Iterative Multi-Tier Management Information Modeling

Jean-Philippe Martin-Flatin, CERN
Divesh Srivastava, AT&T Labs Research
Andrea Westerinen, Cisco Systems

ABSTRACT

The management information models currently used in the Internet have several limitations. Some of them contain errors, are missing important features, or are difficult to understand. Second, standards bodies keep reinventing the wheel, which confuses the terminology (hence customers) and wastes precious time. Third, finding a good balance between too abstract and overly detailed models is a tough challenge, rarely achieved in practice. Last, the learning curve of existing data models is too steep. We propose to alleviate these problems by adopting a new process for designing and standardizing management information models. It is inspired by two techniques from software engineering: the iterative and incremental software development process, which addresses the shortcomings of the waterfall process usually adhered to by the IETF and DMTF; and multi-tier models, which capture different perspectives (e.g., analysis, design, and implementation) of the information model. Our main innovations are management-architecture-neutral universal information models (UIMs), sharing of conceptual models by different standards bodies, and specialization of the people involved in designing the different layers of the models. Our new process takes into account a number of constraints identified in real-life environments.

INTRODUCTION

After more than a decade of large-scale deployment, element management in the Internet is generally considered a success. However, the recent challenge of integrating service management with network and systems management (which leads to what is known as *integrated management*) has shown that management information modeling, one of its building blocks, is still faced with a number of problems.

First, some management information models (MIMs) are simply not good enough. They are often incomplete, sometimes even incorrect, because of a poor trade-off between quality and timeliness, because of fuzzy and changing requirements ("What dials and knobs do customers need/want?"), and because standardization efforts in management often fail to attract the best technology experts in the world.

Second, Working Groups (WGs) in standards organizations keep reinventing the wheel by redefining things their way. Cross-fertilization is often limited between the multitude of organizations involved in the specification and standardization of integrated management:

- Internet Engineering Task Force (IETF)
- Distributed Management Task Force (DMTF)
- International Telecommunication Union (ITU)
- International Organization for Standardization (ISO)
- Open Management Group (OMG)
- The Open Group (TOG)
- TeleManagement Forum (TMF)
- Internet Research Task Force (IRTF)
- Global Grid Forum (GGF)
- And others

This causes a certain degree of confusion in the terminology, hence in the market; and all the duplicate work makes standardization efforts waste precious time.

Third, finding a good balance between an overly abstract model and a model cluttered with low-level engineering details is renowned as a tough challenge, rarely achieved in practice, not only in integrated management, but also in virtually all complex application domains.

Fourth, the learning curve of existing data models is too steep. Newcomers to enterprise management need a better way of understanding the core concepts for managing a given technology.

We propose to alleviate these problems by changing significantly the way MIMs are devised and standardized. Our new modeling

0163-6804/03/$17.00 © 2003 IEEE

process is very general and may serve as the basis for a wide range of management information modeling activities. In this article we restrict our scope to the Internet Protocol (IP) world. We thus propose to change the modeling habits of the two main actors in this world: the IETF, which promotes a management architecture named after its communication protocol, the Simple Network Management Protocol (SNMP) [1]; and the DMTF, which advocates the Web-Based Enterprise Management (WBEM) architecture and its main component, the Common Information Model (CIM) [2]. Other standards bodies may also wish to adopt our proposal.

The core innovations of our information modeling process are twofold. Instead of directly working on competing data models, such as an SNMP management information base (MIB) and a CIM schema, for managing a given technology, the IETF and DMTF WGs should initially join forces to define a single universal information model (UIM), which focuses on the core management issues for that technology. By definition, a UIM is a conceptual model that is independent of any management architecture. Once the UIM is defined, the WGs can split and derive different data models, thereby ensuring that the same semantics and terminology are used. This approach is called *multi-tier information modeling*. By specializing the people who design the different layers of the models, we improve the quality of the models and reduce the risk of missing important aspects for managing a given technology. Our second innovation is to adopt an iterative and incremental software development process for management information modeling. This technique, now fairly common in object-oriented software engineering, addresses several shortcomings of the waterfall process currently used by the IETF and DMTF.

The remainder of this article is organized as follows. First, we summarize how management information modeling is currently performed in the IP world. Then we review the problems encountered in management information modeling. Next, we describe multi-tier MIMs and our iterative modeling process. Finally, we verify that all the issues have been addressed.

Management Information Modeling in the IP World

Although MIMs do not follow the same life cycle at the IETF and DMTF, the processes for defining them are fairly consistent.

