Management of the World-Wide Web

Aiko Pras, Harrie Hazewinkel[1], Eric van Hengstum
Centre for Telematics and Information Technology, University of
PO Box 217, 7500 AE Enschede, The Netherlands
pras@cs.utwente.nl - tel.: +31-53-4893778

Abstract

This paper describes the results of a 'proof of concept Centre (JRC) of the European Commission (EC). The subject of the capabilities of the World Wide Web (WWW). SNMP was selected as the study started there were no SNMP-WWW MIBs available and ment was very limited [1]. As a consequence, we had to define our own MIBs and implement WWW management applications and agents.
We decided to define three different MIBs. The first one is the kind of server, the number of server accesses and the documents the Access Point MIB, which can be used to obtain high level documents over the Internet. The third MIB is the HTTP MIB, and ation of the HTTP protocol.
To test the practicability of these MIBs, agents were implemented servers throughout Europe. The original agent implementation is package, but recently a public domain version, which is based on results of our study have been submitted to the IETF and are being project.

Keywords: WWW, SNMP, MIB, HTTP, Extensible agent, EMANATE, Scotty

1 Introduction

Within Europe there is an interesting and growing market for earth collected from satellites as well as ordinary ground stations that air pollution. To increase the possible use of this data, the a programme called the 'Centre for Earth Observation' which is coordinated by the Commission's Joint Research Centre infrastructure via which potential users of earth observation data such data. The infrastructure will be build on top of the Internet Web' (WWW) as the main application. The infrastructure should the addition of alternative applications like the 'File

WWW is based on a client-server approach. Upon the mouse-click of client requests the WWW server to send one or more multi-media document, the 'viewer part' of the client interprets the within the document and presents the result to its user. WWW 'browsers'; examples come from Netscape, Microsoft and
Well known WWW servers are those from NCSA and Apache. Both such, that they keep copies of all requests and errors in special
A special kind of server is the 'gateway' server, which applications like for instance databases. The advantage of ments need not come form static text files, but can be computed request is made. This guarantees that documents always contain the

As shown in Figure 1, WWW can be decomposed into two functional layers. The upper which is located on top of a conceptual document transfer service, and presentation of documents. To accomplish this, it uses the (HTML). The lower layer takes care of the transparent[2] transfer of HTML documents over underlying TCP connections. The protocol that is used for this Protocol (HTTP) [2].
To investigate the potential management capabilities of this ESYS Ltd. (UK) and the University of Twente (The Netherlands) [3]. Because the skills of both partners were different, it was trate on the manager side of the problem and the University of study was the first one of a series, it was not necessary to problem now; in fact it was sufficient to restrict this first performance and server statistics. Because the study would partners also agreed to use SNMP [4] as management protocol.

The emphasis of this paper is on the agent specific aspects of WWW presents the management architecture and identifies the MIBs that ture of these MIBs is discussed in Section 3. Section 4 explains the way they were implemented.

1 Harrie Hazewinkel is currently working for the Joint Research
2 The term 'transparent' means that the contents of at this layer.

