect, SMIv2 can be an efficient repellant for information modelers that are not versed into SNMP management application programming. The situation can even become intolerable for academics when politics get in their way—and vendor-driven WGs are not always immune to politics…

Fast vs. smart design: Most researchers prefer to do things right than fast. This objective is incompatible with the fact that WGs are mostly driven by vendors, who favor speed over quality.

To conclude with the problems in information modeling, we emphasize that these five problems are *not* specific to the DMTF or IETF. For instance, some of them were not unheard of at the ITU-T, a decade ago, when the management and control of fixed telephone networks were undergoing standardization.

3 From One- To Two-Tier Information Models

In our view, these five problems share two root causes:

- With one-tier information models, we try to do too many things in one step, and require too many skills from the same people. Instead, we propose to adopt two-tier information models. These tiers are devised by different people with different skills.
- The management issues for a given technology are intrinsically independent of the architecture (SNMP or WBEM) used by a customer to manage this technology. Technology-specific information models should therefore contain an architecture-independent core.

Based on these two important remarks, we propose to change the way information modeling is currently performed in the IP world.

3.1 Two tiers: one UIM, several data models

Today, the IETF and DMTF go directly from a high-level description of the managed technology to very detailed SNMP MIBs or CIM schemas. Instead, we propose to split information modeling into two phases (see Fig. 1). First, for each technology to manage, we define a Universal Information Model that is independent of SNMP and WBEM. Second, from this UIM, different WGs derive different data models in the form of SNMP MIBs, PIBs, CIM schemas, etc.

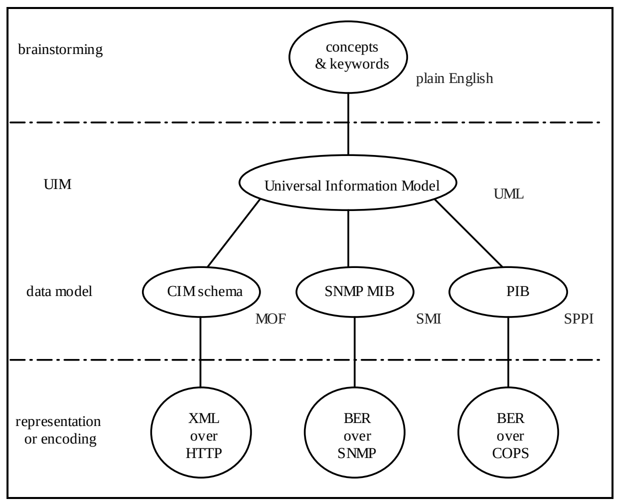

Fig. 1 Two-tier information model

Universal Information Models: Today, the management community needs good information models that management application developers, information modelers, administrators, and operators can easily understand, without having to master the sometimes cryptic notations of SMI, MOF, GDMO, etc. To achieve this, we propose to define a single Universal Information Model (UIM) for each technology to manage. This makes up the first tier of our two-tier information model (see Fig. 1).

A UIM is an object-oriented abstract model for managing a technology. It consists of object-oriented classes and relationships organized in UML diagrams. The management concepts modeled by these diagrams are identified by WGs including some of the best worldwide experts in this technology.

UIMs convey the big picture of the models to people; they ignore the details. They are not meant for machines or compilers, but for people who are familiar with the technologies being managed, e.g. information modelers and administrators. This level of abstraction is independent of the idiosyncrasies of any specific management architecture. In particular, UIMs are not constrained by the rather limited SMIv2 language.

We propose to adopt the UML meta-model [9] for expressing UIMs because it is standard and object oriented. It is also fairly close to the DMTF meta-model.

UIM standardization should be driven by joint IETF and DMTF WGs, and may also involve people from the TMF, the OMG, the W3C, etc. These WGs should not be driven solely by vendors. Academic researchers, who are less prone to having vested interests in standardization activities than people from industry, should get involved too.

