EIDGENÖSSISCHE TECHNISCHE HOCHSCHULE LAUSANNE
POLITECNICO FEDERALE DI LOSANNA
SWISS FEDERAL INSTITUTE OF TECHNOLOGY LAUSANNE

COMMUNICATION SYSTEMS DIVISION (SSC)
CH-1015 LAUSANNE, SWITZERLAND
http://sscwww.epfl.ch

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

IP Network Management Platforms Before the Web

Jean-Philippe Martin-Flatin

Version 1: July 1998
Version 2: December 1998

Technical Report SSC/1998/021

IP Network Management Platforms Before the Web

Jean-Philippe Martin-Flatin
EPFL-ICA, 1015 Lausanne, Switzerland
Email: martin-flatin@epfl.ch Fax: +41-21-693-6610 Web: http://icawww.epfl.ch

Abstract

In this paper, we analyze the characteristics and shortcomings of IP network management platforms before the arrival of Web technologies. In the first part, we give a brief history of IP network management, and summarize the limitations of traditional (i.e., pre-Web and SNMP-based) management platforms. We recall the initial objectives of open network management. We then explain how the early vision of generic management was changed by the industry's natural inclination for market segmentation, and how the market of IP networks evolved from generic to vendor-specific equipment, management GUIs and MIBs. In the second part, we propose a simple model of traditional IP network management platforms, against which new Web-based management solutions can be compared. We introduce the three core functions of such platforms (network monitoring, data collection, and event handling), distinguish regular management from ad hoc management, and explain how SNMP's polling model maps onto these functions.

Keywords: SNMP, Open Network Management, Network Monitoring, Data Collection, Event Handling.

1. Introduction

The attraction of Web technologies has proved irresistible in many segments of the software industry. In IP network management, people began writing HTML forms and CGI scripts as early as 1993-94. These new technologies were typically used to standardize and automate problem reporting, or to replace print-outs of network usage reports with electronic equivalents. In 1995-96, vendors began experimenting with HTTP servers and Java applets embedded in network equipment, as reported by Wellens and Auerbach [26], Bruins [3], and Mullaney [20]. In 1997, many network management platform (NMP) vendors integrated a Web interface into their tool. In the course of 1997 and 1998, people realized that recent Web technologies — e.g., Java applets, servlets, RMI, Object Serialization, or mobile agents — could not only do differently what traditional (that is, pre-Web and SNMP-based) NMPs already did before: they also suggested new ways of collecting data, detecting faults, or distributing the network management application, and more generally, new ways of managing IP networks.

The first aforementioned reasons for the success of the Web in IP network management could be ascribed to fashion, or to the legitimate desire of enterprises to reduce the range of competencies needed to run their business. But the last reason (new ways of managing networks) goes much further, and suggests that Web technologies are here to stay in IP network management, and are more likely to become more pervasive than to fade away when a new fashion comes in.

But how did we come to this point? What were the administrators missing in traditional NMPs that they find today in Web-based management? Why were people so pleased with open network management in the first half of the 1990s, and why do customers currently put so much pressure on

network equipment vendors to have embedded HTTP servers, embedded JVMs, and even secure mobile-agent run-time environments?

The main goal of this paper is to answer these questions. In section 2, we describe the characteristics and shortcomings of traditional NMPs. By placing ourselves into an historical perspective, we show how we came to a point where Web technologies were needed, and how expectations from administrators evolved through the 1990s. We also highlight the fact that among the limitations that we identified, only a fraction can be addressed by simply integrating Web browsers into NMPs.

The second goal of this paper is to give a simple model of NMPs, to serve as a reference to which new models can be compared. This model is depicted and explained in detail in section 3. It is used in particular in two companion papers [14, 15], where we introduce the push and pull models in Web-based network management.

2. A Brief History of IP Network Management Before the Web

In this section, we give a brief history of IP network management before the Web. We explain the initial vision of an open market with generic tools, show how it evolved toward captive markets with vendor-specific tools, and study what the consequences of the advent of Windows-based NMPs were in a market so far dominated by Unix.

2.1. An open market with generic tools

In 1990-91, when NMPs[1] started to sell by the thousand, many people had a vision of open systems and generic network equipment. Unix systems were widely deployed in academia; in the industry, they nearly wiped out mini-computers for scientific work. Proprietary network equipment was gradually replaced with third-party equipment—that is, to interconnect hosts from vendor X, you no longer had to buy routers or bridges from the same vendor X. SNMP compliance became a commercial argument, often mandated by customers. High heterogeneity was considered a good thing, because it allowed substantial financial savings by choosing systematically the best buy of the day: if a network administrator wanted to purchase an IP router today, he would select, say, Cisco, whereas tomorrow he would buy 3Com, as a sheer result of unbiased competition in an open market. Putting a whole network together was as simple as using Lego: generic IP routers, bridges, hubs or FDDI concentrators were interchangeable, just as open systems (i.e., Unix hosts) were—more or less.

Very naturally, this vision was reflected in the management of this equipment: NMPs evolved from proprietary solutions, where a vendor would support only its own equipment (e.g., the first release of SunNet Manager could only manage Sun workstations), to open platforms (e.g., Lexcel's Lance+), which offered generic GUIs to manage all routers alike, all hubs alike, etc.

In those days, the goal was to hide vendor-specific features from operators, who spent their time monitoring large networks and gazing at GUIs. Any interface of any IP router, or any port of any hub, could be reset the same way. Traffic monitoring was independent of the brand of the equipment monitored. Operators resented the idea of having to master dozens of GUIs and dozens of ways of

1. A network management platform encompasses a Network Management Station (NMS)—characterized by a given hardware and a given operating system (e.g. a Sun UltraSparc workstation running Solaris 2.6, or a Compaq Pentium II PC running Windows NT 4.0)—and management software (e.g. Cabletron Spectrum or HP OpenView). For details about the terminology used in this tutorial, see Martin-Flatin *et al.* [13].

managing network devices: the physical heterogeneity had to be masked by an apparent homogeneity, in the logical view of a device offered by a management GUI.

This vision of IP network management was also shared and promoted by the IETF, who issued a number of generic MIBs: MIB-II [17] was released in March 1991, the Token Ring MIB [16] in May 1991, the RMON MIB [25] in November 1991, the FDDI MIB [4] in January 1992, the Hub MIB [18] in October 1992, etc. This effort from the IETF went on for several years.

2.2. Problems with early network management platforms

Over time, this vision of openness, and the very concept of genericity (generic equipment managed via generic GUIs using generic MIBs), did not resist the market reality. The main problems which were encountered were the hidden costs of heterogeneity, the limitations of generic MIBs, the vendors' desire to secure niche markets, and the inexperience of new customers.