Per-Technology Standardization Activities

When a WG sets out to standardize the management of a given technology (e.g., a router, host, database, or service), it begins with a brainstorming phase (Fig. 1). The purpose is to define the concepts, abstractions, and keywords that best capture the core characteristics of the technology. The main outcome of this work is a terminology shared by all the WG members and, later, by the entire market. At the IETF, this terminology is usually defined in the mailing list of the WG. At the DMTF, a similar approach is followed, and the terminology is not captured until the models are formalized in UML class diagrams or in Managed Object Format (MOF, a form of structured text).

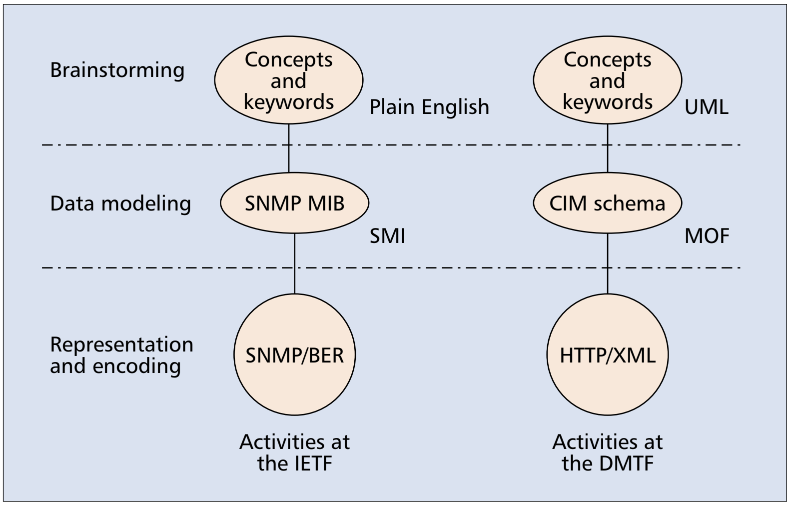


Figure 1. *Current information modeling in the IP world.*

The next phase is the definition and formalization of the MIM per se. At the IETF, the WG in charge of managing a certain technology defines a MIB[1] [1]. At the DMTF, the corresponding WG defines a CIM schema, that is, an object-oriented model (in MOF and UML) subclassing from CIM's core model [2]. In both organizations, the information model consists mainly of a model full of low-level engineering details that frequently hide the big picture of the MIM.[2]

Technology-Independent Standardization Activities

Some standardization activities are independent of the technologies being managed and take place only once. They consist in defining the meta-model shared by all technology-specific models, the language used to express these models, and the way to represent and encode management data in messages exchanged by the communication protocol.[3]

Meta-model: The meta-model defines the basic modeling concepts. In the SNMP management architecture, there is no explicit meta-model. The implicit one is that everything in a MIB is an object identifier (OID), a managed object bearing a name and a value. There is no clean concept of relationships in SNMP, but there are unofficial ways to capture them [6]. Conversely, WBEM encompasses an object-oriented meta-model derived from the UML meta-model. It defines the concepts of class, object, association, qualifier, and so on [2].

Language: In SNMP, MIBs are defined in a language called Structured Management Information (SMI) [7]. This variant of ASN.1 is not object-oriented. In WBEM, CIM

[1] *In the domain of policies, the IETF uses policy information bases (PIBs), which are expressed in a different language. Since most of the IETF's current data models are MIBs, we will ignore PIBs in this article.*

[2] *Our definition of a data model is slightly different from that found in RFC 3198 [3], where the focus is on mapping an information model to a specific data repository. For a detailed discussion on the difference between information and data models, see [4].*

[3] *For more details on these concepts, see [5].*

[4] *To us, representation specifies whether the value of an attribute or a remote method invocation is expressed in XML, as a string, and so on. Encoding indicates whether the data, once represented in a specific format, is further compressed, encrypted, or simply transmitted "as is." In the IP world, there is much confusion between these terms, and many people use them interchangeably.*

Devising a model that is neither too generic nor too detailed is renowned to be a tough problem in software engineering. Integrated management is no exception.

schemas are specified in MOF [8]. This object-oriented language is richer and more expressive than SMI.

Representation and Encoding:[4] In the SNMP realm, management data is represented in SMI, encoded with basic encoding rules (BERs), and transferred via the SNMP protocol. In the WBEM world, CIM attributes and remote method invocations on CIM objects are represented in Extensible Markup Language (XML), encoded with one of the MIME encodings supported by HyperText Transfer Protocol (HTTP), and transferred via HTTP. The DMTF also advocates other representations and encodings, as evidenced by directory-enabled networks (DENs).

