Desktop- vs. Web-based Network Management

[1]
Luca Deri
Finsiel S.p.A.
Via Matteucci 34/b
56124 Pisa, Italy

The advent of powerful desktop environments has greatly simplified the human-machine interaction and allowed people to work with computers using intuitive graphical user interfaces. The desktop is a metaphor through which system resources are accessed and used in a simple and consistent manner. Although every modern operating system comes with a desktop environment, network management applications do not yet take advantage of it.

This paper introduces a novel management paradigm named desktop-based management that enables management of network resources from a desktop environment. It also covers the design and the implementation of SMB_SNMP, a simple desktop-based management system that allows people to manage SNMP resources from a desktop environment. Finally, the desktop-based management paradigm is compared and positioned against the most popular web-based management paradigm.

Keywords: World-wide Web, Network Management, SNMP, Samba.
Topic Area: Human Machine Interfaces.
Paper Type: Research Paper.

Introduction

The introduction of the web has significantly changed the way people work and communicate. Many user interfaces have been modified in favour of the web interface that has been adopted as the privileged way to interact with users. This allowed many different incompatible interfaces to be unified and provided the user a consistent and unique way to work. The main idea is that the web browser is the user interface regardless of the operating system (OS) and the application being used. From one side, this approach has significantly reduced the number of commands and operations that a user had to remember in order to use different applications. On the other hand, differences among OSs have been flattened in favour of the web interface. The *desktop* is a graphical environment that allows users to interact with the operating system by means of icons, windows and mouse. In a few modern desktop environments such as Windows™ 98 [26] and Netscape Constellation [5], the desktop has been integrated inside the web browser and not the other way round. The former desktop environment has been integrated in a web browser hence turned its name into webtop. Prior to this, the preferred way to interact with system resources such as files and printers is through a web browser. Basically every aspect of the operating system has been web-ised even if this process has not been completed because:

1. The OS has been integrated and not adapted to the web.
2. Only the web user interface side has been adopted whereas many other web facilities are missing. In fact, users cannot connect their browser to another networked host running the same OS and then access the host resources (e.g. control panel or network setup) just like the browser does with the local machine where

[1] Email: l.deri@finsiel.it, WWW: http://www.tlcpi.finsiel.it/~deri/.

it runs.

3. The web user interface has often been used in an inconsistent way. For instance, virtually every OS uses the mouse double click to open containers (folders for instance) and execute actions (launch applications) whereas inside the web browser a single click is used.

4. OS resources have not been fully converted to the web interface because only a few of them (basically only the files) are accessible using standard URLs (Uniform Resource Locator) [22].

Network and system management are complicated tasks. In the past years several organisations have attempted to define new tools and paradigm for simplifying management. Probably the most popular activity is web-based management that is the application of web to the management of systems and networks. Although a standard URL notation for network resources has not yet been standardized [6] [10], many network management systems [2] [27] offer today a web interface. Such interface has been selected because it allowed developers to create simple, powerful, client-server applications accessible from virtually every platform. In addition the integration of Java [1] applets inside HTML [24] pages provided the means to create software applications that are portable across platforms, easy to distribute, and accessible through standard web browsers. Beside all the benefits derived from the use of web technology, current web-based applications are not integrated into the desktop, that is the graphical environment daily used by people to interact with computers. However, as discussed, nowadays the web browser has still to be considered as a desktop intruder because its integration into the desktop is very limited and not always consistent.

This paper shows how management applications can profitably take advantage of the desktop environment. In particular, a new management paradigm called desktop-based management is introduced, and a simple desktop-based management application named SMB_SNMP is described. The term SMB stands for Server Message Block Protocol and is part of NetBIOS a networking protocol used by the Windows operating system. Finally, the desktop-based management paradigm is compared and positioned against the most popular web-based management [7] paradigm.

Towards Desktop-based Management

In the management world, the web browser is considered by many people the best user interface for next generation management systems. This is because the use of the web interface has several benefits with respect to non web-based management applications such as:

- Web browsers are provided with almost every OS, and are based on standards such as HTML and HTTP [25].
- Prior to the web client-server architecture, web servers act as central software repositories, reducing maintenance costs whereas standard web browsers can be used as client applications
- Hypertext facilities can be profitably used for on-line help and documentation.
- HTML pages can integrate multiple heterogeneous services and living Java applets.

