

Names should mean What, not Where

James W. O'Toole and David K. Gifford

Abstract

Naming is a fundamental issue in distributed systems that is growing in importance as the number of directly accessible systems and resources grow to the point that it is difficult to discover the names of files or objects of interest. Based upon our recent experience building a system that combines the ideas of location and naming we propose a naming architecture called **content names** that is a departure from a traditional tree structured hierarchy. In a **content name** system a name can include a description of what a user seeks in addition to where an interesting file or object might be found. Our experience with an implementation of **content names** in a **Semantic File System** suggests that they will be widely applicable to future distributed system architectures. In addition to our own experience, we have observed a trend towards **content names** in new distributed services, such as the **Archie** system and the **Altair** daemon. We discuss how a **Semantic File System** can be used to implement **content names** and examine its feasibility in the context of a USENET publishing system.

Keywords: naming, file systems, distributed systems, information systems

1 Naming is a fundamental unifying concept

The difference between a collection of networked centralized computing systems and a true confederation of computing systems lies in the ability of the system to provide a uniform way of accessing and naming resources that is machine independent. As distributed systems have grown from single local area networks to world-wide internetworks the logistics of discovering resources of interest have become correspondingly more complex.

The growing complexity of finding information in one such large network, the Internet, has motivated a wide-variety of new tools for naming support [2, 7, 9, 4]. For example, the Archie system [1] provides an on-line index of the files that are available at over 200 public FTP sites. The Alex system [3] provides a uniform naming system along with caching services that span all of the hosts in the Internet, and interoperates with an underlying file system that provides primitive FTP support.

Authors' affiliations: o'toole@lcs.mit.edu and gifford@cs.mit.edu, Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139.

This research was funded by the Defense Advanced Research Projects Agency of the U. S. Department of Defense and was monitored by the Department of the Army under agreement number 2-15-1992.

A significant impact of distributed computer systems has been to enable rapid inter-user exchange of information. To take advantage of the potential for such interchange we need to anticipate the kinds of sharing that should be supported, and tailor our naming system to support these needs.

We believe that a significant new use of distributed systems will be in the area of information publishing, and future naming systems will need to anticipate the requirements of this application. Information publishing is a natural extension of mailing lists, file transfer, and bulletin board systems. At present, the most common way to publish is to announce to a widely-read mailing list the anonymous FTP location of the publication. In the electronic world, as in the physical world, there exist well-known libraries, usually operated as large FTP sites, which archive a wide variety of useful information such as source code, technical reports, and back issues of moderated mailing lists.

In the physical world, a good library or bookstore typically contains reference materials, some of which provide indexed listings of available documents. To publish information, either in the physical or the electronic domain, requires that the information be made widely available and that the cost of locating the information be made very small.

It is natural to expect that as the amount of information that is published in the Internet increases, query-based access will become as important as names that describe precisely the file that is wanted. Query support needs to be integrated into the naming system in such a way that applications can use high performance protocols to access data of interest. For example, if video files are indexed, then applications may desire to use content names to locate video files of interest, but will need to use high performance protocols to access the files for playback. In this note we suggest how such location and access services can be integrated into a single easy to use protocol.

2 Content Names

A Semantic File System provides flexible associative access to stored information by automatically extracting attributes from stored objects. Associative access is provided via traditional file system interfaces and other access methods. The system block diagram in figure 1 shows the basic building blocks of a prototype semantic file system [5].

This semantic file system implementation uses the Network File System protocol [6], but interprets the file pathnames as Content Names. The dynamic interpretation of content names by the file storage system allows users to communicate with each other more easily. Inter-user communications can involve content names, and these content names are correctly interpreted by the storage system to mean **what** files are desired, not exactly **where** those files are located.

For example, content names can be used to quickly create and browse USENET news articles related to solving problems with a new modem

```
% cd /sfs/news/subject:/v32bis/text:/hangup
% ls
```

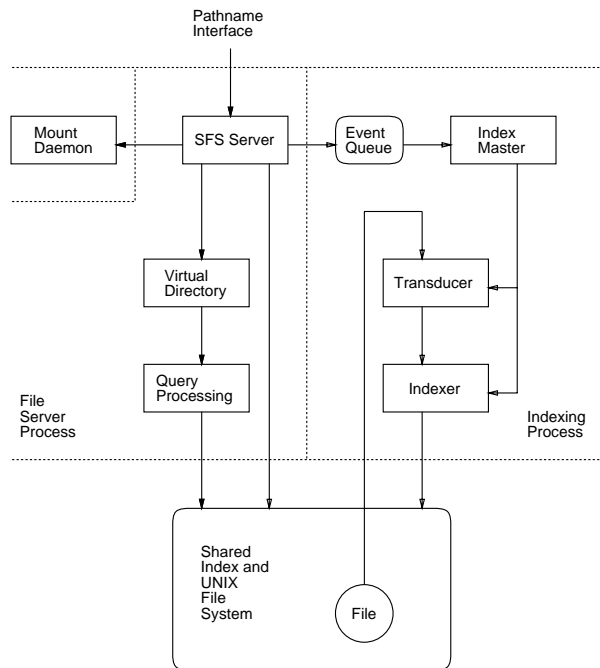


Figure 1: Semantic File System Block Diagram

```

dcom.1      dcom.2      dcom.3      dcom.4
...
uucp.1     uucp.2     zmodem.1
% less dcom.*

```

In the above example, the first command specifies desired properties of USENET news articles, which are then enumerated by the 'ls' command. The desired files are then examined with a simple file browser. The use of NFS permits the content names to be resolved to pathnames via symbolic links. Subsequent access to the files therefore incurs no performance penalty.

