Browsing, Building, and Beholding Cyberspace

New Approaches to the Navigation, Construction, and Visualisation of Hypermedia on the Internet

Keith Andrews

Browsing, Building, and Beholding Cyberspace

New Approaches to the Navigation, Construction, and
Visualisation of Hypermedia on the Internet

Dissertation for the Award of the Academic Degree

Doctor of Technical Sciences

at

Graz University of Technology

submitted by

Keith Andrews

Institute for Information Processing and Computer Supported New Media (IICM),
Graz University of Technology
A-8010 Graz, Austria

September 1996

First Reader:	o.Univ.-Prof. Dr. Dr.h.c. Hermann Maurer
Second Reader:	ao.Univ.-Prof. Dr. Franz Aurenhammer

Das Browsen, Bauen und Betrachten von Cyberspace

Neue Ansätze zur Navigation, Konstruktion und Visualisierung
von Hypermediainformationen im Internet

Dissertation zur Verleihung des akademischen Grades

Doktor der Technischen Wissenschaften

an der

Technischen Universität Graz

vorgelegt von

Keith Andrews

Institut für Informationsverarbeitung und Computergestützte neue Medien (IICM),
Technische Universität Graz
A-8010 Graz

September 1996

Diese Dissertation ist in englischer Sprache verfaßt.

Erster Begutachter:	o.Univ.-Prof. Dr. Dr.h.c. Hermann Maurer
Zweiter Begutachter:	ao.Univ.-Prof. Dr. Franz Aurenhammer

Abstract

The Internet and the World Wide Web form a vast, global information network. This thesis describes new approaches to the navigation, construction, and visualisation of hypermedia on the Internet, as embodied in the Harmony client and authoring tool for the Hyper-G web server.

Harmony's advanced browsing and navigational tools support the concept of *location feedback*, which helps users orient by providing contextual feedback with reference to an explicit structural framework.

Harmony's suite of (remote) authoring facilities includes structure editing, document editing, interactive graphical link editing in all media, attribute editing, interactive uploading, insertion, and deletion, making annotations, and managing user accounts and user groups.

The Harmony 3D Scene Viewer was the first Internet-enabled, 3d hypermedia browser. Its successor, VRweb, is a popular VRML browser with support for collision detection, terrain-following, point-of-interest navigation, and interactive link editing.

Finally, Harmony incorporates two innovative information visualisation tools. The Harmony Local Map is a dynamic, two-dimensional structure map which visualises link and membership relationships. The Harmony Information Landscape is a three-dimensional landscape visualisation, which utilises the three available dimensions to compactly present a *combined* display of both hierarchical structure and hyperlink relationships.

Kurzfassung

Das Internet und das World Wide Web stellen ein gewaltiges, globales Informationsnetz dar. Diese Dissertation beschreibt neue Ansätze zur Navigation, Konstruktion und Visualisierung von Hypermediainformationen im Internet, wie sie im Harmony Client bzw. Editierwerkzeug für den Hyper-G Web-Server integriert worden sind.

Harmony besitzt anspruchsvolle Möglichkeiten zum Browsen und Navigieren. Inbesondere ermöglicht *Location Feedback* den Benutzern eine Orientierungshilfe, indem eine ständige kontextbezogene Rückmeldung erfolgt.

Harmonys Editierwerkzeuge ermöglichen unter anderem das Editieren (auch über das Internet) von hierarchischen Strukturen, Dokumenten, Hyperlinks, Attributen, Annotationen, das Einspielen und Löschen von Dokumenten, und die Verwaltung von Benutzer-Accounts und Benutzergruppen.

Der Harmony 3D Scene Viewer war der erste Browser für 3D-Modelle mit Hyperlinks über das Internet. Sein Nachfolger, VRweb, ist ein beliebter VRML-Browser, samt Collision Detection, Point-of-Interest Navigation und interaktivem Linkeditieren.

Schließlich besitzt Harmony zwei innovative Werkzeuge zur Informationsvisualisierung. Die Harmony Local Map ist eine dynamische 2D Strukturdarstellung der hierarchischen Zugehörigkeit und Hyperlinkverbindungen. Die Harmony Informationslandschaft ist eine 3D Visualisierung, die eine *kombinierte* Darstellung sowohl der hierarchischen Struktur als auch der Hyperlinkverbindungen ermöglicht.

I hereby certify that the work presented in this thesis is my own and that work performed by others is appropriately cited.

Ich versichere hiermit, diese Arbeit selbständig verfaßt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfsmittel bedient zu haben.

Acknowledgements

I am indebted to my friends, colleagues, and students for their help, support, and teamwork over the past five years. Everyone at the IICM and IHM (the Institute for Hypermedia Systems of JOANNEUM RESEARCH) has been ready to provide valuable help and feedback, often at a moment's notice. Above all, Hermann Maurer's leadership, vision, and fund-raising talents, Frank Kappe's technical prowess, and Gerhard Pail's managerial skills combined to make the IICM the ideal place, and Hyper-G the ideal platform, upon which to base my work. Many thanks also to Franz Aurenhammer for agreeing to be second reader of this thesis.

Special mention and thanks go to the members, past and present, of the Harmony and VRweb project teams, and to the current members of my information visualisation team, including: Manfred Brandl, Ingmar Egger, Martin Eyl, Mansuet Gaisbauer, Günther Geiger, Peter Kogler, Gunter Laky, Sieghard Lettner, Bernhard Marschall, Susanne Mayr, Vanessa Mayrhofer, Martin Melcher, Georg Mészáros, Alexander Nußbaumer, Gerbert Orasche, Alexander Passer, Andreas Pesendorfer, Michael Pichler, Peter Pichler, Alexander Rodiga, Karin Roschker, Jürgen Schipflinger, Klaus Schmaranz, Johannes Schmeja, Christine Schuster, Claudia Windisch, and Peter Wolf.

I have benefitted from fruitful discussion with, and have been inspired by, many colleagues from around the world, including: Robert Cailliau, Wolfgang Dalitz, Hugh Davis, Andreas Dieberger, Gitta Domik, Dieter Fellner, Jim Foley, Ed Fox, Nahum Gershon, Ed Grossman, Michael Gruber, Joseph Hardin, Keith Instone, Larry Jackson, Franz Leberl, Mark McCahill, Sougata Mukherjea, Jakob Nielsen, Steve Poltrock, Jenny Preece, Dan Russell, Jodok Schäffler, Wolfgang Schinagl, Ben Shneiderman, and Andy Wood.

Much of the early work on Hyper-G was supported by the Austrian Ministry for Science and Research, by JOANNEUM RESEARCH, and through projects with the European Space Agency. The work on 3d hypermedia and on information visualisation was financed in part by the Anniversary Fund of the Austrian National Bank under project number 5334 "Virtual Information Spaces".

Last, but not least, I would like to thank my wife Elisabeth, and children David and Kristin, for their love and patience while I was travelling and working long weekends, and Rob and Rich Cheese and all my friends in Graz for making it such a great place to live . . .

Keith Andrews
Graz, Austria, September 1996

Credits

- Parts of Chapter 4 are based on papers co-authored with Hermann Maurer, Frank Kappe, and Klaus Schmaranz [Andrews *et al.*, 1995c,e,g].
- Parts of Chapter 7 are based on papers co-authored with Michael Pichler, Gerbert Orasche, Ed Grossman, and Mark McCahill [Pichler *et al.*, 1995; Andrews and Pichler, 1996].
- Part of Chapter 8 is based on a paper co-authored with Michael Pichler and Peter Wolf [Andrews *et al.*, 1996].
- Figure 4.3 was drawn by Frank Kappe.
- Figure 8.2 is copyright ©1996 Ben Shneiderman, University of Maryland, and is used with permission.
- Figure 8.3 is copyright ©1996 Xerox PARC and is used with permission.
- Figure 8.7 is copyright ©1996 Andreas Dieberger and is used with permission.
- Figure 8.10 is copyright ©1995 Andy Wood, University of Birmingham, and is used with permission.
- Figure 8.13 is copyright ©1995 Sougata Mukherjea, Georgia Institute of Technology, and is used with permission.
- Figure 8.14 is copyright ©1996 Tim Bray, Open Text Corporation, and is used with permission.
- Figure 8.15 is copyright ©1996 Communications Research Group, University of Nottingham, and is used with permission.
- Figure 8.16 is copyright ©1996 Communications Research Group, University of Nottingham, and is used with permission.

Contents

List of Figures

List of Tables

1 Introduction

> *"Cyberspace. A consensual hallucination experienced daily by billions of legitimate operators, in every nation, . . . A graphic representation of data abstracted from the banks of every computer in the human system. Unthinkable complexity. Lines of light ranged in the nonspace of the mind, clusters and constellations of data. Like city lights, receding . . . "*
>
> William Gibson, Neuromancer, 1984

In his 1984 science-fiction novel, Neuromancer [Gibson, 1984], William Gibson described a global information network interfaced with a three-dimensional graphical visualisation and coined the term *cyberspace* to describe it. Today, the Internet and the World Wide Web form a vast, global information network. Information visualisation techniques are emerging and powerful graphics hardware is becoming available on desktop computers. In a sense, Gibson's vision of cyberspace is in the making.

In this thesis, I will describe new approaches to the navigation, construction, and visualisation of hypermedia on the Internet — the browsing, building, and beholding of cyberspace, respectively — with which I have been personally involved over the past five years.

The next chapter, Chapter 2, introduces the Internet and describes the information systems which lead up to the "Web" as we know it today. In Chapter 3, I present an overview of hypermedia and hypermedia systems, of which the Web is just one, albeit well-known, example. Chapter 4 describes Hyper-G, the "second generation" web server, developed at Graz University of Technology, which serves as the platform for my work[1].

Chapters 5, 6, 7, and 8 form the body of the thesis and largely describe my original work. Chapter 5 discusses the advanced browsing and navigational tools implemented in the Harmony client and authoring tool for Hyper-G, for which I was project leader until the summer of 1995. Of particular note are the synchronised, correlated navigational views which provide contextual feedback (which I call location feedback) and help users orient themselves in large information spaces.

Chapter 6 covers Harmony's sophisticated structuring, management, and authoring facilities. The notable contribution of Harmony in this respect is its suite of (remote) facilities for structure editing, document editing, interactive graphical link editing in all media, attribute editing, interactive uploading, insertion, deletion, making annotations, and managing user accounts and user groups. Most of these facilities have been available in Harmony since 1993 and 1994 and are only now beginning to become available in other Web packages.

The Harmony 3D Scene Viewer and its successor, the VRweb VRML browser, are discussed in Chapter 7. To my knowledge, my work on 3d hypermedia, combining 3d models and hyperlinking across the Internet, was the first such work world-wide and pre-dates equivalent facilities in VRML (Virtual Reality Modeling Language) by more than two years.

Chapter 8 introduces the field of information visualisation and presents the two innovative visualisation tools developed for Harmony. The Harmony Local Map is a dynamic, two-dimensional structure map to visualise the link and membership relationships in the vicinity of a particular document or collection. The Harmony Information Landscape is a three-dimensional landscape visualisation of the collection structure of a Hyper-G Web server. Particularly significant is the *combined* display of both hierarchical structuring and hyperlink relationships in *one* visualisation, utilising the three available dimensions to compactly present both kinds of structure.

Finally, Chapter 9 discusses current trends, describes work in progress, and outlines some of my ideas for future work and research.

[1] There is a commercial version of Hyper-G called HyperWave, but I will generally use the term Hyper-G, except when referring explicitly to the commercial version.

2 Information Systems and the Internet

This chapter gives an introduction to the Internet and Internet information systems. An earlier version of this chapter was published as Chapter 2 of *HyperWave: The Next Generation Web Solution*, Addison-Wesley, May 1996 [Andrews, 1996d].

2.1 The Internet

The Internet, the worldwide computer network, now connects almost 12.9 million individual computers (July 1996) with a growth rate of around 100% per year [Network Wizards, 1996]. Estimates of the number of regular users of the Internet vary from 20 million to 50 million. If current growth rates were to continue, the whole of the world population would be connected by the year 2004!

Around 18 Terabytes of information (18 000 000 000 000 bytes) traversed the main US backbone of the network, the NSFNET, in the month of November 1994 [Merit NIC, 1996], a figure which was also double that of the previous year[1]. These statistics highlight the immense social and economic significance which the Internet is attaining. More and more businesses are realising the enormous potential of the Internet: in the summer of 1994, the number of commercial sites in the United States overtook the number of educational sites and this trend continues today.

Over the past couple of years, with increasingly powerful desktop computers and heavy investment in network bandwidth, attention has shifted significantly from traditional Internet services such as electronic mail and file transfer to more appealing, but resource-hungry, information systems and interactive services, which now account for more Internet traffic (in bytes) than any other Internet service. The World Wide Web has made the Internet into a household name and turned browser software into a commodity.

The History of the Internet

The Internet began life as a research project of the US Department of Defense's Advanced Research Projects Agency (ARPA). The original purpose was to build a communications system capable of withstanding partial fallout of nodes during a nuclear attack, through the use of automatic re-routing. The first remote terminal connection from the University of California at Los Angeles (UCLA) to the Stanford Research Institute (SRI) was demonstrated in November 1969. From these humble beginnings grew the ARPANET, a fully operational computer network, based on a host-to-host protocol called the Network Control Protocol (NCP), and running services such as remote login (telnet) and file transfer (ftp). Electronic mail (email) was added as an afterthought, after two programmers decided to send each other messages as well as data [Lynch and Rose, 1993].

By 1981 there were some 200 sites on the ARPANET. In order to link in other vendors' physical networks into one seamless network, a new standardised protocol, the Transmission Control Protocol/Internet Protocol (TCP/IP), was developed. On "flag day," 1 January, 1983, the whole ARPANET was switched from NCP to TCP/IP protocol and the foundation for the Internet was laid. The inclusion of the TCP/IP protocol suite into Berkeley UNIX around the same time and its subsequent adoption by the vendor community gave a major boost to the Internet.

In March 1986 the Internet had around 3000 sites, and the US National Science Foundation (NSF) initiated development of the NSFNET, connecting six NSF supercomputer centres, to provide

[1] In December 1994, NSFNET traffic began migrating to the new network architecture, for which no comparable statistics are available.

Time	Stage of Development
Early 1970s	ARPANET, US military network, designed to withstand partial fallouts
Early 1980s	TCP/IP introduced to connect diverse networks together; Ethernet LANs and workstations become popular, Berkeley UNIX supports TCP/IP
Late 1980s	NSFNET backbone
1992	Internet Society founded as ultimate authority
Today	“The Net”, a network of networks

Table 2.1 The development of the Internet.

a major backbone communication service for the Internet. By late 1989, the Internet had grown to over 150 000 sites in North America, Europe, South America, Australasia and beyond.

The Internet Society (ISOC) was founded in January 1992 as a non-profit, voluntary body to oversee development of the Internet. It appoints a council of elders, the Internet Architecture Board (IAB), which coordinates various sub-committees. The Internet Engineering Task Force (IETF) develops new standards and recommends them to the IAB for approval.

Table 2.1 summarises the historical development of the Internet. Today, the Internet is a vast, self-regulating network of national, regional, campus and private networks based on the TCP/IP protocol suite and various connecting gateways. This “network of networks” (currently more than 130 000 individual networks) is funded bottom-up – everyone pays for their part. Although many governments currently finance Internet connectivity for their educational and research institutions, the sheer demands for connectivity mean this is now changing to more direct forms of accounting. Indeed, the Internet is becoming so pervasive that it is often referred to simply as The Net. The recent popularity and visibility of the World Wide Web on the Internet has introduced the term The Web, embracing the sum of Internet services and becoming more or less synonymous with “The Net” and the Internet.

Internet Protocols

The Internet is a packet switched network. The underlying protocol, the Internet Protocol (IP), provides envelopes or packets for up to 1500 bytes of data. Each packet is stamped with a unique source and destination address. Packets are passed through the Internet from router to router until they reach their destination. Like letters through the postal system, they occasionally get damaged or lost or arrive in a different sequence.

The Transmission Control Protocol (TCP) builds on the basic service provided by the IP. TCP breaks messages larger than 1500 bytes into a sequence of small IP packets. On the receiving side, packets are collected, placed in order and their data extracted. Missing and corrupted packets are retransmitted. TCP creates the appearance of a dedicated link between sender and receiver, which Internet applications use to communicate and exchange data. To this end, TCP also defines the concept of contact ports to be used by individual applications or services. The more common services use permanently assigned, “well-known” port numbers by default (but usually can be configured to use an alternative port). Port numbers are assigned by another Internet body, the IANA (Internet Assigned Number Authority).

Internet Addresses

Every computer (or host) on the Internet has a unique address; in the current version of IP (IPv4) Internet addresses are 32-bit numbers, in four 8-bit parts, for example `129.27.153.10`. They are split between the network and host parts: so-called class A networks have a 1-byte network and a 3-byte host part (providing for up to 16 777 214 hosts per network), class B networks a 2-byte network and a 2-byte host part (providing for up to 65 534 hosts per network), and class C networks a 3-byte network and a 1-byte host part (providing for up to 254 hosts per network). Computers connected to the same physical network generally all have the same network part plus their own unique host part.

Theoretically, 32 bits provide over 4 billion addresses. However, the address allocation method in IPv4 is rather wasteful. For most organisations and companies, a class C network with 254 hosts is too limiting, leading to high demand for class B networks. Consequently, class B address space is almost exhausted. Furthermore, whereas IPv4 today serves what might be called the computer market (connecting computers), there is in future expected to be huge demand for IP addresses from all manner of portable and household devices, from telephones and television sets to toasters.

To solve this problem, and to introduce numerous other improvements, a new version of IP (called IPv6 or IPng) is being specified. IPv6 will have 128-bit addresses and a hierarchical addressing mechanism. This is an extremely large address space, corresponding to approximately 665 570 793 348 866 943 898 599 addresses per square meter of the Earth's surface!

Domain Names

Since human beings are notoriously inadequate when it comes to remembering sequences of numbers, a multi-level, hierarchical naming scheme, the Domain Name System (DNS), is available as an alternative way of specifying computers on the Internet. Each level is called a domain and there may be any number of domains in a name (although usually no more than five). Domains in a name are separated by periods and become successively smaller from right to left. For example, the computer named `aw.com` is the machine serving as the main access point to the Addison-Wesley Publishing Group and is in the commercial domain. The machine's Internet address is `192.207.117.2`. The computer sitting on my desk is `fiicmds06.tu-graz.ac.at`, a DECstation at Graz University of Technology, in the academic domain, in Austria. Its address is `129.27.153.10`. There is a unique mapping from domain name to Internet address, but not vice versa. Several domain names (aliases) may, and often do, refer to the same physical machine. Aliases are particularly useful for naming services in a machine-independent fashion. For example, `ftp.aw.com`, Addison-Wesley's anonymous ftp archive, is in fact an alias for `aw.com`, but publicising the alias rather than the real domain name or Internet address means that the ftp service can be moved to another machine simply by changing the address for the alias, without causing any disruption in service.

The top-level (rightmost) domains currently in use consist of the original six domains: `com` (commercial), `edu` (educational), `gov` (government), `mil` (military), `org` (organisations), and `net` (network resources), plus around 150 of the 300 or so international two-letter country codes such as `uk` (United Kingdom) or `at` (Austria). Responsibility for the registration of domain names at the second level has been delegated by the IANA to regional Network Information Centres: the RIPE NCC in Europe, the AP-NIC in the Asia-Pacific region and the InterNIC for the USA and the rest of the world. The name space at each lower level is controlled by the appropriate body, for example domains beneath `tu-graz.ac.at` are assigned by the Computer Centre at Graz University of Technology.

A common misconception is that the parts of a domain name correspond directly to parts of the Internet address – they do not. DNS name servers are used to look up the address corresponding to a particular domain name: if the local DNS server does not know the address itself, it asks the DNS server of the top-level domain for the address of the DNS server responsible for the next lower domain, and so on down the domain hierarchy until the actual Internet address is found.

There has been huge recent demand for domain names, particularly in the commercial domain. In June 1996, InterNIC registered almost 400 000 new second-level names in the `com` domain, despite the introduction of a $50 a year registration fee in September 1995. Major companies which overlooked the emergence of the Internet as a new medium have found their favoured domain names already taken by unknown rivals. Speculators are registering likely names in advance, and trademark lawyers are working overtime to reclaim them. Joshua Quittner, a journalist with Wired magazine, registered `mcdonalds.com` for fun in 1994 [Quittner, 1994] and asked readers to send in suggestions as to what to do with it. It has since been transferred to the hamburger chain.... To ease the demand for names in the `com` domain and to open up the market to competition, IANA is planning to allocate dozens of new top-level domains to several new registration bodies by January 1997 [IANA, 1996].

2.2 Basic Internet Services: Remote Login, File Transfer, Email and Network News

Many Internet applications are two-part, client-server applications. Client programs are started by users on their local machines. The client establishes a connection across the Internet (usually via TCP) to a server on the remote host machine via a particular port. The host/port pair uniquely defines a particular server. If the server is not running, or is listening to another port, no connection can be made. Communication follows according to an agreed protocol for that application; often multiple implementations of a particular service are available.

The four traditionally most widely used Internet applications are remote login, file transfer, electronic mail, and network news. The remote login tool telnet is a client-server application used to log into another computer on the Internet. It provides a basic, terminal-style interface to the remote machine.

The file transfer tool, ftp, is another client-server application, used to retrieve files from a remote machine. Two kinds of access are possible: identified (with account and password) and anonymous (no account or password required). Most public archive sites operate anonymous ftp servers. Until early 1995, ftp data accounted for by far the largest amount of traffic on the Internet (about one third of the total traffic in bytes). A related service, Archie (discussed in Section 2.3), provides an indexing facility for anonymous ftp archives, making it easier to find ftp resources.

Electronic mail (email) is arguably the most popular of all network services, and is not restricted to just the Internet but can be exchanged via gateways with other networks like BITNET and UUCP. Email is not an “end to end” service like the previous applications, but a “store and forward” service: mail is passed from one machine to another until it reaches its destination. There are many mail programs; most of them use the Simple Mail Transfer Protocol (SMTP). On the Internet, SMTP is implemented atop TCP. Commonly supported features include aliases, forwarding, carbon copies and automatic reply. Mailing lists allow mail to be automatically directed to particular user groups. The MIME (Multi-purpose Internet Mail Extensions) specification defines how to include multimedia objects in pieces of email, and ftp-to-email gateways allow files to be retrieved from ftp servers by email.

Network News is the Internet equivalent of a discussion group or bulletin board. News is organised hierarchically into broad topics called newsgroups dealing with particular themes, for example `comp.lang.c++` for the C++ programming language. In general, anyone can *post* a message to a news group. The USENET comprises a set of voluntary rules for passing and maintaining newsgroups in one of seven categories (`comp` being computer science). In addition, many “alternative,” commercial, and local newsgroup hierarchies also exist. News is passed from news server to news server according to bilateral agreements between server administrators using the Network News Transfer Protocol (NNTP). One of the main problems with news is the overwhelming amount of material – a typical news server might carry several thousand newsgroups and receive hundreds of megabytes of news per day! Another problem is that of “spamming”: postings to newsgroups which are off-topic, advertising, or simply rubbish. Some newsgroups are *moderated* for this reason; a human being scans and approves each posting before passing it on to the newsgroup.

Tool	Purpose	Protocol	Assigned Port
telnet	remote login	TELNET	23
ftp	file transfer	FTP	20 (data), 21 (control)
mail	electronic mail	SMTP	25
news	network news	NNTP	119

Table 2.2 The four basic Internet services.

Table 2.2 summarises the four basic Internet services. These tools are now in fact so common, they are almost taken for granted by most Internet users, especially given the rapid spread of multifunctional Web browsers incorporating ftp, email, news, and telnet software.

2.3 Archie

As increasing numbers of ftp archive sites came online in the late 1980s, it became increasingly difficult for Internet users to know which resources and information were available at which sites – word of mouth no longer sufficed. Archie was a first attempt to provide a comprehensive index to information resources on the Net [Emtage and Deutsch, 1992]. Developed at McGill University in Canada, it was made widely available in December 1990. Today, Archie tracks the contents of around 1300 (known) anonymous ftp archive sites by actively visiting each site about once a month and updating its index with the name of every file and directory in the archive. For a site to be indexed, Archie's maintainers have to be informed of the site's existence by email.

Users can perform queries on one of the public Archie servers (there are about 25 publicly accessible Archie servers worldwide) in a number of ways: by telnet, by email, by using a native Archie client and more recently through a WWW gateway to Archie. In January 1992, the creators of Archie founded Bunyip Information Systems to market and develop Archie and the Archie server software became a commercial product, although Archie searches continue to be free of charge.

2.4 WAIS

WAIS, or Wide Area Information Servers, began in 1989 as a joint development by Brewster Kahle of Thinking Machines Corp., and colleagues at Apple Computer and Dow Jones to provide online access to the *Wall Street Journal* [Kahle *et al.*, 1992]. WAIS is a client-server system which supports content-based search of (previously) indexed databases. WAIS is not a general-purpose information system, but purely a search engine – it provides no other means of structuring or navigation.

WAIS clients provide users with the ability to query and retrieve information from remote databases anywhere on the Internet. A WAIS query is typically a simple list of words separated by spaces; WAIS supports "natural language" queries in the sense that a naturally phrased question can also be treated as a simple list of words. On receiving a query, the WAIS server[2] responds by looking up the search terms in its index, calculating a weighted score for each document containing the search terms, and returning a ranked list of documents to the client. Clients present the results to the user using a normalised score of 1000 for the best matching document. Users, through their client software, can then request that a particular document be sent in full.

[2] In WAIS terminology, servers are usually called "Sources" and queries "Questions".

WAIS is not restricted to text documents. Any body of documents which can be indexed (for example, by file name, or by associated attribute fields or descriptions) can be made available through WAIS, including images, PostScript files, video clips and so on.

A unique feature of WAIS is its support for relevance feedback: (parts of) text documents returned by a search which are deemed by the user to be particularly relevant can be used as input for a further search, in effect refining the search by asking WAIS to look for further similar documents. Newer versions of WAIS also support stemming (the automatic truncation of search terms), Boolean expressions, partial word searches, and stop-lists (the automatic exclusion of non-productive filler words).

A problem for users of WAIS is to know which servers exist in the first place. The solution provided was the directory-of-servers database maintained (originally) by Thinking Machines, which contains entries describing all registered WAIS servers. This server can be searched to find other servers with appropriate databases, resulting in a two-level search strategy. Since manually recording server names or repeatedly searching the directory of servers would rapidly become tedious, most WAIS clients provide facilities for saving and re-using personal lists of servers. Most clients also support personal lists of queries, so that accurate and successful queries can be periodically restarted with minimal effort.

Most WAIS servers run on UNIX machines, although server software is also available for VAX/VMS. The WAIS package includes an indexing utility, `waisindex`, which extracts index terms from a body of documents and creates an inverted index structure (an alphabetical list of terms, each with an associated list of documents in which the term appears). A sophisticated data structure (a two-level B-tree) provides fast access to the index. The `waisindex` utility automatically recognises and indexes a wide range of text formats. Other document types, such as images, can be indexed by file name or `waisindex` can be customised to index associated descriptions, attributes, and so on. For standard text documents the inverted index takes up approximately the same amount of disk space as the original data set. Once an inverted index has been created, `waisserver` is started to serve client requests.

WAIS clients and servers communicate using a binary protocol derived from the Z39.50-88 (1988) ANSI Draft Standard for bibliographic retrieval and by default WAIS uses port 210, the port assigned to Z39.50. Communication is stateless, no state being retained by the server between transactions with a client. The Z39.50-88 standard was superseded in 1992 by the much improved, but incompatible, Z39.50-92 protocol, which has now itself been superseded by Z39.50-1995. In the summer of 1992, WAIS Inc. was formed to develop WAIS commercially, and support and development of the public-domain WAIS (called freeWAIS [CNIDR, 1994]) passed to CNIDR (the Clearing house for Networked Information Discovery and Retrieval). Both WAIS Inc. and CNIDR have developed fully compliant Z39.50-92 servers. In deference to the popularity of the WWW, these WAIS servers are also equipped with WWW gateways, allowing users of WWW browsers to search WAIS servers.

Nowadays, the original WAIS protocol is no longer widely used, having been superseded by Z39.50. The free version of the WAIS indexing software, freeWAIS, is still widely used to index web sites and provide searchable interfaces to them. Indicative of the commercial importance of powerful search engines, WAIS Inc. was acquired in May 1995 by the largest US online service provider, America Online and then in June 1996 by Fulcrum Technologies, Inc.

2.5 The Internet Gopher

The Internet Gopher, or Gopher for short, was started in 1991 by Mark McCahill and colleagues as a Campus-Wide Information System (CWIS) at the University of Minnesota [McCahill and Anklesaria, 1995]. It is a client-server TCP/IP application which provides seamless, menu-like access to information resources on multiple hosts. The name Gopher comes from the American nickname for natives of Minnesota: it is actually a burrowing rodent the size of a large rat. Since Gopher's purpose is to "go for" things across the Internet, its name was born!

A Gopher client provides the user with a view into Gopherspace presented as a hierarchy of menus and document nodes, very much like a (virtual) file system of directories and files. Although the menu presentation gives the impression of a tree, Gopherspace is in fact a graph containing many loops, since sub-menus can appear in multiple places. Typically, users begin navigation at the top of the tree for a particular server and traverse down to leaf nodes containing actual data, which may be texts, GIF images, audio files and so on.

Sub-spaces on remote Gopher servers can be seamlessly incorporated by simply including references to them. Gopher servers usually have a menu of other Gopher servers somewhere in their tree. Access to these remote servers is totally transparent to the user; the Gopher client simply fetches the next selected sub-menu from the appropriate server. Special nodes called index nodes provide the interface to search engines at particular levels of the tree. They prompt the user for a search string and then present a virtual menu containing the list of matching items. Other special leaf nodes include telnet sessions and binary archives.

As with WAIS, a meta-level of access became necessary as the number of Gopher servers increased and users lost track of which servers provided which resources. A special set of Gopher servers, the so-called Veronica servers, provide a keyword search of all menu titles in Gopherspace. The index used by Veronica servers is constructed by the Veronica Harvester, a robot which periodically (every 7–14 days) tree walks the whole of Gopherspace collecting menu titles (similar to the way Archie indexes ftp servers).

Gopher clients are available for most common platforms. Most clients provide support for telnet sessions, text and multimedia documents (images, audio, video and so on), the latter usually by starting external viewer programs. Clients also commonly provide a history list of nodes visited during the current session, and bookmarks (pointers left by the user) to return to in future sessions. Nowadays, multi-protocol Internet browsers such as Mosaic, Netscape, and Internet Explorer speak Gopher as well as FTP, HTTP, and other protocols.

The standard UNIX Gopher server `gopherd` is very straightforward to use. It exploits the hierarchical nature of the UNIX file system and implements a simple mapping from UNIX directories to Gopher menus and UNIX files to Gopher documents. Configuration files are used to specify the display names of Gopher entries and to represent entries pointing to other Gopher servers.

An enhanced version of the Gopher protocol, Gopher+, was published in July 1993 [Anklesaria *et al.*, 1993]. It contains extensions for attribute metadata, interactive queries, and additional document types, but was never widely implemented in Gopher clients outside of the University of Minnesota. GopherVR is a three-dimensional user interface (client) to a Gopher server [McCahill and Erickson, 1995] (see Section 8.5). An extension to GopherVR utilises the attribute metadata of a Gopher+ server to assign positional coordinates to the members of a menu [Iacovou and McCahill, 1995].

Gopher owed its popularity to the ease with which a server can be set up and populated with data, particularly under UNIX. The availability of both client and server software for all major platforms and Gopher's interoperability with WAIS and ftp (servers) and WWW (clients) also contributed to its widespread penetration: in January 1995 around 5000 Gopher servers were registered by Veronica. Gopher's hierarchical menus make it easy both to structure and find information. However, Gopher is almost purely hierarchical in nature: it provides no inherent support for hypermedia links and search facilities require explicit configuration. Although the technique of serving HTML pages from a Gopher server does combine hierarchical structure with hyperlinks [Lindner, 1995], Gopher seems now to have been eclipsed by the World Wide Web.

2.6 The World Wide Web

The World Wide Web project (WWW or W3 for short) was initiated by Tim Berners-Lee and Robert Cailliau at CERN (European Laboratory for Particle Physics, the acronym originates from the French name Conseil Européen pour la Recherche Nucléaire), in Geneva in 1989, originally as an information system for the particle physics community [Berners-Lee *et al.*, 1994; Cailliau, 1995]. WWW is a distributed, heterogeneous, hypermedia information system: hypertext links embedded in text documents can be followed to access related textual or multimedia (image, audio, video, . . .) information.

The World Wide Web is defined by three key specifications: HTML, HTTP, and URL. HTML (the HyperText Markup Language) defines how text is marked up with tags to define certain constructs (such as emphasised text, an enumerated list, or a hypertext link). HTML is an SGML-conformant mark-up language; the final presentation of a text document depends on how individual WWW clients render particular HTML constructs. HTTP (the Hypertext Transfer Protocol), the WWW's client-server protocol, uses the MIME Internet email conventions to encode header information associated with documents. Clients send a request message and servers reply with a response message, in a single stateless transaction. Through its URL (Uniform Resource Locator) mechanism, WWW cleverly encapsulates other Internet protocols: by enabling hypertext links to point to any document on any WWW, Gopher, or ftp server worldwide, as well as network newsgroups or telnet sessions, "the Web" has embraced and become almost synonymous with "the Net." In October 1994, the World Wide Web Consortium (W3C) [W3C, 1996] was formed to develop WWW standards and reference code; it currently (August 1996) has around 150 member organisations.

Although first made available in August 1991, WWW only really gained prominence in early 1993 after the release of NCSA's graphical WWW client, Mosaic [Andreessen, 1993]. Mosaic's intuitive point-and-click interface and glitzy inline images made WWW accessible to a wide and diverse user community. This trend was accelerated with the release in October 1994 of the Netscape Navigator [Netscape, 1996]. Co-founded by the original author of Mosaic, Marc Andreessen, and the founder of Silicon Graphics, Jim Clark, Netscape Communications Corporation [Netscape, 1996] has been perhaps the driving force behind the Web's development upto now. Although dozens of WWW browsers are available for all major computer platforms, Netscape still controls about 70 to 80% of the browser market.

The majority of WWW servers, including the two most popular, Apache [Apache, 1996] and NCSA httpd [NCSA, 1996b], simply serve HTML files from the local file system, with only rudimentary access control, little or no information management, and no integrated search facilities. However, the Web's common gateway interface (CGI) allows WWW servers to start arbitrary application programs, for example linking into external databases or implementing complex search algorithms. More sophisticated, commercial servers are also available, one common approach being to put a WWW front end onto an existing information or database system. Professional HTML authoring tools are also now available for all major platforms.

On what became know as "Pearl Harbour Day", 7th December 1995, Microsoft, which had until then almost completely ignored the Internet, announced a complete U-turn of previous strategy [Microsoft, 1995]. In just six months, Microsoft's Internet Explorer [Microsoft, 1996] browser and Internet Information Server [Microsoft, 1996] have become a serious challenge to Netscape's market dominance.

As WWW growth exploded in 1994 and 1995, the problem of finding resources on the Net became particularly acute and was exacerbated by the WWW's reliance on hypertext links and its lack of meta-search facilities. Web crawlers, programs which relentlessly traverse the links of the Web to build centralised indexes of the contents of Web sites, such as Lycos [Lycos, 1996] and Alta Vista [AltaVista, 1996], and manually maintained catalogues of sites such as Yahoo [Yahoo, 1996], have alleviated the situation somewhat, but neither approach is really able to keep up with exponential growth rates.

Version 1.0 of the Virtual Reality Modeling Language (VRML) [Pesce, 1995; Eubanks *et al.*, 1996], a 3d file format for modeling objects and worlds on the Web, was specified in June 1995. VRML allows 3d objects and visualisations to be transferred across the Web in a platform-independent way. Version 2.0 has just been finalised and includes support for scripting and behaviours. VRML is covered in more detail in Chapter 7.

The introduction of Java [Niemeyer and Peck, 1996], a secure, platform-independent, object-oriented language which executes on a "virtual machine", by SunSoft (a division of Sun Microsystems) [JavaSoft, 1996] has brought interactivity to the Web in the form of downloadable mini-applications, or "applets", running in the Web browser. Well-defined, standard APIs allow applets to access standard libraries (often written in native code and optimised for the specific platform on which the virtual machine is running) to perform windowing, network access, 3d graphics, etc. in a platform-independent way. The potential applications of Java applets are vast. Even more intriguing are the possibilities for Java "servlets", small pieces of code which are uploadable to a server to carry out useful work on behalf of the user. To protect against malicious code, applets and servlets have especially strict security controls placed on them by the virtual machine in the browser or server in which they run. However, Java is in fact a generic programming language which can be used to write any kind of application (and not just applets) – applications, in general, do not have or need the same kind of security restrictions. In all, Java is well positioned to succeed C++ as the programming language of choice for the coming decade.

Despite its broad appeal, the World Wide Web has a number of limitations. It does not provide any information structuring facilities beyond hyperlinks; its links are one-way (in general, there is no way of determining which other documents refer to a particular document, leading to inconsistencies when documents are moved or deleted, resulting frequently in "dangling links") and embedded within documents (there are no links from documents such as MPEG video clips, because there is no standard way of embedding links within an MPEG stream). Most WWW servers have no in-built search facilities, but rely on external search engines such as WAIS, leading to patchy server-by-server provision of search facilities by individual sites and no real-time cross-server searches (searches in centralised Web indices such as Alta Vista are of course available). The flexibility provided by CGI is achieved at great cost: the uniformity of the interface disappears, different WWW servers behave differently – resulting in the "Balkanisation" (to quote Ted Nelson) of the Web into independent "WWW Empires." Also, there is little support for the maintenance of large amounts of information, so it is not uncommon to see several WWW servers within a single organisation, each with a fundamentally separate interactive context. The Web today is still very much "read-only," in the sense that information providers prepare data sets in which information consumers can generally only browse. Finally, although its URL mechanism endows WWW with scalability in terms of number of servers, it is not scalable in terms of number of users. Extremely popular WWW servers such as Sun Microsystems' World Cup USA '94 [Sun Microsystems, 1996] site can often become overwhelmed by "flash crowds" [Nielsen, 1995] of tens of thousands of users, necessitating explicit physical mirroring to a number of alternative sites.

Notwithstanding these limitations, the number of WWW servers continues to grow exponentially. Netcraft found 342 081 WWW sites[3] in their August 1996 survey [Netcraft, 1996].

2.7 Further Reading

The number of books about the Internet, its services, and applications seems to be growing faster than the Internet itself. Two popular general guides are *The Whole Internet* [Krol, 1994] and *The Internet Unleashed* [Sams, 1995]. The Electronic Frontier Foundation's (extended) guide to the Internet is available free of charge on the Web [EFF, 1996]. For a comprehensive Internet bibliography consult the *Unofficial Internet Book List* [Savetz, 1996] which lists over 500 titles. Other online resources include the Internet Society's home page [ISOC, 1996], the very extensive *Internet Tools Summary* [December, 1996], and for Internet statistics the *Internet Domain Survey* [Network Wizards, 1996].

The World Wide Web is comprehensively covered at the W3C web site [W3C, 1996]. *WebCompare* provide a summary and comparison of Web server features [WebCompare, 1996] and the *Netcraft Web Server Survey* [Netcraft, 1996] provides a good overview as to which server software is currently most popular. A good introductory text on Java is *Exploring Java* [Niemeyer and Peck, 1996].

There are also a number of magazines which cover the Internet, the Web, and related topics. Most prominent among them are probably *Wired* [Wired, 1996] and *Internet World* [Meckler, 1996].

[3] Netcraft counts domain name aliases which map to the same IP address as separate sites, so this figure is somewhat inflated.

3 Hypermedia Systems

This chapter introduces the hypermedia paradigm. An earlier version of this chapter was published as Chapter 3 of *HyperWave: The Next Generation Web Solution*, Addison-Wesley, May 1996 [Andrews, 1996b].

3.1 What is Hypertext and Hypermedia?

The body of human knowledge is so large and is growing so fast that no single human mind can possibly comprehend its entirety. The best we can hope to do is to provide mechanisms which help us locate, structure, and compile useful information. One approach is hypermedia. The human mind operates by association, jumping from one item of thought to the next, almost instantaneously. The hypermedia paradigm attempts to model this process with associative links between chunks of information.

Unlike the typical printed book, which is read sequentially from beginning to end, hypertext is inherently nonlinear: it is comprised of many interlinked chunks of self-contained text. Readers are not bound to a particular sequence, but can browse through information intuitively by association, following their interests by following a highlighted keyword or phrase in one piece of text to bring up another, associated piece of text. Figure 3.1 illustrates this difference.

The word *self-contained* is important. Whereas in traditional, linear writing, a piece of text has a well-defined context and is embedded within the linear structure of the work, in a hypertext environment a particular piece of text may be reached from any number of contexts, that is, other chunks of text. Hence it becomes important to avoid assumptions of prior knowledge and keep individual pieces of text as self-contained as possible.

Hypermedia is the generalisation of hypertext to include other kinds of media: images, audio clips, and video clips are typically supported in addition to text. Individual chunks of information are usually referred to as documents or nodes, and the connections between them as links or hyperlinks –

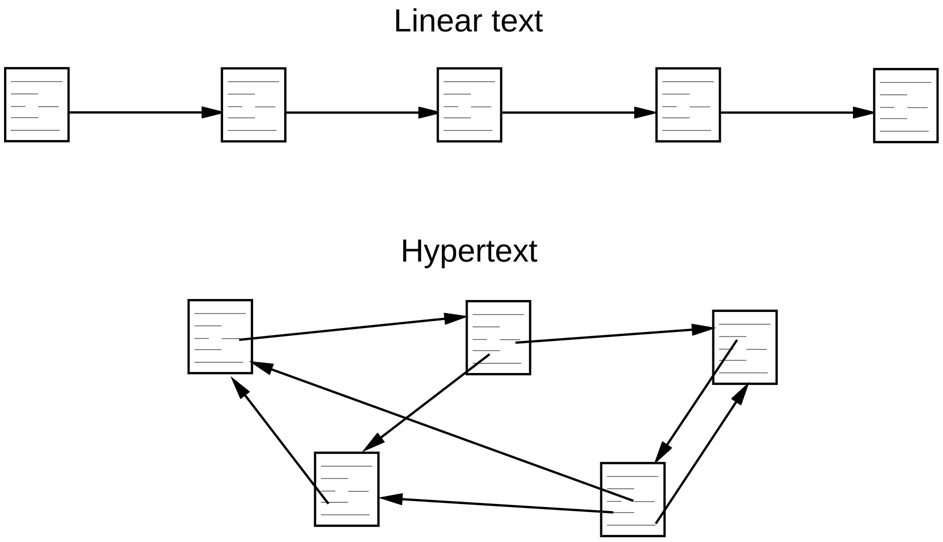


Figure 3.1 Linear text and hypertext.