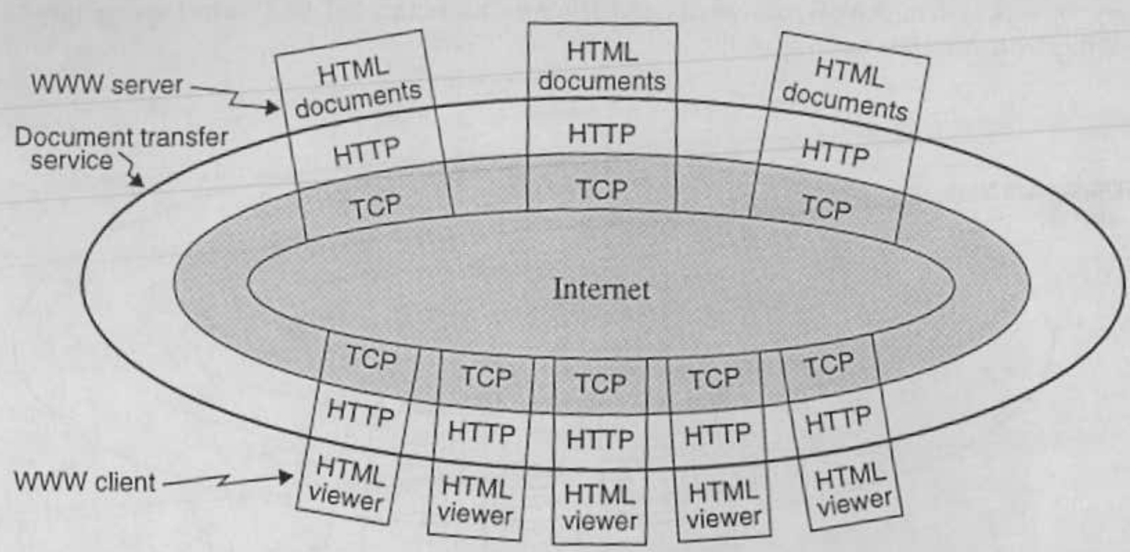

Figure 1: Functional view of the WWW

2 Management architecture

To enable managers to monitor the WWW servers, a special *Server MIB* (S-MIB) had to be developed. This MIB, which had to be implemented within each server information like: the server's name, the contact person, the used, the address of the clients that used this server, the far etc. A detailed description of this MIB is provided in Section Next to monitoring the WWW servers, managers should also be able which documents are being transferred. Ideally managers would *central transfer service MIB*, which provides high level information like: how much information is exchanged between each client-server pair, what is the delay, client perceives etc. Unfortunately, this kind of information is system within the network, but should be computed from rudimentary tered over all client and server systems. The idea of a service tion of a kind of *mediation device* (an intermediate level manager), which interacts with the various client and server systems to populate its service MIB. Since a more complex and is not essential for this proof of concept study, of a central service MIB. Instead, it was agreed to develop a within the manager system. This application collects the *Access Point MIBs* (A-MIB), which were developed within the project and implemented each client and server system (Figure 3). Section 3.2 provides an overview of this MIB.

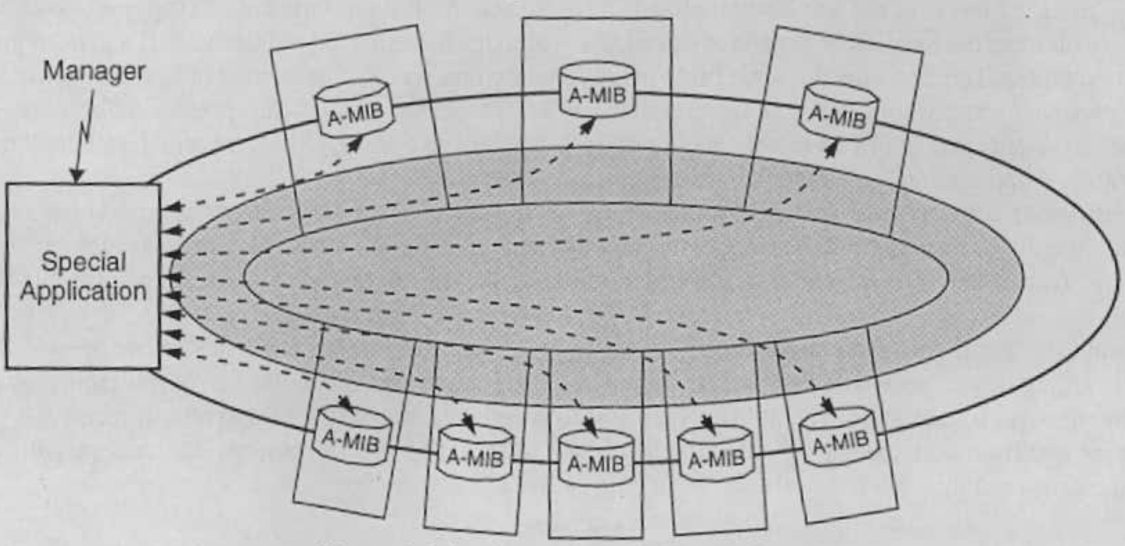

Figure 2: The Access Point MIBs (A-MIB)

Extending each WWW client and server system with an Access Point amount of work and is not feasible within this proof of concept therefore decided to restrict the implementation of the Access number of server systems, which had to be adapted anyway because Server MIBs. It is important to understand however, that despite has not been implemented within the client systems, indications of obtained by using the well known 'ping' and 'echo'

If the manager detects from the Access Point MIBs a problem within the manager should inspect the various protocol MIBs to find the Management information concerning the IP and TCP protocol can be which is usually available within every network system. A MIB for available at the time the project started, thus we had to develop project. The structure of this MIB, which is again only discussed in Section 3.3.