First, customers discovered the hard way that genericity in network equipment is confined to bottom-of-the-range niche markets, such as print servers or terminal servers. But there is no such a thing as a generic router, because vendors endeavor to differentiate their offer from the competition—especially for expensive equipment such as routers that sustain the backbone of a LAN. So, for customers, the cost of training staff to manage equipment from different vendors, added to the cost of maintaining and debugging so many different network devices, quickly turned into a nightmare. The industrial reality proved that, over time, the extra costs ascribable to the heterogeneity of the network equipment often outweigh the immediate financial benefits of going for the best buy of the moment.

Second, vendors were not happy with such fierce competition that cut down their margins significantly. They were not happy either with the concept of generic equipment, which reduced their chances to differentiate their offer from the competition, and thereby justify a higher price for better equipment. So they looked for ways to segment this open market into multiple captive markets.

Third, vendors and customers alike were dissatisfied with the slow pace of the IETF Working Groups responsible for issuing generic MIBs. Several MIBs, especially those related to systems management, were released rather late, including the Host MIB [7] in September 1993, the Mail MIB [11] in January 1994, the UPS (Uninterruptible Power Supply) MIB [5] in May 1994, the Modem MIB [1] and the RDBMS MIB [2] in August 1994, etc.

Fourth, generic MIBs could not stand the comparison with interactive command line interfaces: they tended to be the least common denominator between all existing interfaces, because the IETF did not want to appear to favor one vendor over another, and because it could not keep up with the pace imposed by the vendors in a highly competitive market. As a result, administrators and operators could do with SNMP only a fraction of what they could do via a terminal directly attached to the network equipment, or via a `telnet` session. To give a concrete example, security was not standardized in router management like traffic statistics were in MIB-II. So, although most IP router vendors soon supported per-interface access control lists, no generic MIB allowed to manage them with SNMP.

Consequently, vendors looked for a way to work around the concept of generic network equipment, in order to increase their revenue, better satisfy their customers, and attract new customers. And undeniably, they succeeded in splitting this large open market of IP network management into a mosaic of smaller, captive markets.

2.3. Captive markets with vendor-specific tools

The answer to all these concerns came in the form of vendor-specific MIBs: vendors developed their own MIBs to manage their own equipment. These proprietary MIBs were richer and more comprehensive than the generic MIBs then specified by the IETF. They covered most if not all of the interactive command line interface, whereas generic MIBs covered only a fraction of it. This addressed the fourth issue raised in the previous section. Vendors were also in complete control of their own MIBs, so they could update them at a faster pace than the IETF (e.g., there were three releases of the Cisco MIB between 1991 and 1995). This tackled the third issue.

In order to help their customers manage their network equipment, vendors developed vendor-specific management GUIs, and even device-specific management GUIs, which relied heavily, if not exclusively, on their proprietary MIBs. These GUIs were initially integrated into stand-alone management applications, which were sold separately, and generally run on middle-range PCs. But many customers had to manage heterogeneous networks, and did not want to run multiple management applications in parallel. This also defeated one of the purposes of integrated network management: to be able to manage all network equipment from a single NMP. Instead, customers wanted vendors to integrate vendor-specific management GUIs into their favorite NMP. So they did. Peer-to-peer agreements were signed between NMP vendors and network equipment vendors, which allowed the latter to integrate their proprietary GUIs into existing NMPs. These GUIs were called *add-ons*, because they were optional extensions of the management platforms.

Vendor-specific management GUIs gave vendors the opportunity to gain a commercial edge on the competition: the quality of these GUIs became a valuable commercial argument for salespeople. Management GUIs evolved to look like the real devices, a feature that operators—and often administrators—came to cherish. Beside the marketing argument that their network equipment supported generic MIBs, a warrant of their openness, vendors also put forward the fact that this equipment could be managed with user-friendly and intuitive add-ons, e.g. CiscoWorks by Cisco. Technically, these two arguments were actually opposed; nonetheless, salespeople used them side by side, and successfully.

The peer-to-peer agreements that we mentioned above progressively reshaped the entire landscape of IP network management, by introducing a bias which seriously shook the concept of “open” management. They were based on market shares, business relationships, and mutual commercial interests. And the bias was that some business partners became more equal than others, to put it in an orwellian way.

As far as network equipment vendors were concerned, small companies (especially start-ups) found it very difficult to sign any agreement at all with NMP vendors. They had small market shares, and they did not bring in new customers or new revenue, so they were of no interest to large NMP vendors. For these small companies, the best strategy was to find one large customer willing to pay the high price for the integration of a management GUI into its current NMP (especially to manage cutting edge equipment)... but such customers were difficult to find. Large equipment vendors, conversely, had no problem to integrate their GUIs into any NMP: their names even appeared on the booklets advertising these NMPs. But the timeliness of this integration depended to a large extent on the quality of the business partnership between the equipment and the NMP vendors. Between the time a new equipment was put on the market and the time its management GUI could be integrated in a given NMP, it could take several weeks, or two months, or six months...

As far as customers were concerned, gone were the days of open systems, generic MIBs and NMPs able to manage any network equipment. Gradually, they became prisoners of their NMP or their vendor-specific add-ons.

For customers managing small networks, the relative cost of the NMP was high, compared to the total price of their network equipment. To put it simply, they had two options: they could abide by the rules set by their NMP vendor, and get flashy GUIs in a timely manner from one of its "friends"; or they could take the dangerous route of a start-up company, and try to live with user-unfriendly MIB browsers, additional NMSs, and dissatisfied operators. In practice, in most business settings, they had no choice.

For customers managing large networks, the problem was not so much their NMP as the heterogeneity of their network equipment. They could afford another NMP, if really needed, but they were prisoners of their existing equipment suppliers, and especially their vendor-specific add-ons. Companies could realistically train operators to master a few dozens GUIs, but they could not ask them to know several hundred. Similarly, administrators could not be expected to manage an infinitely large number of different systems. Consequently, companies could not afford to have too many suppliers for their network equipment. When a network administrator wanted to buy a device of a certain type, for which he/she already had one or two suppliers, he/she would always select one of these suppliers, and was prepared to pay a higher price for it than that of the competitors.

Whatever the size of their network, customers could no longer pull the prices down as much as they could in the past. Competition had been hampered: the IP network management market had entered a logic of captive markets. Vendors had managed to address the second issue raised in section 2.2, that is, secure niche markets exposed to little competition.