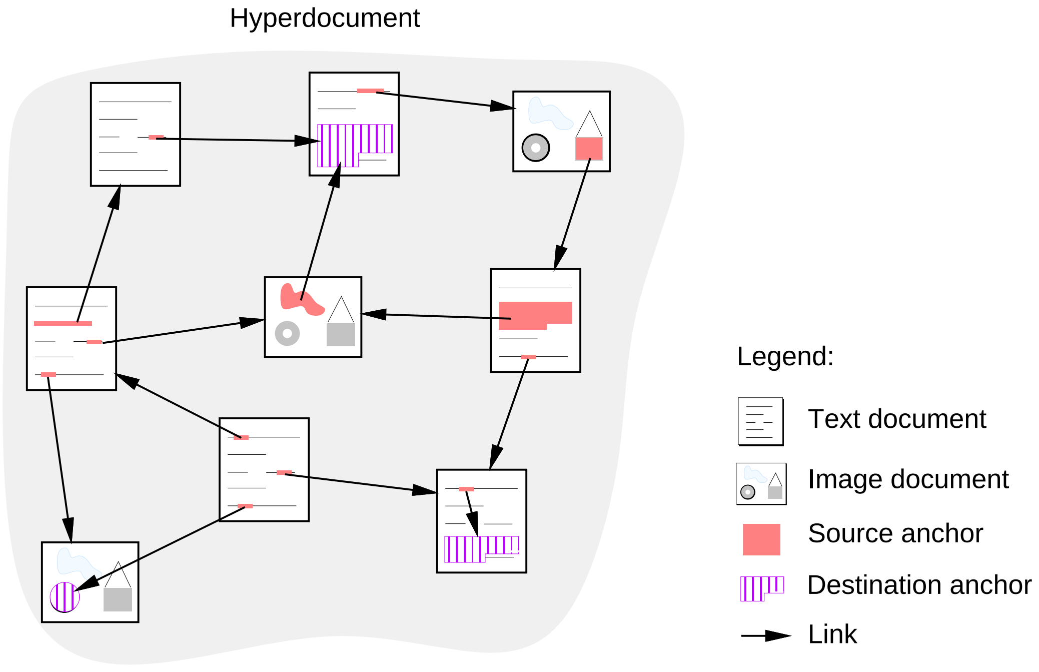


Figure 3.2 Nodes, links, and anchors.

the so-called *node-link hypermedia model*. The entire set of nodes and links forms a graph network. A distinct set of nodes and links which constitutes a logical entity or work is called a *hyperdocument*; a distinct subset of hyperlinks is often called a *hyperweb*.

A *source anchor* is the starting point of a hyperlink and specifies the part of a document from which an outgoing link can be activated. Typically, the user is given visual cues as to where source anchors are located in a document (for example, a highlighted phrase in a text document). A *destination anchor* is the endpoint of a hyperlink and determines what part of a document should be on view upon arrival at that node (for example, a text might be scrolled to a specific paragraph). Often, an entire document is specified as the destination and viewing commences at some default location within the document (for example, the start of a text). Figure 3.2 illustrates these concepts graphically.

Some authors distinguish between *referential* and *organisational hyperlinks*. Referential links are the cross-references distinctive of hypermedia. Organisational links are special links which establish explicit structure by connecting a parent node with its children, forming a tree within the overall node-link graph.

The traditional definition of hypermedia as being “multimedia with links” belies many of the possibilities modern technology now offers. We like to define *real hypermedia* in a broader sense, with two additional components. Firstly, real hypermedia incorporates new technologies like interactive movies, panoramic images, navigable three-dimensional models and virtual reality. Secondly, real hypermedia involves more than read-only browsing: it possesses integral facilities for communication and collaboration such as annotations, structured discussion, user feedback, message passing, and collaborative authoring.

3.2 The Origins of Hypermedia

Vannevar Bush, President Roosevelt’s science adviser during the Second World War, is generally credited as being the first to propose a system similar to what we would now call hypertext. In his 1945 article, “As We May Think,” [Bush, 1945] he described a system called Memex, based on microfilm and mechanical projection equipment:

> *“Consider a future device for individual use, which is a sort of mechanised private file and library. It affords . . . associative indexing, the basic idea of which is a provision whereby any item may be caused at will to select immediately and automatically another.”*

Bush also envisioned the provision of “trails”, sets of links relating information relevant to a specific purpose (i.e. the functional equivalent of hyperwebs or guided tours). Sadly, a functioning Memex was never built and the world had to wait another 20 years until computers became powerful enough to be applied to interactive tasks rather than pure scientific number crunching.

The term hypertext was coined by Ted Nelson in 1965 [Nelson, 1965]. His (proposed) Xanadu [Nelson, 1987] system incorporated a docuverse, a universal repository for all the world’s information and literature ever published. Nothing would ever be deleted, new versions of texts existing beside older versions. The system would be based on transclusions (virtual inclusions) as the fundamental structure, allowing the same material to appear in multiple contexts without being physically copied. This would also allow a business model where authors could be identified and paid pro-rata royalties on any portion of their works used elsewhere (by transclusion). Ted has struggled for 30 years to implement Xanadu, so far without success[1].

The first real working hypertext system was the Hypertext Editing System built at Brown University in 1967 under the leadership of Andries van Dam [van Dam, 1988]. It ran with 128 KB memory on an IBM/360 mainframe and most of the interface was text-based.

Doug Engelbart’s NLS (for oNLine System, part of the Augment project) [Engelbart, 1988], was the first graphical, mouse-based, point-and-click hypertext system. This system in fact laid many of the foundations for modern computing, and was first demonstrated at the 1968 Fall Joint Computer Conference.

The first working hypermedia system was the Aspen Movie Map , developed at the Massachusetts Institute of Technology by Andrew Lippman and his colleagues in 1978 [Lippman, 1980]. The town of Aspen, Colorado, which has a very regular, rectangular street layout, was filmed by driving a truck through every street and taking front, rear, left and right view photographs every three meters. These were transferred to videodisc and linked to follow the street grid. Users sat in front of a vertical monitor showing the street view and a flat monitor showing an overview map and could navigate forwards, backwards and (at a junction) left and right with a joystick. Short video clips of many of the buildings in Aspen were also linked in, so users could stop and explore them.

Intermedia [Haan *et al.*, 1992; Yankelovich *et al.*, 1988], developed at Brown University from 1985 to 1990, introduced the concept of link anchors and the use of a separate database for links (rather than storing them within documents), allowing links to be bidirectional and webs of links (hyperwebs) to be maintained as distinct entities. Intermedia was a multi-user system based on the client-server architecture and combining hypermedia features with information retrieval facilities such as full text search and dictionary lookup. It aroused great interest in the research community and generated numerous academic publications, but only really saw use within Brown University. Promising though it was, due to contractual limitations Intermedia only ran on the Apple Macintosh under UNIX and died a slow death in 1990 due to lack of funding to upgrade to a new operating system version.

The first widely available commercial hypertext product was OWL’s Guide, developed out of the University of Kent’s original UNIX version [Brown, 1992] and released in 1986 for the Macintosh and later for the IBM PC [Weiss-Fersko, 1991]. This system marked the transition of hypertext from a plaything of the research community to a real-world technique for real applications, and has now sold tens of thousands of copies.

The real breakthrough in popularity for hypermedia came in 1987, when Apple decided to bundle HyperCard free with every Macintosh [Williams, 1987; Goodman, 1987]. Within months, everyone seemed to be producing HyperCard “stacks,” as HyperCard documents are called. HyperCard

[1] Ted has, however, stated publicly that Hyper-G comes closest so far to realising Xanadu’s ambitious dreams.

is a frame-based, standalone hypermedia system incorporating a simple, but powerful scripting language called HyperTalk. Links do not have to be hardwired, but can be programmed in HyperTalk and computed dynamically.

The same year also saw Jeff Conklin's hypertext survey article in *IEEE Computer* [Conklin, 1987] and the first conference dedicated exclusively to hypertext, ACM Hypertext'87 at the University of North Carolina [Weiss and Schwartz, 1987]. The conference proved so popular that only half of the 500 people wanting to attend could be accommodated and there were still so many people crammed into the two lecture theatres that some had to sit on the floor!

The following years saw much research and development effort put into hypermedia, with hypermedia systems and projects being announced almost monthly. The Microcosm project, started at the University of Southampton in 1988 and first presented in 1990 [Hall *et al.*, 1996], is an example of an *open hypermedia system*, an external link management facility which provides link services to a wide variety of applications (under MS-Windows). The amount of functionality provided depends on how "aware" the applications are of Microcosm's link protocols: from fully-aware dedicated viewers, through semi-aware viewers which have been extended using their own in-built macro or scripting facilities, to unaware applications which are cloaked with a benign Microcosm-aware parasite.

The next major milestone was the dawn of the World Wide Web (WWW or W3) [Berners-Lee *et al.*, 1994], first demonstrated at the ACM Hypertext'91 conference in December 1991, but whose real breakthrough came in 1993 when the National Center for Supercomputing Applications (NCSA) released Mosaic [Andreessen, 1993], a point-and-click graphical browser for the WWW.

At around the same time as the WWW was conceived, the Hyper-G project was getting underway at Graz University of Technology. Hyper-G recognised the lessons learned from the Intermedia project and adopted the concepts of link anchors and of storing hyperlinks in a separate link database. Hence, Hyper-G supports bidirectional linking and automatic link consistency. Hyper-G also has orthogonal hierarchical structuring and composition facilities and integrated indexing and search facilities. The first Hyper-G server, Graz University of Technology's information system, went online at the end of 1991. In June 1994, the Hyper-G server software was officially released, together with preliminary versions of graphical authoring tools for X Windows and MS-Windows (Harmony and Amadeus, respectively). Hyper-G is now a sophisticated, "second generation" Web server.

Table 3.1 summarises some of the milestones in the history of hypermedia.

3.3 Types of Hypermedia System

Hypermedia systems can be classified according to a number of criteria. An important criterion is how much control over a hyperdocument's look and feel an author should be allowed to exercise: the "form versus content" debate.

Many traditional hypermedia systems were *frame-based*, i.e. documents must fit into a fixed-size frame (screen) and authors have full control over both content and presentation (form). Examples of frame-based systems include KMS, HyperCard [Goodman, 1987], ToolBook [Asymetrix Corporation, 1989], and HM-Card [Maurer and Scherbakov, 1996]. This notion of authoring, inherited from the field of Computer Aided Instruction (CAI), has the advantage that individual authors can create very colourful, interactive hyperdocuments. The HM-Card package, for example, supports vector graphics, animations, and question-answer dialogues within reusable hypermedia chunks. The price of full layout control, however, is that it becomes nearly impossible to maintain interface consistency across a related set of hyperdocuments having multiple authors.

In window-based hypermedia systems, documents may be of any size, and each is displayed in its own (scrollable) window. Authors typically specify only the information content of documents, while style sheets control their presentation. Thus, authors are relegated to the role of information suppliers, supplying marked-up documents devoid of specific presentation attributes. A heterogeneous collection of information delivery tools can then present the information according to their own capabilities and users' preference configurations.

Year	System	Originator	Milestone
1945	Memex	Vannevar Bush	Microfilm-based device
1965	Xanadu	Ted Nelson	Term “hypertext”
1967	Hypertext Editing System	Andy van Dam, Brown University	First working hypertext system
1968	NLS	Doug Engelbart	Mouse, point-and-click hypertext
1978	Aspen Movie Map	Andrew Lippman, MIT	First working hypermedia system
1985	Intermedia	Brown University	Anchors, webs
1986	Guide	OWL	First commercial product
1987	HyperCard	Apple	Free with every Macintosh
1987	Hypertext ’87	University of North Carolina	First ACM conference on hypertext
1990	Microcosm	University of Southampton	Open hypermedia system
1991	WWW	Tim Berners-Lee, CERN	Hypermedia over the Internet
1993	Mosaic	NCSA	Graphical browser for WWW
1994	Hyper-G	Graz University of Technology	Second generation web server

Table 3.1 Milestones in the history of hypermedia.

Modern hypermedia information systems like WWW and Hyper-G provide support for both form-free and form-specific documents. An SGML text mark-up format like HTML caters to information suppliers who are keen to avoid presentation-specific dependencies in their documents. Page description languages like PostScript and PDF cater to authors who want full control over end-user presentation of their material. The Easy viewer for Hyper-G, which is used for kiosk-like public display terminals, takes a hybrid approach. It interprets layout meta-information, such as size and position, which is associated with otherwise form-free documents.

Another distinction is that between standalone, single-author hypermedia systems and distributed, multi-author systems. In a collaborative, multi-author environment, support for document locking and atomic transactions is necessary to prevent authors mutually overwriting one another’s changes. Communication facilities between authors and the sharing of resources are also highly desirable in such an environment.

3.4 When to use Hypermedia

Hypermedia is an enabling technology, from which an extremely wide variety of application domains profit. In his book *Multimedia and Hypertext*, Jakob Nielsen [Nielsen, 1995] lists four golden rules for determining whether an application is suitable for hypermedia:

(1) A large body of information is organised into numerous fragments.

(2) The fragments relate to each other.

(3) The user only needs a small fraction at any time.

(4) The application is computer-based.

Given these requirements, current hypermedia application domains include:

- Help and documentation: Exemplified by the Help system provided as part of MS-Windows, this is a classic application of hypertext. Typically, the various command and function descriptions making up online documentation contain many cross-references, links to more detailed information, links to examples, and so forth.
- Reference works: Electronic dictionaries and encyclopedias are composed of fragmented chunks of material with a high degree of cross-referencing and are hence ideally suited to hypermedia.
- Software engineering: The software engineering development cycle generates a wealth of interrelated documents. Modern CASE tools are also increasingly offering hypermedia features within program code editors, for example to bring up more detailed information about a procedure or variable.
- Organisation of ideas: Hypermedia has been used in systems such as NoteCards [Halasz, 1988] to support the personal organisation of notes and ideas. The gIBIS [Begeman and Conklin, 1988; Conklin and Begeman, 1988] system was used to support debate and argumentation amongst multiple participants: nodes of type Issue, Position and Argument can be manipulated and related.
- Tourist information systems provide tourists with information on accommodation, services, places of interest, and so on. These systems often combine text and images with location and overview maps.
- Interactive fiction: Writers and novelists are now exploring the use of hypertext for interactive, nonlinear fiction. Readers interactively explore an underlying fictional construction.
- Rapid prototyping: An important part of interface design is the creation and evaluation of early mock-up designs. Since mock-ups often consist of interlinked screen designs, hypermedia systems can be used to rapidly generate such prototypes. ToolBook [Duncan, 1990] was used to generate and test early prototypes of a graphical interface to Hyper-G [Laky, 1993].
- Online marketing: In the World Wide Web, it is increasingly common to see glitzy sites offering a mix of advertising and information.
- Teaching support: Courseware has long been distributed in hypermedia form, typically with additional functionality for conditional branching and answer-judging. More recently, distance learning materials are proliferating on the Web.
- Communication and collaboration: Applications such as intelligent email, interactive questionnaires, online order forms and academic reviewing can all benefit from the application of hypermedia technologies.

3.5 Navigating Hyperspace

The primary advantage of hypertext and hypermedia systems over more traditional information systems based on databases is that users need not formulate queries to find information. Instead, graphical point-and-click style interaction is possible, which is easy and intuitive even for computer novices. This is what makes the hypermedia paradigm so attractive to millions of new Internet users.

Studies have shown that users employ a number of different navigation strategies when navigating through a complex information space: [Canter *et al.*, 1985; McAleese, 1989]

- Scanning: covering a large area without depth.
- Browsing: following a path by association until one's interest is caught.

- Searching: striving to find an explicit goal.
- Exploring: finding out the extent of the information space.
- Wandering: ambling along in a purposeless, unstructured manner.

By its very nature, hypermedia is an ideal medium for associative browsing. However, the increased freedom offered by hypermedia brings with it increased complexity. Readers of a linear document have only two choices when looking for information: forwards or backwards. Readers of hyperdocuments, on the other hand, can rapidly become overwhelmed by the plethora of choices available to them. This problem is known as the “user disorientation” or “lost in hyperspace” problem [Conklin, 1987; Edwards and Hardman, 1989].

To help users orient themselves in the information space, many hypermedia systems provide one or more of the following navigational aids in addition to hyperlinks.

Search Facilities

Sophisticated search facilities from the field of information retrieval, such as keyword search, content search, fuzzy (inexact) search, and similarity measures [Salton, 1989] are indispensable for finding specific information once the size of the hypermedia space exceeds browsable proportions (say many hundreds or thousands of nodes and links). As Halasz indicated in his reflections on NoteCards [Halasz, 1988]:

> *“For NoteCards to be useful in managing large heterogeneous networks, search and query needs to be elevated to a primary access mechanism on par with (browsing) navigation.”*

Hypermedia systems which provide information retrieval facilities in addition to hyperlinks are often called *hypermedia information systems* to emphasise this important distinction.

Hierarchical Structure

NoteCards was notable among early hypermedia systems for its provision of Fileboxes, containers which may hold both Notecards (primitive nodes) and other Fileboxes. This proved to be extremely valuable, allowing users to employ hierarchical composition (inclusion relations) in addition to standard referential links when structuring and later navigating through their data.

However, the NoteCards implementation was fundamentally flawed: composite nodes were implemented using standard referential links and not as a separate concept. This caused considerable confusion and led Halasz to conclude:

> *“Inclusion should be implemented within, as opposed to on top of, all hypermedia systems. Moreover, all aspects of hypermedia should support inclusion (or part-of) relations as a construct distinct from standard (reference) links.”*

Fully integrated hierarchical structuring, such as that found in Hyper-G [Maurer, 1996], is not only useful for hierarchical navigation: documents devoid of links can be inserted (and retrieved!), composite nodes can be assigned (searchable) attributes, and perhaps most importantly, the scope of searches can be focused on particular subsets of the database.

Guided Tours

Guided tours were foreseen by Vannevar Bush in 1945 as “trails”. In essence, a guided tour is a predefined path through a subset of hyperspace, generally with no or only a few branches. They are particularly useful for introducing new readers to a hypermedia data set. As long as readers stay on the tour, they can navigate using simple “next” and “previous” commands. The advantage over tourist guided tours is that the reader, unlike the tourist, can leave the tour to explore and return to it at will.

Overview Documents

Overview documents are “hand-crafted” to illustrate the structure of (part of) a hyperweb; they typically provide direct links to documents within the web. Examples include texts containing a list of links to other documents, or an annotated diagram, map, photograph, or a 3d model of a spatial layout.

Such manually edited overviews are only really of use in hyperdocuments which are not expected to change much during their lifetime, since changes to the hyperdocument necessitate manually updating the overview. They also do not scale very well: drawing an overview of a 20 or 30 node hyperdocument is both manageable and often very helpful to users, drawing an overview of a 1000 or 10 000 node hyperdocument by hand is a thankless and futile task. For this reason, overview documents tend to be used for small, stable subsets of hyperspace and standalone, read-only hypermedia systems.

Graphical Browsers

In contrast to overview documents, graphical browsers rely on dynamically generated structure maps. Graphical browsers come in many flavours:

- Hierarchical browsers display hierarchical structures (or organisational links) and allow navigation in a hierarchical fashion.
- Global maps attempt to map the entire hyperspace, but usually run up against problems of sheer scale.
- Local maps show the “vicinity” of the current node in terms of hyperlinks to and from other related nodes.
- Fish-eye views are employed to focus attention on important nodes while retaining a sense of context by deliberately distorting the view (like using a wide-angle “fish-eye” lens).
- Automatic clustering techniques combine similar nodes into larger composites, thus reducing the overall amount of clutter in the structure map.

Backtracking

Backtracking allows users to return to previously visited nodes. This facility is an important confidence builder, since users are encouraged to experiment if they know they can always backtrack. Within the course of a single session, backtracking usually takes the form of a history list of visited nodes, with backwards and forwards functions available to navigate within the list. A more general facility is the history tree [Foss, 1988] which can also visualise backing up and branching in other directions. The Intermedia Web View combines a history list with a local map in one diagram [Utting and Yankelovich, 1989]. Being able to save and edit history lists is an elegant way of finding information again and preparing guided tours.

Bookmarks are pointers to specific nodes in hyperspace which persist between sessions. Users can thus build up a set of direct jumps to their favourite places in hyperspace. Good examples of this are Mosaic’s hotlist and Netscape’s bookmarks. These, however, have the major disadvantage that they are maintained locally by the client. Changing your client software necessitates converting your bookmark list. If you access the Web from any computer other than your regular computer (say at a conference), you must remember to take your bookmarks with you. For these reasons, Hyper-G users have a personal *home collection* or area of the Hyper-G server, within which they can both organise pointers to Web resources (bookmarks) and keep personal or unreleased documents. The personal home collection is accessible through any WWW, Gopher or Hyper-G client from any location, and can be structured arbitrarily like any other Hyper-G collection.

3.6 Further Reading

For a good overall introductory text on hypertext and hypermedia, I highly recommend *Multimedia and Hypertext: The Internet and Beyond* [Nielsen, 1995]. Nielsen also includes a valuable annotated bibliography with pointers to many of the systems mentioned in Section 3.2. Vannevar Bush's 1945 *As We May Think* article was reprinted in 1986 and is also available online [Bush, 1945]. Ted Nelson's definitions of hypertext and hypermedia are described in *Computer Lib/Dream Machines* [Nelson, 1987], his vision of Xanadu in *Literary Machines* [Nelson, 1993]. The *Hypertext/Hypermedia Handbook* [Berk and Devlin, 1991] is an excellent, though somewhat dated, collection of introductory papers; the glossary at the end of the book should be especially useful to newcomers. Conklin's survey article [Conklin, 1987] gives a good introduction to basic principles and an enlightening overview of early systems. More technical coverage of multimedia aspects is given in [Buford-Koegel, 1994].

Good online starting points are *SIGLINK*, the ACM Special Interest Group on Hypertext / Hypermedia [SIGLINK, 1996], and the *Memex and Beyond* web site [Memex, 1996].

4 Hyper-G – A Second Generation Web Server

In this chapter, I present a detailed overview of Hyper-G. Section 4.2 is extracted from the paper *On Second Generation Network Hypermedia Systems*, which won an outstanding paper award at ED-MEDIA 95 [Andrews *et al.*, 1995e]. Section 4.3 is adapted from *Hyper-G: Towards the Next Generation of Network Information Technology*, which appeared in J.UCS, April 1995 [Andrews *et al.*, 1995c]. Section 4.8 is based on material published as *Serving Information to the Web with Hyper-G* in Computer Networks and ISDN Systems, April 1995, as part of the proceedings of the Third International World-Wide Web Conference (WWW'95), Darmstadt, Germany [Andrews *et al.*, 1995g].

4.1 The Evolution of Hyper-G

Hyper-G can be described both as a "second generation" Web server [Andrews *et al.*, 1995e,g; Maurer, 1996] and as a multi-user, structured, hypermedia information system in its own right [Kappe, 1993; Andrews and Kappe, 1994]. Hyper-G development began in 1990 and Hyper-G evolved in parallel with the World Wide Web [Berners-Lee *et al.*, 1994], a WWW gateway always being part and parcel of Hyper-G. Indeed, back in 1991, The Hacker's Jargon collection on Graz University of Technology's Hyper-G server was one of the first major Web resources outside of CERN.

At the time, there was no widely established markup format for hyperlinks in text or 3d models, leading us to define our own hypertext markup format (HTF) [Kappe, 1995b], our own format for hyperlinked 3d models (SDF), and link anchor representations for many other types of document such as images, film clips, PostScript files, etc. Readers of early published articles on Hyper-G sometimes gain the impression that Hyper-G is a closed, proprietary system, outside of and in competition to the WWW. However, as WWW gained in popularity, we soon recognised the importance being part of the wider Web and of providing integrated support for the WWW's formats and conventions within Hyper-G. Today, Hyper-G has full support for the WWW's document formats (HTML and VRML), CGI server scripting, Java applets, etc. In short, the Hyper-G server is a full-blooded Web server, but with a host of extra functionality.

Hyper-G's native clients are sophisticated authoring and management tools, primarily used by information providers; end-users accessing the information on a Hyper-G server generally use a standard Web client such as Netscape or Mosaic. However, the use of Hyper-G's native clients by end-users as powerful browsing tools is also appropriate for certain specialised applications. In June of 1996, a commercial version of Hyper-G became available under the name HyperWave [Maurer, 1996].

I first became involved in the Hyper-G project in the summer of 1991, shortly after Frank Kappe completed the specifications of Hyper-G in his PhD Thesis [Kappe, 1991] and have been involved in shaping and steering its development ever since. In particular, I was responsible for designing Harmony, the graphical browsing, management, and authoring client for X Windows (described in Chapters 5 and 6 of this thesis), for integrating 3d hypermedia into Hyper-G (described in Chapter 7), and for information visualisation tools (described in Chapter 8).

The scientific achievements of Hyper-G lie in its advanced navigational, management, and authoring facilities. In the words of Tim Berners-Lee, inventor of the World Wide Web:

> "*. . . an excellent document-management system for the Web.*"

and Robert Cailliau, co-inventor of the Web:

> *“. . . the most advanced authoring server available.”*

Or as Steve Poltrock, chair of ACM Hypertext’93, says:

> *“Hyper-G achieves a major advance over existing Web technology by integrating document management, link management, and innovative navigational features.”*

A further important contribution is Hyper-G’s sophisticated structuring mechanism in addition to cross-referential hyperlinks. As Jakob Nielsen, hypermedia expert at Sun Microsystems, phrases it:

> *“Finally, some sorely needed structure for the Wild Web.”*

The ability to generate dynamic HTML pages based on explicit structure is also a major benefit. According to Ted Nelson, hypertext guru:

> *“Hyper-G represents a new generation of dynamic Web Servers, serving up dynamic hypermedia combinations in depth.”*

Finally, a separate link database and Hyper-G’s server-server protocol mean that information can be kept consistent *across* server boundaries. As Mark McCahill, creator of Gopher, says:

> *“Hyper-G combines the better parts of Gopher and WWW into one system and solves some fundamental architectural problems that both Gopher and WWW have.”*

4.2 The Need for “Second Generation” Web Systems

As described in Chapter 2, the steady growth of the Internet has made resource discovery and structuring tools increasingly important. Historically first was the introduction of various dictionary servers, Archie [Emtage and Deutsch, 1992] being probably the first and most prominent. WAIS [Kahle *et al.*, 1992] introduced a powerful search engine permitting full-text searches of large data-bases and returning a ranked list, the ranking based on a heuristic approach. Although directory servers like Archie (to locate a database) and search engines like WAIS (to locate a desired item in that database) alleviated the problem of finding information in the Internet somewhat, it soon became apparent that other techniques would also be necessary.

The most important such technique is to emphasise the a-priori organisation of information, rather than try to search for information in a universe of completely different databases. Two efforts in this direction have proven particularly successful: Gopher, originally developed at the University of Minnesota [McCahill and Anklesaria, 1995] and the World Wide Web (WWW, W3, or “The Web” for short) originally developed at CERN in Geneva [Berners-Lee *et al.*, 1994]. In both cases information is stored in a simple structured fashion on servers, and can be accessed via clients available for most major hardware platforms.

Information in Gopher is structured in a hierarchical fashion using menus, an access technique which, though simple to use, has many well-known weaknesses. Information in WWW is structured in documents; documents are linked together according to the hypertext-paradigm: *anchors* within documents are associated with *links* leading to other documents. Although many stand-alone systems using the hypertext-metaphor have emerged since the introduction of HyperCard on the Mac in 1987, WWW can claim to be the first wide-spread hypertext system to be accessible via the Internet.

WWW servers are easy to install and clients are available on all major platforms; much of the software is free and source code is often available. The node-link technique for navigating and finding information is quite appealing, at least for small to medium amounts of data, and the mix of media makes the use of WWW aesthetically pleasing. All this has contributed to the proliferation of WWW — indeed, the term “Web” has come to mean not just WWW, but the entire information space accessible via the Internet. The success of WWW, the number of its proponents and freaks, and its publicity even in non-scientific publications like Time magazine may create the impression that WWW is the solution for most information needs and will remain the dominating system for the foreseeable future.

The reality is slightly different, however. Whilst WWW is undoubtedly a leap forward compared to pre-WWW times, experience shows that much functionality required for sizable applications is missing from WWW. In this sense, WWW should be considered a first generation networked hypermedia system. More advanced “second generation” hypermedia systems are required to cope with the problems currently being encountered on the Web. To give one example, while pure node-link navigation is satisfactory in small systems, it tends to lead to confusion and disorientation, if not chaos, when applied to large amounts of data — for substantial applications, some additional structuring and searching facilities are clearly required [Conklin, 1987; Maurer *et al.*, 1994a]. Similarly, the necessity to keep links separate from, rather than embedded in, documents as is the case in WWW has already been demonstrated in the pioneering work on Intermedia at Brown University [Haan *et al.*, 1992].

Why First Generation Features are not Enough

Information in a hypermedia system is usually stored in “chunks” or “nodes”, as described in Chapter 3. Chunks consist of individual documents which may themselves consist of various types of “media”. Typically, a document might be a piece of text containing a picture. Each document may contain links leading to (parts of) other documents in the same or in different chunks. Typical hypertext navigation through the information space is based on these links: the user follows a sequence of links until all relevant information has hopefully been encountered.

In WWW, a chunk consists of a single document. A typical document consists of textual information and may include pictures (“inline images”) and the (source) anchors of links. Anchors can be attached to textual information and to (parts of) inline images, but not to parts of non-inlined images. Links may lead to audio or video clips which can be activated, but links cannot emanate from such documents (a dead-end document). The textual component of a document is stored as marked-up text in HTML format.

In Hyper-G, the setting is considerably more general. Chunks, called “clusters” in Hyper-G terminology consist of a number of documents. A typical cluster might, for example, consist of five documents: a piece of text, a second piece of text (for example in another language, or a different version of the same text, or an entirely different text), a third piece of text (the same text in a third language perhaps), an image and a film clip. Anchors can be attached to textual information, to parts of images, and even to regions in a film clip or objects within a 3d model. Links are not part of the document but are stored in a separate database. They are both typed and bidirectional; they can be followed forward (as in WWW), but also *backwards* so that references *to* a particular document become directly accessible. The textual component of a document is also stored in HTML format, but any links are extracted upon insertion of the text into the Hyper-G server and are merged back in again, when the document is accessed. The support for multiple pieces of text within a “cluster” allows Hyper-G to handle multiple languages in a very natural way. It also elegantly handles the case where a document comes in two versions, a more technical (or advanced) one and one more suitable for the novice reader. As in WWW, pictures can be treated as inline images, as separate documents, or both (a low resolution inline image pointing to a high quality external image).

Hyper-G provides full support for PostScript documents, including hyperlinking facilities and high-quality PostScript viewers. Almost all word processing packages are capable of producing PostScript files as output and can thus be used to prepare data for Hyper-G. Also, PostScript files allow the incorporation of pictures and formulae; they offer the user the possibility to view documents exactly as if they were printed (given high enough resolution screens), giving Hyper-G the necessary professionalism for high quality electronic publishing of journals, books, and manuals. For one of the major applications of Hyper-G, the Journal of Universal Computer Science, semi-automatic text extraction and link generation tools allow articles submitted in PostScript form to be presented side-by-side with fully-searchable, interlinked hypertext. The use of standard compression techniques allows the PostScript files to be compacted to about the same size as equivalent HTML files. From its beginnings, Hyper-G has supported 3d objects and scenes with embedded hyperlinks.

The native X Windows client for Hyper-G (Harmony) provides four different ways to navigate within such 3d models. Finally, Hyper-G allows the use of documents of a "generic" type, permitting future extensions and the encapsulation of documents otherwise incompatible with Hyper-G.

Let us now turn to the discussion of the philosophy of links in WWW versus Hyper-G. The ability to attach links to parts of a picture is clearly desirable, when additional information is to be associated with certain sub-areas of an image. That links are bidirectional and not embedded in the document has a number of very important consequences. Firstly, links relating to a document can be modified without necessarily having access rights to the document itself. Thus, private links and a certain amount of customisation are possible. Secondly, when viewing a document it is possible to find all documents referring to the current one. This is not only a desirable feature as such, but is of crucial importance for being able to maintain the database. When a document is deleted or modified, all documents referring to it may have to be modified to avoid the "dangling link syndrome", or to avoid being directed to completely irrelevant documents. Hyper-G offers the possibility of automatically notifying the owner of a document that some of the documents that are being referred to have been changed or deleted, an important step to "automatic link maintenance". Thirdly, the bidirectionality of the links allows the graphic display of a *local map* showing the current document and all documents pointing to it and being pointed at, an arbitrary number of levels deep. Finally, the fact that links can have types can be used to show to the user that a link just leads to a footnote, or an annotation, or is a counter argument or supporting argument of some claim at issue – typed links enhance the perception of how things are related and can be used as tool for discussions and collaborative work.

Navigation in WWW is performed solely using the hypertext paradigm of anchors and links. It has become a well accepted fact that structuring large amounts of data using only hyperlinks such that users don't become "lost in hyperspace" is difficult to say the least. WWW databases are large, flat networks of chunks of data and resemble more an impenetrable maze (the "World-Wide Spaghetti Bowl", in the words of Robert Cailliau) than well-structured information. Indeed, every WWW database acknowledges this fact tacitly, by preparing pages that look like menus in a hierarchically structured database. Items are listed in an orderly fashion, each with an anchor leading to a subchapter (subdirectory). If links in WWW had types, such links could be distinguished from others. But as it is, all links look the same: whether they are "continue" links, "hierarchical" links, "referential" links, "footnote" links, or whatever else.

In Hyper-G not only can have links a type, links are by no means the only way to access information. Clusters of documents can be grouped into collections, and collections again into collections in a pseudo-hierarchical fashion. We use the term "pseudo-hierarchical" since, technically speaking, the collection structure is not a tree, but a directed acyclic graph. A collection can have more than one parent: an impressionist picture X may belong to the collection "Impressionist Art", as well as to the collection "Pictures by Manet", as well as to the collection "Museum of Modern Art". The collection hierarchy is a powerful way of introducing structure into the database. Indeed many links can be avoided this way [Maurer *et al.*, 1994a], making the system much more transparent for the user and allowing a more modular approach to system creation and maintenance.

In contrast to WWW, collections, clusters and documents in Hyper-G have a rich set of associated metadata (title, keywords, modification date, etc.), which can be used in Boolean queries to find documents of interest or which are used to map visual attributes of visualisations.

Finally, Hyper-G provides sophisticated, fully integrated full-text search facilities: the textual content of a document is automatically and transparently maintained in a full-text inverted index. Most importantly, the scope of any of such searches can be defined to be the union of arbitrary collections, even if the collections reside on different servers (we will return to this important aspect of Hyper-G as a distributed database below).

Note that many WWW servers also provide full-text search. However, since no standard full-text search engine or API is built into WWW, the functionality is "bolted on". However, adding functionality on top of WWW leads to the "Balkanisation", or the fragmentation of WWW, since different sites implement missing functionality in different ways. Thus, to stick to the example of the full text search engine, the fuzzy search employed by organisation X may have a different syntax and yield entirely different results from the fuzzy search employed by organisation Y, much to the bewilderment of users. Actually, the situation concerning searches in WWW is even more serious. Since documents in WWW do not have attributes, no search is possible on such attributes; even if such a search or a full text search is artificially implemented, it is not possible to allow users to define the scope of the search due to the lack of structure in the WWW database. Full-text searches in WWW always work in a fixed, designated part of the WWW database residing on one particular server. Tens of thousands of WWW servers are currently installed and accessible over the Internet. There is, however, little or no coherence between these servers: users wanting to search for an item on a number of WWW servers have to initiate a new search for each server. Robot-generated search engines such as Lycos [Lycos, 1996] and AltaVista [AltaVista, 1996] have alleviated the situation to some extent, but this approach is not extremely scalable.

Hyper-G, on the other hand, is a genuinely distributed database: servers (independent of geographical location) can be grouped into collections, with the “hyperroot” at the very top. Users can focus the scope of searches to arbitrary sets of collections on arbitrary servers.

Standard WWW servers provide little in the way of information management tools for large information spaces. To be more concrete, suppose 10 departments within a university intend to operate a WWW server. To keep the amount of information manageable, the tendency would be for each department to operate its own server. (Indeed, there are many more WWW servers than there are server sites, clearly demonstrating this phenomenon.) However, if the 10 departments operate 10 different servers and a user from outside wants to look up a person without knowing the department, the user is forced to query all of the servers, one after the other. Hyper-G, with its sophisticated structuring and management tools, offers a much more satisfactory solution. The 10 departments operate a single Hyper-G server. Hyper-G provides various types of access rights and the definition of arbitrarily overlapping user groups, hence different groups can work within the same server without fear of interfering with someone else’s data. Not only can department X not influence the information of department Y, certain parts of the database may have even their read access restricted to certain groups or even to single individuals (private collections). Continuing the example, suppose an outside user looks for a certain person. Accessing the single Hyper-G server operated by the 10 departments with a full-text search will find the information, assuming it is present. However, suppose the departments insist on each operating their own Hyper-G server: by simply defining a collection “Servers of this University”, a single Hyper-G search will still examine all ten databases (assuming they are all in a LAN or on the Internet).

Hyper-G may be accessed anonymously by anyone. However, users having an account on a particular Hyper-G server may *identify* themselves with name and password to gain access to their account and any additional access rights associated with it. Identified users have a “home collection” on the server, where personal documents can be kept (similar to a home directory on a UNIX machine) and a personal view of the information space maintained (pointers to the information most important for them). Since home collections reside on the server rather than being maintained locally (like hotlists in WWW), they are not dependent on using a particular browser on a particular machine.

Every Hyper-G server is also a caching proxy server. If one accesses the Web via a local Hyper-G server, all accesses to other Hyper-G servers, but also WWW and Gopher servers, are routed through the local Hyper-G server. Documents retrieved once are automatically cached for all users of the local Hyper-G server, so they will no longer be retrieved from the remote database next time around. Between Hyper-G servers, a scalable flooding algorithm ensures that cached documents are automatically updated [Kappe, 1995c]. Although WWW servers support caching, the consistency of cached documents is not guaranteed.

The acceptance of a hypermedia system is certainly not only dependent on deep technical features, but above all on the information content and the ease of use. Due to the fact that large hypermedia systems tend to lead to disorientation, second generation hypermedia systems have to try very hard, both at the server and at the client end, to help users with navigational tools. Some navigational tools, like the structuring and search facilities have already been described; others, such as link maps, history lists, specific and personal collections can also be of great help and are available in Hyper-G; a particular speciality of the Harmony client is a 3d “information landscape” depicting collections and documents as blocks spread out across a three-dimensional landscape, over which the user is able to fly.

Hypermedia for Communication and Collaboration

First generation hypermedia systems like WWW have traditionally been seen mainly as (simple) information systems. Most applications currently visible on the Internet support this view: very often WWW servers offer some pleasantly designed general information on the server institution, but only rarely does the information go much deeper. If it does, usually a "hybrid" system is used, WWW with some add-ons using the WWW scripting interface. It is our belief that hypermedia systems acting as simple information systems, where one or more people input information to be read by others, do not offer much potential; they will disappear into obscurity sooner rather than later. To ensure the success of a hypermedia system, it must allow users also to act as authors: allow them to change the database, create new entries for themselves or other users, create a personal view of the database as they need it, and, above all, allow the system to be used also for communication and cooperation.

First generation hypermedia systems like current WWW servers almost entirely lack support for such features. Emerging second generation hypermedia systems are bound to incorporate more and more features of the kind mentioned; Hyper-G provides a start. The native Hyper-G clients Amadeus and Harmony are designed to allow the easy import of data into the server. They are also designed to allow point-and-click link generation; source and destination anchors are defined with simple mouse clicks and drags, and links completed in a confirmation dialogue.

Hyper-G supports annotations with user-definable access rights. In contrast to some WWW clients, which support annotations stored locally by the client, Hyper-G annotations become part of the server database, i.e. they are available when working with other clients, or from another user account or machine. Annotations can themselves be annotated; the network of annotations can be graphically displayed using the local map function of Harmony. Thus, the annotation mechanism can be used as the basis of (asynchronous) computer-conferencing, and has been successfully employed in this fashion. The client-server protocol in WWW is "static" in the sense that the server can only react to requests by the client, but cannot become active itself. In Hyper-G, the client-server protocol is "active" in the sense that the server can contact the client. This can be used for example to send update notification to a client, and provides the first (rudimentary) possibilities for client-client communication, synchronous communication, conferencing, and cooperation.

We believe that many of the features discussed in the area of computer supported cooperative work [Dewan, 1993] will eventually be incorporated into second generation hypermedia systems. The most widely used functions of the Internet are file transfer (ftp) and, in particular, electronic mail. Without leaving their hypermedia environment, users must be able to retrieve files and edit, send, and receive email. Email should automatically be presorted by criteria such as subject, author, and date, and related pieces of email linked together, the local map feature presenting a graphical overview of the flow of discussion. The hypermedia system should also have the possibility to send mail with delays or on certain dates to act as reminder and as an active personal scheduler. A number of relevant ideas are collected in [Kappe and Maurer, 1993] and are currently under implementation in Hyper-G.

Hyper-G – A Smooth Transition from First to Second Generation

As has become clear from the above discussion, first generation hypermedia systems such as WWW do not have enough functionality to serve as a solid and unified basis for substantial multi-user information systems with a strong communicational component. Hyper-G is a first attempt to offer much more basic functionality, yet to continue the path started by and remain fully interoperable with WWW [Andrews *et al.*, 1995g]. Any WWW or Gopher client can be used to access any Hyper-G server; a Hyper-G client may, through a Hyper-G server, access any WWW or Gopher server, indeed providing "free" caching as a by-product. The compatibility of Hyper-G with WWW and Gopher actually goes much further: tools to import complete WWW and Gopher databases into Hyper-G without manual intervention are available. Thus, users of WWW can migrate up to an environment allowing all kinds of searches, access control, etc., without being forced to abandon their current database or their favourite WWW client.

The first Hyper-G server went online at the end of 1991. The Hyper-G software was officially released in June 1994, and already has a substantial following. Clients are available for UNIX (Harmony and hgtv) and MS-Windows (Amadeus) and are free of charge; the server runs on UNIX platforms and is free of charge for educational institutions. The current applications of Hyper-G include: university information systems (e.g. Graz University of Technology and The University of Auckland), organisation-wide information and communication systems (the European Space Agency being the largest user so far), multimedia infrastructure for museums and exhibitions (e.g. The Museum of New Zealand and the AEIOU project), an infrastructure for teleteaching experiments (e.g. the University of Auckland in cooperation with NZ Telecom), electronic publishing (e.g. Meyer/Brockhaus/BI/Langenscheidt for reference books and Springer for the Journal of Universal Computer Science), and an infrastructure for cooperation amongst scientists (German Mathematics Association). A major new market is the corporate Intranet, such as the Hyper-G installations at HILTI, Lichtenstein and German Aerospace.

Whether Hyper-G will ever be as wide-spread as Gopher and WWW, nobody can know. What is clear, is that time is working against first generation systems and in favour of the increased functionality and universality of second generation systems. Hyper-G will certainly contribute to and speed up developments in this direction by demonstrating the feasibility of new concepts and of a new generation of networked multimedia systems.

4.3 The Design of Hyper-G

Based on an analysis of the strengths and weaknesses of existing systems, the following primary design goals were formulated:

(1) Provide orientational and navigational aids.