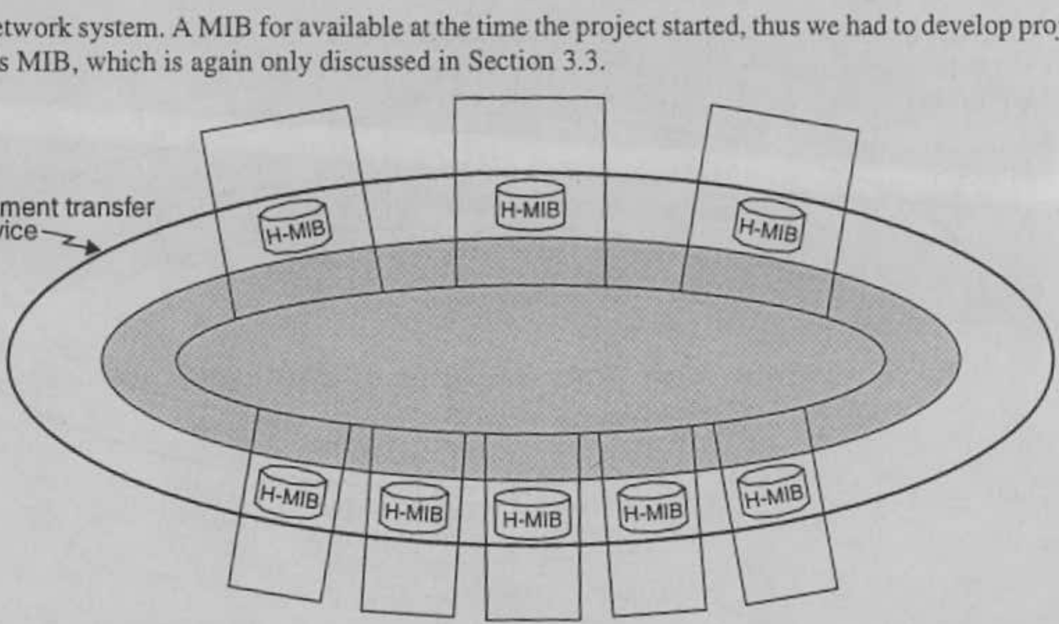

Figure 3: The HTTP MIB (H-MIB)

3 Management information

This Section explains the structure of the three MIBs that were

- the Server MIB (Section 3.1)
- the Access Point MIB (Section 3.2)
- the HTTP MIB (Section 3.3).

3.1 Server MIB

The Server MIB (Figure 5) provides information about the kind of server (the sGeneral number of server accesses (the sAccess group), the documents which (the sDocument group) and the errors that occurred (the sError

The sGeneral group (Figure 6) contains simple variables which holds the server's name, the isation that operates the server, the contact address of the was last initialized and a variable for the supported media types movies). Note that a variable to indicate the location of the already available within the system group of the MIB-II and need group also includes two tables. The first one, the topicTable is provided by this server. The second one, which is needed in server, shows which applications (e.g. an Oracle database) are This table is called the applicationTable and includes the operational status and errors. In fact the table provides similar (SYSAPPL) MIB, for which standardization has just been started applicationTable may therefore be replaced by (parts of) the

The sAccess group informs the manager how the server is being the sAccess group provides three summary figures: the total number bytes received and the number of bytes transmitted (Figure 7). Although the manager could compute these figures from other MIB information, they were of traffic that must otherwise be exchanged for this computation.

The domainTable contains the number of accesses per user domain and where mirror servers are needed. The lastDaysTable shows how accessed during the last couple of days. Like the serTable, the size of this table may be modified by the manager to tables are typical 'TopN' tables; to speed up FrequentUserTable after it receives a corresponding request from set a special *refresh* variable). To avoid superfluous refreshes, an additional variable included to show the time the table was last updated.

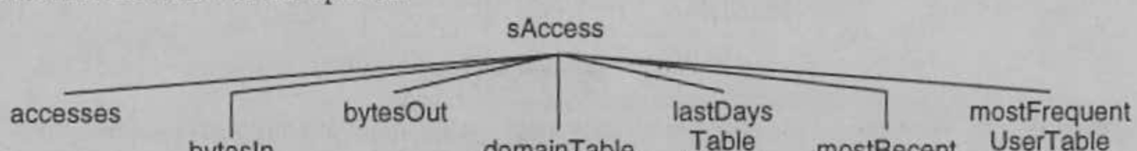

Figure 4: The Access group of the Server MIB

The sDocument group of the Server MIB maintains information ments that are provided by the server. The group consists of a and tableRefresh variables that may be set by the manager. The rate row, which holds the document's name, size, type, time of of accesses and the number of errors.

The sError group keeps track of the local information retrieval (information transfer errors are counted elsewhere). To allow the group is organised as a table. For each error type the table holds a description of the error, the number of such errors and

3.2 Access Point MIB

The Access Point MIB consists of a serviceStatistics group and a serviceStatistics group at the client side keeps track of the at the server side it counts the number of service indications and
The qualityOfService group consists of a delay table, a throughput (Figure 9). The two tables are indexed by the address of the remote system. delay can be determined by measuring the interval between the occurrence of the confirm; at the server side delay values can be or UDP's *echo* function [6][7]. The time-out variable counts how many confirms got lost.