Proprietary MIBs and GUIs were initially resisted by many experienced customers, who knew the drawbacks of captive markets. But from 1992-93, they were quickly adopted by new customers, perhaps more naïve, who did not realize that by doing so, they were contributing to the segmenting of the open market of SNMP into a multitude of captive markets. In other words, they were wrecking the very reason why they had come to SNMP in the first place: its openness. SNMP was no longer an open protocol supporting open management in an open market: it had become an open protocol supporting proprietary management in a mosaic of captive markets. The sole stable guarantees of openness were the use of third-party NMPs, and the conformance to one of the SNMP management frameworks (SNMPv1, SNMPv2, SNMPv3).

To be fair, captive markets were not only bad news for customers. Admittedly, they induced higher purchase costs for customers. But they also lowered the training cost of staff (administrators and operators), and they significantly simplified the maintenance of the network equipment. Indeed, although they were created primarily by vendors for their own sake, captive markets also suited customers in some respects, and, in particular, addressed the first issue raised above.

Peer-to-peer agreements between NMP and network equipment vendors had an even greater impact on the NMPs market. As far as NMP vendors were concerned, the major players knew that they were a must for large network equipment vendors. They did not have to spend much money for all major equipment vendors to port their management GUIs onto their NMP. But smaller players were progressively left out. The less customers an NMP vendor had, the less chances its NMP would have to raise interest from equipment vendors. By snowball effect, smaller players were initially supported one month later than the bigger players, then six months later, and one day, they were not supported at all. If customers wanted to manage a piece of equipment with an NMP that had only gained a small market

share, they were charged for its development... a strong deterrent indeed! For small equipment vendors, as we saw, the situation was also bad; but at least they had a workaround: customers could manage their equipment with a dedicated PC, directly attached to the equipment. Things were much worse for small NMP vendors: without peer-to-peer agreements, they could not survive.

As a result, these agreements caused the number of actors in the NMPs market to reduce drastically. During the years 1993-95, minor players gave up this market, one after the other, and only four NMP vendors managed to retain a significant market share: HP with OpenView, Sun with SunNet Manager (which later became Solstice), Cabletron with Spectrum, and IBM with NetView. These platforms still dominate the IP network management market today. Interestingly enough, only one of these four companies came from the networking industry, while the three others came from the computing industry. The reason was that third-party management software appeared to be a guarantee of impartiality and openness. A company like HP was more trustworthy when it claimed to manage all network equipment alike, whatever its vendor, than the heavy weights of the networking industry like Cisco or 3Com, who had too many business interests at stake[1].

The only type of peer-to-peer agreement we considered so far involved NMP and network equipment vendors. But another type of agreement soon appeared. NMPs initially relied on a single solution to store management data; this could be a proprietary database (like HP OpenView in its early days), a third-party DBMS (relational or object-oriented), or a directory tree of text files. But large customers often had an RDBMS in their intranet; they had already spent fortunes on training its administrators, and were very reluctant to use another DBMS; so they asked NMP vendors to support their existing RDBMSs. Unfortunately, although all RDBMSs spoke SQL, they all offered different APIs (there was no mature ODBC in those days), and the cost of porting from one RDBMS to another was high for NMP vendors. This problem appeared to be solved when peer-to-peer agreements began to be signed by NMP and DBMS vendors. But unlike what happened with NMP and network equipment vendors (all major NMPs soon supported all major equipment vendors, and *vice versa*), the proportion of agreements that were actually signed by NMP and DBMS vendors remained fairly low, compared to the number of actors on these markets. In 1995-96, just before the days of Web-based management, Cabletron still supported only a proprietary OODBMS, whereas HP, Sun and IBM supported only one or two major RDBMSs (Oracle, Sybase, Informix, Ingres). As most customers could not afford to buy a new DBMS for the sole purpose of network management, they often resorted to storing management data into text files, and writing ad hoc scripts to exploit this data.

2.4. The move from Unix-based to Windows-based network management platforms

Before the Web came into the game, a last change occurred in the IP network management market. As this market had gained the reputation of being mature, in 1993-94, many small to medium-sized enterprises (SMEs) decided to make a move toward SNMP-based network management. But for these companies, an expensive NMS running expensive software was simply not an option: there had to be an inexpensive way to manage their relatively small networks. Operators were also out of the question: these companies could not afford staff monitoring constantly the health of their network, which generally consisted of a fairly small LAN, with possibly just a few WAN links. Network management was entirely the responsibility of the network administrator, who was managing the network on an ad hoc basis (typically when a problem showed up, that is, troubleshooting rather than

1. In the PC market, a similar conflict of interests exists between Microsoft vendor of the Windows operating system, Microsoft vendor of applications for Windows, and third-party vendors of applications for Windows. Is Microsoft trustworthy when it claims that Windows treats all applications alike?

monitoring), and did not require permanent access to an NMS. Finally, Unix was generally not an option either, as SMEs were mostly equipped with PCs and Mac's, and often lacked Unix expertise.

In order to take on this new market, which was all the more attractive since the market of larger networks was almost fully equipped, NMP vendors came up with so-called "light" versions of their software. They were "light" in the sense that they were less demanding in terms of CPU, memory and disk resources, they did not offer all the functionalities of the Unix-based NMPs (e.g., no RDBMS to store polled data), and they were less expensive. These were not as inexpensive as customers were hoping for, but they could easily run on middle-range PCs, and they could run under Windows. As a matter of fact, by 1996, with the significant increase in the power of PCs and the generalization of Windows NT in enterprises (that is, a clean and stable operating system a la Unix, which did not repeatedly crash like Windows 3.1), there were no longer any functionality differences between Unix-based and Windows-based NMPs.

This evolution enlarged the network management market significantly. But it also indirectly made the development costs for network equipment vendors skyrocket: not only did they have to integrate their device-specific GUIs into all major Unix-based NMPs, they also had to support these NMPs under new operating systems: Windows 3.1, Windows 95, Windows 98, Windows NT 3.x, Windows 4.x...

2.5. Problems with traditional network management platforms

To summarize the shortcomings of IP network management before the Web, customers had three grievances: (i) they sought less expensive solutions, in terms of both hardware and software; in particular, they did not want to have dedicated hardware to manage small networks; (ii) they wanted to be able to store management data in whatever RDBMS they happened to own, and did not want to be constrained by peer-to-peer agreements signed or not signed between NMP and RDBMS vendors; (iii) those who were supporting a Unix system for the sole purpose of network management wanted to use a PC or a Mac instead, since this kind of expertise was ubiquitous in enterprises; but they did not want to buy a new and expensive NMP for that.