(2) Provide automatic structuring and maintenance.

(3) Reduce fragmentation across servers.

(4) Support user identification and access control.

(5) Support multilinguality.

(6) Maintain interoperability with existing systems.

To help alleviate the disorientation associated with becoming “lost in hyperspace”, Hyper-G provides three closely coupled, orthogonal navigational mechanisms: hierarchical navigation, hyperlinks, and search. The tight coupling allows clients to correlate search results, link maps, and structure overviews, providing a powerful aid to navigation.

Users of WWW are soon confronted with the “dangling link” syndrome inherent in WWW’s data model: when a document is moved or deleted, there is no way of automatically updating or deleting links to that document – following such links produces an error message. In and between Hyper-G servers, the consistency of links is maintained automatically, as are Hyper-G’s aggregate collection structures.

To counter the Balkanisation of databases, Hyper-G provides much functionality integrated into it (and hence uniform in nature) which has to be implemented on top of WWW or Gopher (and hence potentially differing from site to site), in particular powerful search and retrieval facilities which can be performed simultaneously and seamlessly *across* server borders. Information can also be *structured* both within and across server borders.

In order to differentiate between users and provide tailored views of the available information, support for user accounts, user groups, access rights, and access modes is essential. They are also a prerequisite for the implementation of charging facilities and other business models.

Although the Gopher+ and WWW protocols in theory support multi-language versions of documents, the major Web browsers have only recently begun to support languages other than English.

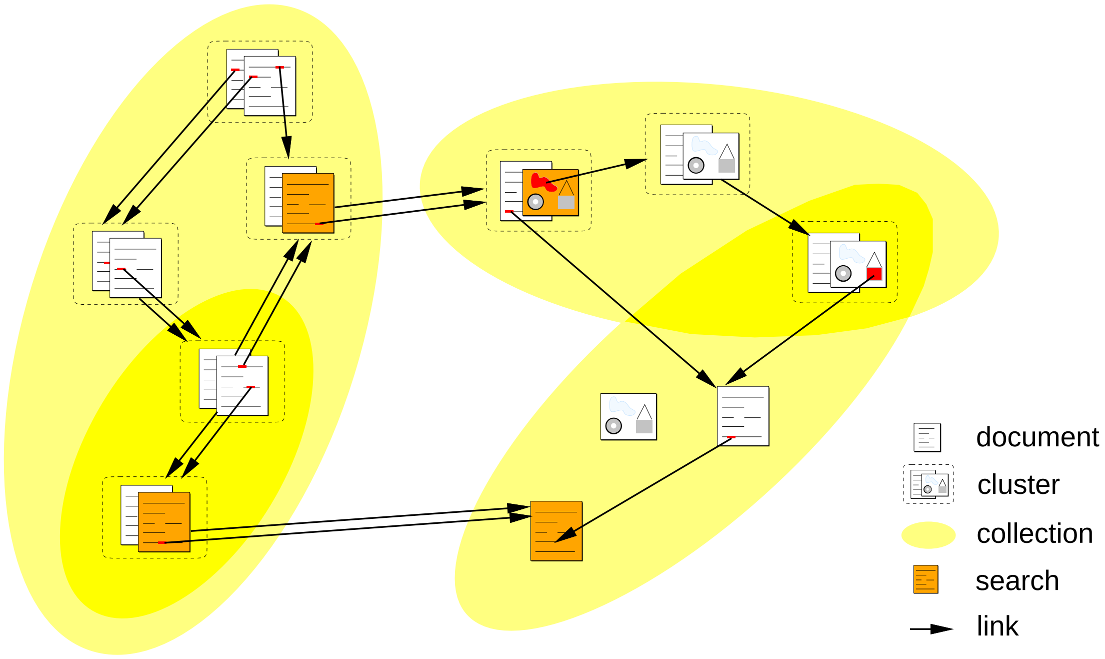


Figure 4.1 The Hyper-G data model.

Full text search in multiple languages is still largely unavailable. Hyper-G and its clients, on the other hand, were specifically designed for multilingual use.

The most fundamental design decision was to provide support for orthogonal yet closely coupled structuring, linking, and search facilities at the server level. The Hyper-G data model shown in Figure 4.1 defines:

- *Structuring*: Documents may be grouped into aggregate *collections*, which may themselves belong to other collections. Every document must belong to at least one collection. Navigation may be performed down through the collection hierarchy (the collection "hierarchy" is, strictly speaking, a directed acyclic graph since collections and documents may be members of multiple parent collections), access rights assigned on a collection-by-collection basis, and the scope of searches restricted to particular sets of collections. Collections may span multiple Hyper-G servers, providing a unified view of distributed resources.

- *Linking*: Hyperlinks connect a *source anchor* within one document to either a *destination anchor* within another document, an entire document, or a collection. Links are not stored within documents (as in WWW) but in a separate link database (as pioneered by Intermedia [Haan *et al.*, 1992]). This has a number of important advantages:

  – Consistency constraints can be met more easily (for example when moving or deleting a document, it is important to know which other documents contain links to it).

  – Source and destination anchors are not limited to text documents, but can be attached to any kind of media (image, audio, film, 3d scene, formatted PostScript document, etc.).

  – Links can be attached to otherwise read-only documents (for example documents on CD-ROM or with read-only access rights).

  – Links can be followed backwards.

  – Links can have attributes of their own, including for example access rights.

  – A local link map can be readily computed and visualised using the link database.

- *Attribute and Content Search*: Documents, collections, and links have an associated set of attributes (author, title, keywords, etc.) which may be searched for, including boolean combinations and term truncation. Full text (content) search facilities include vector and fuzzy boolean queries. Every document and collection is automatically indexed upon insertion into the database – no extra indexing steps are required. The scope of a search may be focussed to one or more collections on one or more servers or may be as wide as all collections on all Hyper-G servers worldwide.

Each of these three orthogonal features works *seamlessly* across server boundaries, reducing fragmentation while at the same time promoting consistency. To achieve scalability in terms of number of users, Hyper-G servers replicate and cache remote objects; updates are propagated to other servers using an efficient, scalable flooding algorithm [Kappe, 1995c].

A special kind of collection, a *cluster*, groups documents into logical entities. Clusters are used both to define multimedia aggregates (for example, a text and an associated image or video clip) which are presented together and multilingual aggregates (for example, English and German translations of a text, audio clip, structure diagram, or any combination of these).

Other design features supported by Hyper-G and include:

- Anonymous and identified user identification modes.
- A hierarchical scheme of user groups and subgroups maintained by the server.
- Access rights for users and user groups on a document or collection basis.
- “Home collections”, personal information spaces for identified users (kept on the server) used to organise personal documents and pointers to resources.
- Language preferences, applied both to document retrieval and to the user interface.
- An underlying object-oriented database, which guarantees the consistency and integrity of data (for example the updating of links when a document is moved or deleted).

4.4 The Architecture of Hyper-G

Figure 4.2 shows the client-server architecture of Hyper-G. Note the interoperability of Hyper-G with Gopher and WWW clients and servers. When accessed by a Gopher client, the Hyper-G server maps the collection hierarchy into a Gopher menu tree. A synthetic search item is generated at the foot of each Gopher menu to allow searching the corresponding collection. When accessed by a WWW client, each level of the collection hierarchy is converted to an HTML document containing a menu of links to its members. HTML text documents residing on the Hyper-G server have their links and inline images merged in at the correct position on-the-fly. Additional Hyper-G functionality such as user identification, language preference selection, and searching are implemented via HTML forms and are accessible by buttons at any time. The exact layout and appearance of the pages is highly configurable, as is discussed later in Section 4.8.

The Hyper-G server stores references to resources on remote Gopher and WWW servers as objects of type Remote, whereby attributes of the object specify the remote resource’s host, port, access protocol, and so forth. This allows the seamless incorporation of information on remote non-Hyper-G servers. Gopher menus are transformed into Hyper-G collections. Interoperability with Z39.50 and ftp clients and servers is planned.

Unlike Gopher or WWW clients which connect to many servers during a typical session, Hyper-G clients talk to a single Hyper-G server for the entire session, typically local to the user’s organisation. Should information from a remote server be needed, the local server acts as a proxy, i.e. it fetches the object and passes it on to the client. This approach has the following advantages:

- Remote information can be cached in the local server for all local users.

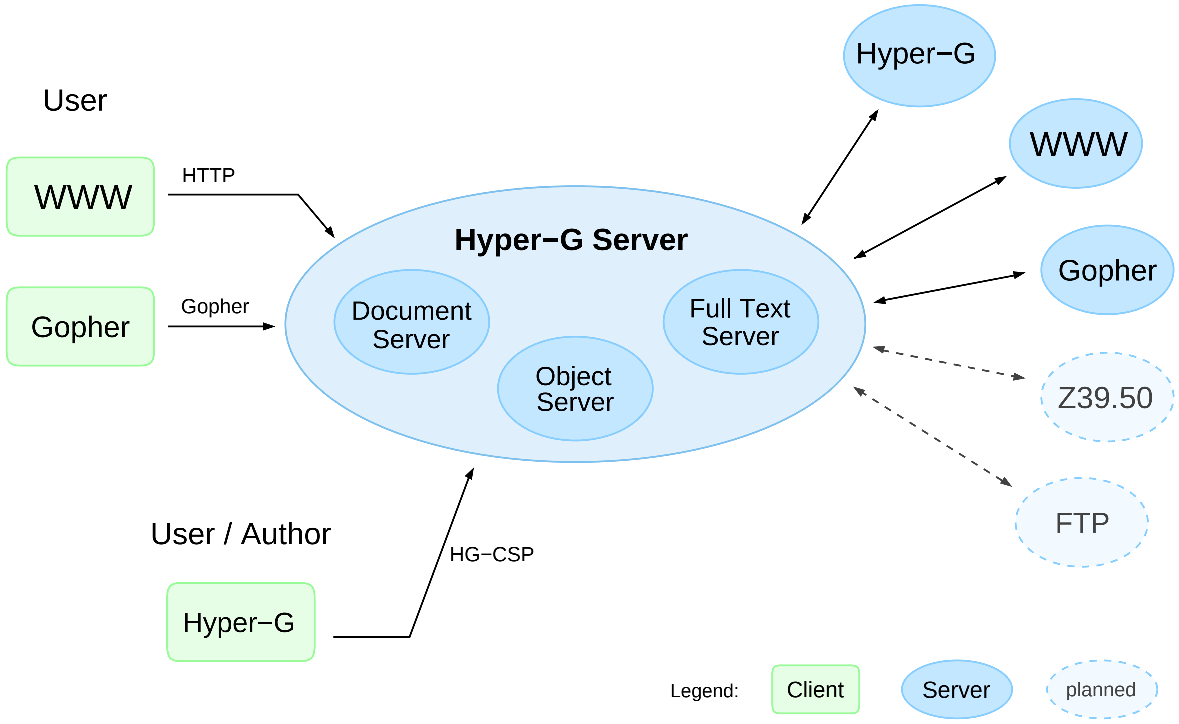


Figure 4.2 The client-server architecture of Hyper-G.

Command	Purpose
GETCHILDREN	get member IDs of given collection
GETOBJECTBYQUERY	get object IDs matching given query
GETANCHORS	get anchor IDs attached to given doc
GETOBJECT	get object record with given ID
GETANDLOCK	get object record and lock
CHANGEOBJECT	modify object attributes
UNLOCK	release lock on given object
PIPEDOCUMENT	retrieve document on given port
INSERTOBJECT	insert new document, collection, or link
DELETEOBJECT	delete object with given ID
COMMAND	status, who, message, etc.

Table 4.1 Some of the commands available in HG-CSP, Hyper-G's client-server protocol.

- User accounts and access rights have only to be maintained on the local server (the user has to identify to one server only).
- Statistics and user profile information can be gathered on a per-session basis.

Hyper-G clients and servers communicate using the Hyper-G Client/Server Protocol (HG-CSP) [Kappe and Pani, 1994]. A Hyper-G client opens a TCP connection to a Hyper-G server using the assigned port number 418 by default. Following protocol version and format (binary or ascii, with or

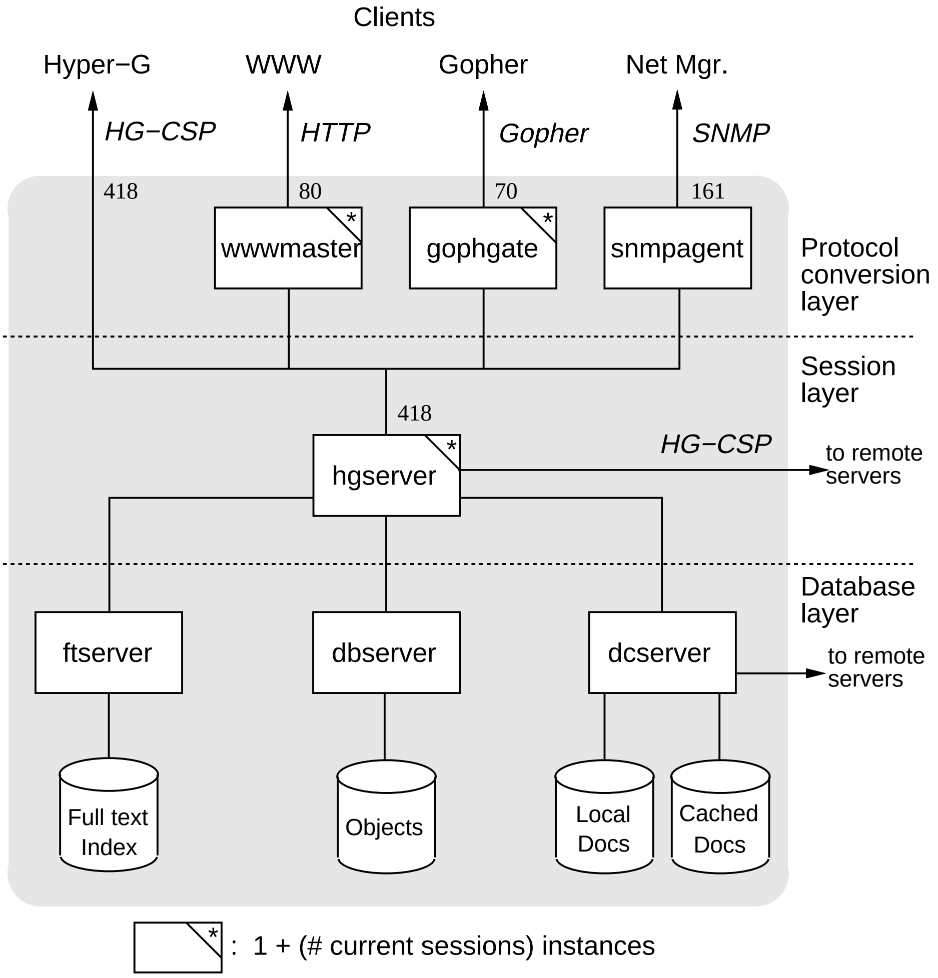


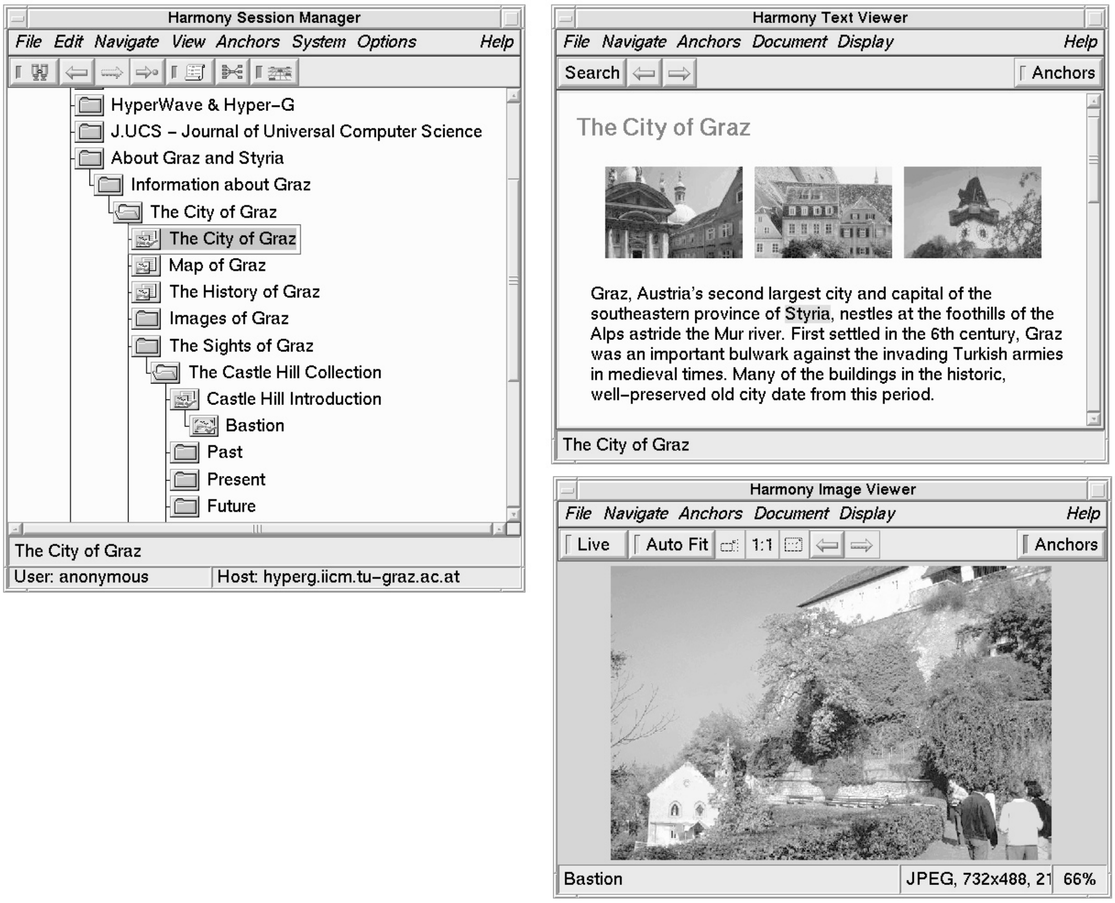
Figure 4.3 The architecture of the Hyper-G server.

without compression) negotiation, messages are exchanged using semantics similar to remote procedure calls (RPCs). Table 4.1 lists some of the commands available in HG-CSP. Control information is sent via port 418, documents are usually sent through separate (parallel) channels.

4.5 The Hyper-G Server

The Hyper-G server comprises many component modules in three layers of functionality, as shown in Figure 4.3 (for a fuller description, the reader is referred to [Maurer, 1996]). The protocol conversion layer transforms other protocols into Hyper-G's HG-CSP, so that lower layers only need to handle one protocol. In addition to the HTTP, Gopher, and SNMP gateways shown in the figure, work is underway on FTP and Z39.50 gateways. The session layer parallelises client requests and maintains state information.

The database layer provides low-level database functionality. It consists of three components: a document manager (`dcserver`), an object manager (`dbserver`), and a full text inverted index (`ftserver`). Hyperlink, collection, and document metadata are stored as objects by the object manager. Hyperlinks themselves are realised internally as relations between objects, as is collection membership. Hence, the object manager can be thought of as a cross between an object-oriented and a relational database.



User: anonymous

Host: hyperg.iicm.tu-graz.ac.at

Graz, Austria's second largest city and capital of the southeastern province of Styria, nestles at the foothills of the Alps astride the Mur river. First settled in the 6th century, Graz was an important bulwark against the invading Turkish armies in medieval times. Many of the buildings in the historic, well–preserved old city date from this period.

Figure 4.4 The Harmony client and authoring tool for X Windows.

In order to guarantee consistency of links and collection membership, Hyper-G servers also communicate with one another. The Server-Server protocol uses a scalable flooding algorithm (P-Flood [Kappe, 1995c]) to propagate update messages between Hyper-G servers.

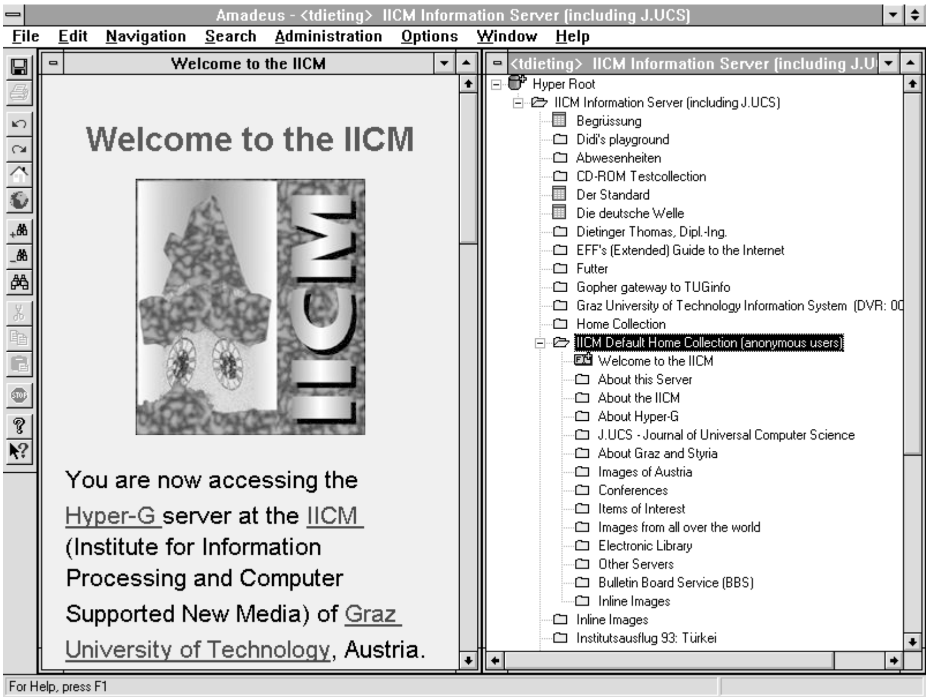
4.6 Hyper-G Clients

The most fully featured Hyper-G client and authoring tool is Harmony for X Windows, shown in Figure 4.4. Harmony takes advantage of Hyper-G's underlying facilities to provide intuitive navigational aids, informative feedback about the location of information, and a full suite of authoring tools. Harmony is covered extensively in Chapters 5 and 6 of this thesis.

Figure 4.5 shows the Amadeus client and authoring tool for Microsoft Windows platforms. To the right is the Tree View, a browser for hierarchical style navigation in the Hyper-G collection hierarchy, modeled after Harmony's collection browser. To the right is a text document containing an inline image and hyperlinks. Figure 4.6 illustrates Amadeus' editing capabilities; here the user is about to select Netscape Gold as the editing tool for car rental text document.

The Hyper-G Terminal Viewer `hgtv`, is a simple terminal-based client for UNIX platforms. It has the advantage, however, that it can be started automatically in response to a telnet connection, hence providing terminal access to the entire functionality of a Hyper-G server[1]. Figure 4.7 shows

[1] For an example, try opening a telnet connection to `www.tu-graz.ac.at`.

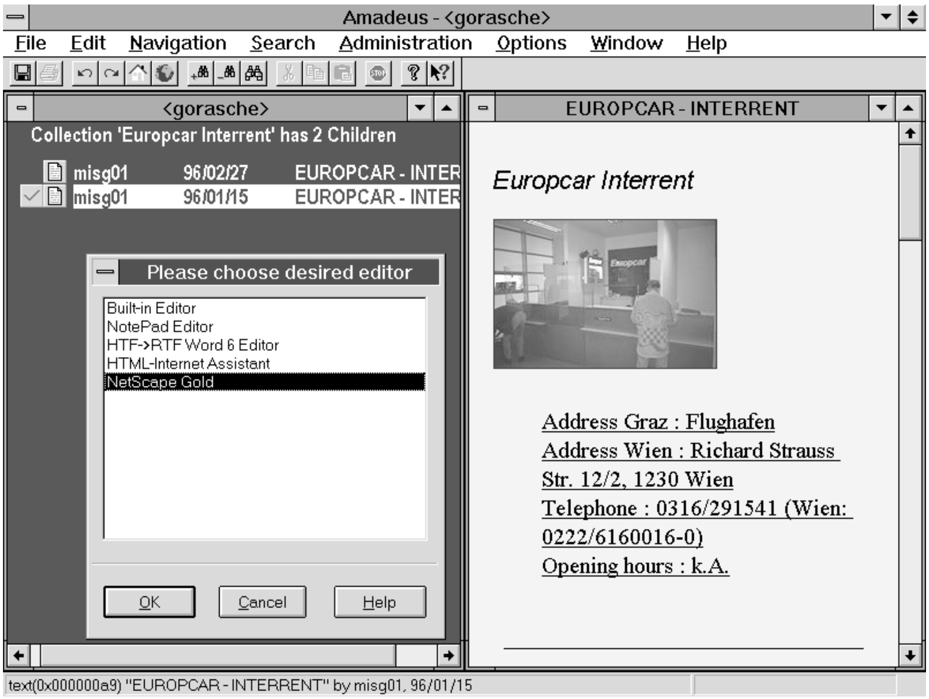

Amadeus - <tdieting> IICM Information Server (including J.UCS)
File Edit Navigation Search Administration Options Window Help
Welcome to the IICM
<tdieting> IICM Information Server (including J.U
For Help, press F1

Welcome to the IICM

You are now accessing the Hyper-G server at the IICM (Institute for Information Processing and Computer Supported New Media) of Graz University of Technology, Austria.

- Hyper Root
  - IICM Information Server (including J.UCS)
    - Begrüssung
    - Didi's playground
    - Abwesenheiten
    - CD-ROM Testcollection
    - Der Standard
    - Die deutsche Welle
    - Dietinger Thomas, Dipl.-Ing.
    - EFF's (Extended) Guide to the Internet
    - Futter
    - Gopher gateway to TUGinfo
    - Graz University of Technology Information System (DVR: 00
    - Home Collection
    - IICM Default Home Collection (anonymous users)
      - Welcome to the IICM
      - About this Server
      - About the IICM
      - About Hyper-G
      - J.UCS - Journal of Universal Computer Science
      - About Graz and Styria
      - Images of Austria
      - Conferences
      - Items of Interest
      - Images from all over the world
      - Electronic Library
      - Other Servers
      - Bulletin Board Service (BBS)
      - Inline Images
    - Inline Images
    - Institutsausflug 93: Türkei

Figure 4.5 The Amadeus client and authoring tool for Microsoft Windows.


Amadeus - <gorasche>
File Edit Navigation Search Administration Options Window Help
<gorasche>
EUROPCAR - INTERRENT
Please choose desired editor
OK
Cancel
Help

Collection 'Europcar Interrent' has 2 Children

misg01	96/02/27	EUROPCAR - INTER
misg01	96/01/15	EUROPCAR - INTER

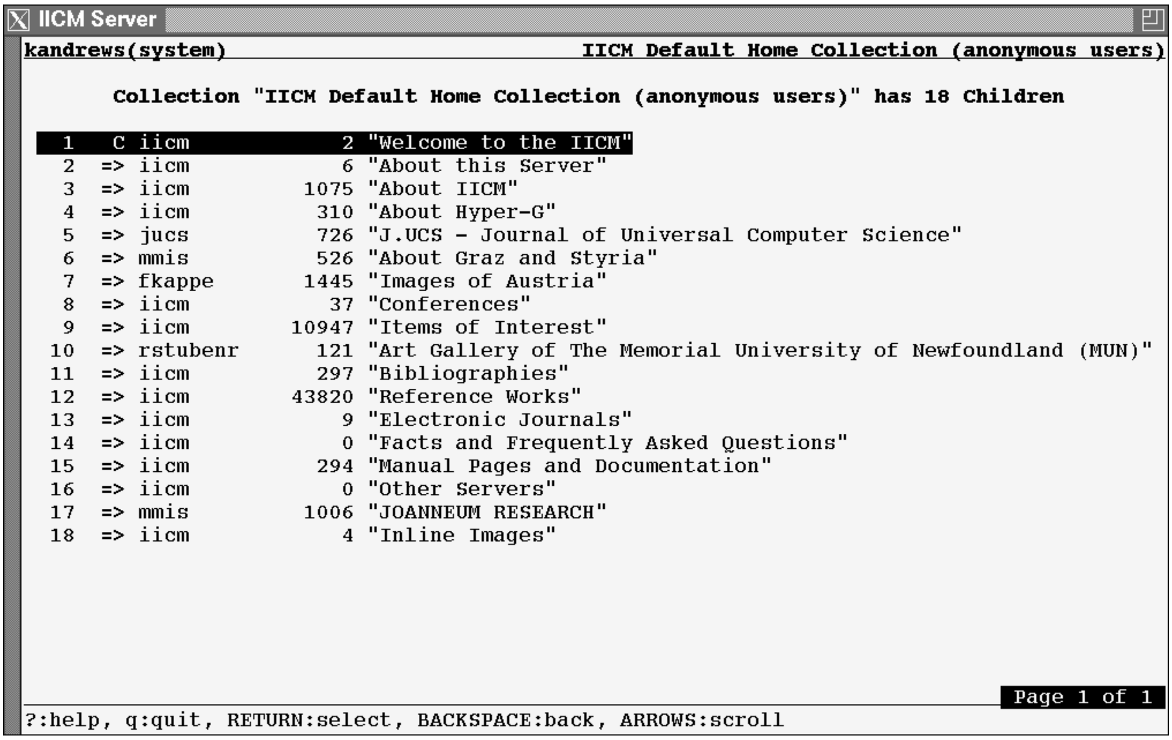
Built-in Editor
NotePad Editor
HTF->RTF Word 6 Editor
HTML-Internet Assistant
NetScape Gold

Europcar Interrent

Address Graz : Flughafen
Address Wien : Richard Strauss Str. 12/2, 1230 Wien
Telephone : 0316/291541 (Wien: 0222/6160016-0)
Opening hours : k.A.

text(0x000000a9) "EUROPCAR - INTERRENT" by misg01, 96/01/15

Figure 4.6 Editing a HTML text document through Amadeus.



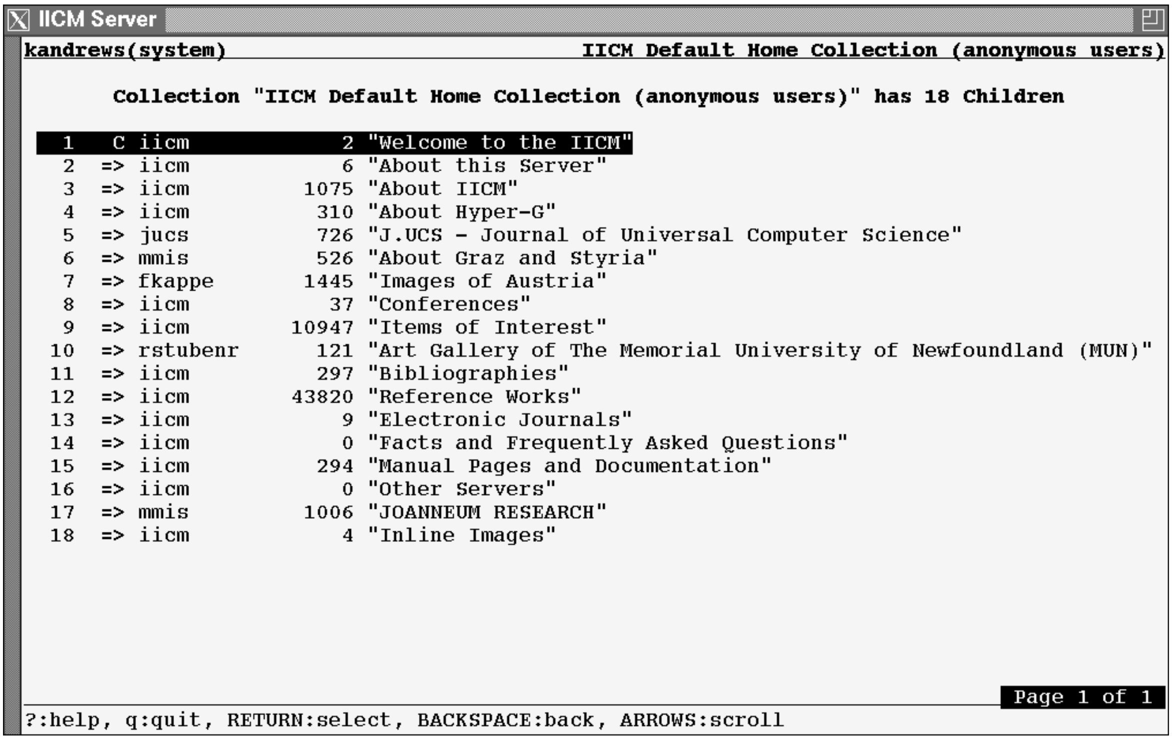
```
kandrews(system)                              IICM Default Home Collection (anonymous users)

      Collection "IICM Default Home Collection (anonymous users)" has 18 Children

  1   C iicm            2 "Welcome to the IICM"
  2  => iicm            6 "About this Server"
  3  => iicm         1075 "About IICM"
  4  => iicm          310 "About Hyper-G"
  5  => jucs          726 "J.UCS - Journal of Universal Computer Science"
  6  => mmis          526 "About Graz and Styria"
  7  => fkappe       1445 "Images of Austria"
  8  => iicm           37 "Conferences"
  9  => iicm        10947 "Items of Interest"
 10  => rstubenr      121 "Art Gallery of The Memorial University of Newfoundland (MUN)"
 11  => iicm          297 "Bibliographies"
 12  => iicm        43820 "Reference Works"
 13  => iicm            9 "Electronic Journals"
 14  => iicm            0 "Facts and Frequently Asked Questions"
 15  => iicm          294 "Manual Pages and Documentation"
 16  => iicm            0 "Other Servers"
 17  => mmis         1006 "JOANNEUM RESEARCH"
 18  => iicm            4 "Inline Images"

                                                                           Page 1 of 1
?:help, q:quit, RETURN:select, BACKSPACE:back, ARROWS:scroll
```

Figure 4.7 The Hyper-G Terminal Viewer `hgtv`. A simple terminal-based Hyper-G client which can be used for example across a telnet connection. Here a Hyper-G collection is being displayed.



IICM Server

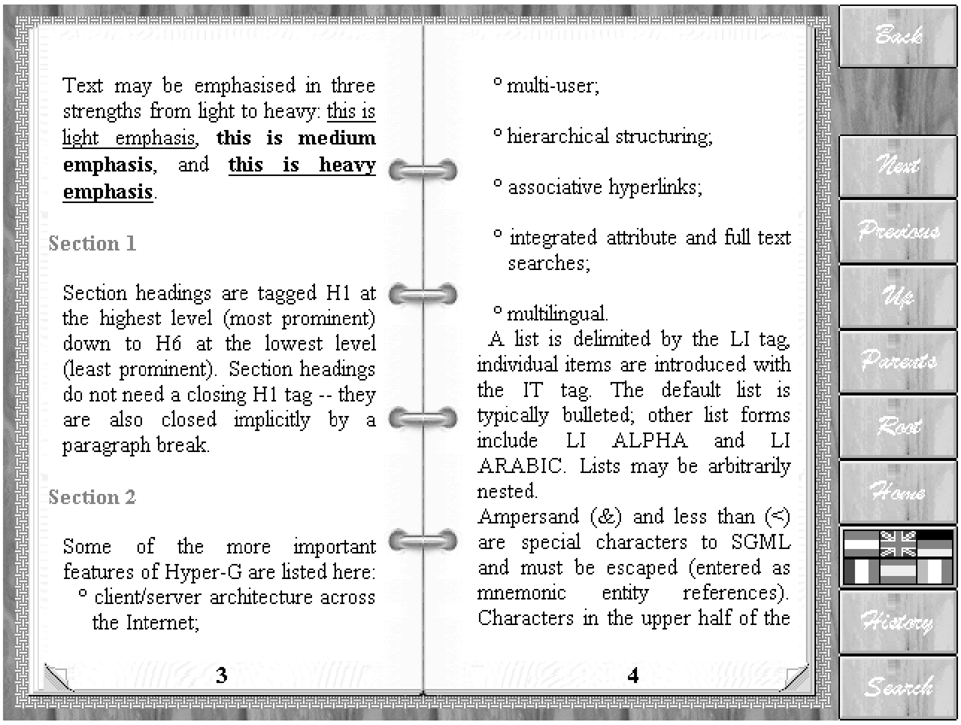
```
kandrews(system)                              IICM Default Home Collection (anonymous users)

      Collection "IICM Default Home Collection (anonymous users)" has 18 Children

  1   C iicm            2 "Welcome to the IICM"
  2  => iicm            6 "About this Server"
  3  => iicm         1075 "About IICM"
  4  => iicm          310 "About Hyper-G"
  5  => jucs          726 "J.UCS - Journal of Universal Computer Science"
  6  => mmis          526 "About Graz and Styria"
  7  => fkappe       1445 "Images of Austria"
  8  => iicm           37 "Conferences"
  9  => iicm        10947 "Items of Interest"
 10  => rstubenr      121 "Art Gallery of The Memorial University of Newfoundland (MUN)"
 11  => iicm          297 "Bibliographies"
 12  => iicm        43820 "Reference Works"
 13  => iicm            9 "Electronic Journals"
 14  => iicm            0 "Facts and Frequently Asked Questions"
 15  => iicm          294 "Manual Pages and Documentation"
 16  => iicm            0 "Other Servers"
 17  => mmis         1006 "JOANNEUM RESEARCH"
 18  => iicm            4 "Inline Images"

                                                                           Page 1 of 1
?:help, q:quit, RETURN:select, BACKSPACE:back, ARROWS:scroll
```

Figure 4.8 A text document displayed by `hgtv`. Note that hyperlinks are followed by typing numbers, since simple terminals do not support mouse clicking.



Text may be emphasised in three strengths from light to heavy: this is light emphasis, **this is medium emphasis**, and **this is heavy emphasis**.

Section 1

Section headings are tagged H1 at the highest level (most prominent) down to H6 at the lowest level (least prominent). Section headings do not need a closing H1 tag -- they are also closed implicitly by a paragraph break.

Section 2

Some of the more important features of Hyper-G are listed here:

- client/server architecture across the Internet;
- multi-user;
- hierarchical structuring;
- associative hyperlinks;
- integrated attribute and full text searches;
- multilingual.

A list is delimited by the LI tag, individual items are introduced with the IT tag. The default list is typically bulleted; other list forms include LI ALPHA and LI ARABIC. Lists may be arbitrarily nested.
Ampersand (&) and less than (<) are special characters to SGML and must be escaped (entered as mnemonic entity references). Characters in the upper half of the

Figure 4.9 Easy, Hyper-G's public terminal viewer, designed with large interface buttons for touch-screen use.

`hgtv` displaying one level of a Hyper-G collection hierarchy. Figure 4.8 shows a text document. Note that since simple terminals do not support mouse clicking, links are activated by typing the number of the link.

Another kind of Hyper-G client is Easy, the Hyper-G public terminal viewer, shown in Figure 4.9. Running on Microsoft Windows platforms, Easy implements an interface designed for use by the general public at information kiosks or terminals with a touch-screen interface. It interprets layout and synchronisation attributes and has a scripting language called SyncScript.

4.7 Hyper-G Tools and Utilities

Server administrators have a wide range of tools available to insert, maintain, and manipulate the information content of a Hyper-G server. Utility programs, converters, and perl scripts are available to insert pre-existing data in a variety of formats. Table 4.2 gives an overview of some of these tools. HIF, the Hyper-G Interchange Format [Kappe, 1995a], is a clear text format used to encode entire Hyper-G collections, including subcollections, documents, links, and their metadata. The `hifexport` and `hifimport` utilities are used to archive and transport entire collections.

`loadwww` is a tool to "roll over" a WWW server tree into a Hyper-G collection tree. Directories are mapped to Hyper-G collections, documents are uploaded, and links are entered into Hyper-G's link database. Most importantly, original URLs remain valid, assuming that the host name remains the same.

A standalone Hyper-G server, called the "local database" implements a subset of Hyper-G server functionality. It is intended for offline reading and authoring. For example, a local database containing a presentation, a catalogue, or conference proceedings can be put onto a CD-ROM for distribution.

Tool	Purpose
hginfo	output information about objects
hgmodify	modify attributes of objects
hginscoll	insert new collection
hginsdoc/hginstext	insert new document
hggetdata	retrieve document
hgdelobj	delete object
hgerase	recursively delete subtree
hifexport	export entire collection (HIF format)
hifimport	import entire collection (HIF format)
loadwww	load WWW directory tree into Hyper-G server
loadgopher	load Gopher directory tree into Hyper-G server
hgw3stat	statistics generator
hgbackup	backup entire server
hgrestore	restore entire server from backup

Table 4.2 Some of the low-level tools and utility scripts available to Hyper-G server administrators.

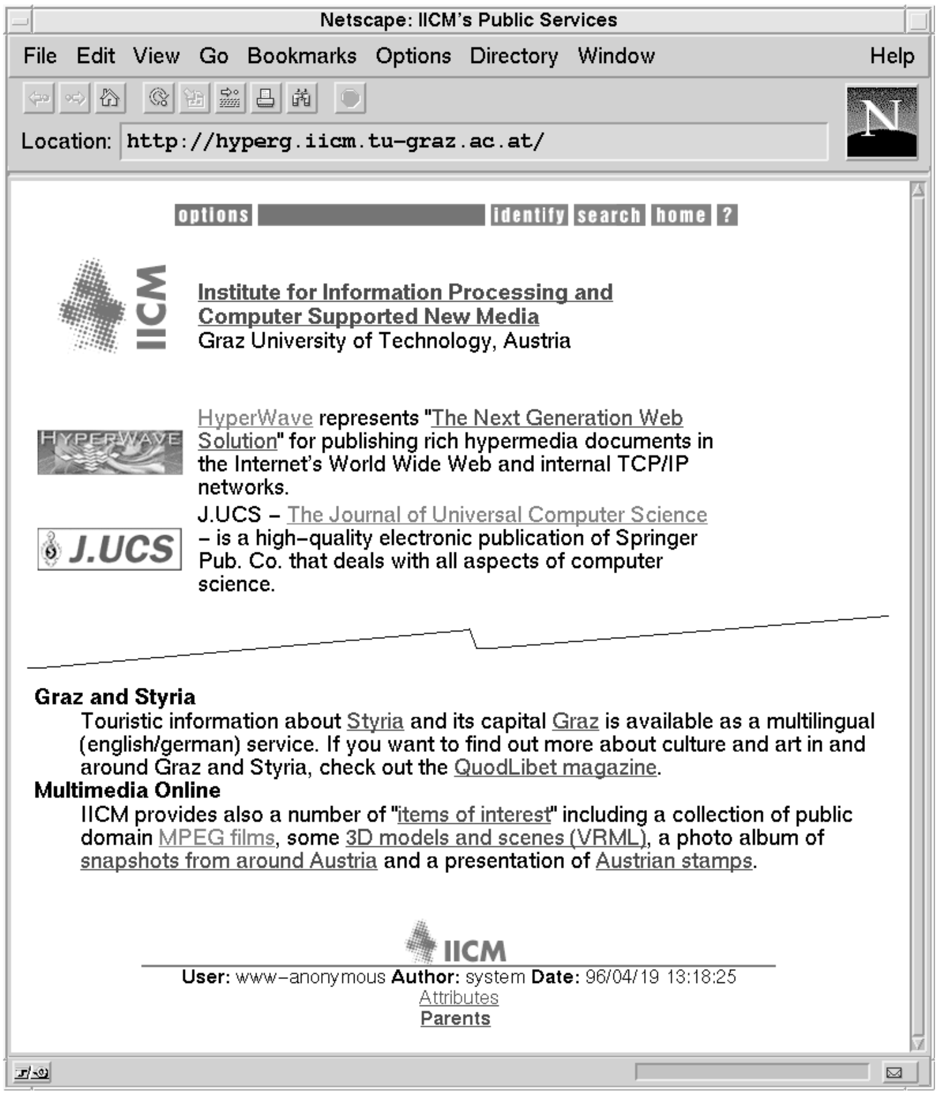
4.8 Serving to the Web with Hyper-G

Although Hyper-G can be considered a complete Internet information system in its own right, its primary use is as a sophisticated Web authoring, serving, and collaborative environment. In this section, we will discuss how a Hyper-G server presents itself to standard Web browsers such as Netscape, Mosaic, or Internet Explorer.