DESCRIPTION OF THE PROBLEMS

Let us now dive into the problems outlined in the introduction and show their interdependencies.

PROBLEM 1: SOME MODELS ARE NOT GOOD ENOUGH

The quality of some MIMs leaves a lot to be desired. One problem is that some important management aspects are missed. For instance, people working in 24 × 7 network operation centers (NOCs) frequently complain that the commands available through a vendor's command line interface (CLI) do not always have a counterpart in an SNMP MIB. Another grievance is that many MIBs contain few, if any, writable OIDs: they are designed to be used in read-only mode. As a result, it is necessary to use `telnet`, the most basic form of management, to perform certain tasks. Lastly, some models are poorly designed. For instance, the Address Translation Group was defined in MIB-I (RFC 1156) and deprecated shortly afterward in MIB-II (RFC 1213) because of a serious design flaw. Some CIM schemas also reveal design problems [9]. So why do we have poor or incomplete models?

Moving Fast in a Competitive Market — WGs are mostly driven by vendors, whose customers demand that a technology be managed before they consider purchasing it. Therefore, vendors (or their contractors) rush to put together management software for their new technology. But in many cases, management is not essential to them: it is just a market constraint required to sell their products.

As a result, management information modelers must meet drastic time schedules, squeezed between the time when the technology is complete and when it can be sold to customers with management software. Some of these modelers also have little technical expertise in the problem domain, or ignore modern software engineering techniques. Under these circumstances, it comes as no surprise that they spend less time designing a MIM than implementing it, and jump too quickly to engineering details!

Standardization WGs Fail to Attract the Best Technology Experts in the Field — Another source of limitations in MIMs is that standardization WGs often fail to attract the best experts in the technologies to be managed. As we once heard, standardization efforts are perceived as "soul destroying" by these people — and by management experts, too. They stay away from MIM standardization because WGs spend most of their time arguing about low-level engineering details. Experts would rather design a sound backbone (the core managed objects and relationships) and let others sort out engineering details. Moreover, to a designer, it is irritating to see MIMs reduced to data models; for example, the poor expressiveness of a modest language such as SMIv2 [7] is not appealing to model the management of a technology.

PROBLEM 2: THE REINVENT THE WHEEL ANTIPATTERN

In 1998, the DMTF decided to enlarge the scope of its work from desktop to enterprise management. Since then, its standardization efforts have gained much momentum. They now embrace a larger scope than the IETF's management activities. Unfortunately, the DMTF's information modeling work has, to a large extent,[5] remained separate from the IETF's, in part due to the operational policies of the IETF and in part due to the *reinvent the wheel* antipattern, also known as the *not invented here* syndrome. Few people belong to WGs in both management communities, and not many have migrated from one community to the other. As a result, cross-fertilization between the IETF and DMTF is rare, although they both address enterprise management in the IP world.

PROBLEM 3: FINDING THE RIGHT LEVEL OF ABSTRACTION

Devising a model that is neither too generic nor too detailed is renowned to be a tough problem in software engineering. Integrated management is no exception.

Since its early days, management information modeling has swung between two extremes. At one end of the spectrum, some theoreticians produce overly complex models that define very abstract concepts and try to be too generic. Most of the time, such models end up being shelved and ignored by the market, even if, hidden within the model, lies a smart solution to a real problem. An example is the OMG's four-tier meta-model architecture [10], which few people fully understand.

At the other end of the spectrum, some developers, concerned only with implementation, clutter MIMs with engineering details that hide the big picture of the model and make it easy to miss important aspects. An example of this was a discussion of the DMTF Events WG on whether a new property[6] called `AdditionalText` should be added to CIM events to complement the existing `Description` property [11]. This is exactly the level of detail an application developer wants to know and an information modeler wants to ignore.

In the middle of the spectrum, the goal is to make a trade-off between the granularity of the

[5] *There are exceptions. For example, in policy-based management, some cooperation occurs between the DMTF and IETF, but it takes place at the individual level rather than the standards bodies level. The DMTF has more officially partnered with other standards bodies in specific management areas such as storage management.*

[6] *A property in Java and DMTF parlance corresponds to an attribute or a state variable in standard object-oriented jargon.*

abstractions necessary for managing a given technology, and the number of classes, associations, and so on that will have to be kept up to date by all implementations of this technology worldwide. Striking a good balance is what makes management information modeling a difficult discipline. *"Everything should be made as simple as possible, but not simpler."*

Problem 4: The Learning Curve Is Too Steep

Another problem is that the learning curve for current data models (SNMP MIBs and CIM schemas) is too steep. First, data models are expressed in a language that is not immediately comprehensible. It takes a while to read SMI or MOF fluently. It is difficult for engineers newly assigned to a management project to discover a new application domain; but it is even worse if they get explanations in a language they hardly understand!