Network equipment vendors were primarily dissatisfied by the huge cost they had to bear to support a myriad of device-specific GUIs running on all existing hardware and operating systems. They also shared two concerns with customers. First, they wanted to reduce the time-to-market of vendor-specific management GUIs. Ideally, they wanted any network equipment to be manageable via a nice GUI (that is, not a mere MIB browser) as soon as it was launched on the market. Start-up companies also wanted to have access to integrated management platforms, in order to attract more customers. Second, both vendors and customers needed a solution to the problem of versioning. From time to time, network equipment vendors release a new version of their proprietary MIBs and GUIs. As it was not possible to upgrade all equipment and the add-ons of the NMP at the same time, customers had to live with several versions of the same MIB at the same time, either temporarily or permanently. But many NMPs were poorly designed in this respect: some did not support multiple versions of the same MIB, while others required the administrator to enter manually, device by device, what version of the MIB was supported. Customers and vendors alike wanted this phase to be automated, e.g. via some kind of MIB-discovery protocol.

In summary, we showed in section 2 how the way of managing IP networks evolved since the inception of SNMP and open network management, and how the initial vision of generic equipment was lost in favor of a more pragmatic, business-oriented approach leading to:

- peer-to-peer agreements between vendors
- captive markets for customers
- vendor-specific management add-ons

3. A Simple Model of Network Management Platforms Before the Web

Now that we have explained how IP networks were typically managed when Web technologies arrived in this field, let us present a simple model of how NMPs were structured before the Web. This model will be used in [14, 15] as a standard against which new models, integrating Web technologies, can be compared and assessed.

Traditional NMPs can all be modeled as follows:

- a dedicated Network Management Station (NMS)
- software to deal with network monitoring and data collection (polling of the agents by the manager[1])
- software to deal with events (SNMP notifications issued by agents)
- GUIs for generic MIBs
- GUIs for vendor-specific MIBs (add-ons)
- a data repository, which can be a third-party RDBMS (Oracle, Sybase...), a proprietary DBMS, or a collection of text files

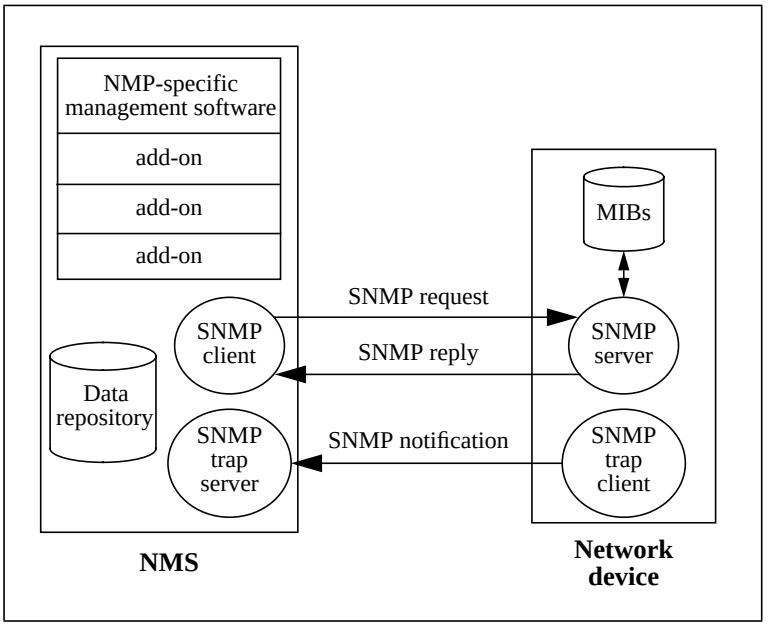

Fig. 1. Network management platforms before the Web

1. Called the management station by some authors [23]

3.1. Data polling vs. event handling

In the IP world, NMPs fulfill three basic tasks: network monitoring, data collection, and event handling (these concepts are presented in detail in Rose [23]). Some NMPs include other tasks such as inventory management, accounting or billing, but these are not at the heart of SNMP-based network management.

Data collection (see Fig. 2) is the process of gathering statistics to build daily, weekly and monthly reports for the administrator. These reports are generated off-line (see Fig. 4). The purpose of *network monitoring* (see Fig. 3), on the other hand, is to check whether each network device is alive, and generate an alarm in case a device is diagnosed as being hung or down.

In pre-Web network management, network monitoring and data collection are implemented with a mechanism known as *data polling* (or simply *polling*), which is based on the request/response model, and the client-server style of communication. It is characterized by the manager sending many SNMP `get`'s to many managed devices (agents) at regular time intervals[1], and receiving separate replies from all agents.