Information on a Hyper-G server is usually structured first into (overlapping) collections and subcollections; both collections and documents may be *re-used* by virtue of their membership of multiple collections. Hyperlinks are used for cross-references, orthogonal to the collection structure. Most Internet browsers today speak multiple protocols, usually at least HTTP, Gopher, and FTP. The Hyper-G server is a multi-protocol server. It responds to access requests with an appropriate representation of the requested information, preserving as much as possible of the rich functionality of Hyper-G in the process. When accessed using Gopher protocol, the Hyper-G server presents its collection hierarchy as a Gopher menu tree (hyperlinks cannot be represented in Gopher). A synthetic search item is generated at the foot of each Gopher menu to allow searching the corresponding collection. When accessed using HTTP, each level of the collection hierarchy is converted to an HTML document containing a menu of links to its members. Hyperlinks are merged into HTML text documents and VRML 3d models on-the-fly at the appropriate places. Additional Hyper-G functionality such as user identification, language preference selection, and searching is implemented via HTML forms which are accessible at any time. Java applets will soon supplement forms-based access to Hyper-G functionality.

Figure 4.10 shows the welcome page of the IICM Information Server through the eyes of Netscape. Access to Hyper-G functionality and contextual and meta information are wrapped around the document proper. Each HTML page passed to the client typically comprises four logical component areas:

- The *Menu Bar* at the top of the page has four buttons, Options, Search, Help, and Home, respectively.
- The *Text Area* displays a text document (and also the head of a collection, if present). Images within a cluster are displayed as inline images.



Netscape: IICM's Public Services
File Edit View Go Bookmarks Options Directory Window Help
Location: http://hyperg.iicm.tu-graz.ac.at/
options identify search home ?

Institute for Information Processing and Computer Supported New Media
Graz University of Technology, Austria

HyperWave represents "The Next Generation Web Solution" for publishing rich hypermedia documents in the Internet's World Wide Web and internal TCP/IP networks.

J.UCS – The Journal of Universal Computer Science – is a high-quality electronic publication of Springer Pub. Co. that deals with all aspects of computer science.

Graz and Styria
Touristic information about Styria and its capital Graz is available as a multilingual (english/german) service. If you want to find out more about culture and art in and around Graz and Styria, check out the QuodLibet magazine.

Multimedia Online
IICM provides also a number of "items of interest" including a collection of public domain MPEG films, some 3D models and scenes (VRML), a photo album of snapshots from around Austria and a presentation of Austrian stamps.

User: www-anonymous Author: system Date: 96/04/19 13:18:25
Attributes
Parents

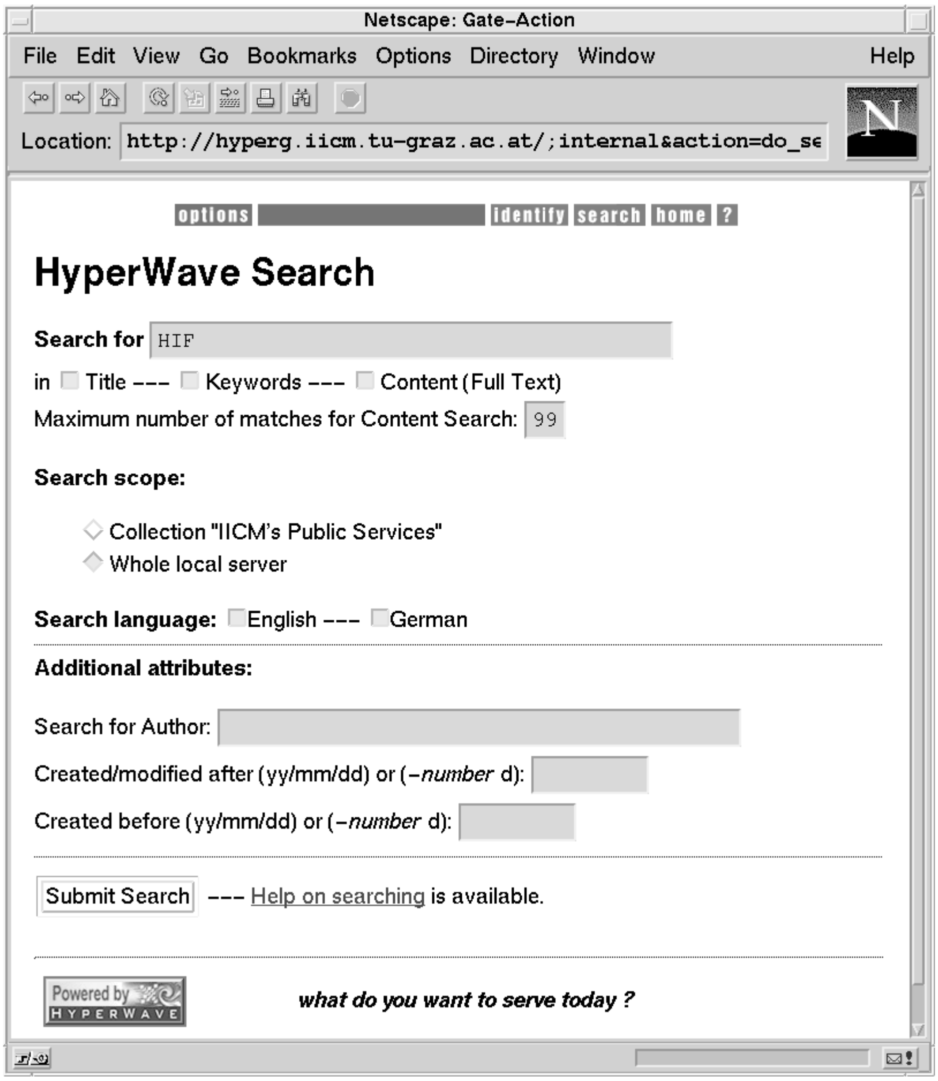
Figure 4.10 The IICM welcome page, seen through the eyes of Netscape. The welcome document is a full collection head, containing embedded links to its siblings and children, hence the collection area is not present (the middle of the page has been clipped out for a better overview).

- The *Collection Area* is present only for collections without a Full Collection Head and displays links to the other members (documents, clusters, and subcollections) of the collection.
- The *Attributes Area* at the bottom of the page provides meta-information about the document or collection, such as author, creation time, access rights, etc. Often, a list of parent collections is also provided here.

Note that the presence and appearance of these areas is highly configurable in the Hyper-G server's WWWgateway, `wavemaster`, using a kind of programmable meta-HTML language called PLACE [PLACE, 1996] [2]. Hence the look of Web servers running Hyper-G can vary widely from server to server depending on their administrator's policy and preferences. Figures 4.14 and 4.15 show, for example, the European Space Agency's Earth Observation Guide and Directory Service.

The Hyper-G Search panel shown in Figure 4.11 is the interface to Hyper-G's powerful attribute and content search facilities. Searches may be restricted in scope to the current collection (and its

[2] The older gateway, `wwwmaster`, was also configurable, but not to the same extent as `wavemaster` is.



Location: http://hyperg.iicm.tu-graz.ac.at/;internal&action=do_se

HyperWave Search

Search for HIF

in □ Title --- □ Keywords --- □ Content (Full Text)

Maximum number of matches for Content Search: 99

Search scope:

◇ Collection "IICM's Public Services"

◆ Whole local server

Search language: □ English --- □ German

Additional attributes:

Search for Author:

Created/modified after (yy/mm/dd) or (–*number* d):

Created before (yy/mm/dd) or (–*number* d):

--- Help on searching is available.

what do you want to serve today ?

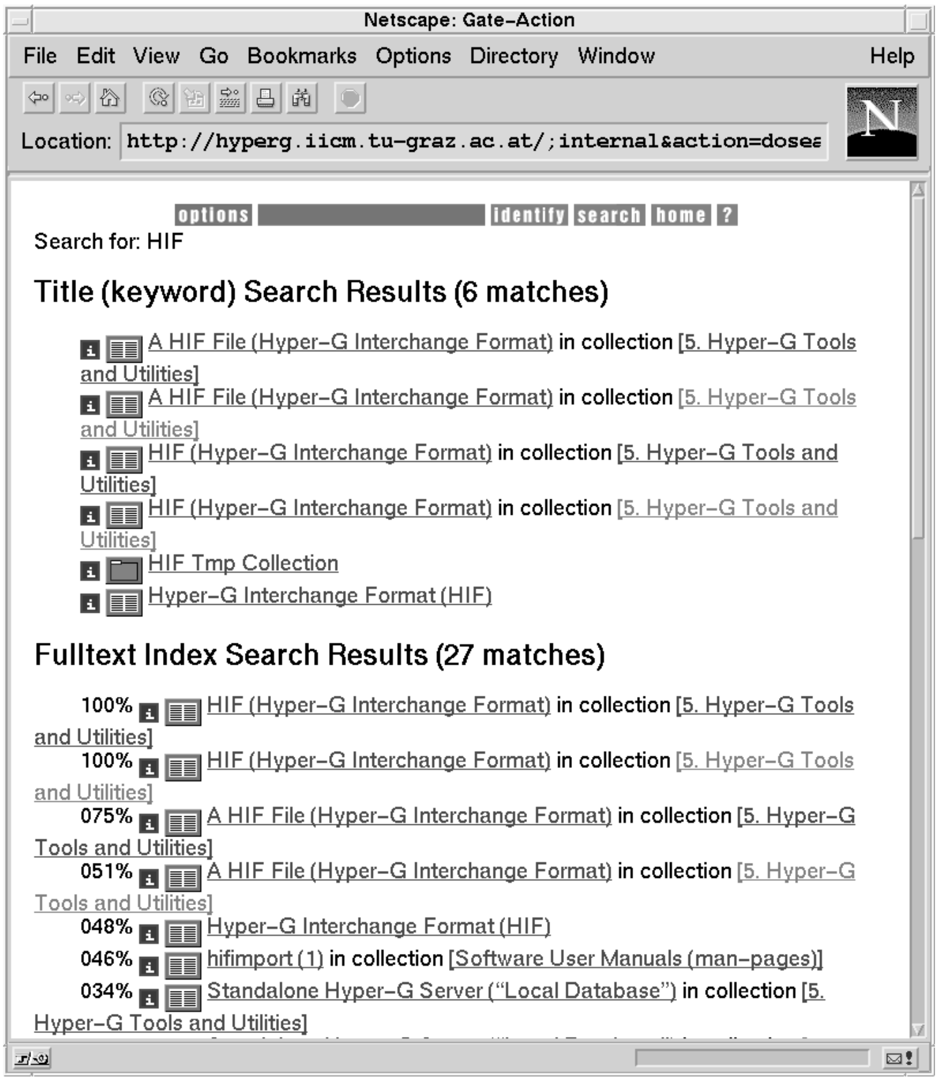
Figure 4.11 The Hyper-G search panel.

subcollections), or may span the whole local server. Here the user is searching for information on Hyper-G's HIF interchange format, in both the title and keyword attribute fields and the textual content of documents. Figure 4.12 shows the list of matching objects returned by the server. Further functionality is embodied by the Hyper-G Options panel, another HTML form, where users can see who else is online, change their language preference, and so forth.

The Hyper-G server stores a reference to a remote (non-Hyper-G) resource, such as a document accessible by HTTP, Gopher, or FTP, as a Remote Object of the appropriate type. The object's attributes contain the host, port, and location of the resource. This allows the seamless incorporation of information on remote, non-Hyper-G servers.

Arbitrary Java applets (complied byte code) may be uploaded and stored on a Hyper-G server in the same way as other documents, and have a full set of attribute metadata. HTML documents reference them by name or ID in the applet tag.

The standard way of bolting extensions to WWW servers is the Common Gateway Interface (CGI). Put simply, CGI is a naming convention for a set of environment variables which pass information and parameters from the calling URL into a program started at the server. The CGI program itself can be written in any programming language, although often a scripting language such as Perl is used. The CGI program generates its output as a HTML page, which is returned back to the request-


Netscape: Gate–Action
File Edit View Go Bookmarks Options Directory Window Help
Location:
options identify search home ?

Location: http://hyperg.iicm.tu-graz.ac.at/;internal&action=dosea

Search for: HIF

Title (keyword) Search Results (6 matches)

- A HIF File (Hyper–G Interchange Format) in collection [5. Hyper–G Tools and Utilities]
- A HIF File (Hyper–G Interchange Format) in collection [5. Hyper–G Tools and Utilities]
- HIF (Hyper–G Interchange Format) in collection [5. Hyper–G Tools and Utilities]
- HIF (Hyper–G Interchange Format) in collection [5. Hyper–G Tools and Utilities]
- HIF Tmp Collection
- Hyper–G Interchange Format (HIF)

Fulltext Index Search Results (27 matches)

100% HIF (Hyper–G Interchange Format) in collection [5. Hyper–G Tools and Utilities]
100% HIF (Hyper–G Interchange Format) in collection [5. Hyper–G Tools and Utilities]
075% A HIF File (Hyper–G Interchange Format) in collection [5. Hyper–G Tools and Utilities]
051% A HIF File (Hyper–G Interchange Format) in collection [5. Hyper–G Tools and Utilities]
048% Hyper–G Interchange Format (HIF)
046% hifimport (1) in collection [Software User Manuals (man–pages)]
034% Standalone Hyper–G Server (“Local Database”) in collection [5. Hyper–G Tools and Utilities]

Figure 4.12 Hyper-G search results. Full text matches are ranked in decreasing order of relevance.

ing client. Among other things, CGI programs are used to parse the results of a HTML form and to provide back-end access to external databases. The Hyper-G server provides full support for CGI.

One of the problems with CGI, however, is that a new process is typically started for each and every CGI call. Particularly when integrating external database packages, this entails substantial overhead. For this reason, the Hyper-G server has a special back-end interface called the Hyper-G Gateway Interface (HGI) [Reisinger and Faschingbauer, 1996], shown schematically in Figure 4.13. HGI allows the server to hold a persistent connection to an external source. A Perl 5 module is available for writing arbitrary HGI gateways and a sample, HGI-conformant SQL gateway is supplied.

4.9 Applications of Hyper-G

The structuring and ease of maintenance of *large* information bases is one of the major strengths of Hyper-G. The IICM Information Server `www.iicm.edu` currently (September 1996) holds about 213 000 documents, organised in almost 40 000 clusters and collections, with 290 000 links, and 905 users organised into 26 user groups – and runs on a PC under Linux. The information resources in-

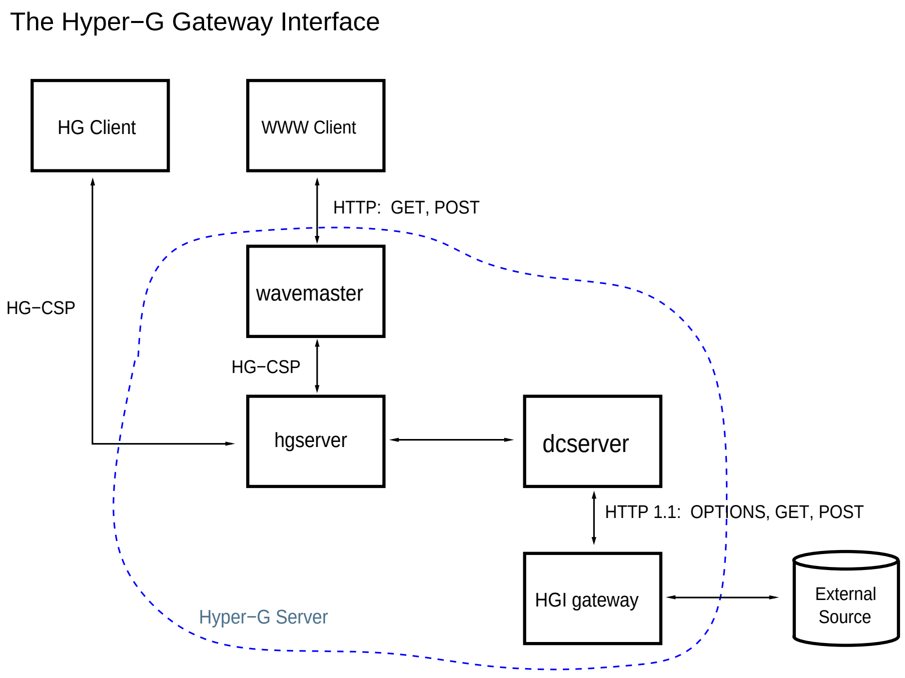


Figure 4.13 The Hyper-G Gateway Interface (HGI).

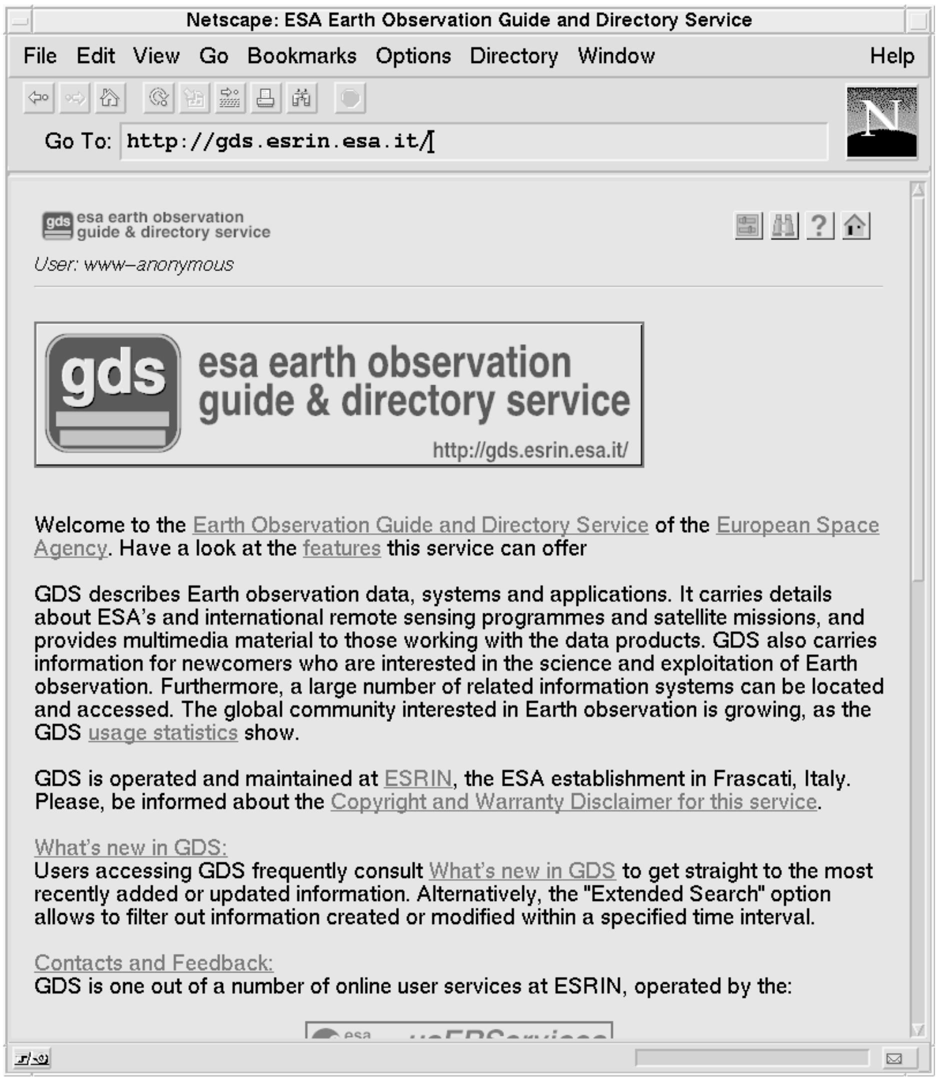
clude online documentation about Hyper-G and HyperWave, information about Graz and Austria, and the complete Meyers Lexikon (a German-language encyclopedia). Graz University of Technology's Hyper-G Information Server contains personnel and course information as well as the complete ACM SIGGRAPH and HCI bibliographies (fully searchable).

The European Space Agency (ESA) uses Hyper-G to organise and publish its earth observation data. Figure 4.14 shows the welcome page of the ESA Earth Observation Guide and Directory Service (GDS), which currently comprises some 88 000 documents. The look and feel of the Hyper-G server is highly customisable – note the replacement of the standard Hyper-G icons by customised ESA versions. The Museum of New Zealand and numerous other companies and institutions have adopted Hyper-G for their own information systems. The German Mathematics Association (DMV) has set up a Hyper-G-based information system spanning several German universities and colleges.

Hyper-G also is being used as the basis for major new electronic publishing ventures. The Journal of Universal Computer Science (J.UCS), supported by Springer Verlag, is the first high-quality, fully-refereed, fully-citable scientific journal to depend primarily on Internet distribution [Maurer and Schmaranz, 1994]. J.UCS went online in January 1995, and has appeared monthly since. One of the largest Austrian daily newspapers, the Kleine Zeitung, uses Hyper-G for its online edition [Medienhaus Styria, 1996].

4.10 Further Reading

The most complete guide to Hyper-G and HyperWave is the recent book *HyperWave: The Next Generation Web Solution* [Maurer, 1996]. A more tutorial-like treatment is given in *HyperWave: The New Generation Internet Information System Based on Hyper-G Technology* [Dalitz and Heyer, 1996]. Good overview articles have also appeared in BYTE [Flohr, 1995], New Media [Sengstack, 1996], and Computer Shopper [Aubrey, 1996] magazines.



Netscape: ESA Earth Observation Guide and Directory Service
File Edit View Go Bookmarks Options Directory Window Help
Go To: http://gds.esrin.esa.it/
gds esa earth observation guide & directory service
User: www–anonymous
gds esa earth observation guide & directory service
http://gds.esrin.esa.it/

Welcome to the Earth Observation Guide and Directory Service of the European Space Agency. Have a look at the features this service can offer

GDS describes Earth observation data, systems and applications. It carries details about ESA's and international remote sensing programmes and satellite missions, and provides multimedia material to those working with the data products. GDS also carries information for newcomers who are interested in the science and exploitation of Earth observation. Furthermore, a large number of related information systems can be located and accessed. The global community interested in Earth observation is growing, as the GDS usage statistics show.

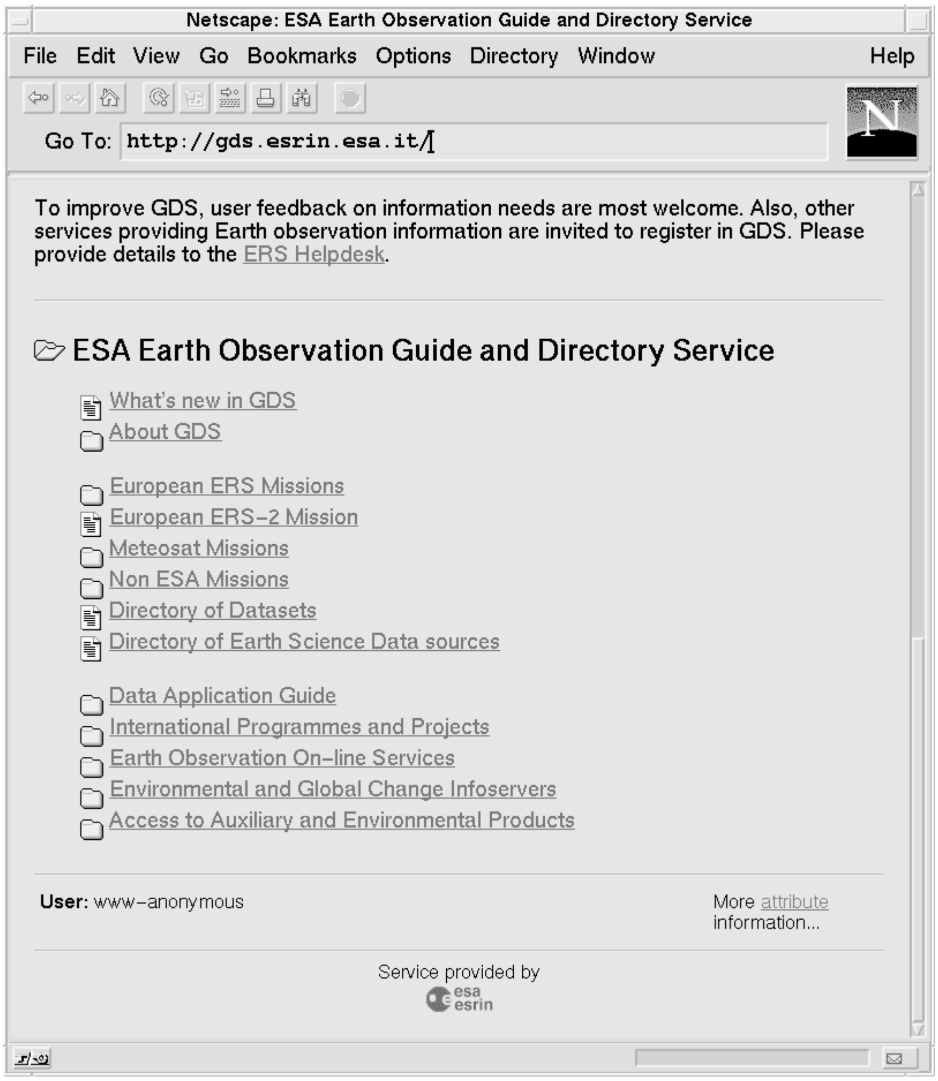
GDS is operated and maintained at ESRIN, the ESA establishment in Frascati, Italy. Please, be informed about the Copyright and Warranty Disclaimer for this service.

What's new in GDS:
Users accessing GDS frequently consult What's new in GDS to get straight to the most recently added or updated information. Alternatively, the "Extended Search" option allows to filter out information created or modified within a specified time interval.

Contacts and Feedback:
GDS is one out of a number of online user services at ESRIN, operated by the:

Figure 4.14 The European Space Agency's Earth Observation Guide and Directory Service (here the top half of the welcome page).

Further information about Hyper-G is available online at the IICM Information Server under `http://www.iicm.edu/hyperg`. Updated slides from my ACM Hypertext'96 course on Hyper-G [Andrews, 1996c] are available online at `http://www.iicm.edu/hgcourse`. The software itself is available by anonymous ftp from `ftp://ftp.iicm.edu/pub/Hyper-G`.


Netscape: ESA Earth Observation Guide and Directory Service
File Edit View Go Bookmarks Options Directory Window Help
Go To: http://gds.esrin.esa.it/

To improve GDS, user feedback on information needs are most welcome. Also, other services providing Earth observation information are invited to register in GDS. Please provide details to the ERS Helpdesk.

ESA Earth Observation Guide and Directory Service

- What's new in GDS
- About GDS

- European ERS Missions
- European ERS–2 Mission
- Meteosat Missions
- Non ESA Missions
- Directory of Datasets
- Directory of Earth Science Data sources

- Data Application Guide
- International Programmes and Projects
- Earth Observation On–line Services
- Environmental and Global Change Infoservers
- Access to Auxiliary and Environmental Products

User: www–anonymous

More attribute information...

Service provided by esa esrin

Figure 4.15 The European Space Agency's Earth Observation Guide and Directory Service (here the bottom half of the welcome page). Note the collection area, listing the members of the current collection.

5 Browsing Cyberspace with Harmony

This chapter presents the browsing and navigational features of the Harmony client for Hyper-G. It is based on material first published in Chapter 15 of *HyperWave: The Next Generation Web Solution*, Addison-Wesley, May 1996 [Andrews, 1996f].

5.1 Designing a Graphical Hypermedia Client

In the summer of 1991, when the World-Wide Web was still unknown, and only a simple terminal client `hgtv` existed for Hyper-G, I assumed responsibility at the IICM for designing, and leading a team to implement, a new graphical hypermedia client for Hyper-G, later to became known as *Harmony*. My intention was to create a powerful graphical browsing and authoring tool, which would bring the sophisticated information structuring and management concepts of Hyper-G to the reach of end users, both locally *and* across the Internet. This chapter deals with Harmony's browsing and navigational facilities. Chapter 6 describes Harmony's powerful editing, management, and structuring tools.

The technical merit of Harmony lies in its *integrated* suite of tools and in the *interplay* between them. As a member of the audience at my CHI'95 talk [Andrews *et al.*, 1995b] told me after the talk:

> *"One of the most impressive things about Harmony is that it all works!"*

Perhaps the most important scientific contribution of Harmony is *location feedback*, alleviating the "lost in hyperspace" problem by helping users retain a sense of position in large hypermedia information spaces. Location feedback is the term I coined to describe the synchronised, correlated, navigational views which Harmony provides. Selecting or activating an object (document or collection) in any of the local map, search result list, history browser, or by link following, *automatically* opens up the path(s) to that object in the collection structure, thus providing immediate contextual feedback. As Jakob Nielsen states in *Multimedia and Hypertext: The Internet and Beyond* [Nielsen, 1995]:

> *"This feature of Harmony is a powerful instrument in the fight against disorientation because users can orient themselves with reference to a fixed structural framework."*

Location feedback is discussed in Sections 5.5 and 5.6. A further important contribution is Harmony's support for multilingual information collections, both in terms of information content and user interface. This is described in Section 5.11 at the end of this chapter.

Since Hyper-G stores hyperlinks in a separate link database, rather than embedded within documents, documents of any media type can function as the source of links. From the beginning, Harmony was intended to take advantage of this feature, by providing viewers and interactive link creation facilities for at least six different types of media: text, images, video, audio, page description files (PostScript), and 3d models. In particular, Harmony's display and linking facilities for 3d models, as described in Chapter 7, pre-date equivalent facilities in VRML (Virtual Reality Modeling Language) [Pesce, 1995] by more than two years [Andrews, 1992].

The rich structure and extensive metadata made available by the Hyper-G server appeared to me to be the perfect infrastructure upon which to build visualisations of the information space, a field which has since become known as "information visualisation". Harmony's Local Map and Information Landscape are the results of my research in this direction and are described in detail in Chapter 8.

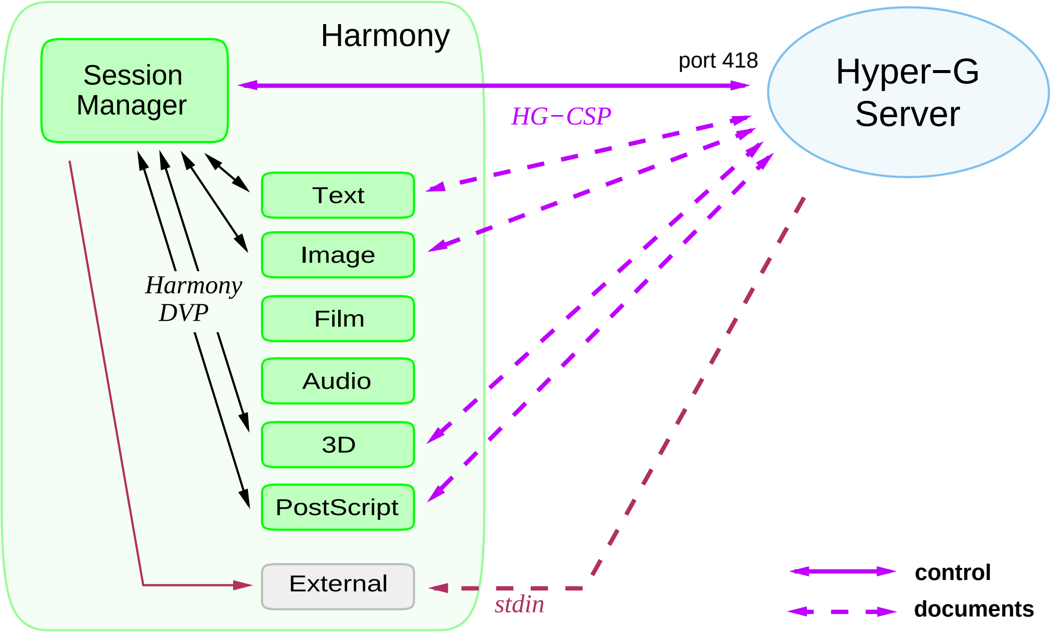


Figure 5.1 The architecture of Harmony client and authoring tool for Hyper-G.

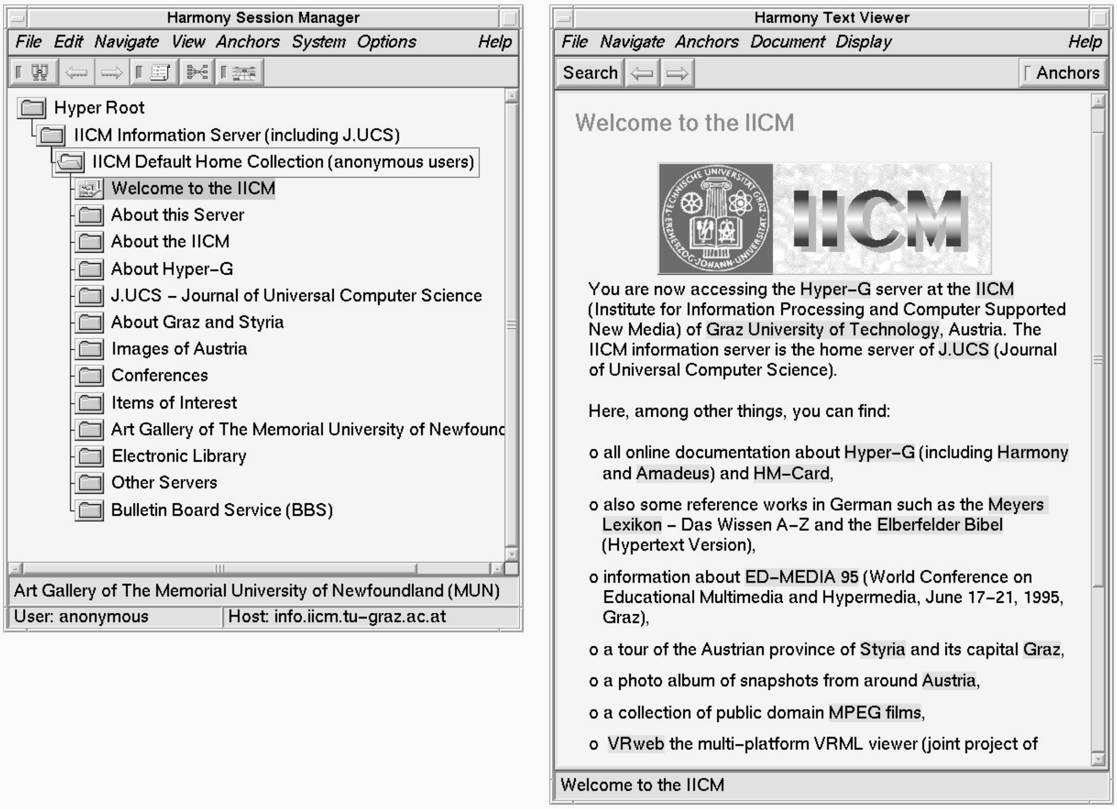
5.2 The Architecture of Harmony

Harmony was designed as a multi-process application, as illustrated in Figure 5.1. The Session Manager is the primary process; it communicates with the Hyper-G server, provides navigational facilities and coordinates all other activities. The Session Manager starts secondary processes, document viewers, of the appropriate type, to display documents as necessary. Native Harmony document viewers communicate with the Session Manager using Harmony's own protocol, the Harmony Document Viewer Protocol (DVP), which defines various browsing, editing, and link functions. There currently exist native Harmony document viewers for text, images, MPEG films, audio, 3d models, and PostScript. Figure 5.2 shows an example screen shot of the Harmony client.

Harmony may be configured to run external programs in place of any native viewer and also for unsupported document types (the document is piped to the standard input of the external program). However, since external programs do not "speak" Harmony's protocol, they are generally unable to support link activation and editing. A Harmony API is also available for external tools which make use of Harmony's functionality and state. One such tool is a generic edit facility, which can be configured to start an external editor for any document type.

The Harmony Session Manager connects to the Hyper-G server on Hyper-G's assigned port number 418. This connection is kept open for the duration of the session and is the channel through which control information (all communication except actual documents) passes in both directions. Documents themselves are not sent from the server via the Session Manager to a document viewer, but directly from server to document viewer. In the original design, the document cache in the Hyper-G server opened a connection back to the viewer, using a port dynamically assigned by the viewer. However, since certain network configurations (for example, machines behind a firewall or using an emulated SLIP connection) do not allow the server to connect back to the client on an arbitrary port, this connection procedure has been turned around so that the viewer connects to the server.

Harmony is written in C++ [Stroustrup, 1991]. Back in 1991, I chose to use the InterViews [Linton *et al.*, 1989] user interface toolkit from Stanford, primarily because it was freely available in source code with no licence restrictions. Over time, the original InterViews library has been so extensively modified, extended, and bug-fixed that we call the version used in Harmony "IICMViews" to distinguish the two!


Harmony Session Manager
File Edit Navigate View Anchors System Options Help
Harmony Text Viewer
File Navigate Anchors Document Display Help
Search
Anchors

Hyper Root
IICM Information Server (including J.UCS)
IICM Default Home Collection (anonymous users)
Welcome to the IICM
About this Server
About the IICM
About Hyper–G
J.UCS – Journal of Universal Computer Science
About Graz and Styria
Images of Austria
Conferences
Items of Interest
Art Gallery of The Memorial University of Newfounc
Electronic Library
Other Servers
Bulletin Board Service (BBS)

Art Gallery of The Memorial University of Newfoundland (MUN)
User: anonymous
Host: info.iicm.tu–graz.ac.at

Welcome to the IICM

You are now accessing the Hyper–G server at the IICM (Institute for Information Processing and Computer Supported New Media) of Graz University of Technology, Austria. The IICM information server is the home server of J.UCS (Journal of Universal Computer Science).

Here, among other things, you can find:

o all online documentation about Hyper–G (including Harmony and Amadeus) and HM–Card,

o also some reference works in German such as the Meyers Lexikon – Das Wissen A–Z and the Elberfelder Bibel (Hypertext Version),

o information about ED–MEDIA 95 (World Conference on Educational Multimedia and Hypermedia, June 17–21, 1995, Graz),

o a tour of the Austrian province of Styria and its capital Graz,

o a photo album of snapshots from around Austria,

o a collection of public domain MPEG films,

o VRweb the multi–platform VRML viewer (joint project of

Welcome to the IICM

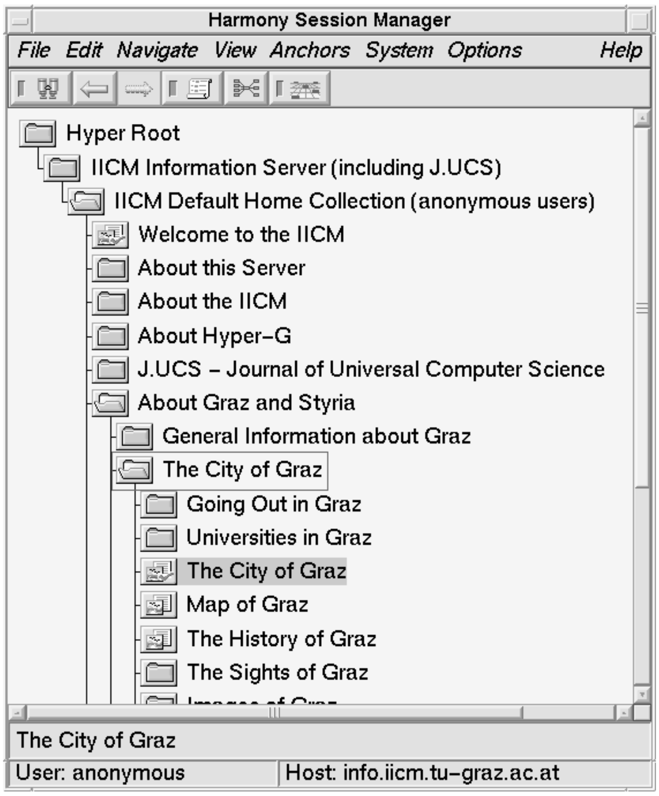
Figure 5.2 The Harmony client and authoring tool for Hyper-G.

5.3 Browsing Collections, Clusters and Documents with Harmony

Figure 5.2 shows Harmony as it is typically configured. On the left-hand side is the Harmony Session Manager, the main Harmony window, which is chiefly used for navigation. On the right-hand side is the Harmony Text Viewer, the window where text documents are displayed. Here we see an anonymous user, without any special access privileges, who has entered the main collection of the IICM Information Server in Graz.

Two broad principles govern the use of the mouse in Harmony. The left mouse button is the one most generally used for navigation: single-left-clicks *select* things, double-left-clicks actually *do* things. In the document viewers, the right mouse button is used to mark regions of interest, for example to define link anchors; in the Session Manager right-clicking diplays an object's attributes.

The Harmony Session Manager embodies most of Harmony's navigational facilities. From top to bottom, the Session Manager is composed of four distinct regions, as is illustrated in Figure 5.3: menu bar, tool bar, collection browser and a two-line status area. The collection browser is the main window of the Harmony Session Manager: it provides intuitive hierarchical navigation through Hyper-G collection structures. Icons are used to indicate collections, clusters, documents, and anchors. Table 5.1 shows icons typically used to represent Hyper-G objects in the Session Manager. A red tick overlaid in the bottom right corner of a document or cluster icon indicates that this object has been visited already during the current session. Ticks may appear in conjunction with any cluster or



Hyper Root
IICM Information Server (including J.UCS)
IICM Default Home Collection (anonymous users)
Welcome to the IICM
About this Server
About the IICM
About Hyper-G
J.UCS – Journal of Universal Computer Science
About Graz and Styria
General Information about Graz
The City of Graz
Going Out in Graz
Universities in Graz
The City of Graz
Map of Graz
The History of Graz
The Sights of Graz

The City of Graz

User: anonymous
Host: info.iicm.tu-graz.ac.at

Figure 5.3 The Harmony Session Manager.

document icon, not only text documents. A green arrow overlaid in the bottom left of a document icon indicates an anchor object (either source or destination) in a document of the specified type. Anchor objects are not currently represented in the collection browser; they can, however, be assigned searchable attributes and thus appear in the search result list.

Hierarchical navigation through Hyper-G collection structures is extremely intuitive, since users are presented with a limited set of visible choices at each level. One popular way of finding information on a Hyper-G server is to start at the top collection and open up sub-collections until the desired information is found.