Information publishing occurs naturally in a Semantic File System because all stored information is automatically "advertised" via the virtual directory mechanism. A "virtual directory" is a directory which is indistinguishable from an ordinary directory, but is computed on demand by a semantic file system. The virtual directories of a semantic file system provide a convenient way for interested viewers to locate previously published information.

Unlike traditional content-based reference materials such as library card catalogs, Books In Print, etc., the virtual directory method used in semantic file systems combines the means of locating materials of interest with the means of accessing them.

3 Example problem Video USENET

USENET is the largest distributed file system in active use today. The primary service provided by USENET to more than 50,000 participating sites is a shared file system known as the "newsgroups". The newsgroup names form a tree-structured directory namespace. The files stored in these directories are the articles submitted by users.

New data is added to the USENET newsgroups at an aggregate rate of approximately 40 megabytes per day. During a two week period, 20,000 users submit about 200,000 new files from more than 50,000 sites. Measurements indicate that the participating sites have approximately 10 million users, of whom about 2 million actively read articles from the system [8].

The current USENET architecture uses a flooding-broadcast distribution method to replicate published data at participating sites. Because of the high volume of new data submission, most sites rapidly delete data. Once locally stored copies of data have been deleted, users desiring to access the data must use ad hoc methods to locate possible archive sites.

In fact, a non-trivial fraction of inter-user communications in the USENET appears to be repetitions of previous traffic, and efforts have been made to avoid this traffic by widely distributing compendia of "Frequently Asked Questions." We believe that the naming methods used in USENET do not serve the primary needs of its users. Users typically submit articles which describe desired information. This indicates that the naming system used in USENET should use Content Names.

The amount of information published per day in USENET continues to increase rapidly, and the increasing use of graphical and video data will lead to even more explosive growth. Compressed graphics currently constitute about 10% of new USENET data, but this represents less than 1% of the new articles. Any slight increase in the use of photographic data will produce a major increase in total USENET traffic volume. The publication of a few minutes of one good full-motion video segment would probably overload the existing USENET infrastructure.

Introducing content names as a fundamental part of the USENET shared file system offers several potential advantages:

- Provides more convenient access to published articles.
- Encourages the embedding of article relationships in the storage system
- Enables the publication of items too large for immediate broadcast.

Content names could be introduced into USENET in many ways. One attractive possibility is to provide "virtual newsgroups" which use content names. These newsgroups could be accessed via the existing USENET file protocol, NNIP. An NNIP server would dynamically interpret content names to create virtual newsgroups on demand. It is possible that the NNIP could also serve as a network file access protocol for articles which have not been broadcast, with a global naming system similar to Alex used to address specific storage locations.

4 Conclusion Basic Primitives for future distributed publishing systems

The basic requirements of information publishing in a distributed storage system require efficient means for locating published information resources based on content. Storage access protocols which integrate basic content-based primitives can be integrated with network-based global naming and local caching. These basic techniques support information publishing in the client-server paradigm by decoupling the storage and naming features of traditional network file systems.

The semantic file system architecture permits information publishing in a distributed system. Loosely coupled network file system protocols such as AFS and NFS can be used to provide convenient access to both content-attribute information and actual contents. Large items such as digitized video segments can be published without unnecessary copying, because network-wide references can be used to uniquely identify objects in virtual directories as in Alex.

References

- [1] P. Deutsch A. Entage. Archie - An Electronic Directory Service for the Internet. In *Proceedings of Usenix*, January 1992. /alex/ca/mcgill/cs/quiche/archie/doc/archie-usenix92-paper.ps.
- [2] Brent Callaghan and Tom Lyon. The Automounter. In *Proceedings of the USENIX Association Winter Conference*, pages 43-51, 1989.
- [3] Vincent Cate. Alex, A Global Filesystem. In *Proceedings of the USENIX File Systems Workshop*, pages 1-12, May 1992. /alex/edu/cm/cs/sp/alex/doc.
- [4] Ferdinando Gallo, Regis Mont, and Ian Thomas. The Object Management System of PCIE as a Software Engineering Database Management System. In *Proceedings of the Second ACM Software Engineering Symposium* pages 12-15, December 1986.
- [5] D. K. Gifford, P. Jouvelot, M. Sheldon, and J. W. O'Toole. Semantic File Systems. In *Proceedings of the Symposium on Operating Systems Principles*, October 1991.
- [6] Sun Microsystems. NFS: Network File System Protocol Specification. Technical Report Request for Comments (RFC 1094) Version 2, Network Working Group, Sunnyvale California, March 1989.
- [7] B. Clifford Neuman. The Virtual System Model: A Scalable Approach to Organizing Large Systems. Technical Report TR-90-05-01, University of Washington, May 1990.
- [8] USENET news.lists contributors. Traffic reports for USENET. March 1992. /alex/net/uu/ftp/usenet/news.lists/USENET*.
- [9] R. Pike, D. Presotto, K. Thompson, and H. Trickey. Plan 9 from Bell Labs. In *Proceeding of UK UUG*, 1990.