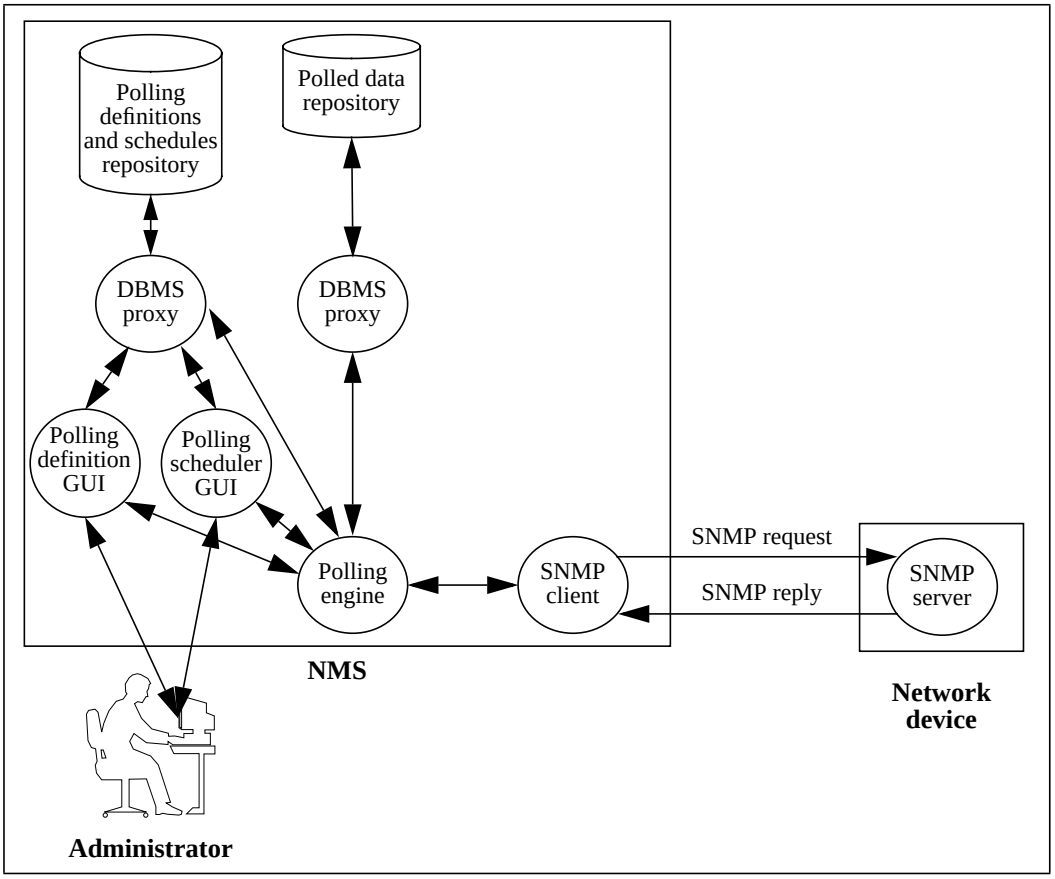

Fig. 2. Data collection before the Web

1. There is a trade-off to be found between the polling frequency, the number of devices to be polled, the acceptable level of network overhead, and the processing power of the NMP. A typical scenario for intranets is to poll all managed devices every 5 minutes. In case the NMP becomes overloaded, and since the polling frequency can be defined on a per-type or per-device basis on most platforms, critical equipment such as IP routers may be polled often, say every 2 minutes, whereas non-critical equipment may be polled less often, say every 15 minutes.

The network monitoring of IP networks is claimed to be based on *trap-directed polling* [23, p. 20]; that is, upon receiving an SNMP notification, a manager performs some polling on the originating agent in order to work out what the problem is with this agent. This can be automated by software, or accomplished manually by an operator. In fact, as SNMP notifications are sent with an unreliable transport mechanism (UDP), and as network equipment may crash without sending any notification, SNMP notifications cannot be relied upon to detect faults; polling is needed at least as a backup [23, p. 20]. In practice, polling generally prevails. According to Lynch *et al.*, "the SNMP management framework primarily utilizes polling, and traps are relegated to a secondary role" [12].

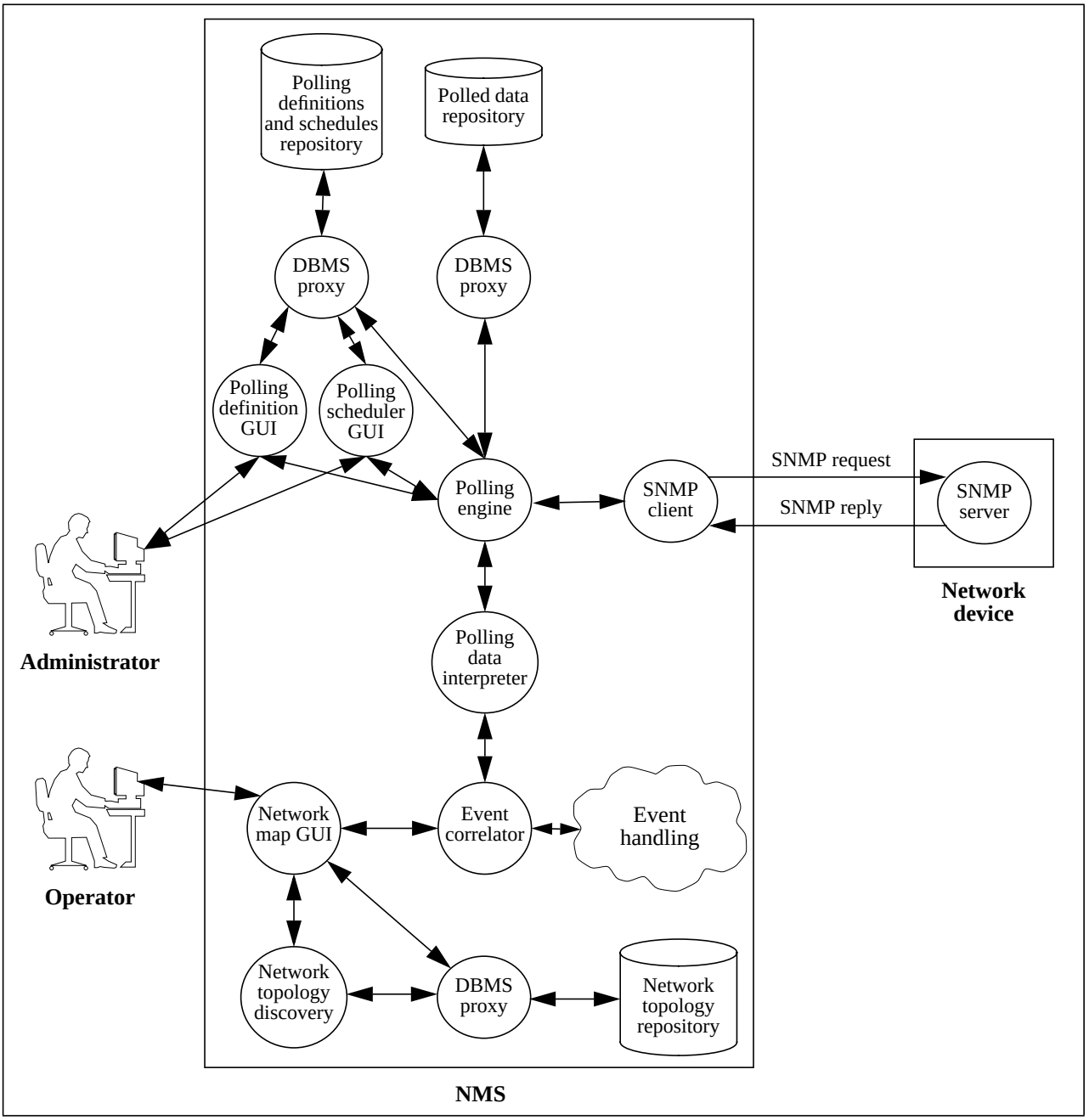

Fig. 3. Network monitoring before the Web

We show on the figures different data repositories. Even though they are logically different, in practice, they can be (and often are) stored in a single DBMS system, be it an RDBMS, an OODBMS, or a simple directory tree composed of plain text files.

The interaction between the operator and the network map GUI works both ways: the GUI displays colored icons (red, green, yellow...) to indicate the current state of the network device, while the operator interacts with the GUI to, for instance, request a time series of a collision rate. Similarly, the interaction between the administrator and other GUIs (e.g., polling definition or polling scheduler) is bi-directional: the GUIs display previously stored entries, while the administrator can type in new ones.