Collections may be opened and closed and documents retrieved in a number of alternative ways: by double-left-clicking, by selecting and using menu commands, and via extensive keyboard shortcuts. The *currently selected object* (collection, cluster, or document) is framed in a thin box. Objects may be selected by single-left-clicking with the mouse, or by using the arrow keys to move the current selection focus. The currently selected object is the target of a number of commands and operations. To provide visual feedback, the most recently displayed document or cluster is underlaid with a solid bar (it is not necessarily the same object as the current selection).

Navigational Functions

The Harmony Session Manager's navigational functions include:

- **Open**
  The Open operation lists a collection's members in the collection browser, displays the collection head (if present) and moves the selection focus to the first object in the collection. Opening a cluster displays the cluster (in the appropriate document viewers), but does not list its members

Icon	Meaning	Icon	Meaning
📁	Collection (closed)	🖼	Image document
📂	Collection (open)	🎞	Film document
🗂	Cluster	🔊	Audio document
🗂✓	Cluster (visited)	📄	PostScript document
📄	Text document	⬡	Scene document
📄✓	Text document (visited)	□	Generic document
📄➔	Anchor in text document	🖥	Telnet session
📄➔✓	Anchor in text (visited)		

Table 5.1 Typical icons used for Hyper-G objects in Harmony.

in the collection browser. Cluster members whose PresentationHints attribute contains the word Hidden are not displayed (this is useful for keeping inline images and textures in the same cluster as the documents in which they are used, without have them also displayed in separate viewers). Clusters may contain a mix of language-dependent and language-independent documents. In this case the rules for presentation are:

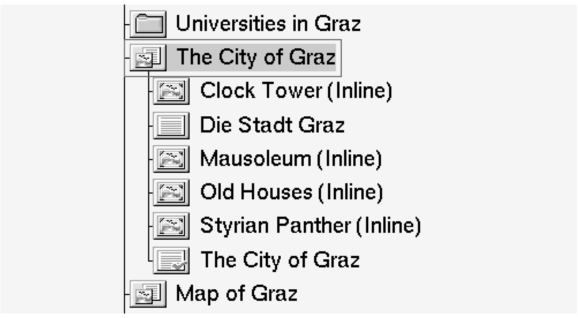
(1) All language-independent members of the cluster are visualised. Language-independent objects are those having multiple Title attributes in different languages.

(2) From the remaining language-dependent members of the cluster, one member of each document type is chosen (based on the user's language preferences) and visualised.

Opening a document simply displays the document.

- **Close**
  When a collection or cluster is closed, its members are no longer listed in the collection browser (they are not removed from the document viewers, however). The Session Manager's Close operation has no effect on document objects.

- **Children**
  The children (members) of the currently selected collection or cluster can be explicitly listed (or hidden) in the collection browser with the Children command. No documents (not even a collection head) are displayed by the Children command.

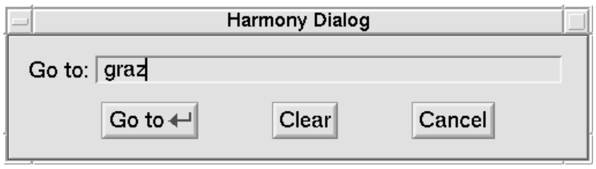


Universities in Graz
The City of Graz
Clock Tower (Inline)
Die Stadt Graz
Mausoleum (Inline)
Old Houses (Inline)
Styrian Panther (Inline)
The City of Graz
Map of Graz

This command is often used to look inside clusters, which are normally considered atomic units as far as display is concerned, to see which documents they really contain. In the example above, looking inside the “City of Graz” cluster reveals an English text, a German text and four images used as inlines. Assuming that the user’s first choice of language is English and that the inline images have PresentationHint Hidden, then activating this cluster would actually only display the English text “The City of Graz” in the Text Viewer.

- **Home**
  With the Home command it is possible to jump to (and open if necessary) your home collection. Anonymous users share a single default home collection, which is the standard point of access to the particular Hyper-G server.

- **Go To**
  The Go To operation allows a specific Hyper-G object or any arbitrary URL to be accessed explicitly. A Hyper-G object can be specified through its object ID, name or Hyper-G URL. In the example below, the user wants to jump directly to the Hyper-G collection named `graz`.

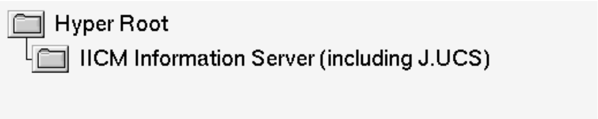


Harmony Dialog

Go to: graz

  Other supported URLs include HTTP and Gopher. For example, to access Addison-Wesley’s WWW welcome page, `http://aw.com/` could be entered into the Go To field. Harmony will then retrieve this WWW page through the Hyper-G server it is connected to.

- **Close All**
  After extensive navigation, the collection browser can become cluttered. The Close All operation tidies up by closing all open collections in the collection browser. This function is often followed by a Home command.



Hyper Root
IICM Information Server (including J.UCS)

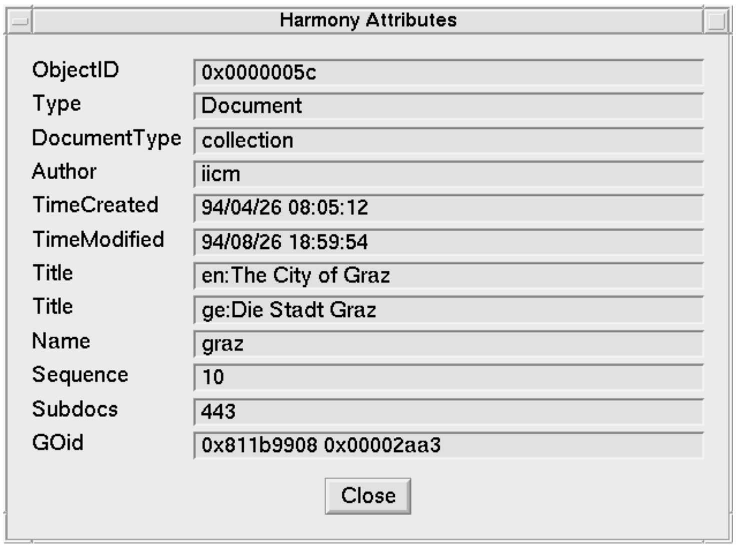
  The example shows just the root collection of the IICM Information Server after issuing a Close All operation.

- **Hold**
  The most recently displayed document or cluster (the one underlaid with a solid bar) can be held on screen with the Hold function. The corresponding document viewer or viewers are frozen and fresh viewers are started for subsequent documents of the same type.

Informational Functions

Informational functions provide additional information about Hyper-G objects. These include:

- **Attributes**
  The Attributes operation displays the Hyper-G attributes (meta-data) of an object in Harmony’s Attributes window. The function is also available by right-clicking on any object in the collection browser, local map, search result list, or information landscape. Figure 5.4 shows the attributes for the collection “The City of Graz.”



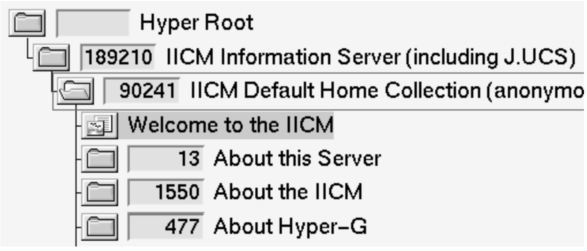
Harmony Attributes

ObjectID	0x0000005c
Type	Document
DocumentType	collection
Author	iicm
TimeCreated	94/04/26 08:05:12
TimeModified	94/08/26 18:59:54
Title	en:The City of Graz
Title	ge:Die Stadt Graz
Name	graz
Sequence	10
Subdocs	443
GOid	0x811b9908 0x00002aa3

Figure 5.4 The Harmony Attributes window.

- **Show Host**
  Show Host displays the host on which the currently selected Hyper-G object resides. This is useful for determining whether an object is located on the local Hyper-G server (the one to which Harmony is connected), or on a remote Hyper-G server. Note that a host's name corresponds to its Internet address, which is incorporated as the first four bytes of the GOid attribute.

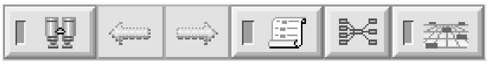
- **Show Count**
  Show Count turns on (or off) display of the document count for collections in the collection browser.



Hyper Root
189210 IICM Information Server (including J.UCS)
90241 IICM Default Home Collection (anonymo
Welcome to the IICM
13 About this Server
1550 About the IICM
477 About Hyper–G

The document count corresponds to the Subdocs attribute of a collection and gives the number of documents contained in the collection and all of its sub-collections (recursively), which are located on the local Hyper-G server. Member documents residing on remote servers are *not* included in the count.

The Toolbar



The Harmony Session Manager toolbar provides direct access to the most frequently used Harmony functions. The Back and Forward buttons are disabled (grayed out) if the end of the history list in the corresponding direction has been reached. Table 5.2 explains the icons typically visible in the Session Manager's toolbar.

Icon	Name	Meaning
	Search	Map/unmap Search window
⇦	Back	Move back in history list
⇨	Forward	Move forward in history list
	History	Map/unmap History window
	Local Map	Map/unmap Local Map window
	Landscape	Map/unmap Landscape window

Table 5.2 Typical icons used in the Harmony Session Manager's toolbar.

Keyboard Support

The Harmony Session Manager has extensive keyboard support. The up arrow (↑) and down arrow (↓) keys move the current selection in the collection browser up and down as you would expect. The left arrow (←) key jumps the current selection back one level in the collection hierarchy to the current selection's parent. The `Return` key opens the current selection if it is a collection, or retrieves and displays the current selection if it is a cluster or document. The right arrow (→) and the plus (+) keys do the same as the `Return` key, except that they also advance the selection into any collection which is opened. The minus (−) key closes the current selection if it is a collection or cluster. The `Page Up` and `Page Down` keys scroll the collection browser display by one page, `Ctrl-l` centers the display around the current selection, `Ctrl-`↑ and `Ctrl-`↓ scroll the display by one line up or down. Table 5.3 gives a list of typical keyboard shortcuts in the Harmony Session Manager.

5.4 Harmony's Document Viewers and Hyperlink Navigation

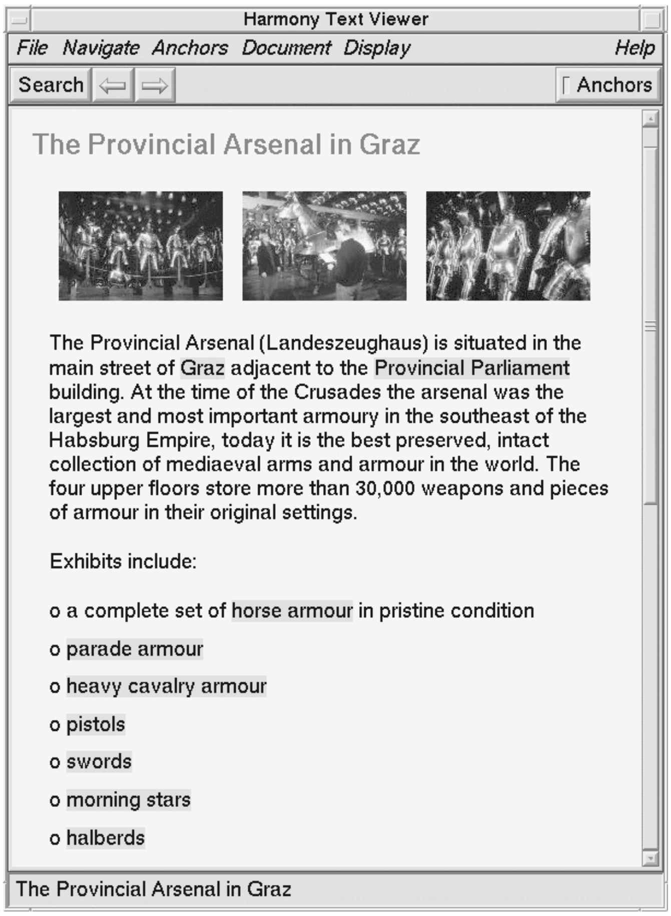
Harmony has native document viewers for six kinds of media: text, image, MPEG films, audio, PostScript, and 3d scenes. Many of the viewers recognise several document encoding formats. Harmony can also be configured to start external viewers for other document types, but external viewers do not generally possess the hyperlink facilities of Harmony's native document viewers.

Associative browsing through hyperlinks is the second main means of navigation in Hyper-G. Users often explore the information space by associative link browsing, either having started out directly from an introductory document with further links, or after having found an interesting document by hierarchical navigation or by searching and beginning from there. Hyperlinks in Hyper-G lead from part of one document to (part of) another associated document or to an associated cluster or collection. The source of a link is typically represented visually as a "clickable" hot region of a document. Since any kind of media may contain hyperlinks in Hyper-G, Harmony provides hyperlink facilities in each of its six document viewers.

In addition to a button to turn link anchor display on and off, all native Harmony document viewers share a number of common navigational facilities such as Hold (to hold the current document on screen), Attributes (to show the attributes of the current document), shortcuts to history functionality such as Back and Forward, and shortcuts to Local Map functionality such as References, Annotations, and Parents.

Key	Function
x	Exit program (confirmation dialogue)
Ctrl-x	Exit program (no confirmation dialogue)
Tab	Next focus (used in dialogues and search)
Return	Open/close collection, display document
+ →	Open collection, goto first child, display document
−	Close collection/cluster
c	Show/hide children of collection/cluster, do not display collection head
Space, n	Next. Display next/current unvisited object
p	Previous. Display current/previous unvisited object
Ctrl-c	Close all collections, open path to root of local server
Ctrl-l	Recenter currently selected object
↑, ↓	Select previous/next object
Ctrl-↑, Ctrl-↓	Scroll one line up/down
Shift-↑, Shift-↓	Select previous/next object, including objects with no title
Page Down, Ctrl-v	Scroll one page down
Page Up, Alt-v	Scroll one page up
←	Select parent collection
a	Show object attributes
e	Edit object attributes
i	Map identify dialogue
s	Map/unmap search dialogue
b	One step back in history
f	One step forward in history
l	Generate local map
L	Unmap local map
R	Show references
A	Show annotations
P	Show parents
Ctrl-a	Annotate object
h	Map/unmap history window
H	Goto home collection
g	Goto dialogue
Ctrl-d, Delete	Remove object from collection
Ctrl-i, Insert	Insert object into collection
Ctrl-s	Display system status
M	Move object
C	Copy object
Alt-s	Display server where the object resides

Table 5.3 Keyboard support in the Harmony Session Manager.



The Provincial Arsenal in Graz

The Provincial Arsenal (Landeszeughaus) is situated in the main street of Graz adjacent to the Provincial Parliament building. At the time of the Crusades the arsenal was the largest and most important armoury in the southeast of the Habsburg Empire, today it is the best preserved, intact collection of mediaeval arms and armour in the world. The four upper floors store more than 30,000 weapons and pieces of armour in their original settings.

Exhibits include:

o a complete set of horse armour in pristine condition

o parade armour

o heavy cavalry armour

o pistols

o swords

o morning stars

o halberds

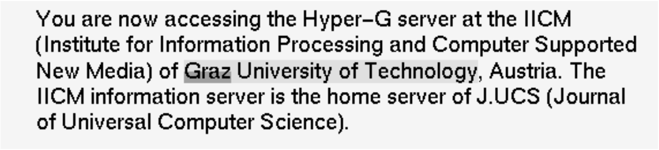
The Provincial Arsenal in Graz

Figure 5.5 The Harmony Text Viewer displaying a HTML document.

The Harmony Text Viewer

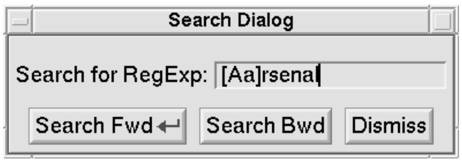
The Harmony Text Viewer displays documents in both HTML [Berners-Lee and Conolly, 1993; Musciano and Kennedy, 1996] and Hyper-G's original, but now obsolete, HTF [Kappe, 1995b] format. Inline images in GIF, JPEG, TIFF, and PNG formats are supported. Figure 5.5 shows the Text Viewer displaying a text concerning the Graz Arsenal. In 1995 and 1996, the Harmony Text Viewer was completely redesigned and re-implemented by a team lead by Mansuet Gaisbauer, to better support HTML 3.x constructs such as tables and free-flowing text around images.

In the Text Viewer, a hyperlink is represented either by underlining, colour coding, or as a solid area underlying the extent of the source anchor, such as "Graz" or "Provincial Parliament" in Figure 5.5. A common technique is to create an itemised list in the text document and link members of the list to associated documents, as has been done for the main exhibits in the Arsenal. Anchors are selected by a single-left-click and activated by double-left-click. The `Tab` key can be used to cycle through the anchors and the `Return` key to activate the currently selected anchor. In HTF texts, up to six levels of overlapping link anchors (overlapping links are possible in HTF, but not currently in HTML) are displayed as separate horizontal layers within the solid anchor area. In the example below, links emanate both from the word "Graz" and from the phrase "Graz University of Technology":



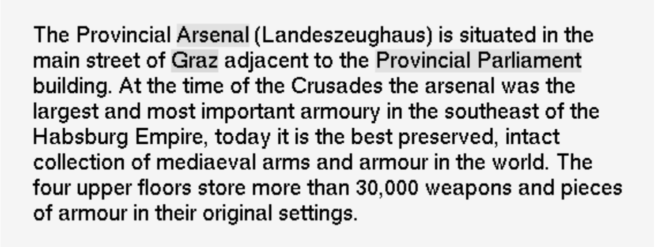
You are now accessing the Hyper–G server at the IICM (Institute for Information Processing and Computer Supported New Media) of Graz University of Technology, Austria. The IICM information server is the home server of J.UCS (Journal of Universal Computer Science).

- **Search in Text**
  It is possible to search for regular expressions within a text document. The regular expression syntax is similar to that of the UNIX commands `awk` and `egrep`. For example, searching for `[Aa]rsenal`:



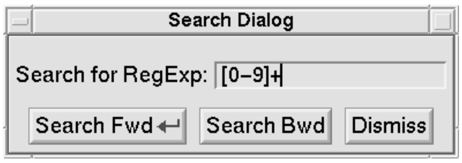
Search for RegExp: [Aa]rsenal

  finds both capitalised and non-capitalised occurrences of the word:



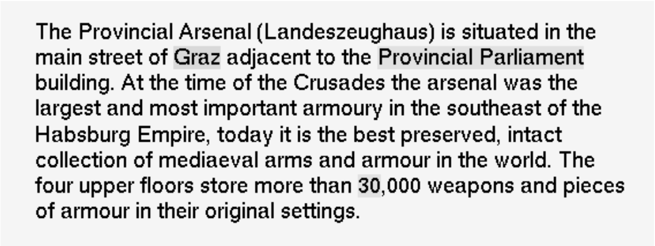
The Provincial Arsenal (Landeszeughaus) is situated in the main street of Graz adjacent to the Provincial Parliament building. At the time of the Crusades the arsenal was the largest and most important armoury in the southeast of the Habsburg Empire, today it is the best preserved, intact collection of mediaeval arms and armour in the world. The four upper floors store more than 30,000 weapons and pieces of armour in their original settings.

  The regular expression `[0-9]+` matches one or more adjacent numeric digits:



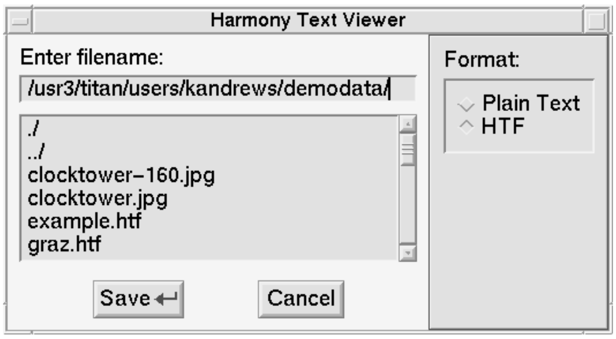
Search for RegExp: [0–9]+

  finding the digits 30 in the example below:



The Provincial Arsenal (Landeszeughaus) is situated in the main street of Graz adjacent to the Provincial Parliament building. At the time of the Crusades the arsenal was the largest and most important armoury in the southeast of the Habsburg Empire, today it is the best preserved, intact collection of mediaeval arms and armour in the world. The four upper floors store more than 30,000 weapons and pieces of armour in their original settings.

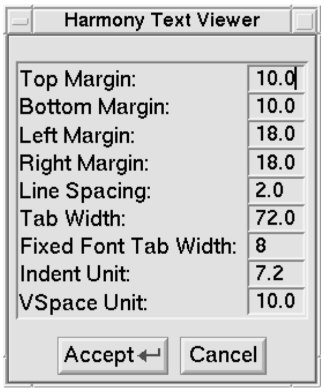
- **Save As**
  The current text document can be saved to your local UNIX file system, either as plain ascii text or in its original marked-up hypertext format (HTML or HTF).



Enter filename:
/usr3/titan/users/kandrews/demodata/

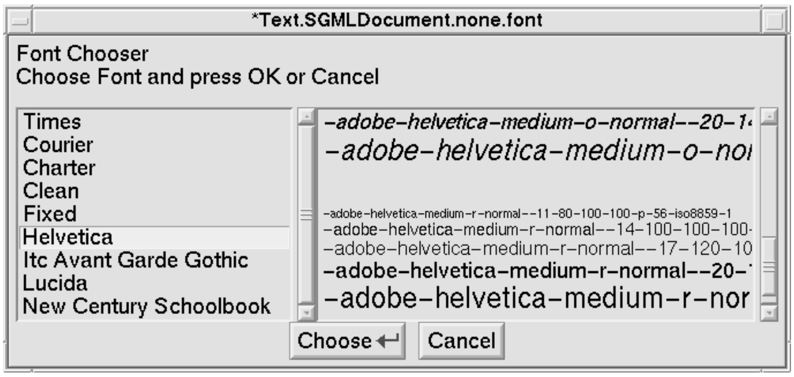
```
./
../
clocktower–160.jpg
clocktower.jpg
example.htf
graz.htf
```

Format:
○ Plain Text
○ HTF



Top Margin: 10.0
Bottom Margin: 10.0
Left Margin: 18.0
Right Margin: 18.0
Line Spacing: 2.0
Tab Width: 72.0
Fixed Font Tab Width: 8
Indent Unit: 7.2
VSpace Unit: 10.0

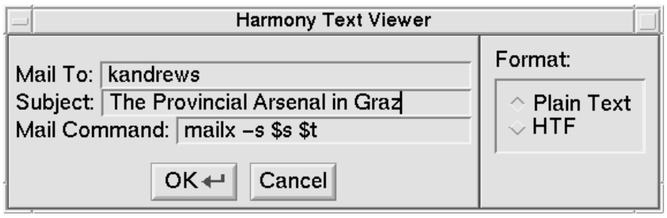
Figure 5.6 The Text Viewer Settings Panel.


*Text.SGMLDocument.none.font
Times
Courier
Charter
Clean
Fixed
Helvetica
Itc Avant Garde Gothic
Lucida
New Century Schoolbook
-adobe-helvetica-medium-o-normal--20-1
-adobe-helvetica-medium-o-no
-adobe-helvetica-medium-r-normal--11-80-100-100-p-56-iso8859-1
-adobe-helvetica-medium-r-normal--14-100-100-100-
-adobe-helvetica-medium-r-normal--17-120-10
-adobe-helvetica-medium-r-normal--20-
-adobe-helvetica-medium-r-nor
Choose
Cancel

Font Chooser
Choose Font and press OK or Cancel

Figure 5.7 The Text Viewer Font Chooser.

- **Mail To**
  The current text document can be sent as email, either as plain ascii or in marked-up hypertext format.

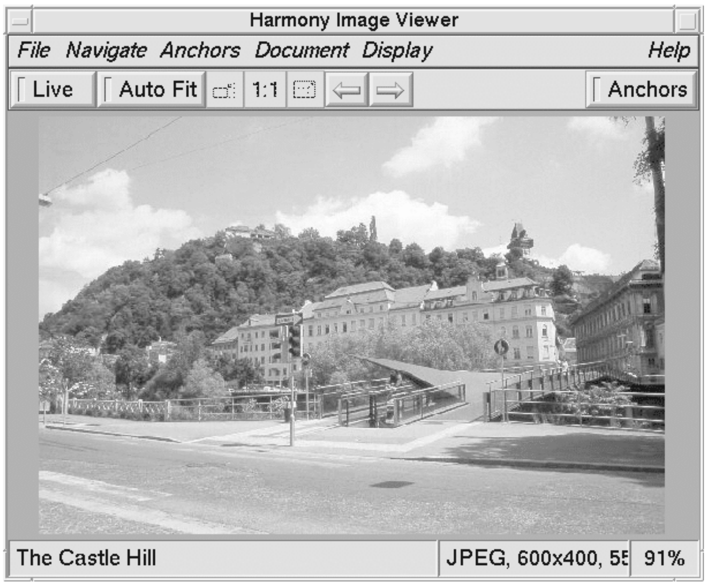


Mail To: kandrews
Subject: The Provincial Arsenal in Graz
Mail Command: mailx -s $s $t

Format:
○ Plain Text
○ HTF

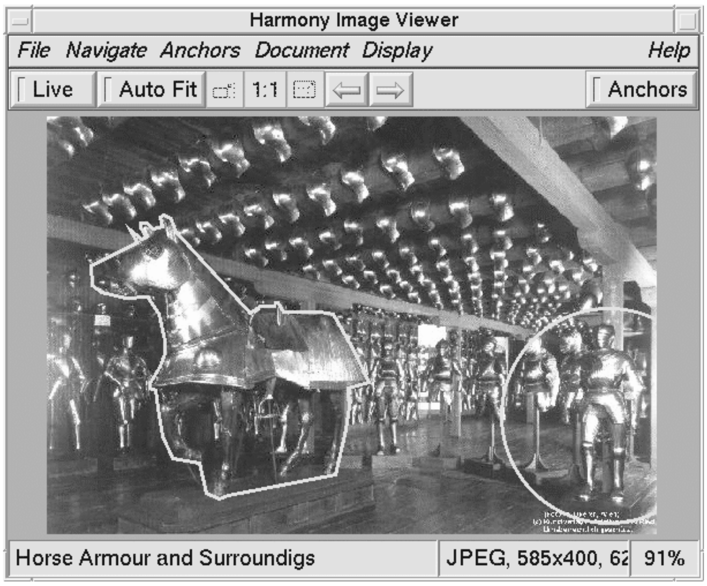
- **View Source**
  The original marked-up source text can be viewed in Harmony's Source Browser.

Various Text Viewer settings such as margin sizes and spacing can be configured interactively in the Settings Panel shown in Figure 5.6. The fonts used to display various kinds of text can be configured in the viewer's font chooser. The example in Figure 5.7 shows a user selecting a Helvetica font for standard text. Similarly, the colours of text, link anchors and the background can be specified using a colour chooser.



The Castle Hill | JPEG, 600x400, 55 | 91%

Figure 5.8 The Harmony Image Viewer.



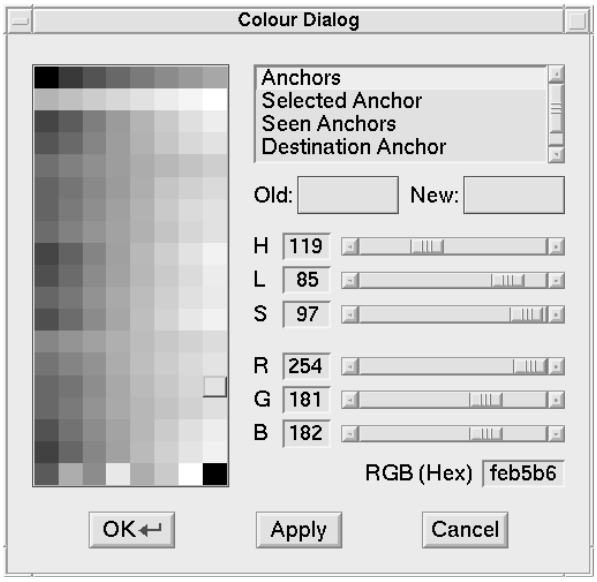
Horse Armour and Surroundigs | JPEG, 585x400, 62 | 91%

Figure 5.9 Links in image documents.

The Harmony Image Viewer

The Harmony Image Viewer displays images in a number of common formats, including GIF, JPEG, TIFF, and PNG. Figure 5.8 shows the Image Viewer displaying a JPEG image of the castle hill in Graz city centre. The status line at the bottom of the viewer indicates the image’s title, format, resolution, size, and current magnification.

Link anchors in image documents may be rectangular, circular, elliptical or polygonal in shape (this applies to both source and destination anchors). The image of Graz Arsenal in Figure 5.9 has two source anchors: a polygonal anchor enclosing the horse armour to the left and a circular anchor enclosing the heavy cavalry armour to the right. Anchors are selected by a single-left-click and activated by double-left-click. The `Tab` key can be used to cycle through the anchors and the Return



Colour Dialog

Anchors
Selected Anchor
Seen Anchors
Destination Anchor

Old: New:

H 119
L 85
S 97

R 254
G 181
B 182

RGB (Hex) feb5b6

Figure 5.10 Image Viewer colour configuration.

key to activate the currently selected anchor (this is useful for overlapping link anchors). The colours used to represent link anchors can be configured using the Colour Dialogue shown in Figure 5.10.

The Live button determines whether or not images are displayed progressively as they are received. AutoFit causes images to be scaled (larger or smaller) so that they fit the current image display area. Turning ShrinkOnly on will only scale images down and not enlarge them if autofitting is enabled. When AutoFit is disabled, the smaller ⧉ (−25%), original size 1:1 (100%) and larger ⧉ (+33%) buttons and menu items for zooming in and zooming out become accessible.

The Harmony Film Player

The Harmony Film Player plays film clips in MPEG-1 format. Figures 5.11 and 5.12 show the Film Player displaying a video clip taken from the Graz Arsenal. Analogous to the Image Viewer, the status line at the bottom of the Film Player indicates the film clip's title, format, resolution, size, and current magnification. VCR style control buttons are provided for playing, pausing, rewinding, and single-stepping through the film. The Live button determines whether or not films are played back as they are received. Loop causes the film to be played continuously. MPEG audio is not currently supported; if an audio stream is present, it is silently ignored.

Link anchors in films may be rectangular, circular, or elliptical in shape (this applies to both source and destination anchors). Their position and size are defined for specific key frames and are interpolated in between. Source anchors may be activated even during playback. The image in Figure 5.11 shows a circular source anchor around the cavalry helmet in the first part of the film. The image in Figure 5.12 illustrates a destination anchor covering the last segment of the clip which deals with horse armour.

The Film Player's Settings Panel, depicted in Figure 5.13, allows various settings to be configured interactively. Particularly useful are the brightness and the frame rate. By default frames are drawn as fast as possible; the frame rate is only synchronised upon specific request. The colours used to represent link anchors in the Film Player are configurable interactively using a Colour Dialogue similar to that of the Image Viewer.

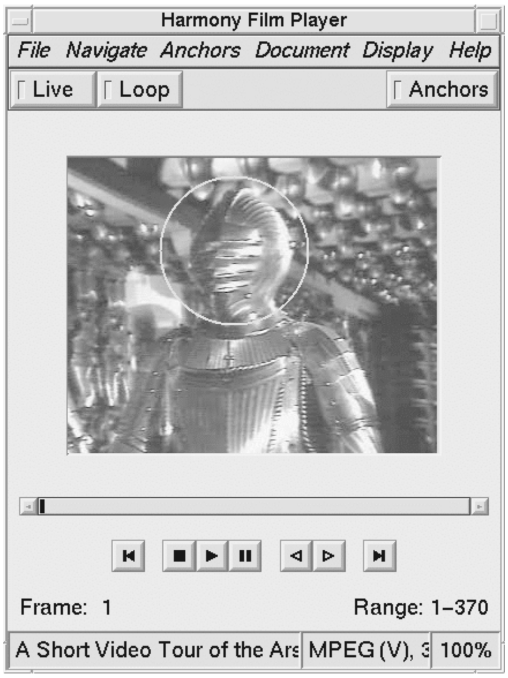

Harmony Film Player
File Navigate Anchors Document Display Help
Live
Loop
Anchors
Frame: 1
Range: 1–370
A Short Video Tour of the Ars
MPEG (V), 3
100%

Figure 5.11 The Harmony Film Player.

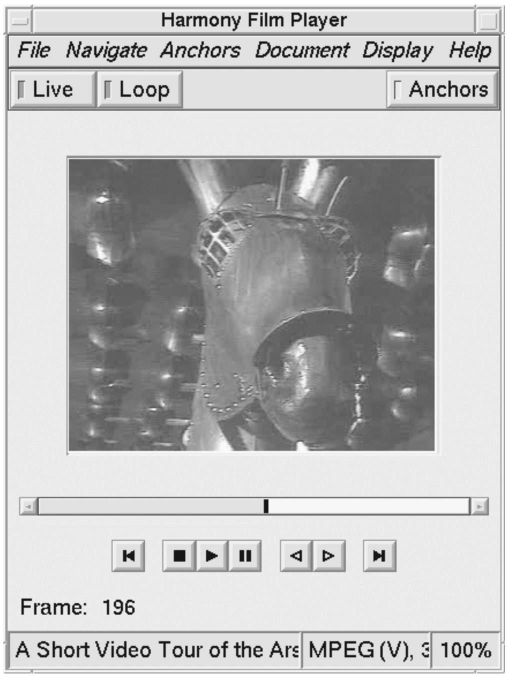
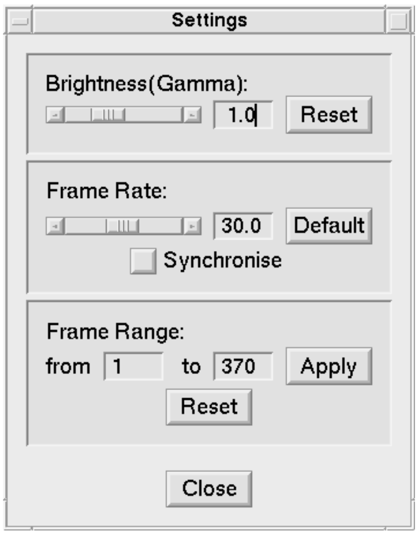

Harmony Film Player
File Navigate Anchors Document Display Help
Live
Loop
Anchors
Frame: 196
A Short Video Tour of the Ars
MPEG (V), 3
100%

Figure 5.12 A destination anchor extending across the second half of a film clip.

The Harmony Audio Player

The Harmony Audio Player plays audio clips. The supported audio formats depend on the underlying platform and on the Audio Player's configuration. By default, the Audio Player does not look into the audio stream itself, but simply passes it on to the client machine's native audio play command. In this

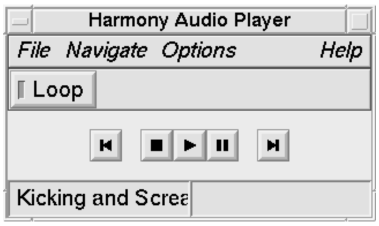


Brightness(Gamma): 1.0

Frame Rate: 30.0

Frame Range: from 1 to 370

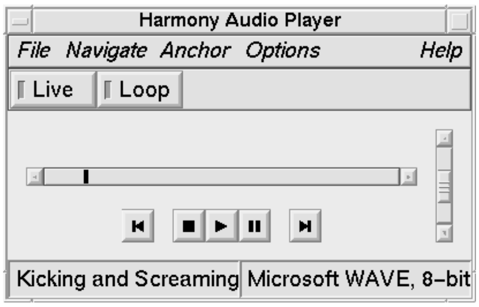
Figure 5.13 The Film Player Settings Panel.



Kicking and Screa

Figure 5.14 The Harmony Audio Player in native audio mode.

mode, illustrated in Figure 5.14, it is not possible for the Audio Player to resolve link anchors and the supported audio formats are exactly those supported by the client platform.

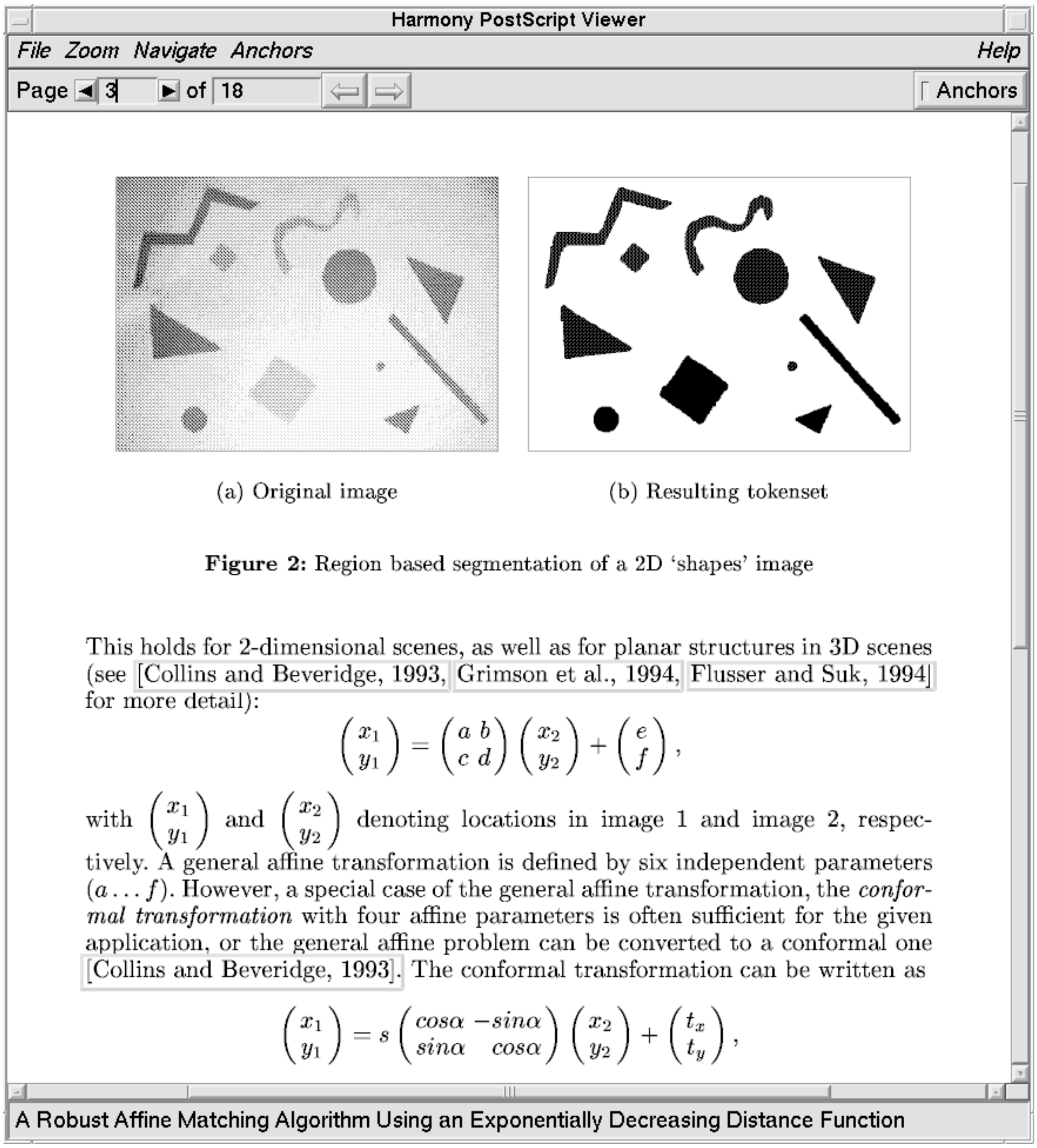


Kicking and Screaming | Microsoft WAVE, 8-bit

Figure 5.15 In NAS mode, the Harmony Audio Player also supports volume control and link anchors.

If the Network Audio System (NAS) [Fulton and Renda, 1994] is installed on your system or on another local machine, the Audio Player can be configured to use NAS. In this case, illustrated in Figure 5.15, the Audio Player examines the audio stream and performs audio output itself; it also supports volume control and link anchors. The status line at the bottom of the player indicates the audio clip's title and format, resolution, sampling rate, and size.

The Harmony PostScript Viewer



(a) Original image

(b) Resulting tokenset

Figure 2: Region based segmentation of a 2D 'shapes' image

This holds for 2-dimensional scenes, as well as for planar structures in 3D scenes (see [Collins and Beveridge, 1993, Grimson et al., 1994, Flusser and Suk, 1994] for more detail):

$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix},$$

with $\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}$ and $\begin{pmatrix} x_2 \\ y_2 \end{pmatrix}$ denoting locations in image 1 and image 2, respectively. A general affine transformation is defined by six independent parameters $(a \ldots f)$. However, a special case of the general affine transformation, the *conformal transformation* with four affine parameters is often sufficient for the given application, or the general affine problem can be converted to a conformal one [Collins and Beveridge, 1993]. The conformal transformation can be written as

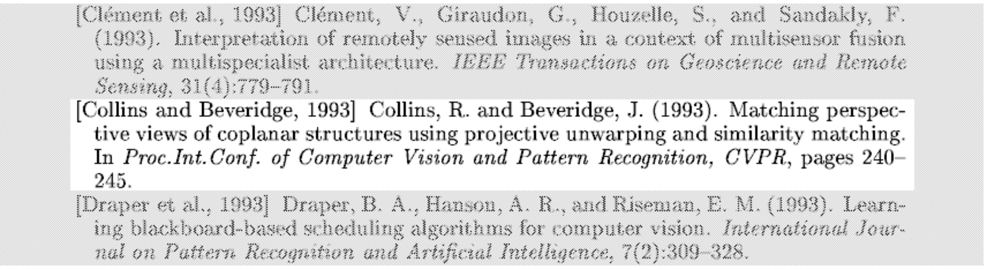
$$\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = s \begin{pmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix},$$

A Robust Affine Matching Algorithm Using an Exponentially Decreasing Distance Function

Figure 5.16 The Harmony PostScript Viewer.

The Harmony PostScript Viewer displays documents described in the PostScript page description language. PostScript documents compressed with GNU `gzip` are automatically decompressed by the viewer. GhostScript (`gs`) is used to generate the individual page images (note that `gzip` and `gs` must both be installed on your system and be in your path for the PostScript viewer to function properly). Figure 5.16 shows the PostScript Viewer displaying an 18-page article which appeared in the *Journal of Universal Computer Science* [Maurer and Schmaranz, 1994].

Link anchors in PostScript documents are rectangular areas overlaid on the page, such as the four links to references later in the article in Figure 5.16. A destination anchor is highlighted, or rather the rest of the page is “low”lighted, as shown in Figure 5.17. Links can, of course, point to arbitrary objects like any other Hyper-G links; they are not restricted to being within a single document.



[Clément et al., 1993] Clément, V., Giraudon, G., Houzelle, S., and Sandakly, F. (1993). Interpretation of remotely sensed images in a context of multisensor fusion using a multispecialist architecture. *IEEE Transactions on Geoscience and Remote Sensing*, 31(4):779–791.

[Collins and Beveridge, 1993] Collins, R. and Beveridge, J. (1993). Matching perspective views of coplanar structures using projective unwarping and similarity matching. In *Proc.Int.Conf. of Computer Vision and Pattern Recognition, CVPR*, pages 240–245.