Second, the big picture of most existing data models is completely hidden by the details. It takes newcomers a long time to fight their way through these details. Even experienced information modelers with an SNMP/SMI, WBEM/CIM, or OSI/GDMO background can find it difficult to map their knowledge onto another's MIMs. The situation is aggravated when a well established terminology is changed: people misinterpret the semantics of a data model until they notice the terminological change, which can take some time.

Analysis of the Problems

With all these problems in mind, let us now summarize the rationale that led us to define a new process for devising MIMs.

Causes of These Problems

In the early days of SNMP, when management was mostly about testing the reachability of remote network devices and collecting traffic statistics per interface, information modeling could get by with simple ad hoc approaches such as those currently used by the IETF. But this is no longer adequate. Integrated management has made information modeling more complex. Dealing with this complexity calls for the use of more advanced methodologies.

We identify below four root causes behind all these problems. The good news is that they constitute well-known issues in software engineering, and we have well-known methods to solve or alleviate them.

First, with one-tier MIMs the IETF and DMTF try to do too many things at a time, and require too many skills from the same people. Instead, we propose to adopt multi-tier MIMs, with tiers devised by different people with different skills. This change adheres to a principle that the software engineering community has been advocating for years: the split between conceptual, specification, and implementation models. Conceptual models are more appropriate than data models to help newcomers grasp the core issues for managing a given technology.

Second, going from one architecture (e.g., SNMP) to another (e.g., WBEM) does not make the management issues any different for a given technology. By isolating the architecture-independent core (i.e., the invariant) from the architecture-specific part, we render the design cleaner, facilitate reuse, and decrease the risk of terminological chaos.

Third, leaving management information modeling only to vendors does not lead to a balanced design. The quality of MIMs would benefit from the participation of researchers (most notably academic researchers, with supposedly no partisan interests), independent consultants, end users, and independent employees of standards bodies. We follow here the principle that software quality is best assured by finding the right people to fulfill each task throughout the software development process.

Finally, in software engineering, it is well known that the waterfall approach (usually adopted by the IETF and DMTF) works fine in simple cases, but is not suitable for complex projects. As management issues become more complex, it is necessary to migrate to an iterative and incremental form of modeling [12, 13].

Consequently, we need to change the way that management information modeling is currently performed in the IP world.

Constraints from Real Life

To be effective in real life, our new process should take into account three constraints in integrated management.

Standards-based: Standards prevent the proliferation of noninteroperable and proprietary MIMs.

Timely: Vendors operate in very competitive markets. Any new modeling process must allow them to release new technologies with minimum delay, and allow customers to buy high-quality management software at the time they purchase hardware, software, or services (and not later).

High-quality and stable design: Redeploying a new version of a MIM in a large installed base poses major logistic problems and is costly for customers and vendors alike. When MIMs are designed, every effort should be made to avoid the need for redeployment.

Multi-Tier Models

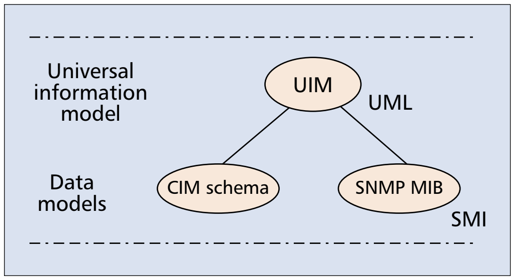
Instead of jumping directly from a high-level description of the technology being managed (brainstorming phase) to very detailed SNMP MIBs or CIM schemas (data modeling phase), we propose to split management information models (MIMs) into multiple tiers. Let us start by describing two-tier models; next, we will generalize to multi-tier models.

Instead of jumping directly from a high-level description of the technology being managed (brainstorming phase) to very detailed SNMP MIBs or CIM schemas (data modeling phase), we propose to split management information models into multiple tiers.

One UIM per Technology

The first tier of our two-tier MIM is the universal information model (Fig. 2). A UIM is an object-oriented abstract model for managing a technology. It is independent of any management architecture, and thus of any data repository or communication protocol. For a given technology, there is only one UIM.