Data Models: The second tier of our information model consists of data models derived from the UIM (see Fig. 1). In SNMP, data models are usually SNMP MIBs (or PIBs in the case of policies). In WBEM, they are CIM schemas. A data model is a level of abstraction well suited to developers of management applications. It is specific to a given management architecture and contains many details that should be hidden to the end-users of these applications—especially administrators and operators in NOCs.

Data models directly reflect the constraints of the language used to express them. For instance, the expressiveness of MOF is greater than SMIv2's, so CIM schemas can be more expressive than SNMP MIBs. Another example is that data models need not necessarily be object oriented, although UIMs are. Note that our proposal does not prescribe the language that should be used for defining a data model: this is left to the management architecture.

The main actors for defining data models should be vendors developing real-life management applications (managers or agents).

3.2 Advantages of using two-tier models

Our proposal alleviates the five problems described earlier.

Between overly generic and cluttered models: By adding the UIM-design phase between the brainstorming and data-modeling phases, we make sure that information modelers are not hampered by low-level details when they devise a new model. This decreases the risks of making a bad design mistake in the information model, and of having to update it once it is already deployed.

The* reinvent the wheel *antipattern: We make it possible to solve Problem 2 by devising good, solid UIMs that are independent of any specific management architecture and can live for many years, without undergoing major upheavals. When a new management architecture is defined, only data models need to be defined. By doing this, we help build on past experience and help prevent old problems that have already been solved from resurfacing in the future. We also prevent customers from being confused by constant changes in the terminology.

Some models are not good enough: We alleviate this problem by addressing the next two.

Fast design vs. good design: We solve Problem 4 by having vendors compete on the fast implementation of data models rather than on the definition of UIMs. Mapping a UIM onto a data model should not take much time, especially as the DMTF and IETF WGs develop some know-how in doing so.

The best technology experts are rarely involved in standardization: We offer a solution to Problem 5 by making standardization efforts a lot more attractive to experts, especially academic researchers. With UIMs, they are no longer drown by low-level details and can focus on getting their models right reasonably fast. And by bringing more experts into standardization, we increase the chances of devising good models in the first place, and decrease the risks of having to update an already-deployed model.

4 A Research Agenda for UIMs

For UIMs, the first item on the research agenda is probably to systematically formalize existing SNMP MIBs in the form of UIMs, to abstract UIMs out of the existing DMTF UML diagrams (most notably, by making them CIM independent), and to attempt to merge these pairs of UIMs, technology by technology. The primary outcome of this work would be to make it possible for information modelers not interested in the idiosyncrasies of SNMP and CIM to study, and possibly improve, these UIMs, and to document different, incompatible approaches between different management architectures. Another outcome would be to highlight the areas where the DMTF's CIM schemas are lagging behind the IETF's SNMP MIBs, and those where the DMTF expands on existing SNMP MIBs. These two outcomes would be particularly useful to customers, as they would enable them to compare different approaches to a single problem, and to make an educated guess when selecting SNMP or CIM for deploying a new management solution.

It would also be interesting to assess whether deriving several data models from a single UIM eventually facilitates the translation between these data models. Past work has shown the difficulty to translate managed objects expressed in SMI, GDMO, and IDL, and the semantic losses that can occur [7]. Do shared UIMs alleviate these problems?

Another point worth investigating is whether UIMs need the equivalent of the CIM Core Model in order to share some building blocks. Should derived types such as Ipv4Address, Ipv6Address, Date, and TimeSeries be defined once and for all, and shared by all UIMs?

5 Related Work

Our proposal can be viewed as a generalization of a mechanism used by the DMTF WGs. The main differences are fourfold:

– These WGs generally spend much more time specifying the nitty-gritty of the data models than putting together smart UML diagrams. Some WGs only occasionally update their UML diagrams. In our approach, the definition of the UIM is a goal *per se* that usually takes more time than deriving the data models.