[Draper et al., 1993] Draper, B. A., Hanson, A. R., and Riseman, E. M. (1993). Learning blackboard-based scheduling algorithms for computer vision. *International Journal on Pattern Recognition and Artificial Intelligence*, 7(2):309–328.

Figure 5.17 A destination anchor in the Harmony PostScript Viewer. The anchor region is highlighted by dimming the rest of the page.

The page images are calculated once at 300 dpi (corresponding to 300% in the Zoom menu) and are scaled down appropriately to the current zooming factor, which is typically set at 100% or 75%. The document can be saved and printed directly from the viewer. The middle mouse button of the PostScript Viewer acts as a magnifying glass, magnifying the area immediately beneath the cursor.

The Harmony VRweb 3D Viewer

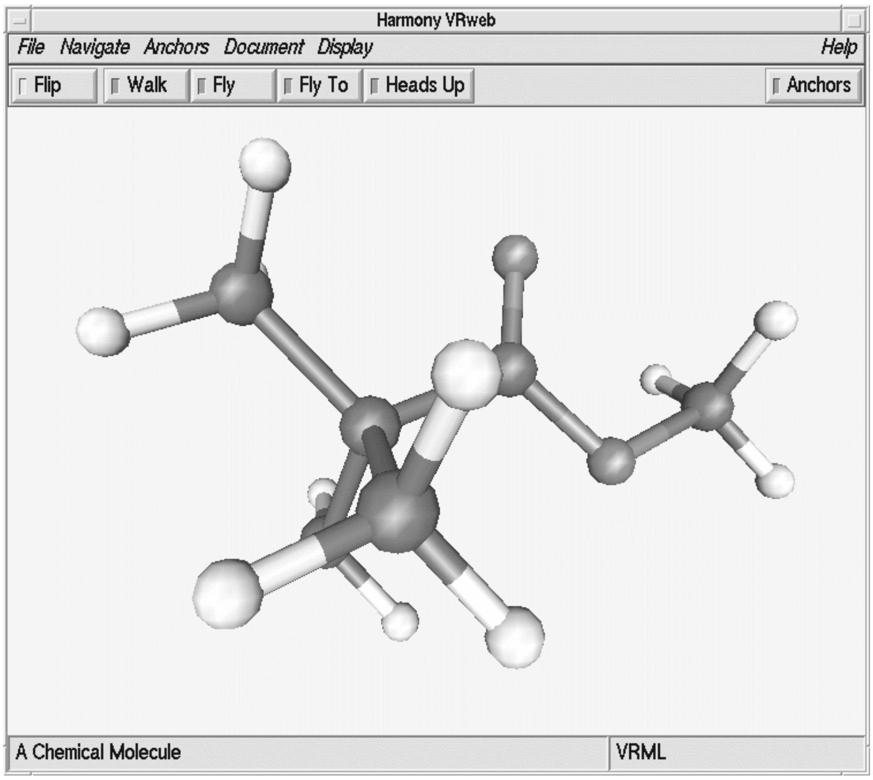
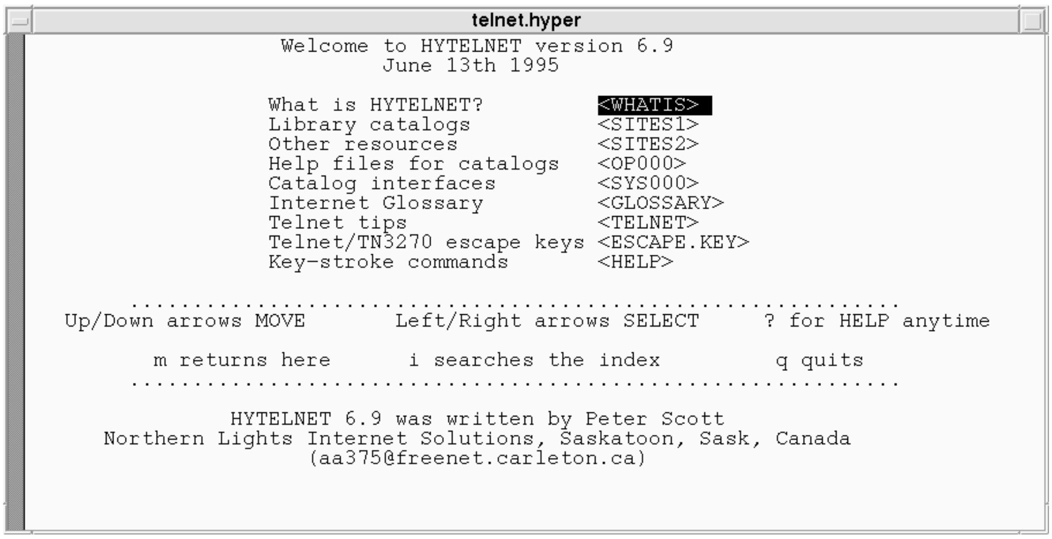


Figure 5.18 The Harmony VRweb 3D viewer.

VRweb is Harmony’s 3d viewer, which can display 3d models modeled in the Virtual Reality Modeling Language (VRML) [Pesce, 1995]. VRML is the 3d model description format standardised



```
                  Welcome to HYTELNET version 6.9
                          June 13th 1995

                 What is HYTELNET?          <WHATIS>
                 Library catalogs           <SITES1>
                 Other resources            <SITES2>
                 Help files for catalogs    <OP000>
                 Catalog interfaces         <SYS000>
                 Internet Glossary          <GLOSSARY>
                 Telnet tips                <TELNET>
                 Telnet/TN3270 escape keys  <ESCAPE.KEY>
                 Key-stroke commands        <HELP>

      ............................................................
 Up/Down arrows MOVE       Left/Right arrows SELECT     ? for HELP anytime

        m returns here      i searches the index         q quits
      ............................................................

              HYTELNET 6.9 was written by Peter Scott
    Northern Lights Internet Solutions, Saskatoon, Sask, Canada
                      (aa375@freenet.carleton.ca)
```

Figure 5.19 A Harmony Telnet session to HYTELNET.

by the WWW community [Bell *et al.*, 1996a,b]. Figure 5.18 shows Harmony VRweb displaying a VRML model of a chemical molecule. Flip navigation mode, active here, enables the model itself to be moved and rotated, without changing the user's position. VRweb's other four navigation modes (Walk, Fly, Fly To, and Heads Up) all involve the user moving through the model.

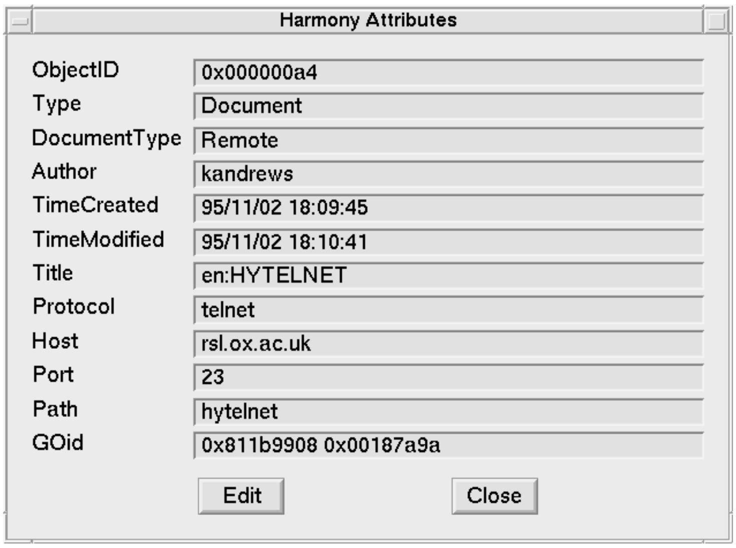
VRweb is described in detail in Chapter 7, so here I will concentrate on the additional functionality provided in the Harmony version of VRweb for Hyper-G. One of the main benefits of using a Hyper-G server is its link management. When a 3d model in VRML is uploaded onto Hyper-G server, its hyperlinks, inline links, and texture links are extracted and are entered into Hyper-G's link database, in the same way as links and inline images in HTML texts. When the model is accessed, the appropriate nodes are merged back in to the VRML file. Using Harmony VRweb, links can be edited interactively. It will also soon be possible to edit models within VRweb: not at the polygon level, but by importing and including models and by moving, rotating, and scaling existing models. Once this functionality is stable and released, a Hyper-G server will become a powerful system for keeping track of models, their components, and textures. In other systems, renaming a model file or moving a texture file to a different directory can have dire consequences, since each and every embedded reference has to be changed to match.

Harmony Telnet Sessions

Telnet sessions can be started from within Harmony by activating a Telnet object. Figure 5.19 shows a telnet session to `HYTELNET`. Upon activation, the Harmony Session Manager opens a new X terminal (xterm) and starts the program `telnet.hyper`, a modified telnet program which accepts login names and passwords as command line parameters. The Hyper-G object representing this particular telnet service is a document of DocumentType Remote and Protocol telnet, as can be seen in Figure 5.20.

5.5 The Harmony Local Map

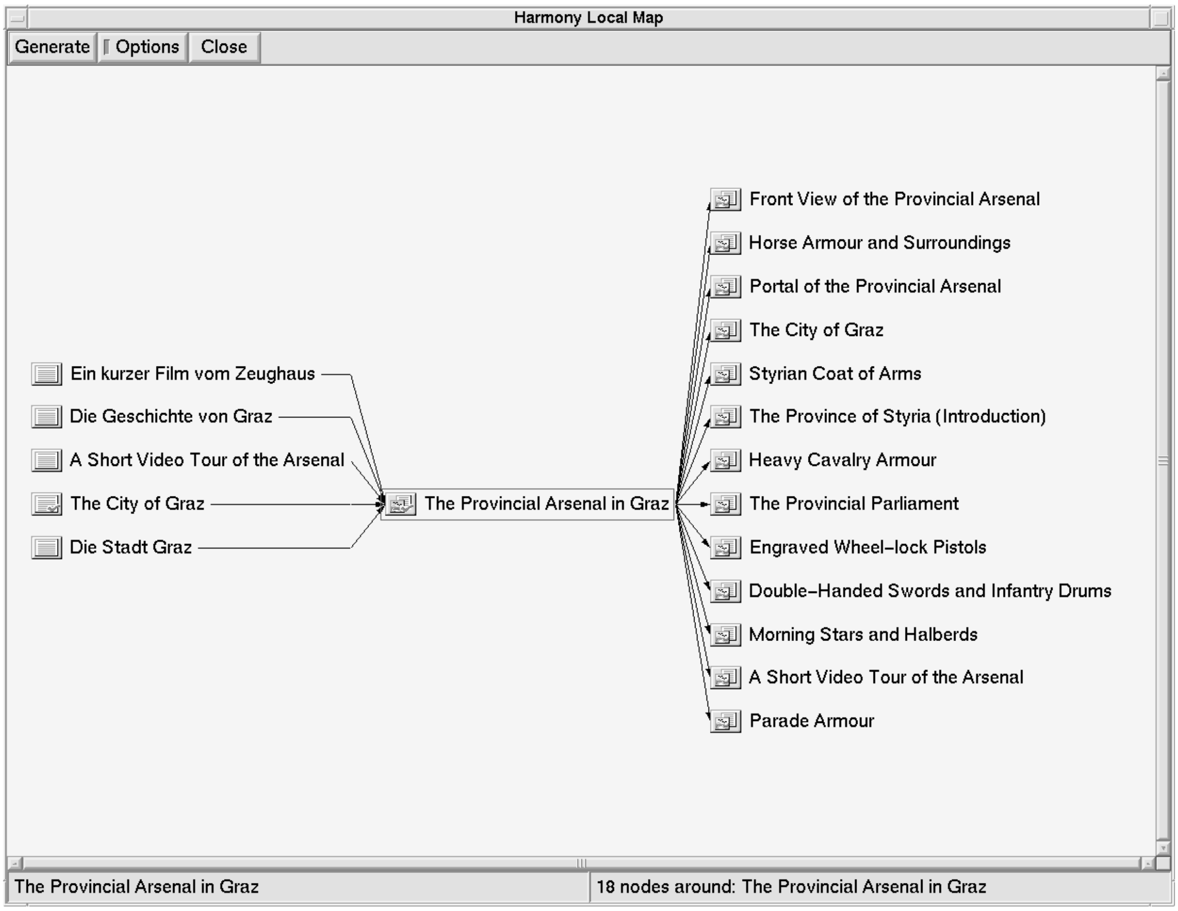
The Harmony Local Map is a dynamic 2d structure map which provides a kind of short-range radar, generating on request (dynamically) a map of the hyperlink neighbourhood of a chosen document.



Harmony Attributes

ObjectID	0x000000a4
Type	Document
DocumentType	Remote
Author	kandrews
TimeCreated	95/11/02 18:09:45
TimeModified	95/11/02 18:10:41
Title	en:HYTELNET
Protocol	telnet
Host	rsl.ox.ac.uk
Port	23
Path	hytelnet
GOid	0x811b9908 0x00187a9a

Figure 5.20 A Hyper-G Telnet object.



Harmony Local Map
Generate
Options
Close
Ein kurzer Film vom Zeughaus
Die Geschichte von Graz
A Short Video Tour of the Arsenal
The City of Graz
Die Stadt Graz
The Provincial Arsenal in Graz
Front View of the Provincial Arsenal
Horse Armour and Surroundings
Portal of the Provincial Arsenal
The City of Graz
Styrian Coat of Arms
The Province of Styria (Introduction)
Heavy Cavalry Armour
The Provincial Parliament
Engraved Wheel–lock Pistols
Double–Handed Swords and Infantry Drums
Morning Stars and Halberds
A Short Video Tour of the Arsenal
Parade Armour

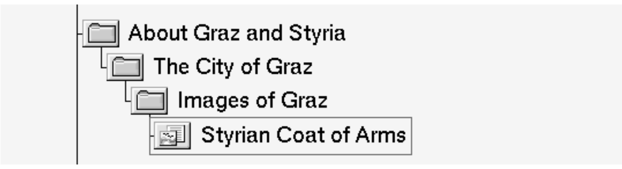
The Provincial Arsenal in Graz

18 nodes around: The Provincial Arsenal in Graz

Figure 5.21 The Harmony Local Map.

Figure 5.21 shows a local map of the hyperlink vicinity of the cluster “The Provincial Arsenal in Graz.” The Harmony Local Map is covered in detail in Chapter 8. As described more fully in Section 8.7, Harmony’s local map can also show other relationships, such as collection membership, annotations, inline images, and the textures applied to a 3d model. The number and type of incoming and outgoing relationships can be interactively configured in the Options panel.

The Local Map is closely coupled with the Harmony Session Manager’s collection browser (see Section 5.3). When an object is selected in the Local Map and is not already visible in the collection browser, a path to that object is opened up automatically. Thus, users can gain an understanding of the context in which the object is embedded through its location in Hyper-G’s collection structure, *prior* to any decision to view it. I coined the term *location feedback* to describe this feature of Harmony. For example, selecting “Styrian Coat of Arms” in Figure 5.21 opens up the following path in the collection browser:



About Graz and Styria
The City of Graz
Images of Graz
Styrian Coat of Arms

Users can navigate within the local map by generating a new map around any selected object; objects are accessed by double-clicking. One practical use of the Local Map, for example, is to check whether a particular inline image is used in multiple text documents before changing or deleting the image.

5.6 Searching in Harmony

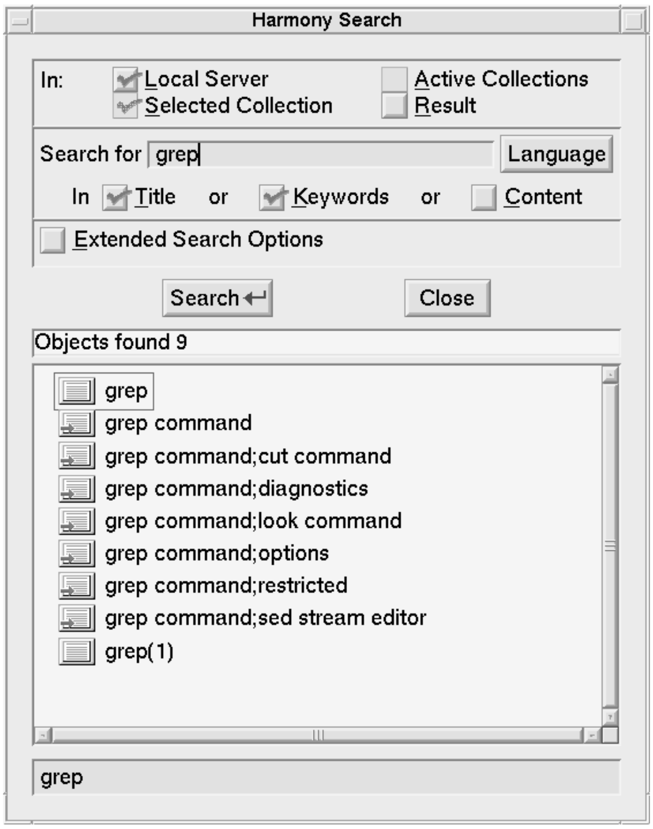
The third main means of navigation in Hyper-G, in addition to hierarchical browsing of the collection structure and following hyperlinks, are Hyper-G’s sophisticated search facilities. In contrast to WWW or Gopher, Hyper-G has *fully integrated* attribute and content search mechanisms.

Attribute search provides for searching in those attribute fields of Hyper-G which are indexed by the Hyper-G server. These typically include Title, Keywords, Author, and TimeModified. Content search utilises the Hyper-G server’s full text inverted index to search for words or phrases contained in the content of a text document. Every text document’s content is automatically indexed upon insertion into the server and the index is updated when a document is edited or deleted.

Harmony’s interface to Hyper-G search facilities is the Harmony Search Dialogue, shown in Figure 5.22. In this example, the user has searched for the word “grep” in the Title and Keywords attributes of all objects (collections, clusters, documents and anchors) on the local Hyper-G server. The languages in which to search can be specified in a separate dialogue. The result here is a list of nine matching objects; the first and last in the list are text documents, the remaining seven objects are link anchors.

Single-left-clicking an object in the search result list selects the object and activates location feedback (as it does in the Local Map). The value of location feedback is illustrated in Figure 5.23. Here we see from the collection context that the first grep text document in the result list belongs to the collection “G” in the “Hacker’s Jargon” and the second belongs to “(1) User Commands” in the “ULTRIX 4.2” manual pages. Assuming that the user was looking for a description of the UNIX `grep` command, it is now possible to make an educated choice between the two documents, before having retrieved either of them. Right-clicking an object in the search result list brings up the attributes of that object (as it does in the Collection Browser and Local Map) and double-left-clicking retrieves or activates an object.

Harmony’s Extended Search Options allow a search to be further constrained to a particular author (owner) and to particular (ranges of) modification times. In the example in Figure 5.24, a



In: ☑ Local Server ☑ Selected Collection □ Active Collections □ Result

Search for grep

In ☑ Title or ☑ Keywords or □ Content

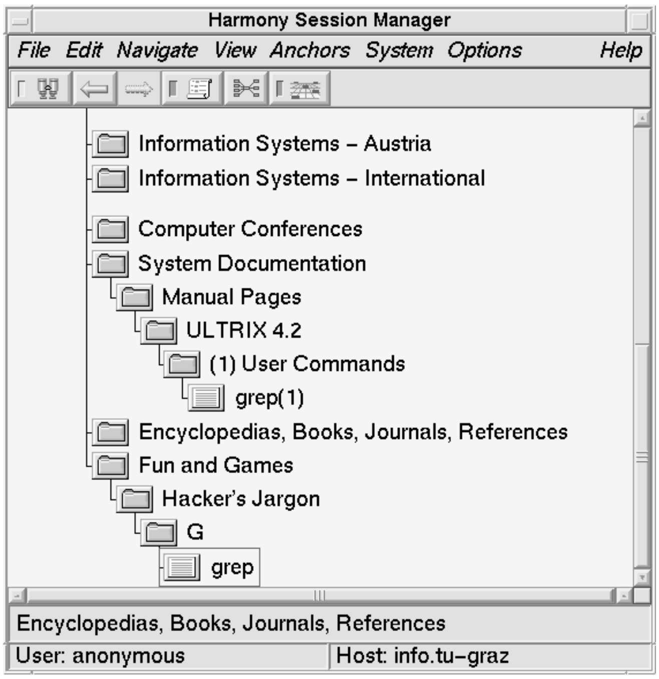
□ Extended Search Options

Objects found 9

- grep
- grep command
- grep command;cut command
- grep command;diagnostics
- grep command;look command
- grep command;options
- grep command;restricted
- grep command;sed stream editor
- grep(1)

grep

Figure 5.22 The Harmony Search Dialogue.



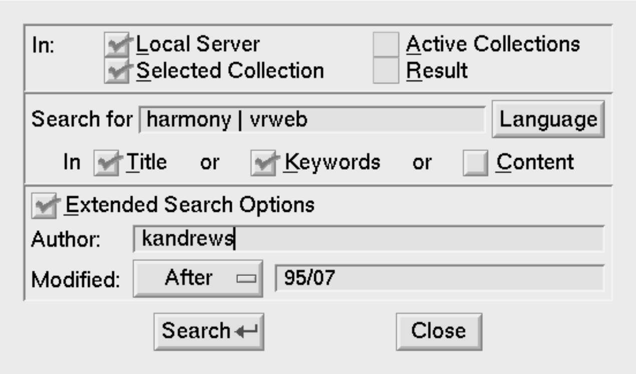
- Information Systems – Austria
- Information Systems – International
- Computer Conferences
- System Documentation
  - Manual Pages
    - ULTRIX 4.2
      - (1) User Commands
        - grep(1)
- Encyclopedias, Books, Journals, References
- Fun and Games
  - Hacker's Jargon
    - G
      - grep

Encyclopedias, Books, Journals, References

User: anonymous

Host: info.tu–graz

Figure 5.23 Location feedback from search results.


In: ☑ Local Server ☑ Selected Collection □ Active Collections □ Result

Search for harmony | vrweb

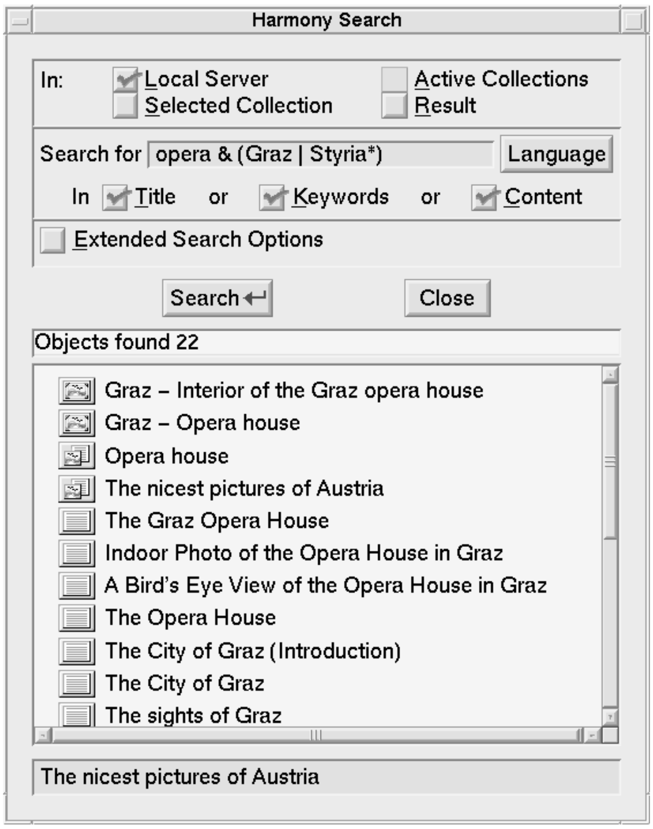
In ☑ Title or ☑ Keywords or □ Content

☑ Extended Search Options

Author: kandrews

Modified: After 95/07

Figure 5.24 Harmony's extended search options.



In: ☑ Local Server □ Selected Collection □ Active Collections □ Result

Search for opera & (Graz | Styria*)

In ☑ Title or ☑ Keywords or ☑ Content

□ Extended Search Options

Objects found 22

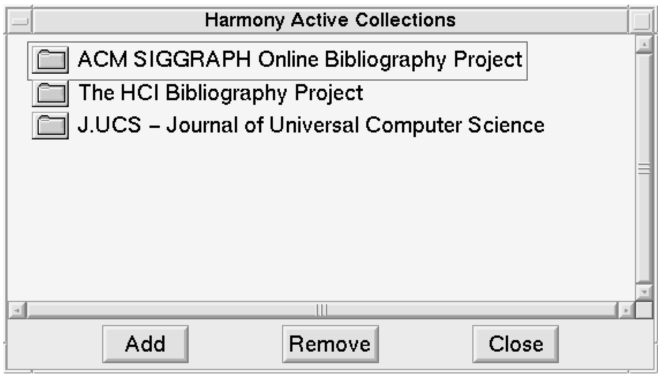
- Graz – Interior of the Graz opera house
- Graz – Opera house
- Opera house
- The nicest pictures of Austria
- The Graz Opera House
- Indoor Photo of the Opera House in Graz
- A Bird's Eye View of the Opera House in Graz
- The Opera House
- The City of Graz (Introduction)
- The City of Graz
- The sights of Graz

The nicest pictures of Austria

Figure 5.25 Combined content and attribute search.

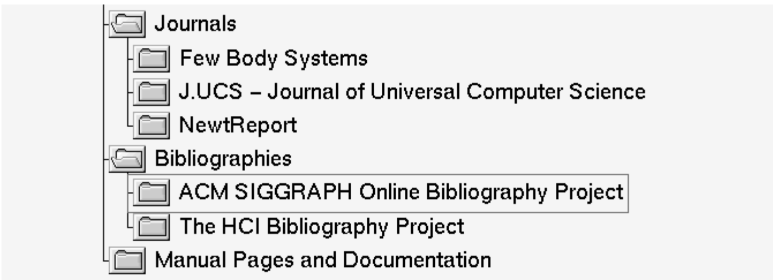
search is being made for documents on the local server about Harmony or VRweb, authored by user `kandrews`, after the 1st July, 1995.

Figure 5.25 illustrates the effect of combining content search with attribute search. An initial search in the Title and Keywords only resulted in four matching objects, whereas turning on content search as well found a further 18 documents. The example in Figure 5.25 also illustrates the syntax of search expressions with the example `opera & (Graz | Styria*)`. Multiple search terms are implicitly `AND`ed together; an ampersand (&) can be used to explicitly express an `AND` operator. A vertical bar (|) represents the `OR` operator, which has lower precedence than `AND`. Terms may be grouped using parentheses. There is no `NOT` operator. Prefix search is supported using a trailing



- ACM SIGGRAPH Online Bibliography Project
- The HCI Bibliography Project
- J.UCS – Journal of Universal Computer Science

Figure 5.26 Set of collections activated for search.



- Journals
  - Few Body Systems
  - J.UCS – Journal of Universal Computer Science
  - NewtReport
- Bibliographies
  - ACM SIGGRAPH Online Bibliography Project
  - The HCI Bibliography Project
- Manual Pages and Documentation

Figure 5.27 Activating local and remote collections for search.

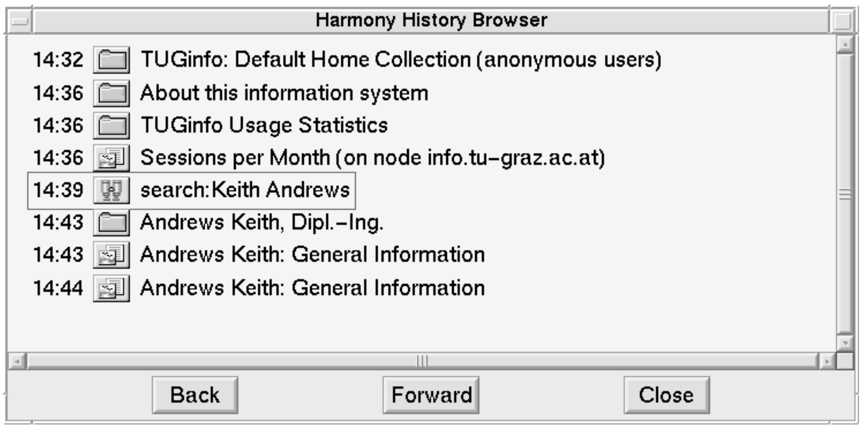
asterisk. Note that search terms in Hyper-G are principally case-insensitive.

A major strength of Hyper-G is the ability to control the scope of searches. As can be seen in Figure 5.25, searches may be restricted to objects residing on the local server, members of the current set of active collections, or only members of the currently selected collection. A powerful additional feature is to search in the current search result list, which allows users to successively refine their searches. The Harmony Session Manager's Active Collections window displays the current set of active collections. The currently selected collection can be added to or removed from the active set in the Session Manager. The example in Figure 5.26 shows three active collections from the part of the IICM Information Server collection structure shown in Figure 5.27.

Searches are not restricted to objects on the local Hyper-G server. Collection members are sometimes located on remote servers; including such collections in the scope of a search initiates simultaneous searching on multiple servers. This is the case in the example, where the two bibliographic collections in fact reside on a remote Hyper-G server.

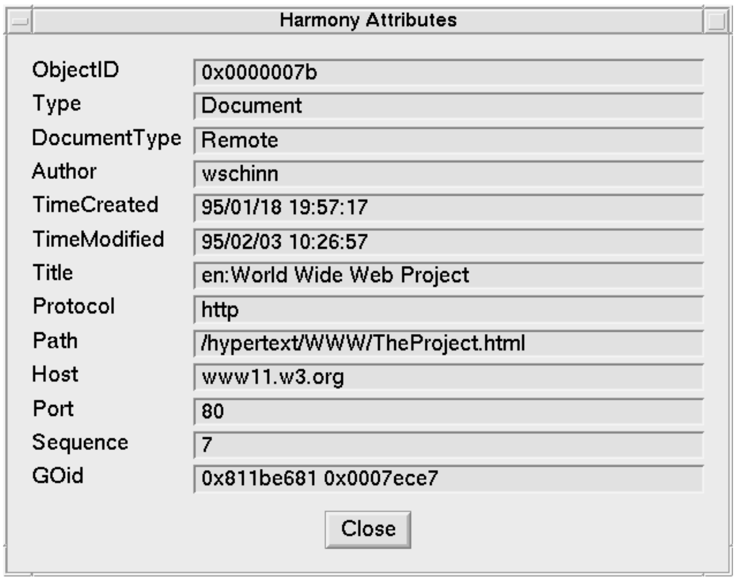
5.7 Harmony's History Browser

The Harmony History Browser shown in Figure 5.28 maintains a time-line of past interactive waypoints during browsing. Collections, clusters, and documents visited are time-stamped and entered into the history list. Search queries are also entered into the history, enabling users to modify and reissue previous searches. Users can step backward and forward through the history list using the Back and Forward arrow buttons available in the Session Manager and in the document viewers. Within the History Browser, single-left-clicking an entry initiates location feedback, double-left-clicking activates an object, and single-right-clicking displays an object's attributes.



14:32 TUGinfo: Default Home Collection (anonymous users)
14:36 About this information system
14:36 TUGinfo Usage Statistics
14:36 Sessions per Month (on node info.tu-graz.ac.at)
14:39 search: Keith Andrews
14:43 Andrews Keith, Dipl.-Ing.
14:43 Andrews Keith: General Information
14:44 Andrews Keith: General Information

Figure 5.28 The Harmony History Browser.



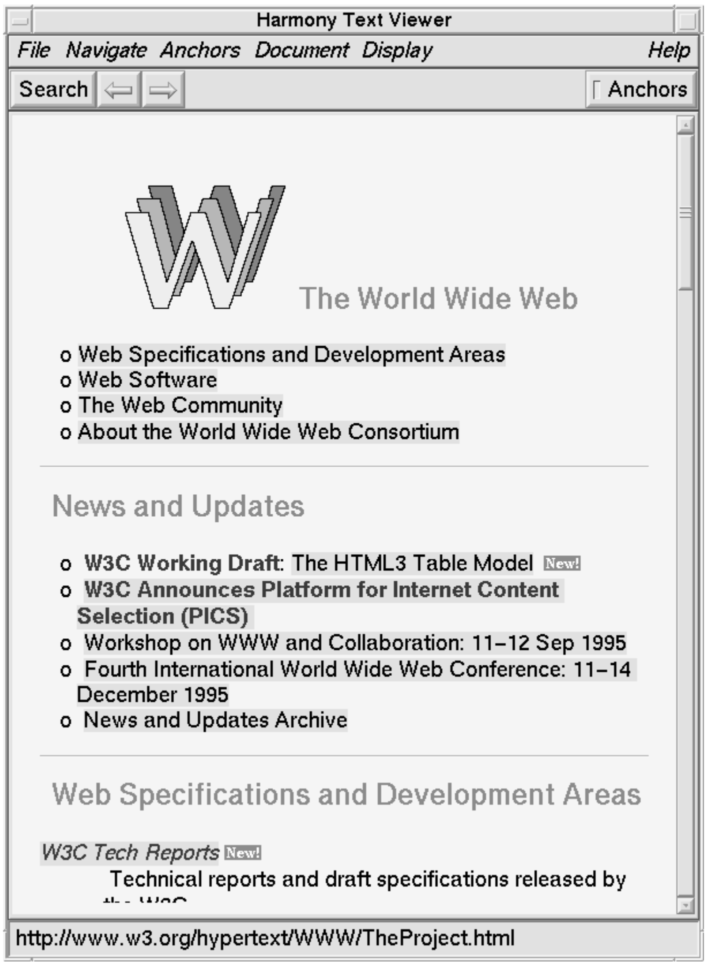
ObjectID	0x0000007b
Type	Document
DocumentType	Remote
Author	wschinn
TimeCreated	95/01/18 19:57:17
TimeModified	95/02/03 10:26:57
Title	en:World Wide Web Project
Protocol	http
Path	/hypertext/WWW/TheProject.html
Host	www11.w3.org
Port	80
Sequence	7
GOid	0x811be681 0x0007ece7

Figure 5.29 A Hyper-G remote object representing the WWW project page at W3C.

5.8 Harmony for WWW and Gopher Sites

Documents residing on remote WWW servers can be referred to using a Hyper-G remote object. They are displayed with an icon appropriate to their type in Harmony and can be accessed transparently as if they were local documents. Figure 5.29 shows a Hyper-G remote object representing the WWW project page at W3C, Figure 5.30 shows Harmony displaying the page.

In a similar fashion, remote Gopher menus and documents can be referred to using a remote Gopher object on a Hyper-G server. Figure 5.31 shows the remote Gopher object corresponding to part of the University of Minnesota’s Gopherspace. Figure 5.32 shows the Minnesota server, as seen in Harmony. Note that Gopher menus are mapped transparently to Hyper-G collections, Gopher documents are mapped to documents of the same type in Hyper-G.



The World Wide Web

- o Web Specifications and Development Areas
- o Web Software
- o The Web Community
- o About the World Wide Web Consortium

News and Updates

- o W3C Working Draft: The HTML3 Table Model New!
- o W3C Announces Platform for Internet Content Selection (PICS)
- o Workshop on WWW and Collaboration: 11–12 Sep 1995
- o Fourth International World Wide Web Conference: 11–14 December 1995
- o News and Updates Archive

Web Specifications and Development Areas

W3C Tech Reports New!
Technical reports and draft specifications released by

http://www.w3.org/hypertext/WWW/TheProject.html

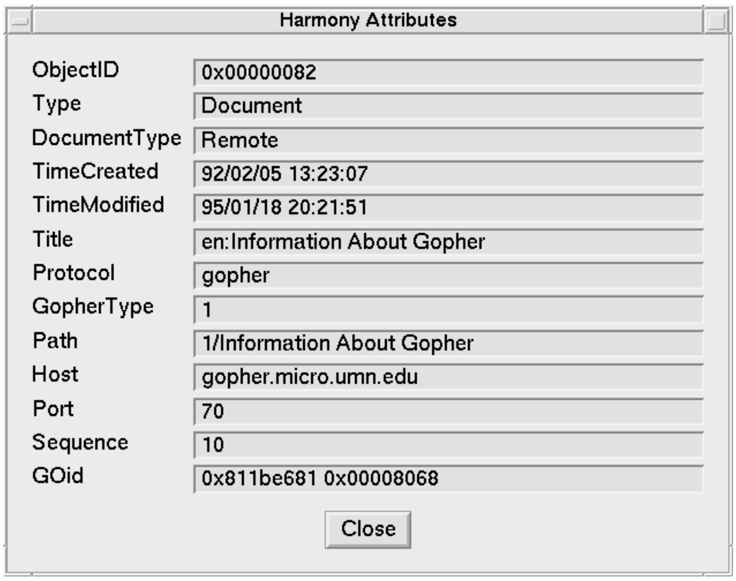
Figure 5.30 The WWW project page at W3C displayed in Harmony.

5.9 Harmony System Functions

Harmony's system-oriented functions include user identification, changing passwords and explicitly clearing Harmony's cache. The Harmony Identify dialogue, shown in Figure 5.33, allows anonymous users to identify themselves to the server by entering their Hyper-G username and password. Identified users can make themselves anonymous by entering anon in the Login field. Identified users with system rights can identify themselves as any other user simply by entering the username in the Login field, without being required to enter a password.

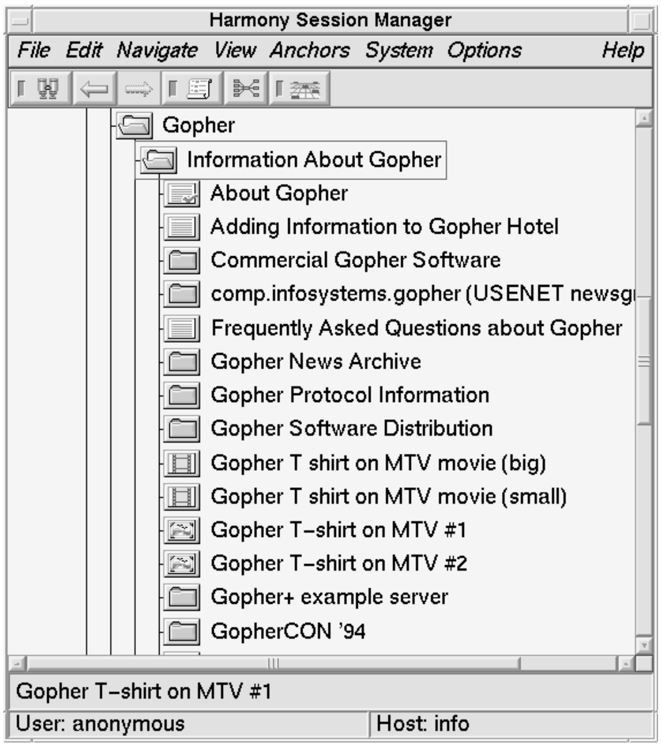
The Harmony Change Password dialogue, illustrated in Figure 5.34, allows identified users to change their password. The new password is entered twice to guard against typing errors. For simplicity, a new password can also be entered in pre-encrypted form, for example when cutting and pasting from the UNIX password file.

Harmony maintains its own client-side cache, which may be explicitly cleared. Clearing the cache forces Harmony to (re-)fetch subsequently accessed documents from the server, rather than relying on cached copies.



ObjectID	0x00000082
Type	Document
DocumentType	Remote
TimeCreated	92/02/05 13:23:07
TimeModified	95/01/18 20:21:51
Title	en:Information About Gopher
Protocol	gopher
GopherType	1
Path	1/Information About Gopher
Host	gopher.micro.umn.edu
Port	70
Sequence	10
GOid	0x811be681 0x00008068

Figure 5.31 A Hyper-G remote object representing a Gopher menu at the University of Minnesota's Gopher server.



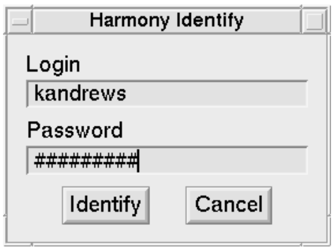
Gopher
Information About Gopher
About Gopher
Adding Information to Gopher Hotel
Commercial Gopher Software
comp.infosystems.gopher (USENET newsgı
Frequently Asked Questions about Gopher
Gopher News Archive
Gopher Protocol Information
Gopher Software Distribution
Gopher T shirt on MTV movie (big)
Gopher T shirt on MTV movie (small)
Gopher T-shirt on MTV #1
Gopher T-shirt on MTV #2
Gopher+ example server
GopherCON '94

Gopher T-shirt on MTV #1
User: anonymous Host: info

Figure 5.32 The Gopher menu displayed as a collection tree in Harmony.

5.10 Harmony Options and Preferences

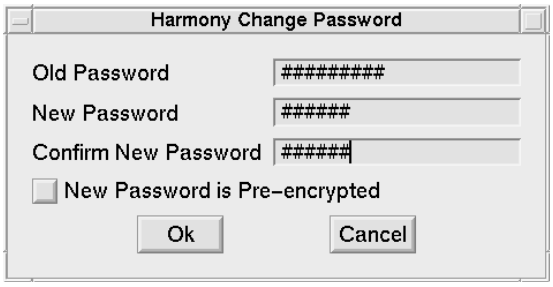
Harmony options and preferences include language settings and the default sort order for members of a collection. Harmony's Language Preference dialogue allows users to specify a prioritised list of preferred languages.



Login
kandrews

Password
#########

Figure 5.33 The Harmony Identify dialogue.



Old Password #########

New Password ######

Confirm New Password ######

□ New Password is Pre-encrypted

Figure 5.34 The Harmony Change Password dialogue.

In the example in Figure 5.35, English is currently the first choice language, followed by German. The language preference determines both Harmony's user interface and the selection of documents available in multiple languages. Languages can be selected and deselected by double-clicking and by dragging and dropping between the boxes. The preference order of selected languages can be changed by dragging and dropping inside the Selected box.

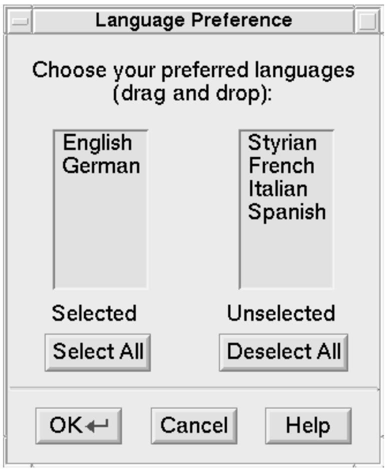
The Default Sort Order dialogue, depicted in Figure 5.36, allows users to specify the sort order used by default when listing members of a collection in the collection browser. The actual sort order can be overridden by setting a collection's SortOrder attribute. Hence the default sort order determines the order in which members of a collection are listed, *only* insofar as no particular sort order has been specified by the author of the collection.

In the Default Sort Order dialogue, individual sort criteria can be selected and deselected both by double-clicking and by dragging and dropping between the boxes. The order of selected sort criteria is significant. Each individual criterion can be applied in descending rather than ascending order by clicking on the ▤ button which precedes it. In the example above, the members of collections for which no particular sort order has been defined will henceforth be listed first by ascending order of sequence number, then in ascending alphabetical order of title.