Beside all these benefits, the use of the web interface as alternative to platform native applications has shown some limitations such as:

1. Limited Operating System Integration
   Contrary to a common belief, the desktop has been web-ised and not the other way round. The consequence is that in desktop environments such as Windows 98 every resource has a unique URL accessible using the browser. Instead it is not possible to access the same resources using a non-web application such as the shell or a text editor. For instance, it is not possible to type on a shell[2] more http://www.kde.org/ or date > file:///tmp/actualDate.
2. No Scriptability and Task Automation
   One of the advantages of shell-based operating systems is the ability to write short programs using a script language such as the shell script, Perl [28] and TCL [19]. With the use of the web browser, every activity has to be performed graphically by filling in forms and navigating through hyperlinks. On the other hand, it is not possible to easily write an application that, exploiting a web-based management system, notifies the user when something bad has happened. This is because the output of those systems (HTML) is meant for human display and not for batch processing.
3. Resource Integration vs. Resource Composition
   HTML allows people to easily integrate etherogenous information and Java applications in a single page. Languages such as JavaScript [8] allow different web resources to interact and communicate in a very limited fashion. Resource composition (software composition) [18] instead, allows developers to compose different resources such as software applications and documents, and produce yet another composable resource.

Basically, most of the problems above arise from the fact that the web browser has been used as a desktop replacement. This is because HTML pages and Java applets live uniquely inside the web browser and every interaction with the OS is mediated by the browser itself. All this has been the driving force for going beyond web-based management, towards a new management paradigm capable of exploiting web technologies while being integrated into the desktop. This novel paradigm is named desktop-based management.

Welcome to Desktop-based Management

Desktop-based management is the activity to manage networks and systems using standard desktop tools (for instance a text editor), methods (drag and drop) and paradigms (trashcan). The principles behind desktop-based management are:

- Every management resource/information must be accessible from the desktop.
- The composition of management information (for instance weekly router usage) should be accessible from the desktop, hence virtually from every application.
- Every management resource should be visible at desktop level (for instance into

[2] Very few programs (for instance kfm part of KDE [15]) accept (a subset of) the URL notation as program arguments.

the filesystem) and accessible from several applications.

The main difference between web and desktop-based management systems is that in the first case network management services and applications are accessible only within the web browser whereas in the latter are available to every application that has access to the desktop. This concept is similar to the Linux [20] /proc [14] filesystem that is a dummy filesystem whose files contain some information about the OS, such as network connections, devices information and logged users. Both desktop-based management and /proc export at desktop level some information that instead would be accessible only using ad hoc applications (for instance a web browser in the case of web-based management).
In order to clarify the differences between web and desktop-based management systems, suppose we want to write a log printer application that is capable of printing all the log records contained inside a log [13] instance part of an OSI [12] agent. A web-based log printer would show a few HTML pages containing the log records and allow users to print those records using the browser printing facilities. Instead a desktop-based log printer would show on the desktop a new icon representing the log that could be printed by dropping it on a desktop printer or viewed by double clicking on it. The goal of both applications is the same (the log records are printed on paper) however a desktop-based log printer is much more flexible because once a resource/service is available at desktop level it can be manipulated with other desktop tools and applications ranging from the shell to powerful scripting languages like Perl. In fact, if the user daily needs to save the content of the log on a disk, in the web-based case the use has to manually open the browser and choose “save HTML file” from the web browser menu, whereas in the second case the icon representing the log could be dragged onto the destination disk and copied as a normal file. Because almost every OS allows repetitive tasks to be performed automatically, in the case of the desktop-based log printer the copy operation could be performed automatically by the OS without any user intervention, whereas in the case of the web a human intervention is probably needed. The previous example shows that web and desktop-based management applications are very similar. Web-based management can be seen as a special case of desktop-based management where the desktop is restricted to the browser. The following section covers the design and the implementation of SMB_SNMP, a simple desktop-based SNMP [21] management system.

SMB_SNMP: a Desktop-based SNMP Management System

The idea to develop SMB_SNMP derived from the author need to manage some SNMP-based devices from different hosts running Unix™ and Windows™. Although SNMP is a relatively simple protocol, command-line SNMP managers often have different command syntax depending on the platform they are running on. In addition, because some devices support proprietary MIBs, all the hosts used for management needed to have these MIBs installed and configured.
NetBIOS [17] is a software interface developed by IBM and implemented in some operating systems such as IBM OS/2™ and Windows™, that allows computers to share resources such as disks and printers. Samba [3] is a public domain implementation for Unix system of the SMB protocol. It allows Unix systems to share Windows disks and printers and vice-versa. SMB_SNMP[3] is a bridge [9] between SNMP and SMB, implemented as an extension to the Samba system. SMB_SNMP represents SNMP MIB variables as files and folders contained under the /SNMP directory tree.