– Usually, at the DMTF, the same people are in charge of defining the UML diagrams and the data models, which requires a mix of skills rarely found in practice. In our approach, these two teams are different.
– The UML diagrams that are produced by the DMTF WGs are completely CIM specific. In two-tier information modeling, the UIM is independent of CIM.
– Finally, the DMTF is using UML diagrams only for improving the quality of the models. We define UIMs also for sharing them with other standards bodies such as the IETF.

6 Conclusion

We have exposed five problems associated with information modeling in the IP world, and proposed to alleviate them by splitting information modeling into two phases. In the first phase, designers and technology experts define a Universal Information Model (UIM) that is independent of any specific management architecture. A UIM focuses on the big picture of the management issues for a given technology. It is expressed in UML. In the second phase, different people derive multiple data models from a single UIM. In SNMP, these data models are MIBs written in SMI, or PIBs written in SPPI; in WBEM, they are CIM schemas written in MOF. For a given technology, a UIM and its associated data models constitute what we call a *two-tier information model*.

Some work is currently under way at AT&T Labs Research to define UIMs for policy-based management and IP-router management. In particular, we have begun reverse-engineering and merging the existing data models of the IETF and DMTF.

Acknowledgments

Several ideas presented here were debated at the 8th meeting of the Network Management Research Group (NMRG) of the Internet Research Task Force (IRTF). The clarity of this paper benefited greatly from these discussions. The author would also like to thank C. Kalmanek for his useful comments on this article.

References

1. W.J. Brown, R.C. Malveau, H.W. McCormick III, T.J. Mowbray. *Antipatterns: Refactoring Software, Architectures, and Projects in Crisis*. Wiley, New York, NY, USA, 1998.
2. W. Bumpus, J.W. Sweitzer, *et al*. Common Information Model: Implementing the Object Model for Enterprise Management. Wiley, New York, NY, USA, 2000.
3. DMTF. Common Information Model (CIM) Specification. Version 2.2. June 1999.

4. DMTF Events Working Group. *Common Information Model (CIM) Indications (Final Draft). Version 2.5 Final Draft*. DMTF, December 14, 2000. Available at: http://www.dmtf.org/var/release/Whitepapers/DSP0107.htm
5. J.P. Martin-Flatin. *Web-Based Management of IP Networks and Systems*. Wiley, Chichester, UK. To appear in 2001.
6. J.P. Martin-Flatin. *Toward Universal Information Models in Enterprise Management*. Technical report TD-4XURBN, AT&T Labs Research, February 2001.
7. S. Mazumdar. "Inter-Domain Management Between CORBA and SNMP". In Proc. 7th IFIP/IEEE International Workshop on Distributed Systems: Operations & Management (DSOM'96), L'Aquila, Italy, October 1996.
8. K. McCloghrie, D. Perkins, and J. Schoenwaelder (Eds.). *RFC 2578. Structure of Management Information Version 2 (SMIv2)*. IETF, April 1999.
9. OMG. OMG Unified Modeling Language Specification. Version 1.3. OMG, March 2000.
10. C. Shaw (Ed.). Minutes of the DMTF Events Working Group meeting of Jan 11, 2001.
11. R. Sprenkels and J.P. Martin-Flatin. "Bulk Transfers of MIB Data". *The Simple Times*, 7(1):1-7, 1999.
12. W. Stallings. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. 3rd edition. Addison-Wesley, Reading, MA, USA, 1999.

Biography

J.P. Martin-Flatin is a principal technical staff member with AT&T Labs Research. He holds a Ph.D. degree in CS from the Swiss Fed. Inst. of Technology, Lausanne (EPFL). He has worked 9 years in industry in the areas of network and systems management, software development, security, and Web engineering. He is the author of a book, *Web-Based Management of IP Networks and Systems*, to appear in 2001, and was a guest editor for a special issue of the *Journal of Network and Systems Management* on the same topic. The bottom line of his research is to build an interdisciplinary bridge between enterprise management and software engineering. He is a member of the IEEE, ACM, IRTF Network Management Research Group, and DMTF WBEM Interop and SysDev Working Groups.