Other options include Scan Attributes and Global Viewers. Turning Scan Attribute on brings up Harmony's Attribute window and continually updates it to reflect the Hyper-G object currently beneath the mouse cursor. This allows quick scanning of objects in the collection browser and their associated attributes. Turning Global Viewer off instructs the Session Manager principally to start a new document viewer every time a document is retrieved, resulting in multiple document viewers. Turning Global Viewer on instructs the Session Manager to reuse existing document viewers by displaying documents in an existing viewer of the same type (if one is already available).

5.11 Multilingual Support

Hyper-G's cluster mechanism allows authors to easily create and manage multiple language versions of hyperdocuments. Harmony supports Hyper-G's multilingual clusters: it displays the most appropri-



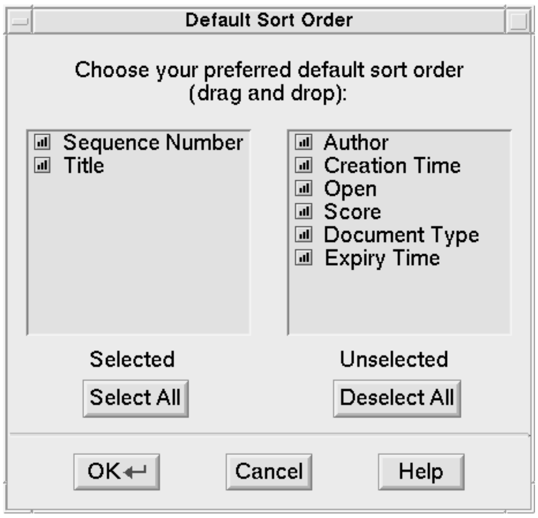
Language Preference

Choose your preferred languages (drag and drop):

Selected: English, German

Unselected: Styrian, French, Italian, Spanish

Figure 5.35 The Harmony Language Preference dialogue.



Default Sort Order

Choose your preferred default sort order (drag and drop):

Selected: Sequence Number, Title

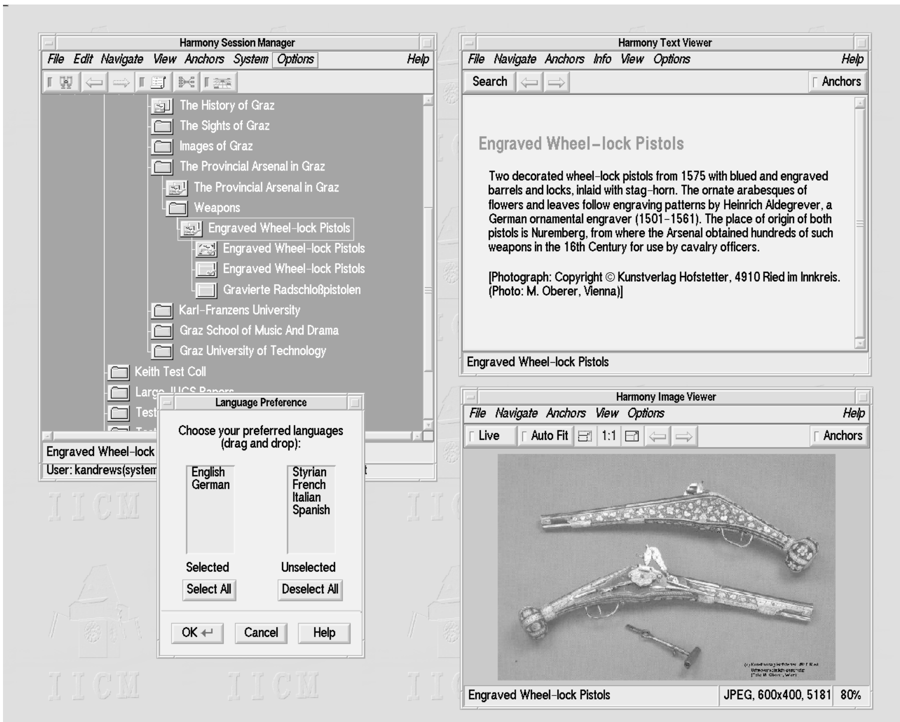
Unselected: Author, Creation Time, Open, Score, Document Type, Expiry Time

Figure 5.36 The Harmony Default Sort Order dialogue.

ate language version of each type of document in a cluster according to the user's language preference setting. Harmony also implements its user interface in multiple languages. Changing one's language preference changes both the user interface and the information content (as far as they are available in the chosen language, of course). This is illustrated in Figures 5.37 and 5.38. In Figure 5.37, the user is about to change the default language priority from first English then German to the reverse. Figure 5.38 shows the effect of this change: the German titles of objects are displayed where available, German language documents in clusters are displayed in preference to English ones, and Harmony's user interface components have changed to German.

5.12 System Status and Communication with Other Users

The Harmony Status Browser, depicted in Figure 5.39, displays the current system status of the Hyper-G server Harmony is connected to. Status information includes the server name, up time, a performance index, and a list of users currently connected to the server. By default, the status

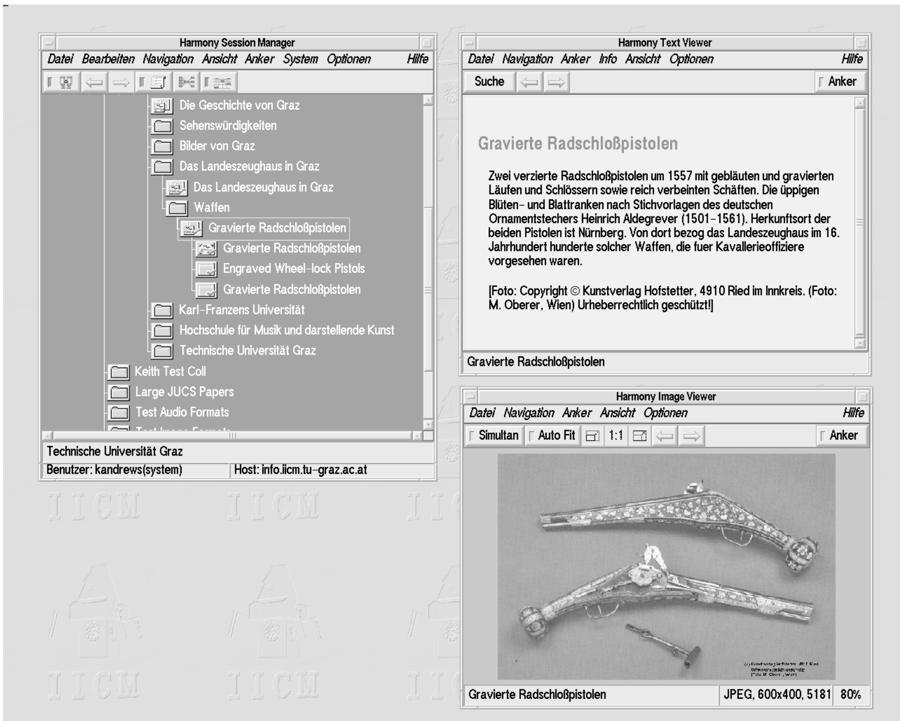

Harmony Session Manager
File Edit Navigate View Anchors System Options Help
The History of Graz
The Sights of Graz
Images of Graz
The Provincial Arsenal in Graz
The Provincial Arsenal in Graz
Weapons
Engraved Wheel-lock Pistols
Engraved Wheel-lock Pistols
Engraved Wheel-lock Pistols
Gravierte Radschloßpistolen
Karl-Franzens University
Graz School of Music And Drama
Graz University of Technology
Keith Test Coll
Engraved Wheel-lock
User: kandrews(system
Language Preference
Choose your preferred languages (drag and drop):
English
German
Styrian
French
Italian
Spanish
Selected
Unselected
Select All
Deselect All
OK
Cancel
Help
Harmony Text Viewer
File Navigate Anchors Info View Options Help
Search
Anchors
Engraved Wheel-lock Pistols
Harmony Image Viewer
File Navigate Anchors View Options Help
Live
Auto Fit
1:1
Anchors
Engraved Wheel-lock Pistols
JPEG, 600x400, 5181
80%

Engraved Wheel-lock Pistols

Two decorated wheel-lock pistols from 1575 with blued and engraved barrels and locks, inlaid with stag-horn. The ornate arabesques of flowers and leaves follow engraving patterns by Heinrich Aldegrever, a German ornamental engraver (1501–1561). The place of origin of both pistols is Nuremberg, from where the Arsenal obtained hundreds of such weapons in the 16th Century for use by cavalry officers.

[Photograph: Copyright © Kunstverlag Hofstetter, 4910 Ried im Innkreis. (Photo: M. Oberer, Vienna)]

Figure 5.37 Harmony's multilingual support. The user is about to change the language preference order, currently first English then German, to the reverse.

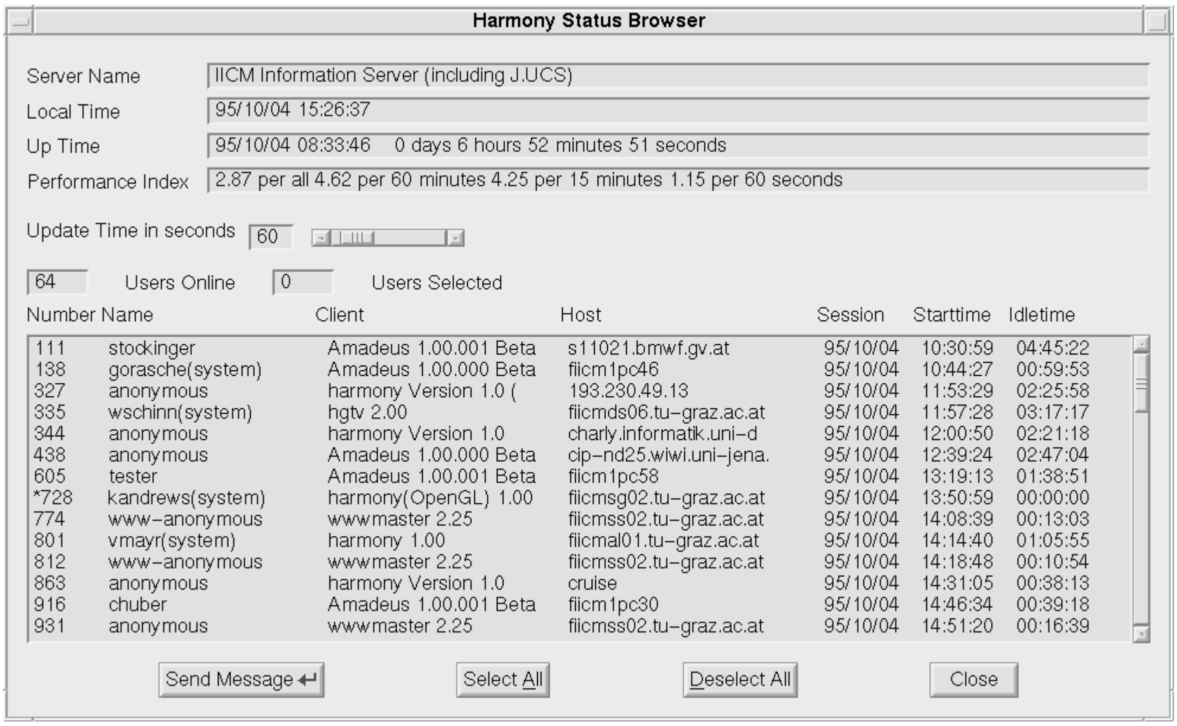

Harmony Session Manager
Datei Bearbeiten Navigation Ansicht Anker System Optionen Hilfe
Die Geschichte von Graz
Sehenswürdigkeiten
Bilder von Graz
Das Landeszeughaus in Graz
Das Landeszeughaus in Graz
Waffen
Gravierte Radschloßpistolen
Gravierte Radschloßpistolen
Engraved Wheel-lock Pistols
Gravierte Radschloßpistolen
Karl-Franzens Universität
Hochschule für Musik und darstellende Kunst
Technische Universität Graz
Keith Test Coll
Large JUCS Papers
Test Audio Formats
Technische Universität Graz
Benutzer: kandrews(system)
Host: info.iicm.tu-graz.ac.at
Harmony Text Viewer
Datei Navigation Anker Info Ansicht Optionen Hilfe
Suche
Anker
Gravierte Radschloßpistolen
Harmony Image Viewer
Datei Navigation Anker Ansicht Optionen Hilfe
Simultan
Auto Fit
1:1
Anker
Gravierte Radschloßpistolen
JPEG, 600x400, 5181
80%

Gravierte Radschloßpistolen

Zwei verzierte Radschloßpistolen um 1557 mit gebläuten und gravierten Läufen und Schlössern sowie reich verbeinten Schäften. Die üppigen Blüten- und Blattranken nach Stichvorlagen des deutschen Ornamentstechers Heinrich Aldegrever (1501–1561). Herkunftsort der beiden Pistolen ist Nürnberg. Von dort bezog das Landeszeughaus im 16. Jahrhundert hunderte solcher Waffen, die fuer Kavallerieoffiziere vorgesehen waren.

[Foto: Copyright © Kunstverlag Hofstetter, 4910 Ried im Innkreis. (Foto: M. Oberer, Wien) Urheberrechtlich geschützt!]

Figure 5.38 Harmony's multilingual support. Both the user interface and the information content have now changed to German.



Harmony Status Browser

Server Name: IICM Information Server (including J.UCS)

Local Time: 95/10/04 15:26:37

Up Time: 95/10/04 08:33:46 0 days 6 hours 52 minutes 51 seconds

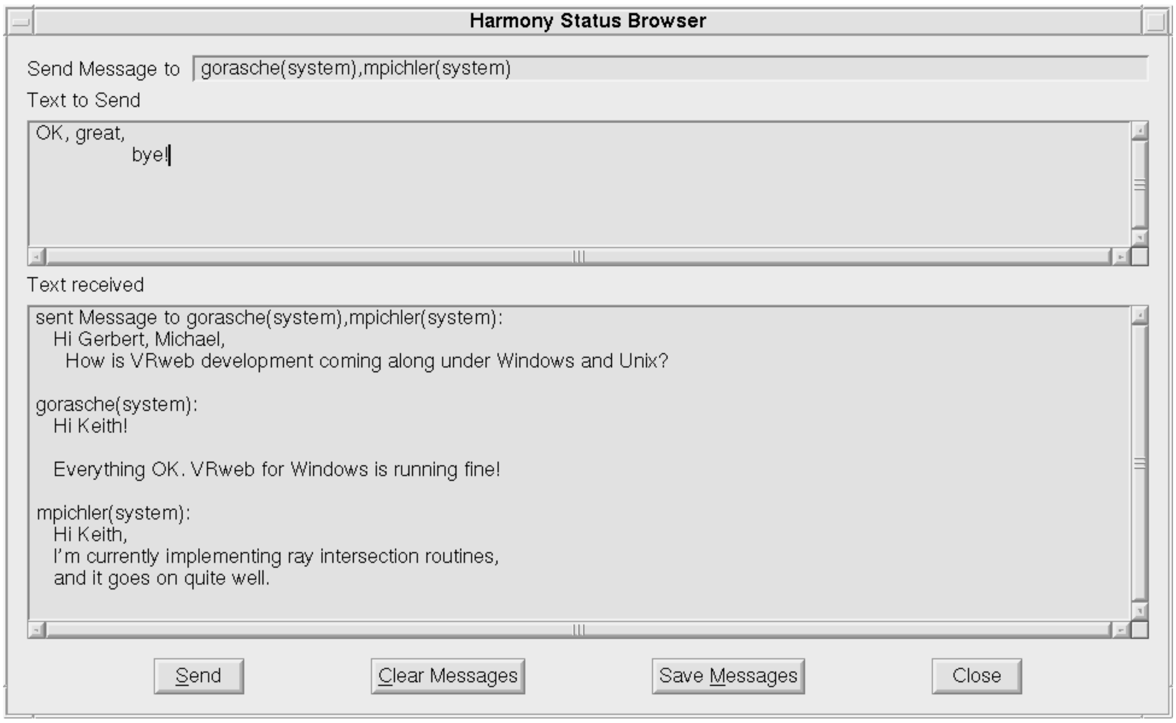
Performance Index: 2.87 per all 4.62 per 60 minutes 4.25 per 15 minutes 1.15 per 60 seconds

Update Time in seconds: 60

64 Users Online 0 Users Selected

Number	Name	Client	Host	Session	Starttime	Idletime
111	stockinger	Amadeus 1.00.001 Beta	s11021.bmwf.gv.at	95/10/04	10:30:59	04:45:22
138	gorasche(system)	Amadeus 1.00.000 Beta	fiicm1pc46	95/10/04	10:44:27	00:59:53
327	anonymous	harmony Version 1.0 (	193.230.49.13	95/10/04	11:53:29	02:25:58
335	wschinn(system)	hgtv 2.00	fiicmds06.tu–graz.ac.at	95/10/04	11:57:28	03:17:17
344	anonymous	harmony Version 1.0	charly.informatik.uni–d	95/10/04	12:00:50	02:21:18
438	anonymous	Amadeus 1.00.000 Beta	cip–nd25.wiwi.uni–jena.	95/10/04	12:39:24	02:47:04
605	tester	Amadeus 1.00.001 Beta	fiicm1pc58	95/10/04	13:19:13	01:38:51
*728	kandrews(system)	harmony(OpenGL) 1.00	fiicmsg02.tu–graz.ac.at	95/10/04	13:50:59	00:00:00
774	www–anonymous	wwwmaster 2.25	fiicmss02.tu–graz.ac.at	95/10/04	14:08:39	00:13:03
801	vmayr(system)	harmony 1.00	fiicmal01.tu–graz.ac.at	95/10/04	14:14:40	01:05:55
812	www–anonymous	wwwmaster 2.25	fiicmss02.tu–graz.ac.at	95/10/04	14:18:48	00:10:54
863	anonymous	harmony Version 1.0	cruise	95/10/04	14:31:05	00:38:13
916	chuber	Amadeus 1.00.001 Beta	fiicm1pc30	95/10/04	14:46:34	00:39:18
931	anonymous	wwwmaster 2.25	fiicmss02.tu–graz.ac.at	95/10/04	14:51:20	00:16:39

Figure 5.39 The Harmony Status Browser.



Harmony Status Browser

Send Message to: gorasche(system),mpichler(system)

Text to Send

```
OK, great,
          bye!
```

Text received

```
sent Message to gorasche(system),mpichler(system):
  Hi Gerbert, Michael,
    How is VRweb development coming along under Windows and Unix?

gorasche(system):
  Hi Keith!

  Everything OK. VRweb for Windows is running fine!

mpichler(system):
  Hi Keith,
  I'm currently implementing ray intersection routines,
  and it goes on quite well.
```

Figure 5.40 Sending messages between users in Harmony.

information is updated every 60 seconds, but this period can be changed by the user. The user's own current session is marked in the list with an asterisk (for example, `*728 kandrews(system)` in the figure).

It is possible to send arbitrary messages to one or more current users via the Hyper-G server. Messages are displayed on the user's terminal by their own Hyper-G client. This powerful facility can be used by a server administrator to notify all connected users of a pending server shutdown, for example. It is also used to send certain system messages. In Figure 5.40, three users are undertaking a three-way conference. Such conferences can involve any number of users logged on to the Hyper-G server from anywhere in the world.

The messaging facilities hint at one possible direction for future extension of Harmony: collaborative work. The Hyper-G server already has facilities to keep track of which users are logged on and which documents and collections they are currently accessing. Expanding on this to allow group discussions of particular documents, interactive guided tours, and the provision of collaborative tools such as shared whiteboards would be extremely valuable.

5.13 Further Reading

The *Harmony User Guide* [Mayrhofer and Andrews, 1996], included in the Harmony distribution, is a practical how-to guide to Harmony. Published articles on Harmony include [Andrews and Kappe, 1994; Andrews, 1995a]. Harmony is covered in more tutorial style in *HyperWave: The New Generation Internet Information System Based on Hyper-G Technology* [Dalitz and Heyer, 1996].

Good handbooks on HTML include [Flynn, 1995; Musciano and Kennedy, 1996]. An authoritative reference for image and video formats is [Murray and van Ryper, 1996]. The standard reference for the PostScript language is [Adobe Systems Inc., 1990].

Information on the Virtual Reality Modeling Language (VRML) is available at the VRML Repository [Eubanks *et al.*, 1996] or in Mark Pesce's book [Pesce, 1995].

6 Building Cyberspace with Harmony

This chapter deals with Harmony's extensive information management and authoring facilities. It is a revised version of material first published in Chapter 23 of *HyperWave: The Next Generation Web Solution*, Addison-Wesley, May 1996 [Andrews, 1996a].

6.1 Moving Beyond the Provider-Consumer Paradigm

Although it was designed with collaboration and remote authoring in mind, the World Wide Web today is primarily an environment where providers provide information and users consume information[1]. Hyper-G goes a step beyond this paradigm, blurring the distinction between users and authors: depending on their access rights, every user with an account on the Hyper-G server is also an author.

In addition to browsing and navigating, my second major goal for the Harmony client was to provide extensive in-built authoring facilities, which would be available to authors working on remote machines via the Internet and thus open up all kinds of possibilities for collaborative authoring and virtual communities. Given appropriate access permissions, for example, I can work on a document collection at the University of Auckland, or can grant you the reader write access to certain collections or documents on our Hyper-G server here in Graz.

The notable contribution of Harmony in this respect is its suite of (remote) authoring and management facilities, including:

- Structure editing in collections and clusters.
- Document editing while preserving links.
- Interactive graphical link editing in all media.
- Attribute editing.
- Interactive uploading, insertion, and deletion.
- Making annotations to documents and collections.

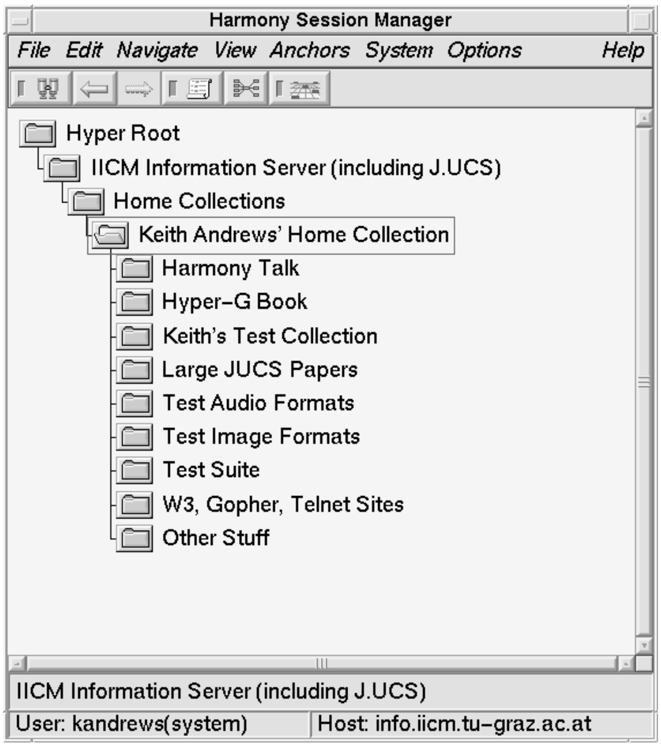
Most of these facilities have been available in Harmony since 1993 and 1994 and are only now beginning to become available in commercial Web packages such as Netscape LiveWire [Netscape, 1996], Microsoft FrontPage [Microsoft, 1996], and Adobe SiteMill [Adobe, 1996]. However, as InformationWeek reporter Jason Levitt says [Levitt, 1996]:

> *"Unfortunately, due to a lack of Web standards, none of these tools allows you to manage Web servers from a variety of vendors. Specifically, there is no agreed-upon method for remote management of Web servers–that is, to retrieve Web pages over a network, edit them, and then write them back to the server using HTTP."*

Interactive link editing and document editing facilities similar to those in Harmony were demonstrated as early as 1986 in the Intermedia system [Haan *et al.*, 1992], but Intermedia ran only on Apple A/UX and its clients did not communicate with the server via the Internet.

A second novel contribution of Harmony is the remote management of user accounts and user groups using a graphical tool called HarAdmin, which is described in Section 6.9.

[1] Tim Berners-Lee's original Web browser on the Next platform did, in fact, support editing, but issues of security and authentication have prevented widespread adoption of editing functionality over HTTP.



Hyper Root
IICM Information Server (including J.UCS)
Home Collections
Keith Andrews' Home Collection
Harmony Talk
Hyper–G Book
Keith's Test Collection
Large JUCS Papers
Test Audio Formats
Test Image Formats
Test Suite
W3, Gopher, Telnet Sites
Other Stuff

IICM Information Server (including J.UCS)
User: kandrews(system)
Host: info.iicm.tu–graz.ac.at

Figure 6.1 Keith Andrews' home collection, where objects will be inserted.

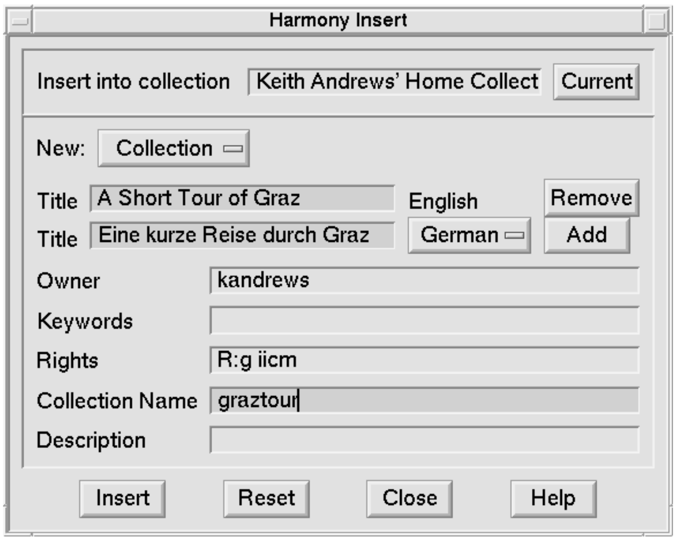
6.2 Object and Document Creation

For the purposes of the following explanation, let us assume that a hypermedia collection containing a short tour of the city of Graz is to be created. A Hyper-G user `kandrews` has been created and this user has a home collection and full access rights to it. It is common practice to work on draft versions of hyperdocuments in one's home collection, possibly protected from outside access, and later move or plug them into an appropriate location in the collection structure and make them visible. Figure 6.1 shows Keith Andrews' home collection just prior to starting work on this task.

Creating Collections and Clusters

The Harmony Insert Dialogue is used both to create completely new Hyper-G objects and to upload files accessible on the UNIX file system onto the server. Figure 6.2 shows the Insert Dialogue being used to create a collection for the tour of Graz. The collection is called `A Short Tour of Graz` in English and `Eine kurze Reise durch Graz` in German. The author (or owner) of the collection is `kandrews`, and the collection is protected so that only members of the group `iicm` can read it initially. A reasonably intuitive symbolic name, `graztour`, has been assigned to the collection; this name will later appear in URLs referencing the collection.

By default, objects are inserted into the collection current at the time the dialogue was activated. A different collection can be specified by selecting the desired collection and pressing the Insert Dialogue's `Current` button. Keywords and a description could also be associated with the collection, but these have been left empty for the time being. When the `Insert` button is pressed, this new empty collection is inserted as a sub-collection of `Keith Andrews' Home Collection`:



Insert into collection: Keith Andrews' Home Collect

New: Collection

Title: A Short Tour of Graz — English

Title: Eine kurze Reise durch Graz — German

Owner: kandrews

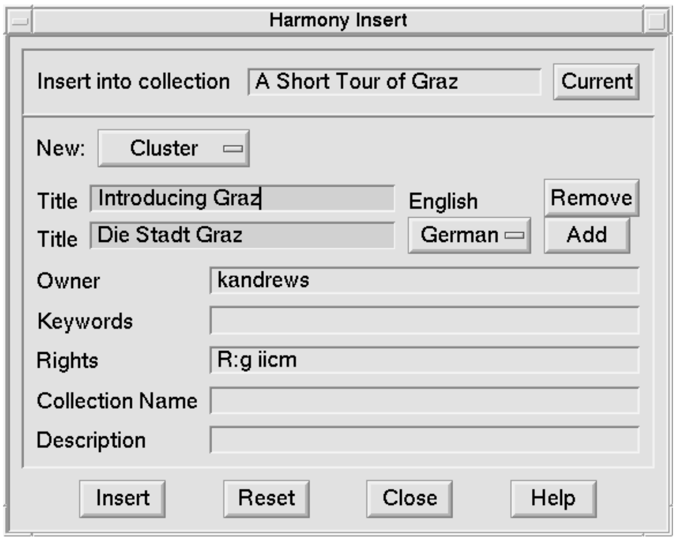
Keywords:

Rights: R:g iicm

Collection Name: graztour

Description:

Figure 6.2 Inserting a collection with the Harmony Insert Dialogue.



Insert into collection: A Short Tour of Graz

New: Cluster

Title: Introducing Graz — English

Title: Die Stadt Graz — German

Owner: kandrews

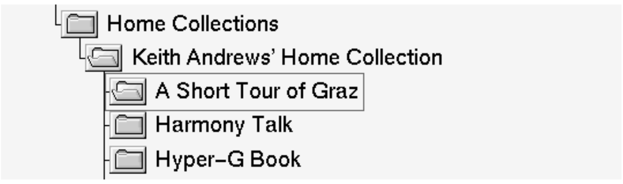
Keywords:

Rights: R:g iicm

Collection Name:

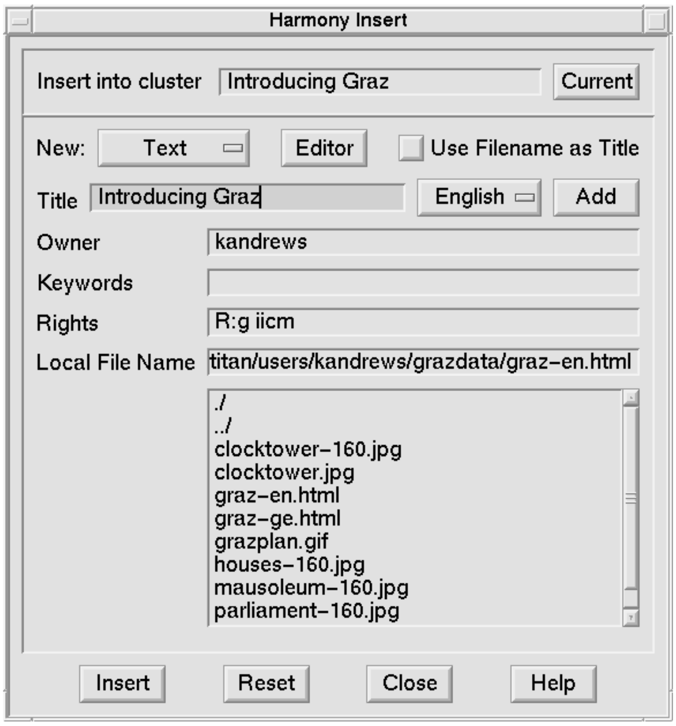
Description:

Figure 6.3 Inserting a cluster.



Home Collections
Keith Andrews' Home Collection
A Short Tour of Graz
Harmony Talk
Hyper–G Book

Creating clusters is equally straightforward. In Figure 6.3 a cluster is being inserted into the previously created collection. The cluster has been given both English and German titles and has inherited the same access rights as its parent collection (by default, the Insert Dialogue initialises the Rights attribute to the same value as the new object's parent collection). Titles in additional languages could be assigned using the `Add` button. The cluster is created and inserted into the collection `A Short Tour of Graz`:



Harmony Insert

Insert into cluster: Introducing Graz [Current]

New: Text [Editor] □ Use Filename as Title

Title: Introducing Graz | English [Add]

Owner: kandrews

Keywords:

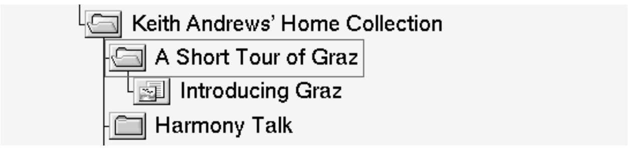
Rights: R:g iicm

Local File Name: titan/users/kandrews/grazdata/graz-en.html

```
./
../
clocktower-160.jpg
clocktower.jpg
graz-en.html
graz-ge.html
grazplan.gif
houses-160.jpg
mausoleum-160.jpg
parliament-160.jpg
```

[Insert] [Reset] [Close] [Help]

Figure 6.4 Uploading a text document.



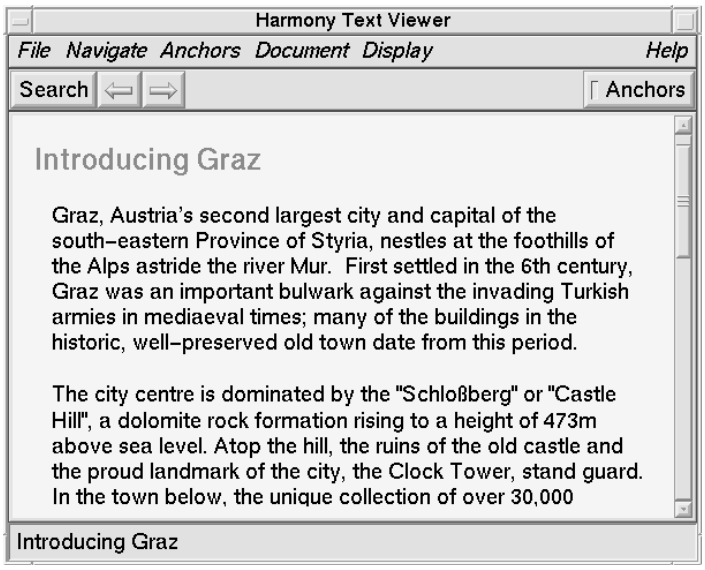
- Keith Andrews' Home Collection
  - A Short Tour of Graz
    - Introducing Graz
  - Harmony Talk

Of course, at this point the newly created cluster is still empty — it is time to learn how to upload documents onto the server.

Inserting Documents

By choosing `New: Text` in the Insert Dialogue's selector, text documents can be inserted into the Hyper-G server. The text can either be uploaded from an existing file or created from scratch in an editor. Figure 6.4 shows the text document `graz-en.html` being uploaded from the user's UNIX home directory into the previously created cluster. Text documents are generally written in HTML, although Hyper-G's original text format HTF is still supported. In the future, support will be available for arbitrary SGML documents with appropriate DTDs and style sheets. Clicking on the `Editor` button rather than specifying a filename would cause Harmony to start an empty editor for the new document, rather than uploading it from a file.

Since this particular text is written in English, only an English title has been assigned. After repeating the same Insert procedure for the equivalent German text `graz-ge.html`, the cluster now contains two member text documents:



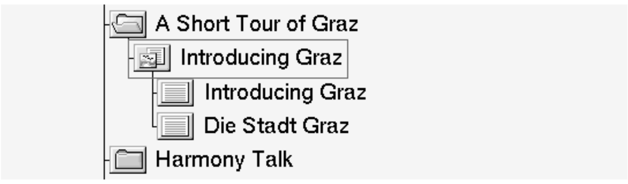
Introducing Graz

Graz, Austria's second largest city and capital of the south–eastern Province of Styria, nestles at the foothills of the Alps astride the river Mur. First settled in the 6th century, Graz was an important bulwark against the invading Turkish armies in mediaeval times; many of the buildings in the historic, well–preserved old town date from this period.

The city centre is dominated by the "Schloßberg" or "Castle Hill", a dolomite rock formation rising to a height of 473m above sea level. Atop the hill, the ruins of the old castle and the proud landmark of the city, the Clock Tower, stand guard. In the town below, the unique collection of over 30,000

Introducing Graz

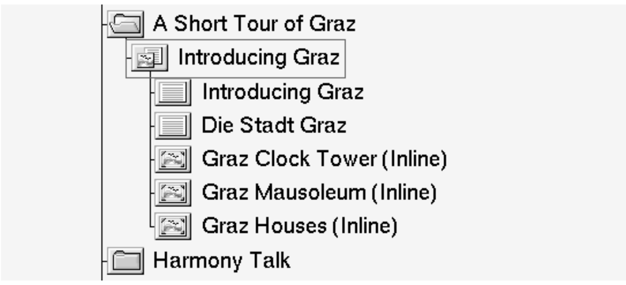
Figure 6.5 The newly uploaded text document.



A Short Tour of Graz
Introducing Graz
Introducing Graz
Die Stadt Graz
Harmony Talk

These documents can be displayed in the Harmony Text Viewer like any other text documents by simply double-clicking their icons. Figure 6.5 shows the first two paragraphs of the English text.

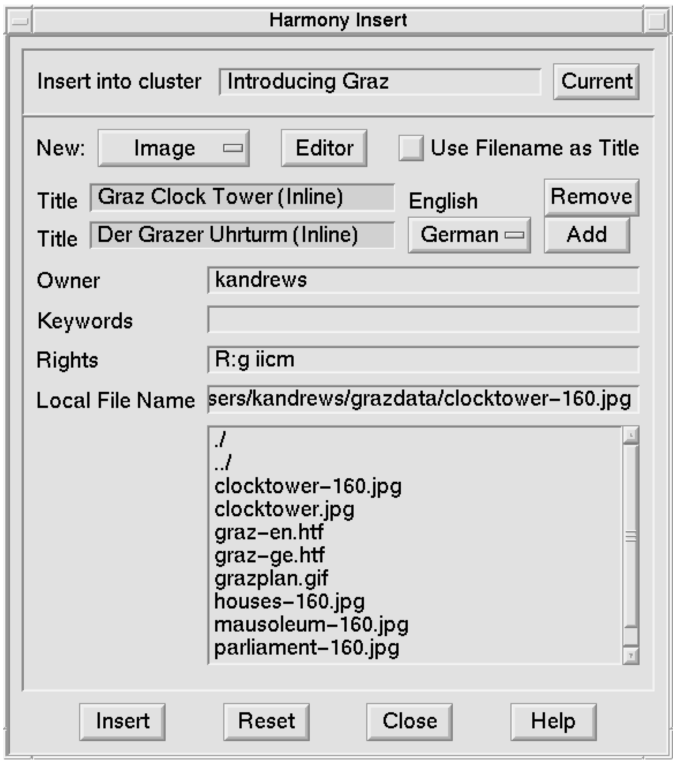
Let us continue the example collection about Graz. Three small JPEG images will now be uploaded into the previously created cluster `Introducing Graz` to be used later as inline images. The Insert Dialogue for uploading an image looks like the one in Figure 6.6. Since these images are independent of language, they are given titles in both English and German. The word `Inline` in the titles is purely for the author's information, it is not interpreted by the system in any special way. After uploading all three JPEG images, the cluster now has five member documents:



A Short Tour of Graz
Introducing Graz
Introducing Graz
Die Stadt Graz
Graz Clock Tower (Inline)
Graz Mausoleum (Inline)
Graz Houses (Inline)
Harmony Talk

At this point, each image can be inspected in the Harmony Image Viewer, as illustrated in Figure 6.7 for the Clock Tower image. Note that the user has turned the AutoFit feature off, so as not to distort the small inline image by overly enlarging it.

The Insert Dialogue can, in fact, be used to insert any kind of document or collection. The new object's type is simply chosen from among the choices in the `New` selector. The Harmony Session Manager performs a rudimentary magic number check on documents prior to uploading them, so as to detect when a document would be uploaded as the wrong type, for example an image declared as a text document. In such cases a warning dialogue appears.



Harmony Insert

Insert into cluster: Introducing Graz

New: Image

□ Use Filename as Title

Title: Graz Clock Tower (Inline) — English

Title: Der Grazer Uhrturm (Inline) — German

Owner: kandrews

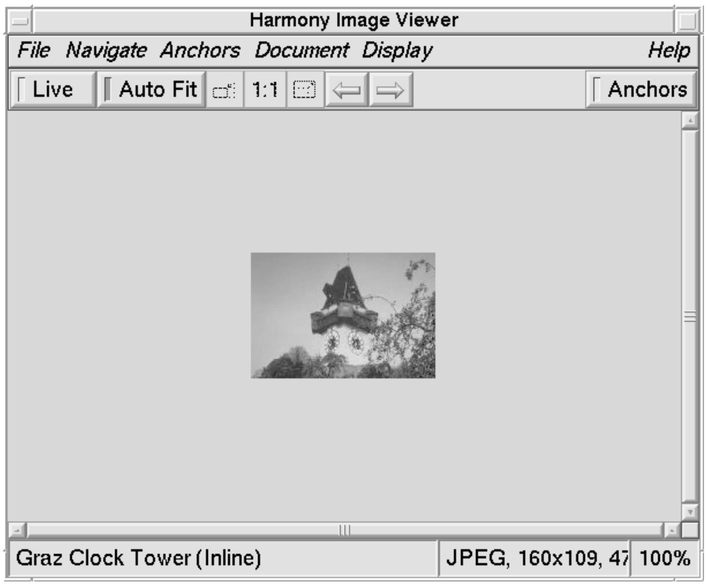
Keywords:

Rights: R:g iicm

Local File Name: sers/kandrews/grazdata/clocktower−160.jpg

```
./
../
clocktower−160.jpg
clocktower.jpg
graz−en.htf
graz−ge.htf
grazplan.gif
houses−160.jpg
mausoleum−160.jpg
parliament−160.jpg
```

Figure 6.6 Inserting an image document.

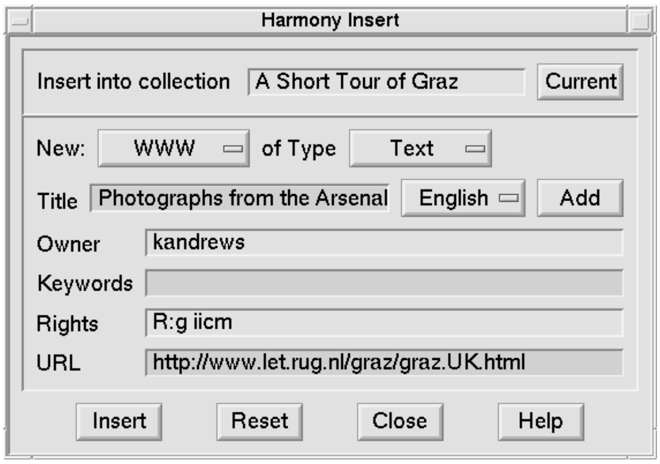


Harmony Image Viewer

Graz Clock Tower (Inline) | JPEG, 160x109, 47 | 100%

Figure 6.7 Inspecting the new image document.

PostScript documents and 3d models, which may be very large in uncompressed form but usually respond well to compression, are generally compressed (GNU gzip or UNIX compress) externally by the user before inserting them into the Hyper-G server. Transparently to the end user, clients retrieve the compressed files across the Net, hence saving bandwidth, and decompress them locally.



Harmony Insert

Insert into collection: A Short Tour of Graz

New: WWW of Type Text

Title: Photographs from the Arsenal — English

Owner: kandrews

Keywords:

Rights: R:g iicm

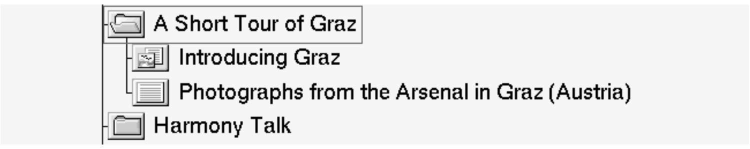
URL: http://www.let.rug.nl/graz/graz.UK.html

Figure 6.8 Inserting a remote WWW document.