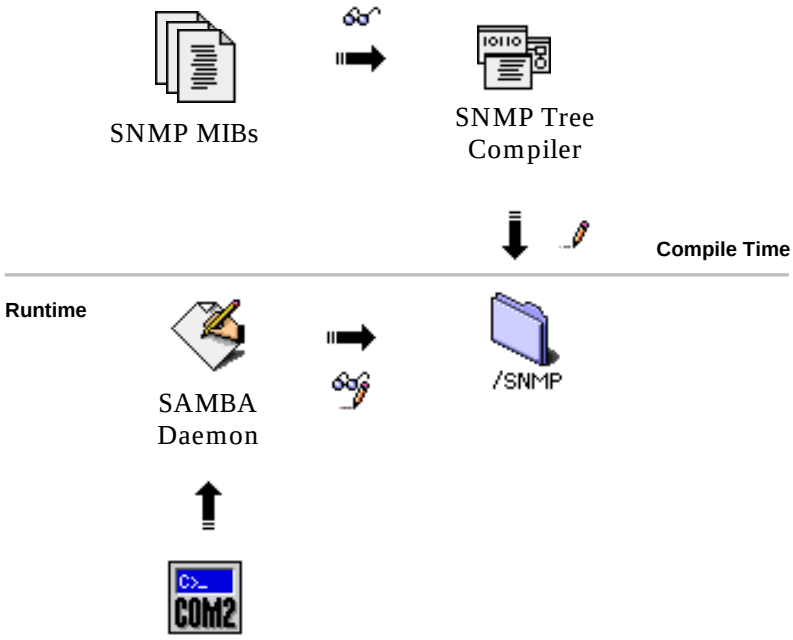

Figure 1 - SMB_SNMP Architecture

Figure 1 illustrates the SMB_SNMP architecture that is composed of a MIB compiler called SNMPTree and an extended Samba daemon. During the configuration phase, SNMPTree reads the available SNMP MIBs and creates a tree of empty files and directories under /SNMP, that represent the tree structure as specified by the MIB variable object identifiers. These generated files take almost no space on the filesystem because they are used just as placeholder for representing the MIB hierarchy. The tree hierarchy is structured as follows /SNMP/SNMP_agent_address@SNMP_agent_port/MIB_variables. For instance `sysDescr.0` contained in the SNMP agent running on host `penguin.finsiel.it` at port 300 is represented as /SNMP/penguin.finsiel.it@300/system/sysDescr.0. If the standard 161 port is used the port value can be omitted. At runtime, the Samba daemon intercepts all the calls for files contained inside the /SNMP directory and handles them properly. Basically, the original Samba daemon has been extended in order to handle the files contained under /SNMP in an appropriate way.

File Operations	Issued SNMP Requests
read	SNMP GET
create/write	SNMP SET
delete	SNMP SET to NULL value

Table 1: File Operations vs. SNMP Primitives

In fact, every time one of those files is accessed, some SNMP requests are issued transparently to the user who does not notice any difference between a real file and one contained under /SNMP. Table 1 contains the mapping of file operations with SNMP primitives. The original Samba daemon handles all the usual file management tasks, and passes the control to SMB_SNMP whenever needed. Errors encountered during

[3] The SMB_SNMP package is publicly available under GPL (GNU Public Licence) from the author home page or can be downloaded from *ftp://sunsite.unc.edu/Linux/system/network/management/*.

SNMP operations (for instance access denied) are mapped to SMB errors and then presented to the user. Files created by SNMPTree have the read / write permission flags set properly according to the relative MIB, hence errors related to users who attempt to set a read only variable are filtered directly by SMB and do not arise at SMB_SNMP level. SNMP community values are specified in a configuration file, and security and access control is enforced by Samba itself that prevents unauthorised users from accessing the /SNMP directory. The /SNMP directory can be mounted as a real directory as shown in figure 2.