UIMs leverage UML to organize graphically, into object-oriented classes and relationships, Whatever the experience of model designers, they will always get it wrong the first time a complext domain is modeled. In all but simple cases, they need to model once, learn from their mistakes, and then redo the model correctly.



■ **Figure 2.** *A two-tier MIM.*

the concepts the worldwide experts in this technology deem necessary for managing it. UIMs are primarily expressed in the form of class diagrams and whitepapers. If need be, they can also use other types of UML diagram (e.g., use cases and sequence diagrams).

UIMs focus on conveying the big picture in a clear and easy manner to people, not machines or compilers. They ignore low-level engineering details. Their high-level semantics make them well suited to the way people think. They can be easily understood by those who are familiar with the technologies being managed.

In the IP world, we recommend that UIM standardization be driven by joint IETF and DMTF WGs. Other standards bodies may get involved as the problem domain warrants. These joint WGs should not be driven solely by vendors. Academic researchers, end users, and independent consultants, who are less prone to having vested interests in standardization activities than people from industry, should also get involved.

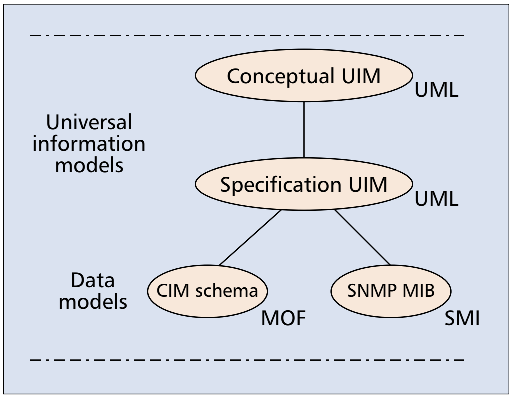
Multiple Data Models Per Technology

The second tier of our two-tier MIM consists of one or several data models derived from the UIM (Fig. 2). A data model is specific to a given management architecture. It is full of low-level engineering details and includes constructs that are specific to a data modeling language (e.g., `RowStatus` in SMI). Today's typical data models are SNMP MIBs and CIM schemas. We can also have directory schemas, relational database schemas, and so on.

Our proposal does not prescribe the language that should be used for defining a data model. This is left to the management architecture. In particular, data models need not be object-oriented.

Data modelers are specialists in the languages used to express the models. They need not know anything about the technology being managed. As a result, the same people can devise data models for all the technologies on the market. Indeed, only a handful of people really need to understand the intricacies of SMI, MOF, and the like in order to devise smart data models.

In our view, most of the data modelers should come from industry, especially from vendors developing management applications (managers or agents). They are more likely than technology experts, academics, or end users to test out the feasibility and suitably of a MIM in real life.



■ **Figure 3.** *A three-tier MIM.*

More Than Two Tiers

When management issues are complex, it can be useful to further split the UIM. For instance, object-oriented software engineers often produce three models: a conceptual model at the analysis phase, a specification model at the design phase, and an implementation model at the coding phase [13]. Some people further break up the design phase into high-level and low-level design.

Implementation details are specific to each data model, because of constraints from the management architecture and the language used to express the data model. We therefore expect implementation UIMs to be irrelevant to integrated management most of the time. Figure 3 depicts a common scenario: a three-tier MIM comprising a conceptual UIM (analysis), a specification UIM (high-level design), and two data models (low-level design). Implementation models are those used by vendors to support the MIM in their equipment or applications.

Iterative Process

We have just seen that multi-tier information modeling improves the chances of devising good-quality MIMs. But in the form presented so far, this technique presents three shortcomings. First, it increases the standardization time for MIMs, which clashes with an important constraint mentioned earlier. Vendors want to develop management software very quickly, because it is the last hurdle before they can sell a new technology to the market. So we must find a way for vendors to make money before the MIM is fully standardized.

The second problem is a fact of life in software engineering: Whatever the experience of model designers, they will always get it wrong the first time a complex domain is modeled — in our case, when they model the management of a new, complex technology. In all but simple cases, they need to model once, learn from their mistakes, and then redo the model correctly.

The third problem is that requirements can change over time as technologies evolve. MIMs must be able to evolve to incorporate these changes.

In the software engineering community, the usual solution to the last two problems is to go

[7] *The iterative development process, which promotes an incremental approach to design, can be viewed as a refinement of the spiral development process.*

from a waterfall (top-down) development process to an iterative process [12, 13], which addresses the scalability issues of large complex models by combining top-down and bottom-up approaches.[7] Fortunately, this solution also solves our first problem.