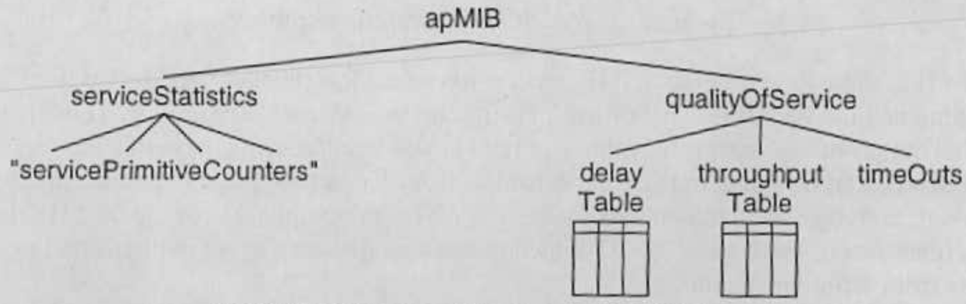

Figure 5: Structure of the Access Point MIB

3.3 HTTP MIB

The HTTP MIB (Figure 6) consists of three groups: one for general information (the group), one for statistical information (the httpStatistics group) Outs group).
The system group contains information such as the vendor of the number, the address of the entity and its uptime. Since the same cable to clients as well as servers, the system group contains a To allow the coexistence of multiple HTTP entities within a single the HTTP MIB is structured into tables; rows within the tables the same HTTP entity.

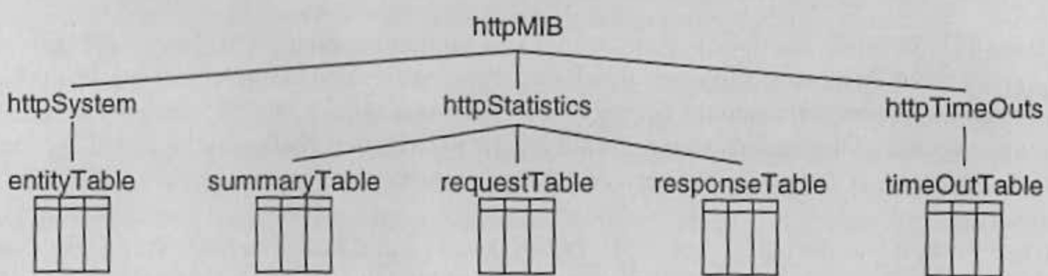

Figure 6: Structure of the HTTP MIB

The summary table within the statistics group provides a quick ance; it shows the total number of requests and responses that (depending on the entity's role), the number of incoming and of errors. Although the manager could compute part of this summary table was included to reduce SNMP network load.

Detailed information, such as timestamps indicating the last found in the request and response tables. These tables have for type a separate entry and can be easily extended to accommodate
The timeOutTable is a TopN table and contains per remote system as the time of the last time-out. The size of the table can be

4 Implementation

After the definition of the MIBs was complete, the MIBs were agent. To test the MIBs, the agent was incorporated into several UNIX systems.

A common way of implementing a SNMP agent within a UNIX UNIX process which interacts with the process that is being The interaction between server and agent processes is usually Communication (IPC) mechanism (left side of Figure 11). Such mechanism allows the agent to *monitor* the server's behaviour by fetching information from the server's behaviour by changing its variables.

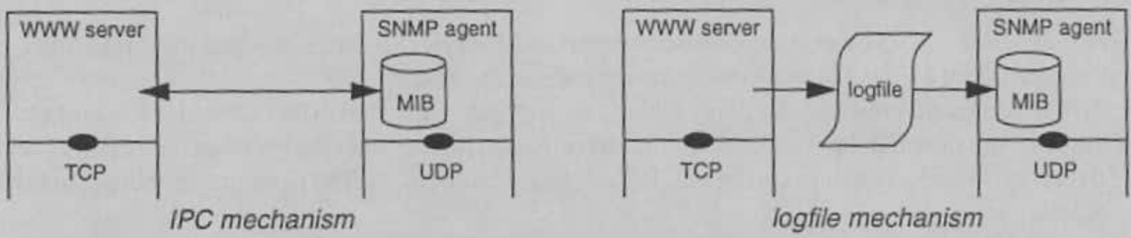

Figure 7: Different implementation approaches

In principle, the IPC strategy allows the SNMP agent to exercise Unfortunately, the server implementations that existed at the time include such IPC capabilities. This implies that we had to modify was clearly undesirable. However, within our proof of concept study the requirement was to *influencing* the behaviour was not a requirement. It is therefore sufficient information exchange mechanism from server to agent, a required. An elegant way to implement this one-way exchange, is to maintained by each WWW server (right side of Figure 11). The format of these logfiles is described in a de facto standard [8], which means that the same agent implementation can be used to manage servers from different vendors.
After the advantages and disadvantages of both approaches were decided to adopt the logfile approach.