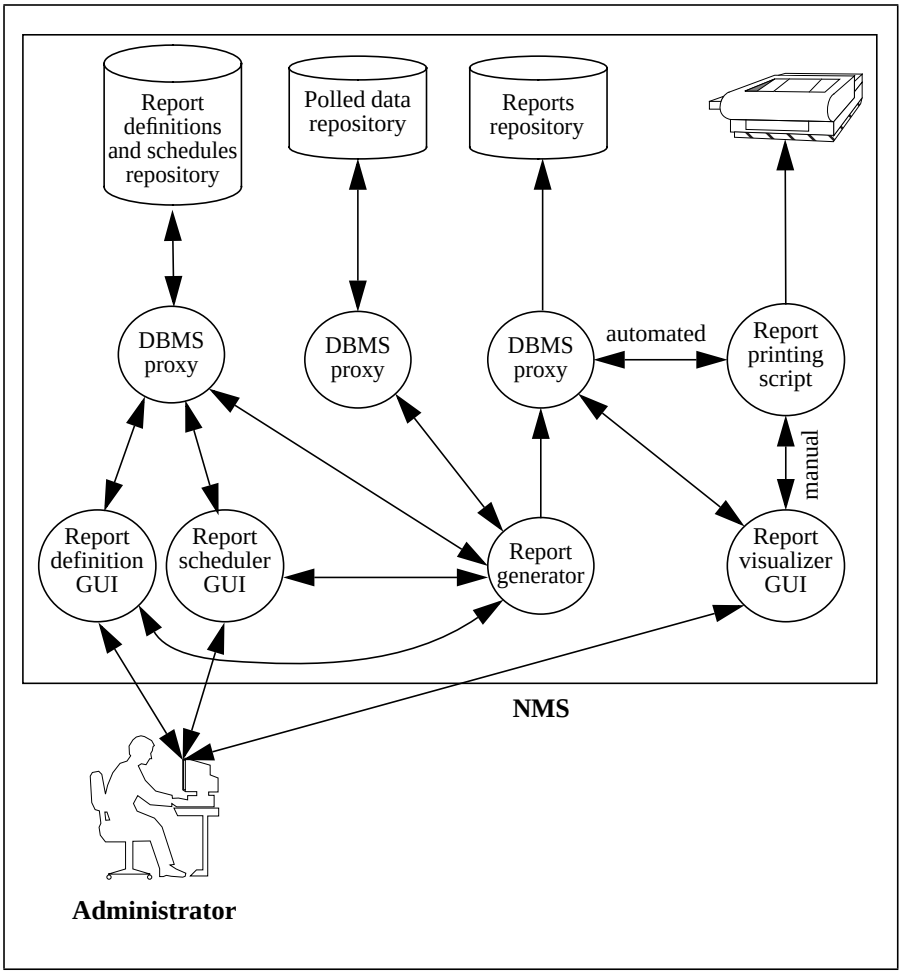

Fig. 4. Report generation before the Web

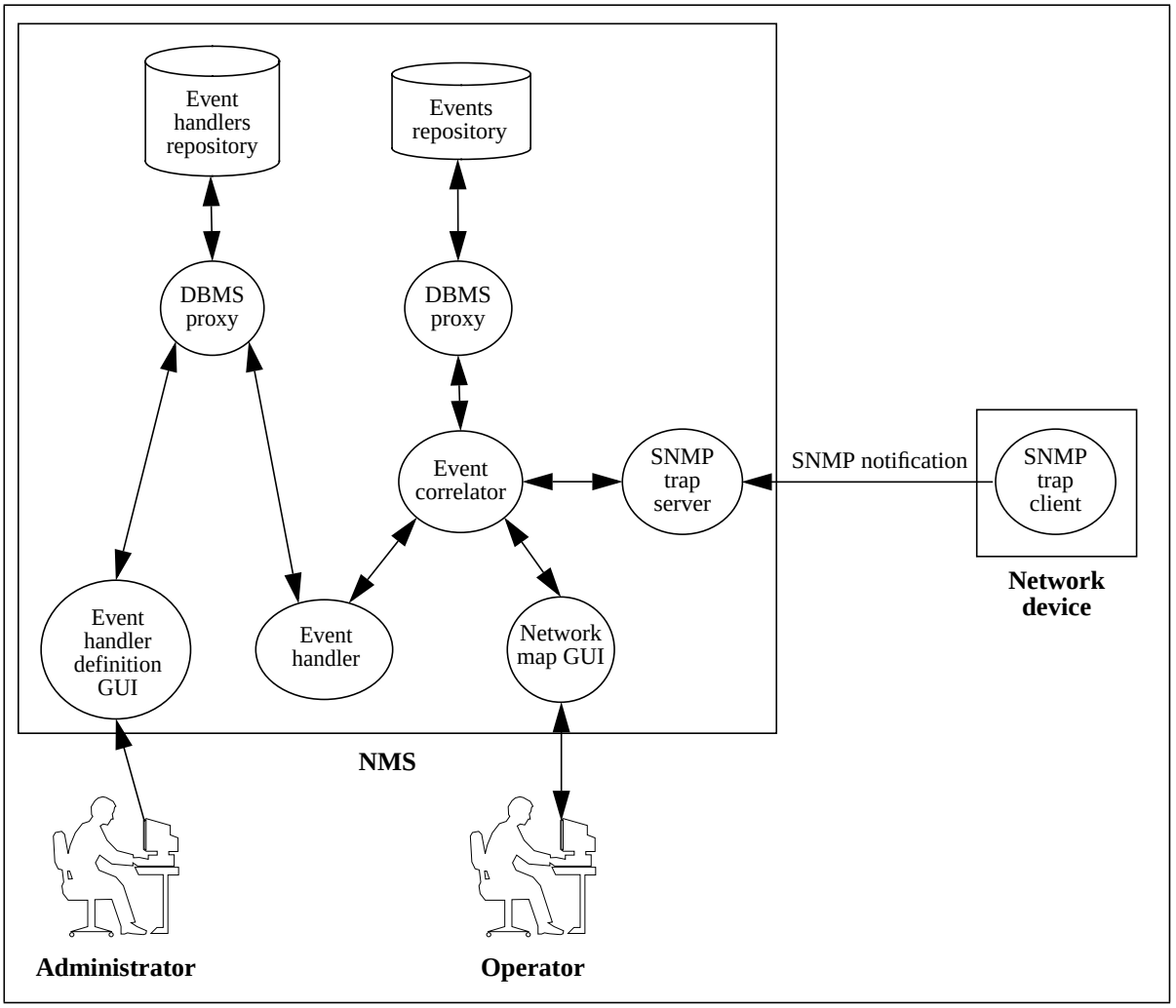

Fig. 5. SNMP notification delivery and event handling before the Web

Event handling, conversely, is not based on the request/response model. There are two sources of network events. First, when an agent detects an internal condition that it deems abnormal (e.g., an interface going down, or a thermometer reading going above a maximum threshold), it sends an unacknowledged asynchronous alarm to its manager [23]. These alarms are known as *SNMP traps* in the SNMPv1 framework, and as *SNMP notifications* in the SNMPv2 and SNMPv3 frameworks [22]. The second source of network events is the polling data interpreter (see Fig. 3), when it infers, based on the polling, that a network device is down or hung.