In its final form, our new process consists of several iterations. Only the first two are mandatory.

ITERATION 1: PROTOTYPING

The purpose of the first iteration is fourfold: identify the core management issues for a given technology, prototype a MIM, test out this MIM in the field, and get feedback from real users. Different actors are involved at different stages (Fig. 4).

First, a small team, gathering worldwide experts in a given technology, devises a simple conceptual model known as a lightweight UIM (LUIM). This LUIM consists of a small set of UML class diagrams, complemented by annotations in plain English and documentation such as use cases (e.g., in the form of a whitepaper). The purpose of this LUIM is to define a terminology, the core classes to manage this technology, and the main relationships between these classes. An annotation helps capture the rationale that led the WG to make a certain trade-off between technical alternatives. The characteristic of an LUIM is that it focuses on the bottom line and ignores the details.

Second, specialists of SMI and MOF derive data models from the LUIM. Third, vendors prototype their management software and put this new technology on the market as soon as possible. Fourth, vendors integrate one (or several) of the data models in real-life equipment or applications. Finally, beta-test customers volunteer to test out the new management software in the field, report bugs, and get software upgrades that are not shipped to ordinary customers.

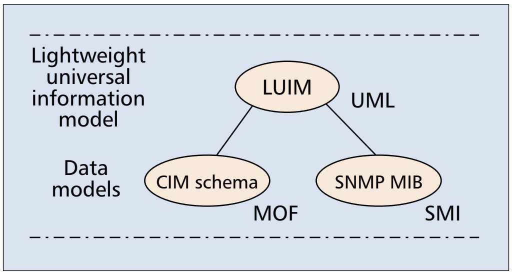


Figure 4. *Iteration 1: lightweight UIM.*

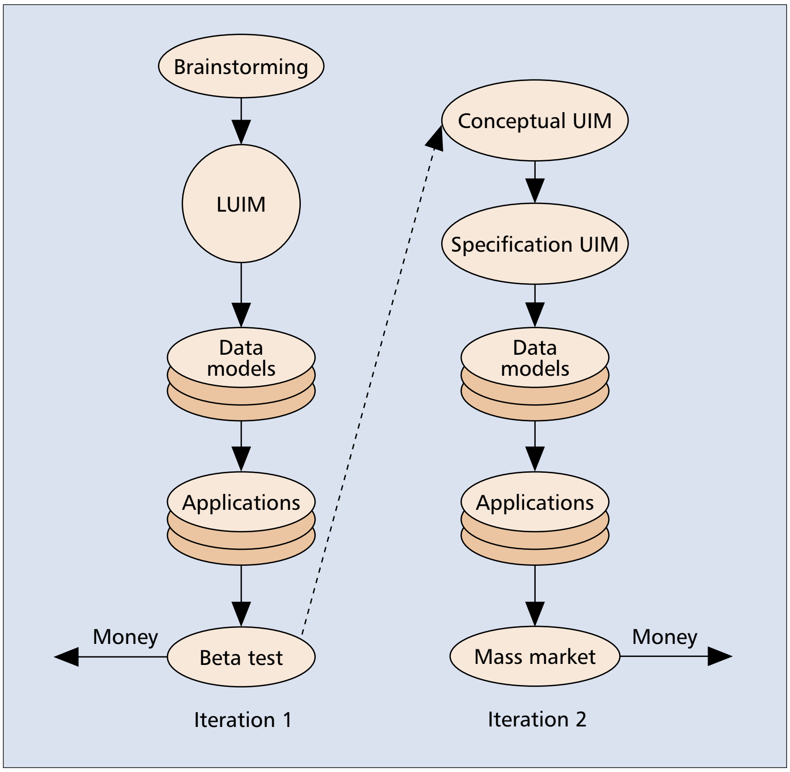


Figure 5. *Two iterations.*

ITERATION 2: REFINEMENT

In the second iteration, the objectives are to learn from the mistakes made in iteration 1 and to improve the MIM. It is important to make this MIM reach a professional grade, because the success of the real-life deployment of the new technology depends (to a certain extent) on the quality of its MIMs. The actors involved in this iteration remain hopefully the same as those in iteration 1.

First, based on the feedback from the developers and beta-testers involved in iteration 1, information modelers improve the LUIM into a full-blown UIM. If the management issues are complex, this model can be specified as a collection of UML diagrams (e.g., conceptual and specification class diagrams, possibly complemented by interaction diagrams [10, 13]) in a whitepaper. When the feedback from the field leads to significant changes in the conceptual model, it should be formalized in writing (e.g., in the form of annotations in the whitepaper).