method	advantage	disadvantage
IPC approach	• complete control over server (set operations are possible)	• server software need to be modified
logfile approach	• no need to modify server software • can be used with different servers	• server can only be monitored (set operations are not possible)

Figure 8: Comparison between IPC and logfiles

Next to the three WWW MIBs, the SNMP agent should also support the ment of the lower layer protocols, such as TCP and IP. It may be MIBs must be supported too. It was therefore decided to implement [9]. With such agents, the MIB specific functions are implemented the common functions, such as the Basic Encoding Rules (BER), are *master agent*. The master agent and the various subagents are separate UNIX new MIB must be added, it is only necessary to develop or buy a gents, as well as the master agent, remain intact.

To speed up our work, it was decided to use the EMANATE extensible Research [10]. This package includes a master agent, several subagents (for and a subagent development kit, which was used to implement the This subagent monitors the tail of the logfiles, where the server the Server MIB (S-MIB), Access Point MIB (A-MIB) and HTTP MIB from the same logfiles, all WWW MIBs are implemented within a
The implementation of the WWW-subagent has about 12000 lines of generated by the EMANATE subagent development kit. The size of the which is slightly more than the master agent (1.1 Mbyte). The of 5 Mbyte and could return around 8 variables per second.

5 Conclusions

The agent has been installed on a number of European servers for currently being tested. To allow others outside the CEO programme we have also developed a public domain WWW manager and agent. The TKINED [11] and can, together with the Scotty based agent [12], be downloaded from our Simpleweb server [13].

The Server MIB, Access Point MIB and HTTP MIB were presented as where a particular interest existed in the latter MIB. The as one of the starting points for the HTTP MIB working group [14]. To get an agreement on an HTTP

MIB, the MIB as presented in this paper will require a number accommodate the results of the other IETF group which has recently an Application MIB (SYSAPPL-MIB).

The MIBs and prototypes that are presented in this paper, will DESIRE project; which is a TELEMATICS project within the fourth EC.

Acknowledgements

We would like to thank the CEO programme of the JRC, in particular proof of concept study possible. We would also like to thank Mark members of the HTTP mailing list, in particular Carl Kalbfleisch

References

[1] Picoto C., Veiga P.: "*Management of a WWW server using SNMP*", University of Lisbon, Portugal, 1995
[2] Berners-Lee, T., Fielding R. and Frystyk, H.:, "*Hypertext Transfer Protocol -- HTTP/1.0*", RFC 1945, 1996.
[3] Hazewinkel H., Hengstum F.P.H. v., Pras A.: "*Results of the CEO Project - WWW Management*", CTIT Technical Report Series No. 96-18, ISSN 1381-3625, University 1996
[4] Case J.D., Fedor M., Schoffstall M.L., Davin C.: "*Simple Network Management Protocol (SNMP)*", RFC 1157, 1990.
[5] McCloghrie K., Rose M.: "*Management Information Base for Network Management of TCP/IP-based internets (MIB-II)*", RFC 1213
[6] Postel J.: "*User Datagram Protocol (UDP)*", RFC 768, 1980.
[7] Postel J.: "*Internet Control Message Protocol (ICMP)*", RFC 792, 1981.
[8] Common Logfile Format: http://www.w3.org/hypertext/WWW/Daemon/User/
[9] Linde H. v.d.: "*Evaluation and new Specification of the Extensible Agent Protocol*", M.Sc. thesis, University of Twente, 1995
[10] SNMP Research: "*EMANATE Subagent Development Kit*", SNMP Research International Inc., Knoxville, Tennessee, 1994
[11] Schönwälder J., Langendörfer H. "*INED - An Application Independent Network Editor*", In Proc. World Conference on Tools and Techniques for System Arlington, 1993
[12] Schönwälder J., Langendörfer H.: "*Tcl Extensions for Network Management Applications*", In Proc. 3rd Tcl/Tk Workshop, page 279-288, Toronto, 1995
[13] The Simpleweb: http://wwwsnmp.cs.utwente.nl/
[14] HTTP Mailing list: http://www.onramp.net/~cwk/http-mib/email.html