The notations we used in Fig. 1 and Fig. 5 might seem confusing at first: why would an SNMP *trap* client send an SNMP *notification* to an SNMP *trap* server? Why mix SNMPv1 and SNMPv2/v3 concepts? Actually, both figures use the latest terminology. In the Unix world, the `/etc/services` file, or the `services` map in an NIS administrative domain, are supposed to use the latest port names and numbers assigned by the IANA [8]. At the time of writing, ports `162/tcp` and `162/udp` were still assigned the name `snmptrap`. This name has not changed since the SNMPv2 framework was released, which explains why people still refer to *SNMP trap clients* and *SNMP trap servers*, even when they no longer use SNMPv1-based software.

3.2. Regular management vs. ad hoc management

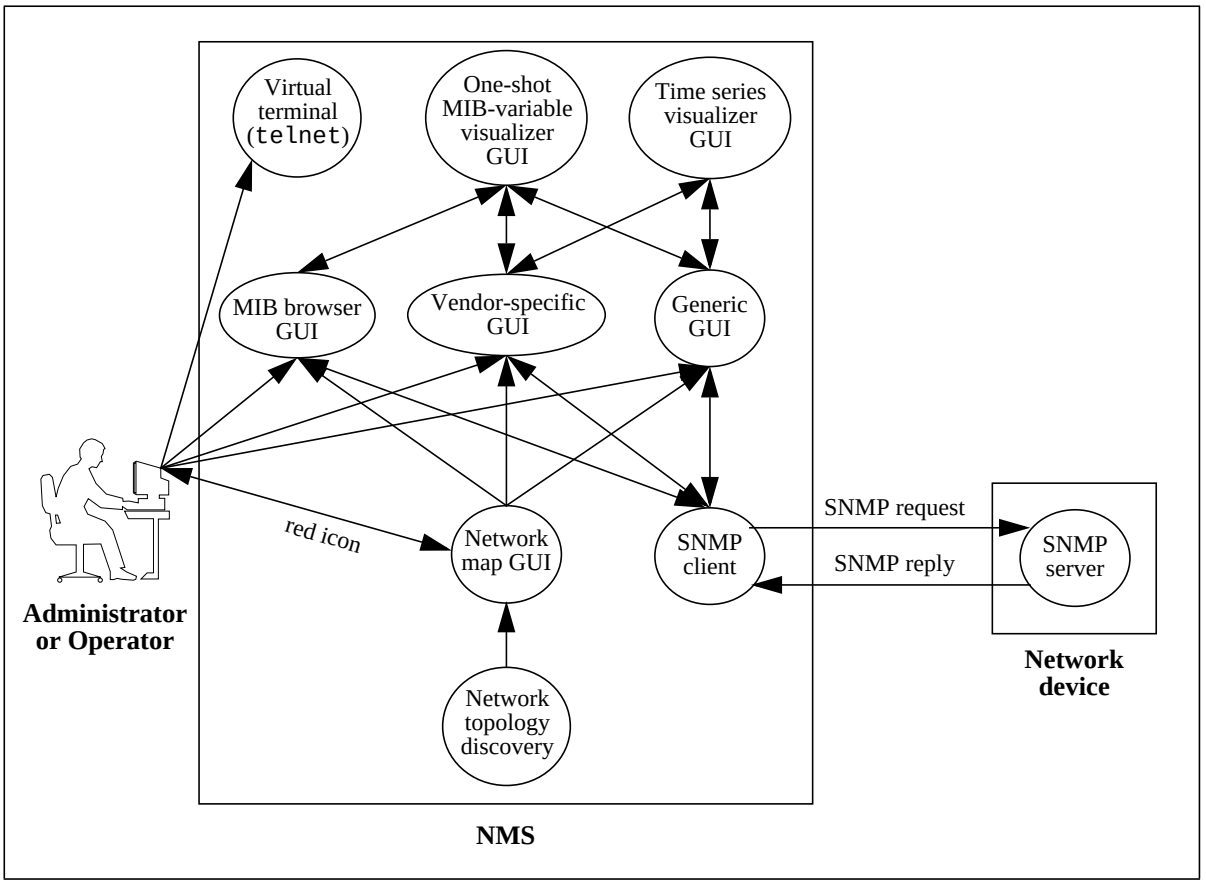

Fig. 6. Ad hoc management before the Web

NMPs can be used for both regular and ad hoc management. Regular management requires a dedicated, permanently available NMS. It encompasses network monitoring, data collection and event handling. In large networks, or in environments where the network is critical to the smooth running of the enterprise business, regular management is typically performed by staff dedicated to network monitoring, often called *operators*. In some SMEs, there is no regular management at all. Others rely on fully automated management: when a critical fault is detected, the administrator is automatically paged; when a minor fault is detected, the administrator is sent an email. Regular management operates over two different time scales: network monitoring typically occurs every few minutes, and aims at solving problems quickly (before users complain); whereas reports generation generally takes place every day, week and month, and is performed off-line, as a background task.

Ad hoc management takes place in virtually all organizations. In the case of a large network, an administrator may occasionally want to pop up a GUI to configure a router, or an operator may take a look at a time series of error rates while investigating a network problem. In this case, ad hoc management is complementary to regular management. In the case of a small network, ad hoc management often replaces regular management. In this case, there is no operator and no dedicated NMS: the management software is only used once in a while when a network problem shows up, or when a new equipment is put in place and needs to be configured. Ad hoc management generally consists in troubleshooting (i.e., there is a problem with the network, and the administrator tries to identify and repair the problem manually), but can also occasionally consist in monitoring (e.g., the administrator checks the configuration of a router, or checks whether error rates are reasonably low).

In short, regular management is automated to a large extent, whereas ad hoc management is always manual and requires a person (administrator or operator) to interact with the management software via some GUIs.

4. Conclusion

The goal of this technical report was to summarize the characteristics and limitations of pre-Web SNMP-based network management platforms, and thus to prepare the grounds for two companion papers [14, 15] that present different Web-based network management solutions.