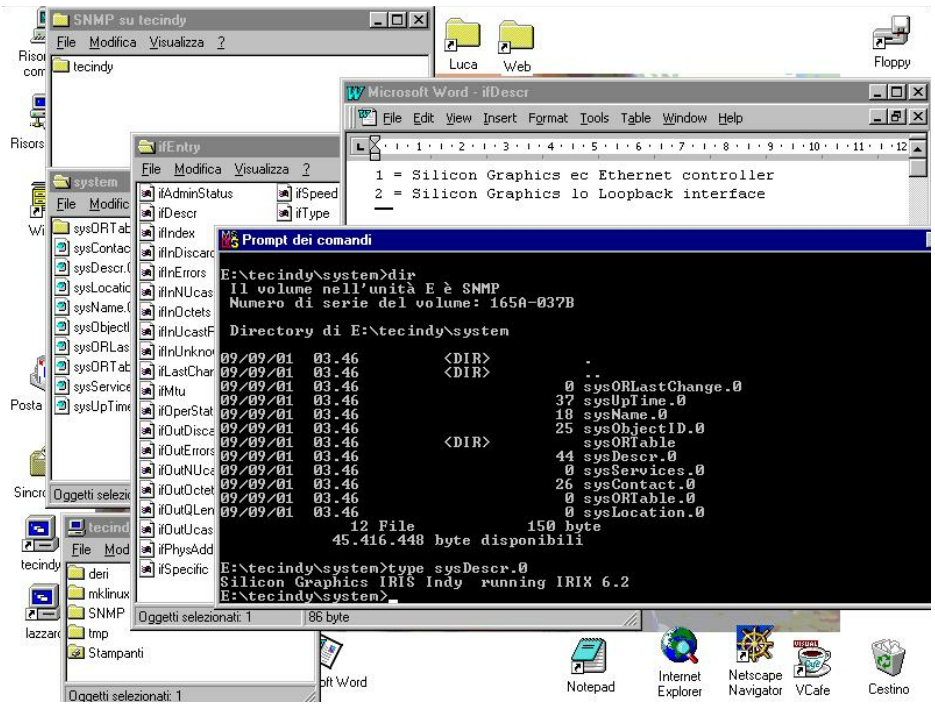

Figure 2 - Simple SMB_SNMP work session

Variables can be set / read / created both using the command line interface and other applications such as text editors. SNMP tables are represented with a few different files that contain the SNMP table in several formats. Default formats include TEXT, HTML and SYLK in order to allow different applications such as web browsers and spreadsheets to handle table values in a native format without any further conversion. Applications can modify / delete / create SMB_SNMP files. Doing this, the values of the corresponding variables in the agent MIB are transparently handled by SMB_SNMP. SNMP traps can be sent by dropping a file containing the trap value over the special /SNMP/SNMP_agent_address/sendTraps directory. In this case, SMB_SNMP transparently sends the trap to the port 162 of the remote SNMP_agent_address SNMP agent.

Samba comes with some tools that allows SMB-based filesystems to be managed on Unix systems. In particular, SMBFS is a kernel module, that allows Linux hosts to mount SMB filesystems that can be then further exported via NFS [23] and AppleShare (both Netatalk [16] and CAP [4] public domain packages can be used for this purpose) hence to be shared with systems running either Unix™ or MacOS™. Thanks to the Samba flexibility, SMB_SNMP enabled users of popular operating systems to have access to desktop-based management at no cost and without having to install additional software on their client machines. This is one of the major advantages of desktop-based management over other management paradigms. Due to the deep integration between the desktop and management resources, desktop-based network

management systems are invisible to users because they are completely integrated into the OS and in order to operate them do not require any additional application such as a web browser.
It is worth to remark that although both SMB_SNMP and Inferno's SNMP agents [11] have some similarities they are very different. This is because:

- Inferno SNMP namespaces have been used uniquely for efficiently implementing SNMP agents with no mean of visibility at desktop level;
- SMB_SNMP mapping between files and management resources have been designed for simplifying the integration between the desktop environment and management resources.

What's new with Desktop-based Management?

Desktop-based management offers several advantages with respect to current management paradigms, including but not limited to:

1. Ease of Use
   Every management detail is hidden by the desktop that presents management resources in a user friendly way. For instance the value of a CMIP object instance can be represented as a file, and a SNMP trap can be sent by dropping a file containing the trap information into a shared folder pointing to a remote host where a SNMP trap daemon is running.

2. Complete Operating System Integration
   Every management resource is embedded into the operating system. This allows virtually every application to be management-enabled. For instance, people who want to draw a graph showing the network utilisation do not need to write / purchase a specialised application. They can use their favourite spreadsheet and write a simple macro that draws a graph by reading some files served by SMB_SNMP.

3. No Specialised Software Needed on Client Hosts
   Contrary to web-based management, desktop-based management systems do not require the installation of additional software applications (for instance a web browser) in order to operate.