Next, specialists of SMI or MOF map the UIM onto their respective data models. Then developers implement the required changes in the management software, and testers make a final check on the real-life equipment or software supporting the new technology. Finally, the new management software can be sold to a large customer base.

FURTHER ITERATIONS

In general, two iterations should be sufficient to devise and standardize MIMs. Occasionally, further iterations may be needed for maintenance and refinement.

Maintenance: Extra iterations are needed when the management issues change over time. This is the classic problem of changing requirements, another reason why the iterative software development process was invented [12].

Refinement: A second reason for adding an extra iteration is when a design flaw is discovered in a MIM after its large-scale deployment. As mentioned in the constraints section, this should be avoided whenever possible, but it is necessary to allow for such corrections.

In both cases, it is recommended that the rationale that led to changes in the MIM be fully documented.

By defining conceptual models, we make it much easier for newcomers to get started with the management of a given technology. Multi-tier models provide them with gradual levels of complexity.

TIME MANAGEMENT: A CONDITION FOR SUCCESS

Undeniably, our process is more complex than the current practice in the IP world. MIMs may therefore require more time to be standardized. Thus, an important condition for our new process to be efficient, and accepted by industry, is that time be managed at each step of the standardization process: at each tier of the MIM and at each iteration. Deadlines must be defined for putting together the LUIM and UIM of a specific technology, and the chairperson of each WG must ensure that these deadlines are met.

ADVANTAGES OF OUR NEW PROCESS

Our new approach to information modeling solves or alleviates all of the problems described earlier:

Some models are not good enough: We alleviate this problem in three ways:

- By devising multi-tier MIMs, step by step, instead of jumping directly to data models
- By adding a prototyping phase in iteration 1, to learn from experience gathered in the field
- By making standardization efforts more attractive to the best worldwide experts in the technologies, especially top-notch researchers and independent consultants

Reinventing the wheel: By having a single UIM to manage a given technology, and deriving multiple data models from this UIM, we encourage WGs not to reinvent the wheel. When a new meta-model, a new data-model language, or a new way of representing data is invented, only data models need to be defined. All existing UIMs can be leveraged immediately, without any change, which is a tremendous saving. We build on past experience, help prevent old problems from resurfacing, and ensure that the terminology remains consistent over time.

Right level of abstraction: By adding one or several tiers to the MIM, we allow information modelers to capture different things. LUIMs and conceptual UIMs focus on the bottom line, while data models concentrate on the engineering details. If need be, intermediary models such as specification UIMs can help clarify complex management issues.

Steep learning curve: By defining conceptual models, we make it much easier for newcomers to get started with the management of a given technology. Multi-tier models provide them with gradual levels of complexity, and allow them to put things in perspective when they eventually begin working on data models. This also has a positive impact on the quality of the data models devised by these people. Documenting changes in the MIM helps newcomers familiarize themselves with the management idiosyncrasies of a given technology.

Iterative multi-tier management information modeling satisfies the three constraints mentioned earlier. It also brings new benefits:

- If the technology itself changes after the LUIM is devised in iteration 1, we still have a chance to adapt the UIM accordingly in iteration 2, before large-scale deployment.
- Having UIMs shared by the IETF and DMTF helps vendors cut their software development costs when they support both an SNMP MIB and a CIM schema for managing a given technology.
- By imposing strict time management, we put an upper bound on the time-to-market for the first iteration. For marketing, this is an important factor.

CONCLUSION

We have described four problems associated with management information modeling in the IP world, and proposed to alleviate them by adopting a new process. This process combines multi-tier management information models with iterative software development. For customers, it ensures that MIMs are better designed, and that the terminology for managing a given technology remains consistent over time and throughout the IP world. For vendors, it decreases their software development costs and preserves their ability to sell a new technology at a very early stage of management standardization. For both vendors and customers, our new process reduces the risks of having to redeploy a MIM to correct a design flaw or to support a feature that was initially missed.

A number of management information modeling issues are worth investigating in the future. What happens when we have not one but several UIMs for managing a given technology? How do we integrate UIMs that have been independently devised by different WGs? Another issue is whether we need a core conceptual model, à la DMTF, from which all conceptual models should inherit. A third issue is the migration path for the IETF and DMTF from their current information modeling habits to our new process.