To achieve this, we first presented a brief history of IP network management. We described the early vision of generic tools for interchangeable network equipment. We then showed how the industrial drive for market segmentation caused the loss of this vision. Finally, we built a new picture of an SNMP-based management framework, based on add-ons, proprietary MIBs, and differentiated network equipment. Throughout this first part, we pointed out the limitations of pre-Web network management, taking alternately a customer's perspective, a vendor's perspective, or both.

In the second part, we proposed a simple model of IP network management platforms. We studied and illustrated the three basic tasks fulfilled by these platforms: network monitoring, data collection, and event handling. We showed how these tasks are fulfilled in the SNMP management framework, and distinguished regular management from ad hoc management.

Acknowledgments

This research was partially funded by the Swiss National Science Foundation (FNRS) under grant SPP-ICS 5003-45311. The author wishes to thank H. Cogliati for proofreading this paper.

Acronyms

API	Application Programming Interface	NMP	Network Management Platform
ATM	Asynchronous Transfer Mode	NMS	Network Management Station
CGI	Common Gateway Interface	ODBC	Open DataBase Connectivity
CPU	Central Processing Unit	OODBMS	Object-Oriented DataBase Management System
DBMS	DataBase Management System	PC	Personal Computer
FDDI	Fiber Distributed Data Interface	RDBMS	Relational DataBase Management System
GUI	Graphical User Interface	RFC	Request For Comment
HTML	HyperText Markup Language	RMI	Remote Method Invocation
HTTP	HyperText Transfer Protocol	RMON	Remote MONitoring
IETF	Internet Engineering Task Force	SME	Small to Medium-sized Enterprise
IP	Internet Protocol	SNMP	Simple Network Management Protocol
JVM	Java Virtual Machine	SQL	Simple Query Language
LAN	Local-Area Network	UDP	User Datagram Protocol
MIB	Management Information Base	UPS	Uninterruptible Power Supply
NIS	Network Information Service	WAN	Wide-Area Network

References

[1] J. Barnes, L. Brown, R. Royston and S. Waldbusser (Eds.). *RFC 1696. Modem Management Information Base (MIB) using SMIv2*. IETF, August 1994.

[2] D. Brower, B. Purvy, A. Daniel, M. Sinykin and J. Smith (Eds.). *RFC 1697. Relational Database Management System (RDBMS) Management Information Base (MIB) using SMIv2*. IETF, August 1994.

[3] B. Bruins. "Some Experiences with Emerging Management Technologies". In *The Simple Times*, 4(3):6–8, 1996.

[4] J. Case (Ed.). *RFC 1285. FDDI Management Information Base*. IETF, January 1992.

[5] J. Case (Ed.). *RFC 1628. UPS Management Information Base*. IETF, May 1994.

[6] J. Case, K. McCloghrie, M. Rose and S. Waldbusser (Eds.). *RFC 1902. Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)*. IETF, January 1996.

[7] P. Grillo and S. Waldbusser (Eds.). *RFC 1514. Host Resources MIB*. IETF, September 1993.

[8] IANA. *Protocol Numbers and Assignment Services*. Available at <URL:http://www.iana.org/numbers.html>. This Web site updates RFC 1700 which is now obsolete.

[9] ITU-T. *Recommendation X.690. Information Technology—ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*. ITU, Geneva, Switzerland, July 1994.

[10] ITU-T. *Recommendation X.691. Information Technology—ASN.1 Encoding Rules: Specification of Packed Encoding Rules (PER)*. ITU, Geneva, Switzerland, April 1995.

[11] S. Kille and N. Freed (Eds.). *RFC 1566. Mail Monitoring MIB*. IETF, January 1994.

[12] D.C. Lynch and M.T. Rose. *Internet System Handbook*. Addison-Wesley, Reading, MA, USA, 1993.

[13] J.P. Martin-Flatin, S. Znaty and J.P. Hubaux. "A Survey of Distributed Enterprise Network and Systems Management". In *Journal of Network and Systems Management*, 7(1):9–26, 1999.

[14] J.P. Martin-Flatin. *Push vs. Pull in Web-Based Network Management*. Technical Report SSC/1998/022, SSC, EPFL, Lausanne, Switzerland, July 1998.

[15] J.P. Martin-Flatin. *The Push Model in Web-Based Network Management*. Technical Report SSC/1998/023, SSC, EPFL, Lausanne, Switzerland, July 1998.

[16] K. McCloghrie, R. Fox and E. Decker (Eds.). *RFC 1231. IEEE 802.5 Token Ring MIB*. IETF, May 1991.

[17] K. McCloghrie and M.Rose (Eds.). *RFC 1213. Management Information Base for Network Management of TCP/IP-based internets: MIB-II*. IETF, March 1991.

[18] D. McMaster and K. McCloghrie (Eds.). *RFC 1368. Managed Objects for IEEE 802.3 Repeater Devices*. IETF, October 1992.

[19] N. Mitra. "Efficient Encoding Rules for ASN.1-Based Protocols". In *AT&T Technical Journal*, 73(3):80–93, 1994.

[20] P. Mullaney. "Overview of a Web-based Agent". In *The Simple Times*, 4(3):8–12, 1996.

[21] G. Neufeld and S. Vuong. "An overview of ASN.1". In *Computer Networks and ISDN Systems*, 23:393–415, 1992.

[22] D.T. Perkins. "Questions Answered". In *The Simple Times*, 6(1):13–17, 1998.

[23] M.T. Rose. *The Simple Book: an Introduction to Networking Management*. Revised 2nd edition. Prentice Hall, Upper Saddle River, NJ, USA, 1996.

[24] M.T. Rose and K. McCloghrie (Eds.). *RFC 1155. Structure and Identification of Management Information for TCP/IP-based Internets*. IETF, May 1990.

[25] S. Waldbusser (Ed.). *RFC 1271. Remote Network Monitoring Management Information Base*. IETF, November 1991.

[26] C. Wellens and K. Auerbach. "Towards Useful Management". In *The Simple Times*, 4(3):1–6, 1996.

Biography

J.P. Martin-Flatin is currently preparing for a Ph.D. thesis at EPFL. From 1990 to 1996, he was with the European Centre for Medium-Range Weather Forecasts in Reading, England, where he worked in network and systems management, security, Web management and software engineering. From 1988 to 1990, he worked on the Geographic Information System of a large city in France. In 1986, he received an M.Sc. in EE and ME from ECAM, Lyon, France. His main research interest is in distributed network management. He is a member of the IEEE and the ACM.