4. Better Solutions to Known Problems
   Thanks to desktop-based systems such as SMB_SNMP, known problems such as CMIP / SNMP agent persistency / replication can be solved from a different perspective:
   - A simple backup / synchronisation software is able to store on a tape the CMIP / SNMP agent state and keep two agents synchronised without either human intervention or modification of existing agents.
   - Contrary to CMIP, SNMP does not offer facilities that allow administrator to locate MIB variables containing a given value. Suppose for instance to write an application that prints all the known systems located in Pisa. Using desktop-based systems like SMB_SNMP it is possible to write this application in a of code `grep Pisa /SNMP/*/system/sysLocation.0`.

In other words, desktop-based management is a novel management paradigm that allows non skilled operators to manage network resources from their daily working en-

vironment, namely the desktop. Prior to this, standard applications can be profitably used to manage network resources as shown in the previous example. This allows network administrators to solve management problems in a way much simpler than the one offered by the current generation of management systems.

Future Research: How to Further Improve the Desktop/OS

While developing SMB_SNMP, it became obvious to the author that the desktop environment has not changed very much from the early days of Macintosh™. In practice companies have added colour icons, 3D look, and an integrated web browser but have not radically enhanced the initial desktop vision[4]. The main thing lacking is the ability to turn the desktop into an active entity and not to consider it just as graphical OS shell. In particular some desktop enhancements, useful in the context of network management, have been identified:

1. Active Desktop
   Icons used to represent desktop resource are usually dead entities. For instance, if a file is modified the user is not notified nor custom actions can be triggered automatically. Relevant desktop resources should be represented in a way that actions could be triggered whenever those resources change state. Actions include visual notification (the icon changes or the label flashes) and execution of simple scripts / applications.

2. Desktop-to-Desktop Communication
   Hosts working in a networked environment have limited ways to communicate. The most advanced facilities are offered by Unix systems that allow applications to run remotely and export the display to distant hosts. In general, relevant resources ranging from files to processing power should be sharable. In the context of network management, SMB_SNMP offers a way to share management resources by mapping them to files similar to the approach used by the `/proc` filesystem. Future desktop-based systems may allow users to remotely print the content of a log instance by dragging its icon over a shared printer or drop the icon of a remote OSI agent over a local a disk optimiser that removes unneeded instances.

3. Operating System / Application Introspection
   Every resource / application should offer some facilities that allow users to introspect them. For instance it should be possible to create a dummy file (a sort of super-alias) that points to information such as number of times a specific application has been launched, memory used by a process, free disk space, or current network traffic. Combining this information with an active desktop, simple management applications could be created in a matter of mouse clicks.
   Another example of introspection is the following. If TCP ports would be accessible from the desktop, users could create symbolic links between ports like with plain files. This would allow people to write on the command line the following commands:

   1. `ln -s /dev/ip/tcp/80 /dev/ip/tcp/80@server.company.com` redirects every request received on the local port 80 (WWW) to the port 80 of server.company.com. Ba-

[4] Probably the only really new desktop is Enlightenment (http://www.rasterman.com/).

sically this symbolic link is a simple web proxy.

2. `ln -s /dev/ip/tcp/13@timeserver.company.com /tmp/WhatTimeIsIt` allows users to display the actual time by printing the `/tmp/WhatTimeIsIt` file.

In conclusion, many current desktops/OSs lack functionality that have to be implemented at application level. Enhancing the desktop not only makes the user-desktop/OS interaction smoother, but also allows developers to implement selected applications in a fraction of time by relying on some desktop/OS services. For instance a network topology application might be implemented by following an approach similar to SMB_SNMP: hosts/resources are represented with files, grouped in folders, embedded inside windows whose background represents the place where those hosts/resources are physically located. If every file is an active entity, as proposed above, the file icon could flash or change colour whenever the status of the host/resource represented by the file change. This simple topology application can be implemented in a short time and with very limited development effort.

Final Remarks

This paper presented a novel management paradigm whose goal is to transparently embed management resources into the desktop. By doing this, users can manipulate management resources as conventional files without the need to install or purchase specialised management applications. Although advanced network managers may still prefer to use classic management applications, desktop-based management can be profitably used to allow non skilled people to monitor and interact with selected management resources common tools. Desktop-based management cannot be considered as an alternative to web-based management but as an evolution of it. This is because, in the case of the web, management resources could be manipulated exclusively with a web browser whereas desktop-based management allows every desktop tool (hence the OS) to directly manipulate management resources as the `/proc` filesystem does with other resources. Finally, SMB_SNMP -available free of charge on the Internet- shown that desktop-based management is both feasible and easy to implement, hence validating the overall architecture.