The DMTF is beginning to use this approach to devise its CIM schemas. Mappings of IETF MIBs and conceptual models into CIM schemas already exist (e.g. in the areas of policy, DiffServ, and IPsec policy).

ACKNOWLEDGMENTS

This research was conducted while J.P. Martin-Flatin was with AT&T Labs Research. Early versions of this work were published in the proceedings of DBTel 2001 [9] and NOMS 2002 [14]. The authors would like to thank C. Kalmanek, E. Lupu, and the members of the IRTF Network Management Research Group for their useful comments on drafts of this article.

REFERENCES

[1] W. Stallings, *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*, 3rd ed., Addison-Wesley, 1999.

[2] W. Bumpus *et al.*, *Common Information Model: Implementing the Object Model for Enterprise Management*, Wiley, 2000.

[3] A. Westerinen *et al.*, "RFC 3198: Terminology for Policy-Based Management," IETF, Nov. 2001.

[4] A. Pras and J. Schoenwaelder, "RFC 3444: On the Difference between Information Models and Data Models," IETF, Jan. 2003.
[5] J. P. Martin-Flatin, *Web-Based Management of IP Networks and Systems*, Wiley, 2002.
[6] J. Schoenwaelder and A. Mueller, "Reverse Engineering Internet MIBs," *Proc. IM 2001*, Seattle, WA, May 2001.
[7] K. McCloghrie, D. Perkins and J. Schoenwaelder, "RFC 2578: Structure of Management Information Version 2 (SMIv2)," IETF, Apr. 1999.
[8] DMTF, "Common Information Model (CIM) Specification," v. 2.2, June 1999.
[9] J. P. Martin-Flatin, "Toward Universal Information Models in Enterprise Management," *Proc. DBTel 2001*, Rome, Italy, Sept. 2001, pp. 167–78.
[10] OMG, "OMG Unified Modeling Language Specification," v. 1.3, Mar. 2000.
[11] C. Shaw, Ed., Minutes of the DMTF Events WG meeting of Jan. 11, 2001.
[12] G. Booch, *Object-Oriented Analysis and Design with Applications*, 2nd ed., Addison-Wesley, 1994, pp. 231–33.
[13] M. Fowler and K. Scott, *UML Distilled*, 2nd ed., Addison-Wesley, 2000.
[14] J. Schott *et al.*, "Common Information vs. Information Overload," *Proc. NOMS 2002*, Florence, Italy, Apr. 2002.

BIOGRAPHIES

JEAN-PHILIPPE MARTIN-FLATIN [SM] (jp.martin-flatin@ieee.org) is technical manager of the European DataTAG project (www.datatag.org) at CERN. He coordinates research activities in networking and middleware for data-intensive transoceanic Grids. Prior to that, he was a principal MTS with AT&T Labs Research. He holds a Ph.D. degree in computer science from EPFL, Switzerland. He has worked 10+ years in industry and government agencies in the areas of network and systems management, software development, network security, and Web engineering. His research interests include integrated management, distributed systems, and software engineering. He is on the editorial board of the *Journal of Network and Systems Management*. He is a co-chair of the GGF Data Transport Research Group and a member of the IRTF Network Management Research Group.

DIVESH SRIVASTAVA [A] (divesh@att.com) is head of the Database Research Division at AT&T Labs Research. He received his Ph.D. degree in computer science from the University of Wisconsin, Madison. He has been a researcher with AT&T since 1993, and has headed the database group since 1999. He has published extensively in the field of database management systems, and holds several patents in this field. His research focuses on the topics of data modeling, query language design, and query processing. He has applied this expertise recently to improving the functionality and performance of IP data warehouses, LDAP directories, and XML databases. He is on the editorial board of the *ACM SIGMOD Digital Review* and was on the editorial board of the *ACM SIGMOD Digital Symposium Collection* (1998–2002).

ANDREA WESTERINEN [A] (andreaw@cisco.com) is a senior architect and manager of information modeling at Cisco Systems. She has worked in the computer industry for more than 20 years, the last 10 in the areas of enterprise, system, network, storage, and policy-based management. She is an active participant in the DMTF and other standards organizations, such as the Storage Networking Industry Association (SNIA), The Open Group, and the Global Grid Forum. She is vice president of technology at the DMTF. Before joining Cisco, she was employed by Microsoft, Intel, IBM, and NCR. She has a B.S. in physics and mathematics from Marquette University, and an M.S. in computer science from Nova University.

Integrated management has made information modeling more complex. Dealing with this complexity calls for the use of more advanced methodologies.