Acknowledgments

The author acknowledges Bela Ban (bba@cs.cornell.edu) for his suggestions and valuable discussions during the design and development of SMB_SNMP. In addition, he would like to thank Finsiel S.p.A. that allowed SMB_SNMP to be freely distributed under the GNU public licence.

References

1. K. Arnold and J. Goslin, *The Java Programming Language*, 2nd Edition, ISBN 0-201-131006-6, December 1997.
2. F. Barillau, L. Deri and M. Feridun, *Network Management Using Internet Technologies*, Proceedings of INM '97, San Diego, April 1997.
3. J. Blair, *Samba: Integrating UNIX and Windows*, ISBN 1-57831-006-7, 1998.
4. *Columbia AppleTalk Package Cookbook v.6.0*, http://www.cs.mu.oz.au/appletalk/cap.html, June 1996.

5. Netscape Communications Corp., *Netscape Constellation*, http://live.netscape.com/comprod/tech_preview/, 1997.
6. L. Deri, *HTTP-based CMIP/SNMP Management*, Internet Draft, November 1996.
7. L. Deri, *Surfin' Network Management Resources Across the Web*, Proceedings of 2nd Int. IEEE Workshop on Systems and Network Management, Toronto, June 1996.
8. D. Flanagan and D. Shafer, *JavaScript: The Definitive Guide*, ISBN 1-56592-392-8, June 1998.
9. E. Gamma, R. Helm, R. Johnson and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, ISBN 0-20163-361-2, July 1995.
10. B. Harrison, P. Mellquist and A. Pell, *Web Based System and Network Management*, Internet Draft, June 1996.
11. Inferno Network Software Solutions, *Implementation of SNMP Agent by Inferno*, (http://www.lucent-inferno.com/Pages/Developers/Documentation/White_Papers/snmp.html), Lucent Technologies, 1997.
12. ISO/IEC, *Information Processing Systems - Open Systems Interconnection - Systems Management Overview*, IS 10040.
13. ISO/IEC, *Information Technology - OSI, System Management - Part 6: Log Control Function*, IS 10164-6.
14. M. Johnson, *Linux Kernel Hacker's Guide*, draft 0.6, Linux Documentation Project, April 1995.
15. The KDE Team, *K Desktop Environment: User Guide*, http://www.kde.org/documentation/en/userguide/, 1997.
16. University of Michigan, *Netatalk User's Guide*, http://www.umich.edu/~rsug/netatalk/, December 1997.
17. IBM Corporation, *NetBIOS Application Development Guide*, S68X-2270-00, April 1987.
18. O. Nierstrasz and D. Tsichritzis, *Object-Oriented Software Composition*, ISBN 0-13-220674-9, 1995.
19. J. Ousterhout, *The Tcl Language and the Tk Toolkit*, ISBN 0-20163-337-X, 1994.
20. R. Petersen, *Linux: The Complete Reference*, ISBN 0-07882-461-3, February 1998.
21. J. Case, M. Fedor, M. Schoffstall and C. Davin, *Simple Network Management Protocol (SNMP)*, RFC 1157, May 1990.
22. T. Berners-Lee, L. Masinter and M. McCahill, *Uniform Resource Locators (URL)*, RFC 1738, December 1994.
23. B. Callaghan, B. Pawlowski and P. Staubach, *NFS v3 Protocol Specification*, RFC 1813, June 1995.
24. T. Berners-Lee, *HyperText Markup Language Specification Version 2.0 (HTML)*, RFC 1886, December 1995.
25. R. Fielding and other, *HyperText Transfer Protocol (HTTP/1.0*), RFC 2068, January 1997.
26. B. Russell, *Introducing Windows 98*, ISBN 1-57231-630-6, September 1997.
27. Free Range Media Inc., *Web-Based Enterprise Management Initiative*, http://wbem.freerange.com/, July 1996.
28. L. Well, T. Christiansen and R. Schwartz, *Programming Perl*, 2nd Edition, ISBN 1-56592-149-6, September 1996.

Biography

Luca Deri is currently working as network engineer at Finsiel S.p.A.. He received his

Ph.D. in Computer Science with a thesis on Software Components from the University of Berne in 1997. He previously worked as research scientist at the IBM Zurich Research Laboratory, and as research fellow at the University College of London. His professional interests include network management, software components and object-oriented technology. His Web page address is http://www.tlcpi.finsiel.it/~deri/.