Creating Remote Objects

Remote objects include documents and directories on WWW, Gopher and ftp servers, and telnet sessions to remote sites. Access to such remote objects from within a Hyper-G context is achieved by creating a Hyper-G remote object to represent the remote entity and specifying its attributes (such as Host and Port) accordingly.

To continue with our example, Bert Bos of the University of Groningen in Holland (and now with W3C) has, on his WWW server, a set of photographs of the Arsenal in Graz, which he took during his 1993 visit. Figure 6.8 shows the Insert dialogue being used to create a reference to the remote WWW page introducing this set of photographs. The object appears in the collection structure as a new text document:



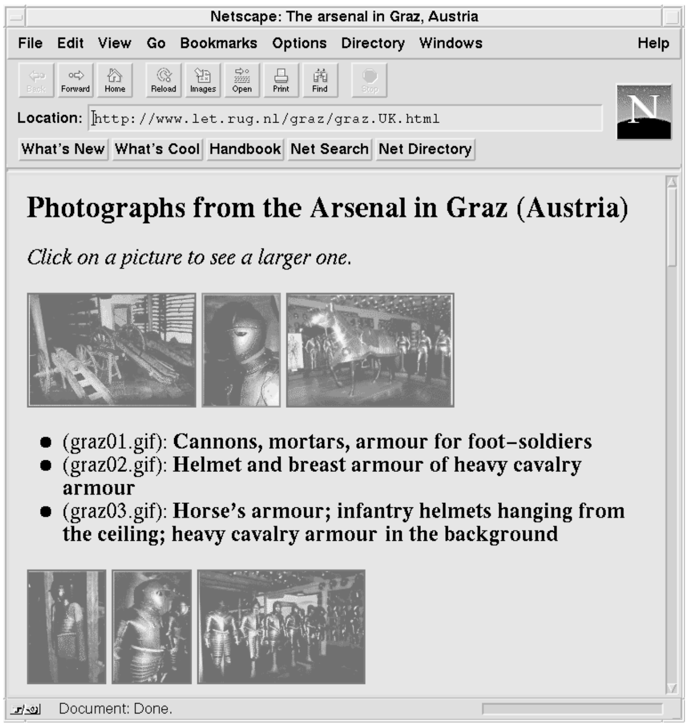
- A Short Tour of Graz
  - Introducing Graz
  - Photographs from the Arsenal in Graz (Austria)
- Harmony Talk

The object can now be accessed with Harmony or any other Web browser, as shown in Figure 6.9 using Netscape.

6.3 Document Editing

As well as facilities for uploading documents, Harmony provides facilities for interactively editing documents already on a Hyper-G server. There are two mechanisms in Harmony for document editing. First, the Text and Image viewers are equipped with an Edit function. When the `Edit` function is activated, the document viewer writes a temporary copy of the document it is displaying to a file and starts an external editor to edit the temporary file. Modifications to the temporary file (after saving to disk) can be optionally previewed in the document viewer. When the external editor is exited, the document viewer presents a dialogue box asking the user whether or not the modified version of the document should be uploaded back to the Hyper-G server, replacing the original.

As an alternative, an *Edit API* is available, which can be called through Harmony's Tools menu to edit any type of document. The Edit process runs alongside Harmony and its small Edit window is always available to edit the currently selected document.



Location: http://www.let.rug.nl/graz/graz.UK.html

Photographs from the Arsenal in Graz (Austria)

Click on a picture to see a larger one.

- (graz01.gif): **Cannons, mortars, armour for foot–soldiers**
- (graz02.gif): **Helmet and breast armour of heavy cavalry armour**
- (graz03.gif): **Horse's armour; infantry helmets hanging from the ceiling; heavy cavalry armour in the background**

Document: Done.

Figure 6.9 Viewing the remote WWW object just inserted.

Editing Text Documents

The Harmony Text Viewer's Edit function starts the Emacs editor by default. Figure 6.10 shows Emacs being used to edit the English text about Graz. Having corrected a spelling mistake and emphasised the words *second largest*, the user has saved the file to disk (without exiting Emacs) and in is previewing the modified document in the Text Viewer.

Harmony can, of course, be configured to use any editor for text documents, for example a dedicated HTML authoring tool such as AsWedit (as shown in Figure 6.11) or Symposia. To specify the desired editing tool, use the Text Viewer's `editcommand` X resource:

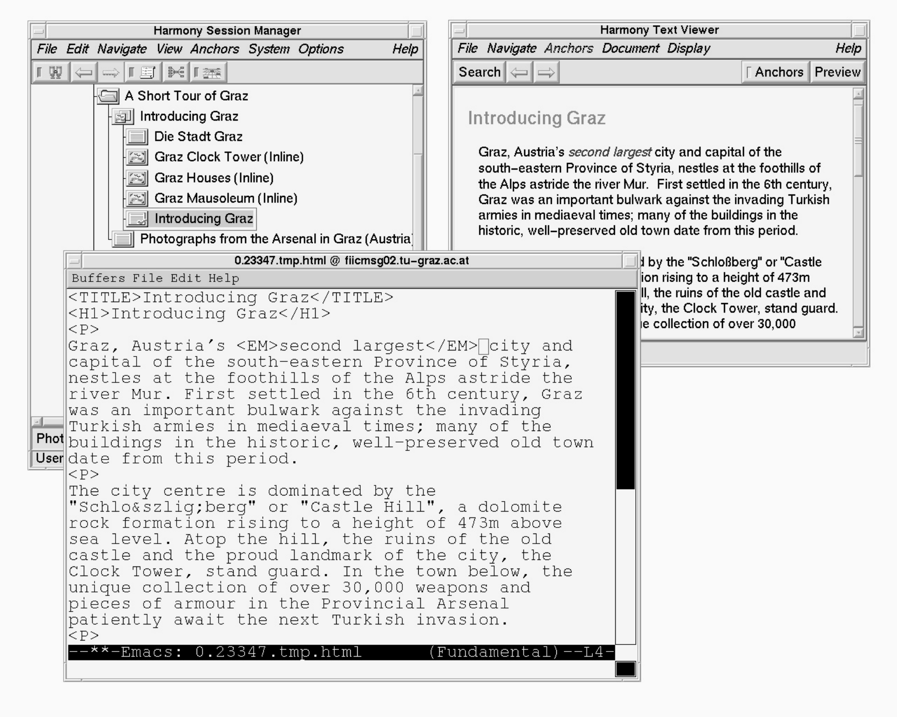
```
Harmony.Text.editcommand:        asWedit
```

The name of the temporary file is appended to the command automatically. When editing is finished and the external editor has been exited, the Harmony Text Viewer presents a dialogue box asking the user whether or not to insert the modified version of the text document.

Hyperlinks in Hyper-G are not stored embedded in documents but externally in a separate link database. Anchor regions in a text document are defined by character offsets. To ensure that link anchors in a text remain associated with the same text region (word, phrase and so on) when a file is edited, they are merged into the temporary text file prior to editing. Hence, link anchors slide around with their associated regions as the text is edited. When editing is finished, the links are extracted and re-inserted into the Hyper-G link database, and the document per se is inserted back into the Hyper-G document server.

Editing Image Documents

Interactive editing of images is analogous to text editing. The image is written to a temporary file by the Harmony Image Viewer and an external editor is started. Figure 6.12 shows an image used as a


Harmony Session Manager
File Edit Navigate View Anchors System Options Help
A Short Tour of Graz
Introducing Graz
Die Stadt Graz
Graz Clock Tower (Inline)
Graz Houses (Inline)
Graz Mausoleum (Inline)
Introducing Graz
Photographs from the Arsenal in Graz (Austria)
Phot
User
Harmony Text Viewer
File Navigate Anchors Document Display Help
Search
Anchors
Preview
0.23347.tmp.html @ fiicmsg02.tu-graz.ac.at
Buffers File Edit Help

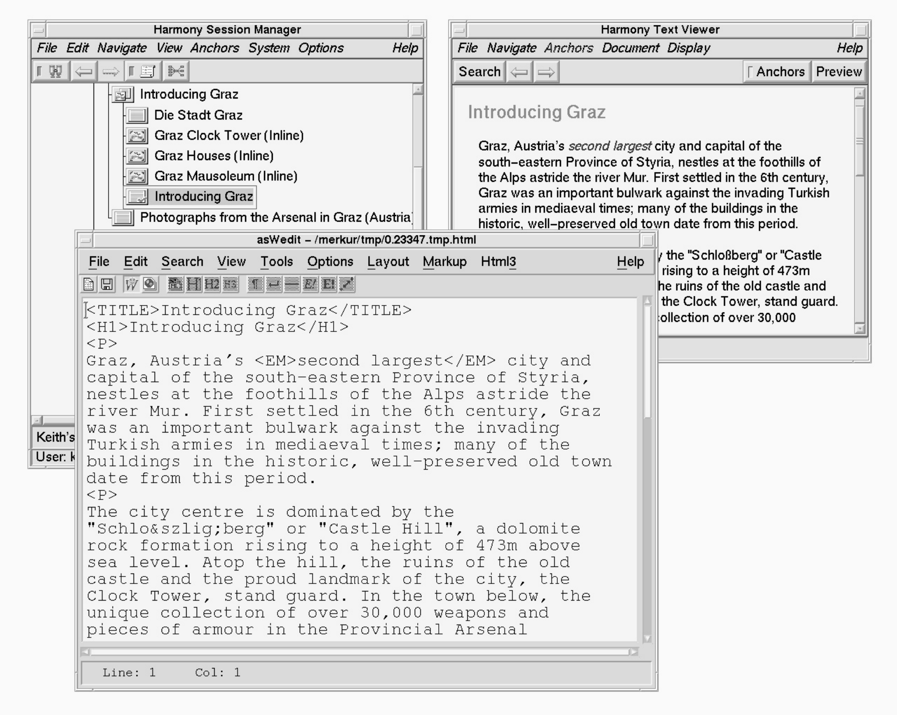
Introducing Graz

Graz, Austria's *second largest* city and capital of the south–eastern Province of Styria, nestles at the foothills of the Alps astride the river Mur. First settled in the 6th century, Graz was an important bulwark against the invading Turkish armies in mediaeval times; many of the buildings in the historic, well–preserved old town date from this period.

d by the "Schloßberg" or "Castle ion rising to a height of 473m ll, the ruins of the old castle and ity, the Clock Tower, stand guard. e collection of over 30,000

```
<TITLE>Introducing Graz</TITLE>
<H1>Introducing Graz</H1>
<P>
Graz, Austria's <EM>second largest</EM> city and
capital of the south-eastern Province of Styria,
nestles at the foothills of the Alps astride the
river Mur. First settled in the 6th century, Graz
was an important bulwark against the invading
Turkish armies in mediaeval times; many of the
buildings in the historic, well-preserved old town
date from this period.
<P>
The city centre is dominated by the
"Schlo&szlig;berg" or "Castle Hill", a dolomite
rock formation rising to a height of 473m above
sea level. Atop the hill, the ruins of the old
castle and the proud landmark of the city, the
Clock Tower, stand guard. In the town below, the
unique collection of over 30,000 weapons and
pieces of armour in the Provincial Arsenal
patiently await the next Turkish invasion.
<P>
--**-Emacs: 0.23347.tmp.html      (Fundamental)--L4-
```

Figure 6.10 Editing a text with Emacs from the Harmony Text Viewer.


Harmony Session Manager
File Edit Navigate View Anchors System Options Help
Introducing Graz
Die Stadt Graz
Graz Clock Tower (Inline)
Graz Houses (Inline)
Graz Mausoleum (Inline)
Introducing Graz
Photographs from the Arsenal in Graz (Austria)
Keith's
User: k
Harmony Text Viewer
File Navigate Anchors Document Display Help
Search
Anchors
Preview
asWedit – /merkur/tmp/0.23347.tmp.html
File Edit Search View Tools Options Layout Markup Html3 Help
H2 H3

Introducing Graz

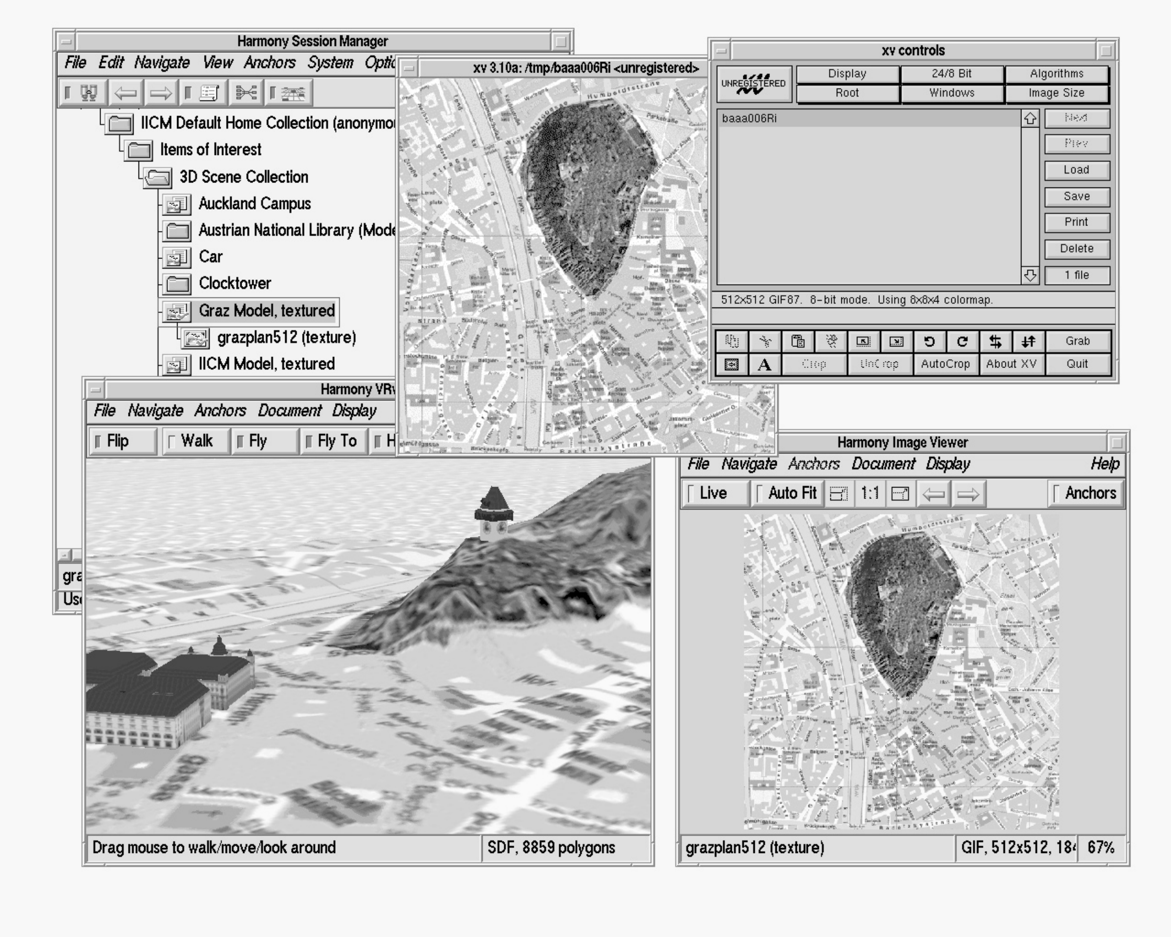
Graz, Austria's *second largest* city and capital of the south–eastern Province of Styria, nestles at the foothills of the Alps astride the river Mur. First settled in the 6th century, Graz was an important bulwark against the invading Turkish armies in mediaeval times; many of the buildings in the historic, well–preserved old town date from this period.

y the "Schloßberg" or "Castle rising to a height of 473m he ruins of the old castle and the Clock Tower, stand guard. collection of over 30,000

```
<TITLE>Introducing Graz</TITLE>
<H1>Introducing Graz</H1>
<P>
Graz, Austria's <EM>second largest</EM> city and
capital of the south-eastern Province of Styria,
nestles at the foothills of the Alps astride the
river Mur. First settled in the 6th century, Graz
was an important bulwark against the invading
Turkish armies in mediaeval times; many of the
buildings in the historic, well-preserved old town
date from this period.
<P>
The city centre is dominated by the
"Schlo&szlig;berg" or "Castle Hill", a dolomite
rock formation rising to a height of 473m above
sea level. Atop the hill, the ruins of the old
castle and the proud landmark of the city, the
Clock Tower, stand guard. In the town below, the
unique collection of over 30,000 weapons and
pieces of armour in the Provincial Arsenal

  Line: 1     Col: 1
```

Figure 6.11 Editing a HTML text with AsWedit from the Harmony Text Viewer.



512x512 GIF87. 8-bit mode. Using 8x8x4 colormap.

Drag mouse to walk/move/look around | SDF, 8859 polygons

grazplan512 (texture) | GIF, 512x512, 18 | 67%

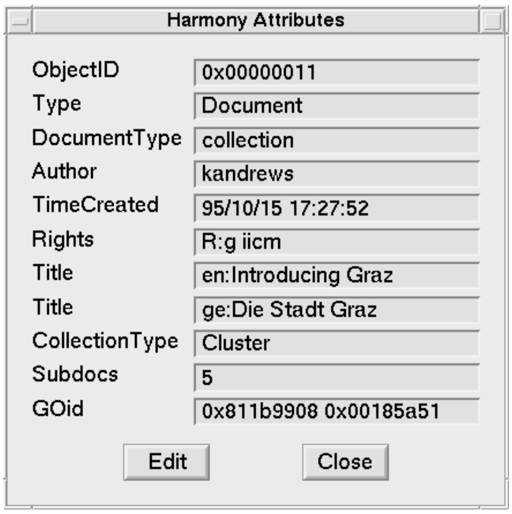
Figure 6.12 Editing an image with XV.

texture in a 3d cityscape of Graz being edited with XV. When the editor is exited, the Image Viewer asks whether or not to upload the new image in place of the original version.

Hyperlinks in image documents are defined in terms of a virtual coordinate space, where (0,0) is the lower left corner of the image and (1,1) the upper right. Operations which do not change the aspect ratio of the image, such as making it brighter or saving in a different format, are perfectly safe. However, cropping or otherwise changing the aspect ratio of the image will cause link anchors in the image to slide, since the Image Viewer does not attempt to remap the positions of link anchors. In such cases, it is necessary to manually check and, if necessary, reposition any anchors in the image.

6.4 Modifying Attributes

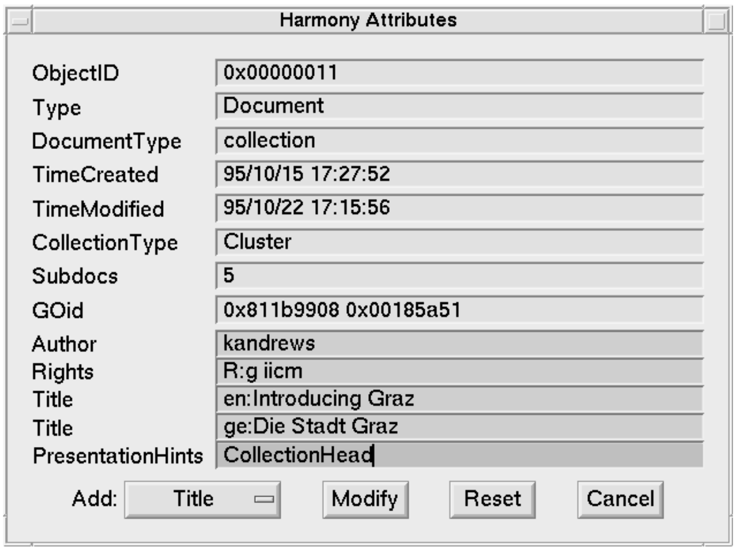
The attributes of a collection, cluster, or document can be displayed in Harmony's Attribute window by right-clicking the corresponding object in the collection browser or alternatively via Session Manager or document viewer menus. Staying with our Graz example, the attributes for the cluster `Introducing Graz` are shown in Figure 6.13. This is the cluster we created previously as an introduction to our collection `A Short Tour of Graz`:



Harmony Attributes

ObjectID	0x00000011
Type	Document
DocumentType	collection
Author	kandrews
TimeCreated	95/10/15 17:27:52
Rights	R:g iicm
Title	en:Introducing Graz
Title	ge:Die Stadt Graz
CollectionType	Cluster
Subdocs	5
GOid	0x811b9908 0x00185a51

Figure 6.13 The Harmony Attribute Window.

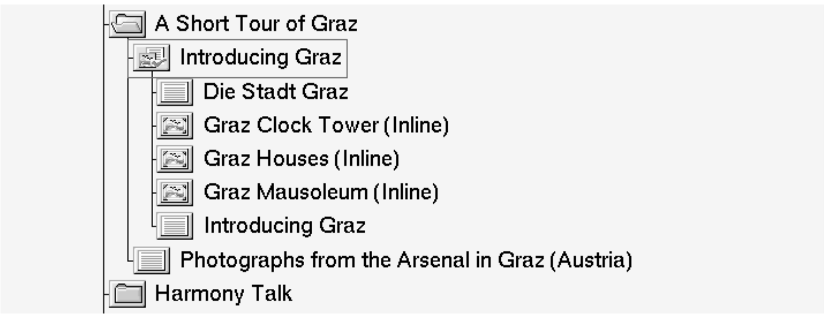


Harmony Attributes

ObjectID	0x00000011
Type	Document
DocumentType	collection
TimeCreated	95/10/15 17:27:52
TimeModified	95/10/22 17:15:56
CollectionType	Cluster
Subdocs	5
GOid	0x811b9908 0x00185a51
Author	kandrews
Rights	R:g iicm
Title	en:Introducing Graz
Title	ge:Die Stadt Graz
PresentationHints	CollectionHead

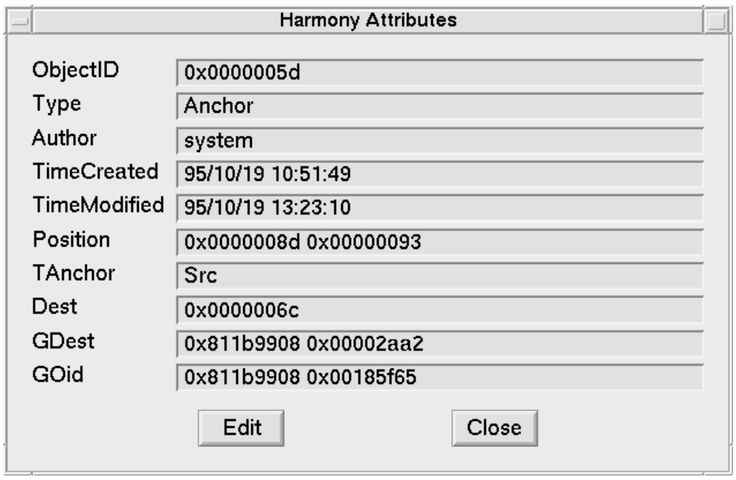
Add: Title

Figure 6.14 Editing attributes in the Attribute Window.



- A Short Tour of Graz
  - Introducing Graz
    - Die Stadt Graz
    - Graz Clock Tower (Inline)
    - Graz Houses (Inline)
    - Graz Mausoleum (Inline)
    - Introducing Graz
  - Photographs from the Arsenal in Graz (Austria)
- Harmony Talk

Clicking the `Edit` button puts the Attribute window into edit mode, as can be seen in Figure 6.14. The attributes are rearranged automatically, so that those which may be modified are at the bottom of the list. In the figure, the cluster is being specified as the collection head for the collection to



Harmony Attributes

ObjectID	0x0000005d
Type	Anchor
Author	system
TimeCreated	95/10/19 10:51:49
TimeModified	95/10/19 13:23:10
Position	0x0000008d 0x00000093
TAnchor	Src
Dest	0x0000006c
GDest	0x811b9908 0x00002aa2
GOid	0x811b9908 0x00185f65

Figure 6.15 Attributes of an anchor object.

which it belongs. This is achieved for any collection by setting the `PresentationHints` attribute `CollectionHead` for one member of the collection. From now on, when the collection `A Short Tour of Graz` is opened, the introductory cluster `Introducing Graz` will automatically be displayed.

A common structuring technique is to place small images being used as inline images inside the cluster where they are used, as was done in Section 6.2. However, when the cluster is accessed, the default presentation rules of a cluster would interpret each inline image as a separate image for display purposes, starting an Image Viewer for each one. This default behaviour can be overridden by assigning the inline images the `Hidden` characteristic in the `PresentationHints` field. Henceforth, the inline images will only be displayed inline in the text and not also as images in their own right. Other Hyper-G attributes, such as `Title` and `Rights`, are assigned and modified analogously.

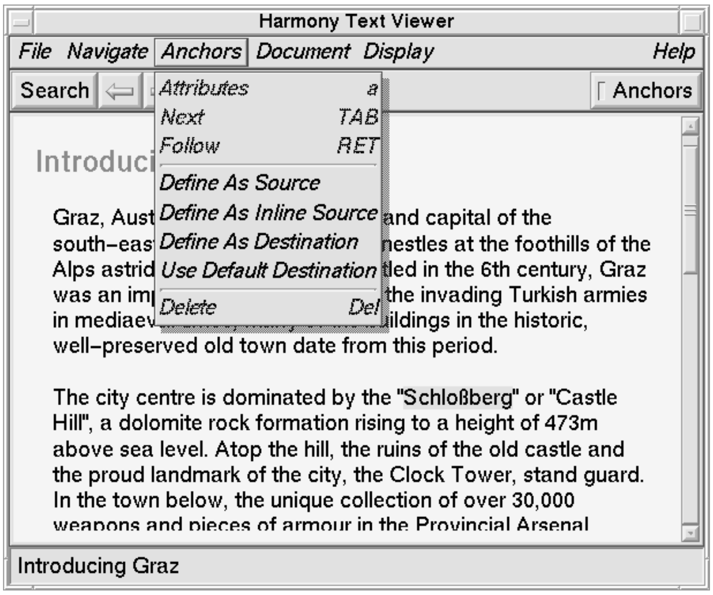
6.5 Interactive Link Editing

Harmony has extensive features for the interactive creation of hyperlinks. Links in Hyper-G emanate from a source anchor (sensitive region) in a document and may point to a collection, a cluster, a whole document, or a specific destination anchor in a document. The Harmony document viewers generally represent source anchors in pink and destination anchors in blue by default.

Anchors are Hyper-G objects in their own right and may possess attributes such as `Title`, `Author`, and `Rights`. The position of an anchor is specified in the `Position` attribute of the anchor object. The format of entries in this field depends upon the type of document containing the anchor. An example of the attributes for a source anchor in a text document is shown in Figure 6.15. The link creation process comprises four steps: defining a source anchor for the link, defining a destination anchor, assigning any link attributes, and then completing the link.

Creating Links in Text Documents

Anchor regions in text documents are defined in terms of character offsets. In the example in Figure 6.15, the source anchor extends from character 141 to character 147 in the text (8d and 93 in hex respectively). Link anchors are defined interactively in the Harmony Text Viewer by selecting a region of text (for example, by dragging the mouse with the left or right button depressed) and defining the region as a source, destination, or inline source anchor. This is illustrated in Figure 6.16, where

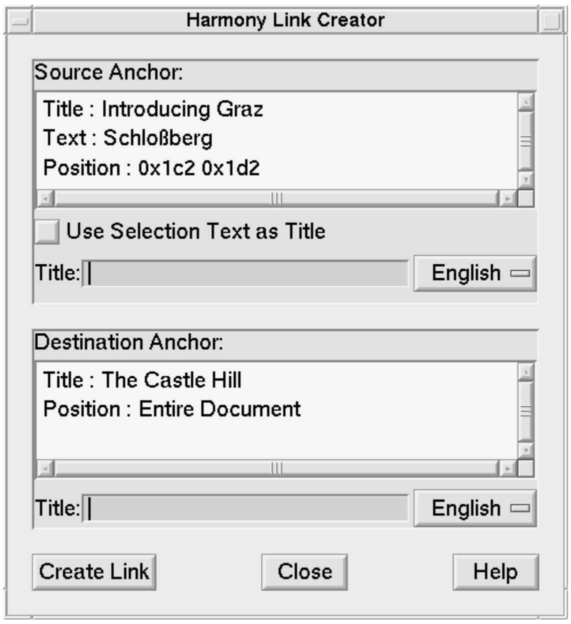


Graz, Aust … and capital of the
south–eas … nestles at the foothills of the
Alps astrid … tled in the 6th century, Graz
was an im … the invading Turkish armies
in mediaev … ildings in the historic,
well–preserved old town date from this period.

The city centre is dominated by the "Schloßberg" or "Castle Hill", a dolomite rock formation rising to a height of 473m above sea level. Atop the hill, the ruins of the old castle and the proud landmark of the city, the Clock Tower, stand guard. In the town below, the unique collection of over 30,000 weapons and pieces of armour in the Provincial Arsenal

Introducing Graz

Figure 6.16 Interactive definition of a source anchor in a text document.



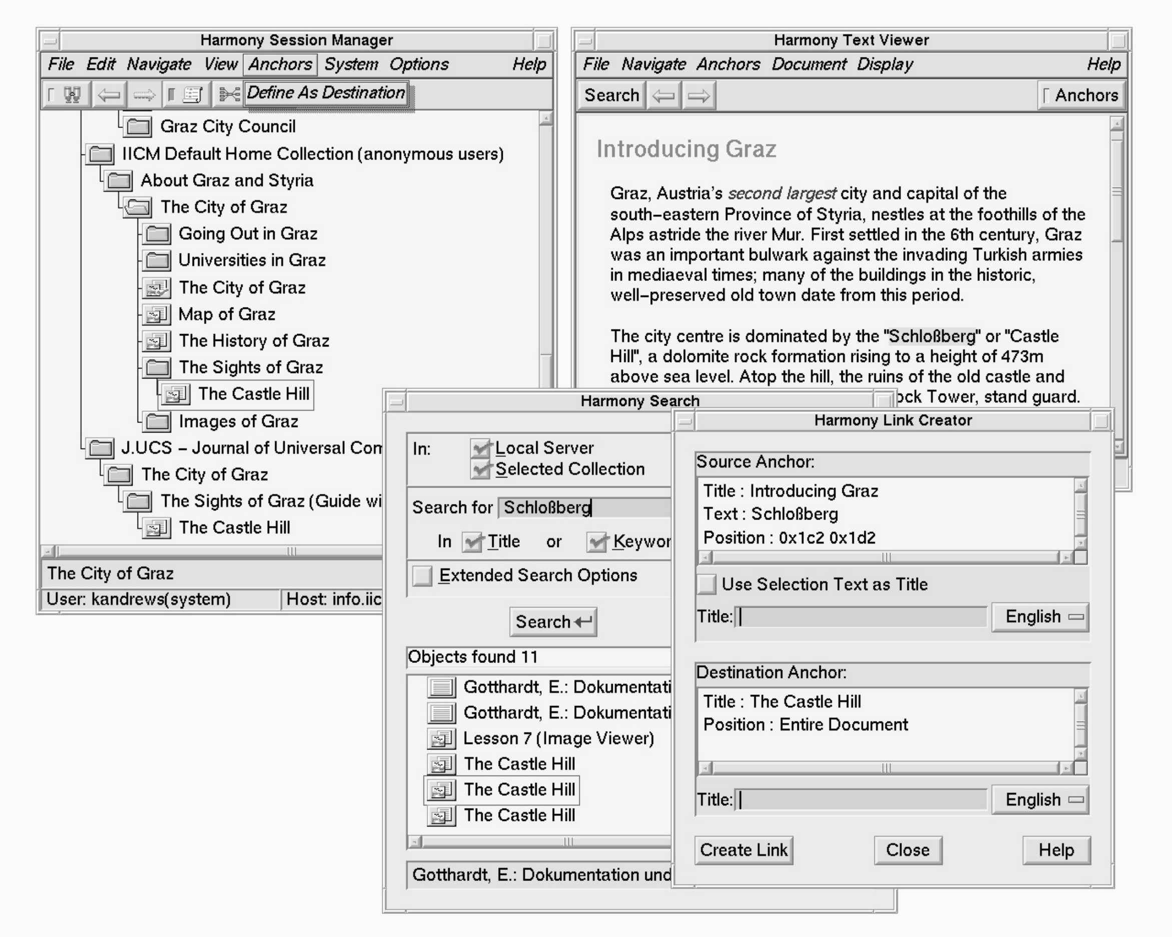
Title : Introducing Graz
Text : Schloßberg
Position : 0x1c2 0x1d2

Title : The Castle Hill
Position : Entire Document

Figure 6.17 The Harmony Link Creator.

a source anchor is being specified from the word “Schloßberg” in the text. Activating the function Define as Source brings up Harmony’s Link Creator, depicted in Figure 6.17. The source of the link has been entered in the corresponding Source Anchor box. The user has also defined the destination of the link to be a cluster called “The Castle Hill.” Clicking the button `Create Link` completes the link and it can be followed immediately thereafter.

An appropriate destination was found in this case by searching the Hyper-G server, as can be seen in Figure 6.18 which illustrates the entire course of this sample link creation process. The actual definition of a destination anchor proceeds in a manner similar to defining a source anchor, with the special case that, by default, a destination anchor in a text document covers the whole document.



Graz City Council
IICM Default Home Collection (anonymous users)
About Graz and Styria
The City of Graz
Going Out in Graz
Universities in Graz
The City of Graz
Map of Graz
The History of Graz
The Sights of Graz
The Castle Hill
Images of Graz
J.UCS – Journal of Universal Con
The City of Graz
The Sights of Graz (Guide wi
The Castle Hill

The City of Graz
User: kandrews(system) Host: info.iic

Introducing Graz

Graz, Austria's *second largest* city and capital of the south–eastern Province of Styria, nestles at the foothills of the Alps astride the river Mur. First settled in the 6th century, Graz was an important bulwark against the invading Turkish armies in mediaeval times; many of the buildings in the historic, well–preserved old town date from this period.

The city centre is dominated by the "Schloßberg" or "Castle Hill", a dolomite rock formation rising to a height of 473m above sea level. Atop the hill, the ruins of the old castle and ...ock Tower, stand guard.

In: ☑ Local Server ☑ Selected Collection
Search for: Schloßberg
In ☑ Title or ☑ Keywor
□ Extended Search Options

Objects found 11
Gotthardt, E.: Dokumentati
Gotthardt, E.: Dokumentati
Lesson 7 (Image Viewer)
The Castle Hill
The Castle Hill
The Castle Hill

Gotthardt, E.: Dokumentation und

Source Anchor:
Title : Introducing Graz
Text : Schloßberg
Position : 0x1c2 0x1d2
□ Use Selection Text as Title
Title: English

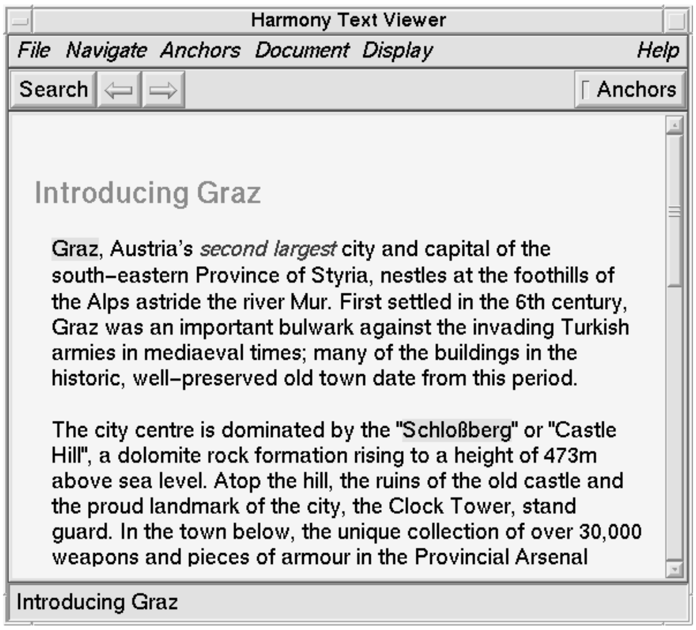
Destination Anchor:
Title : The Castle Hill
Position : Entire Document
Title: English

Figure 6.18 The entire link creation procedure in Harmony.

Placing Inline Images in Text Documents

Hyper-G implements inline images in text documents simply as a special link of type `inline`. The procedure for placing inline images is identical to that for creating links. A region of the text is marked, immediately preceding which the image will be placed. In Figure 6.19, the word `Graz` has been selected and the image will be inserted directly before it. From the Text Viewer menu, this selection is then defined to be an inline source anchor, bringing up the Harmony Link Creator shown in Figure 6.20.

The destination of the inline link (that is, the image to be inlined) is now specified, and the inline link is created. The resulting text with inlined image is shown in Figure 6.21. The inline image can be centered by editing the text document and adding the appropriate tag. Assuming that the user creates inline links to all three small images previously uploaded, the finished introductory document looks something like that shown in Figure 6.22 when accessed with a standard Web browser such as Netscape.



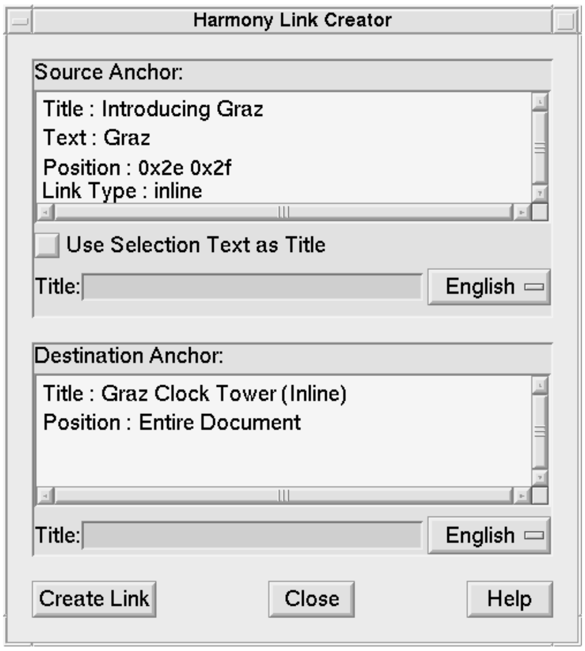
Introducing Graz

Graz, Austria's *second largest* city and capital of the south–eastern Province of Styria, nestles at the foothills of the Alps astride the river Mur. First settled in the 6th century, Graz was an important bulwark against the invading Turkish armies in mediaeval times; many of the buildings in the historic, well–preserved old town date from this period.

The city centre is dominated by the "Schloßberg" or "Castle Hill", a dolomite rock formation rising to a height of 473m above sea level. Atop the hill, the ruins of the old castle and the proud landmark of the city, the Clock Tower, stand guard. In the town below, the unique collection of over 30,000 weapons and pieces of armour in the Provincial Arsenal

Introducing Graz

Figure 6.19 Placing an inline image. The inline image is inserted directly preceding the selected region, here the word "Graz".



Source Anchor:

Title : Introducing Graz
Text : Graz
Position : 0x2e 0x2f
Link Type : inline

□ Use Selection Text as Title

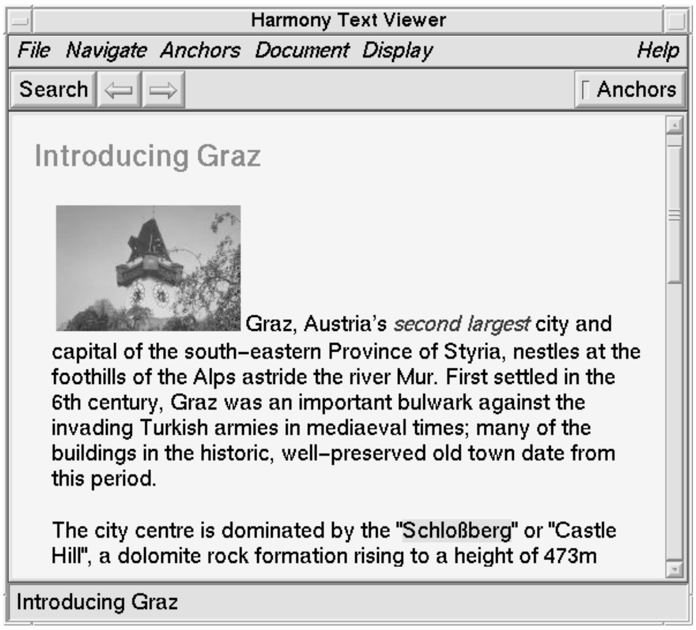
Title: English

Destination Anchor:

Title : Graz Clock Tower (Inline)
Position : Entire Document

Title: English

Figure 6.20 The Link Creator for an inline image.

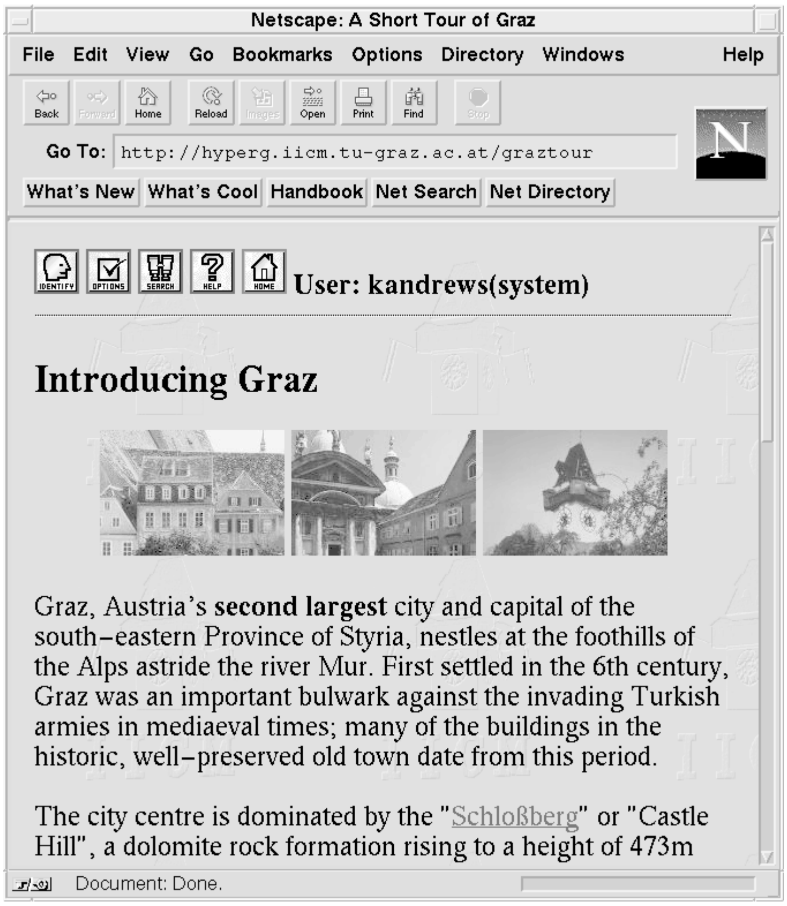


Introducing Graz

Graz, Austria's *second largest* city and capital of the south–eastern Province of Styria, nestles at the foothills of the Alps astride the river Mur. First settled in the 6th century, Graz was an important bulwark against the invading Turkish armies in mediaeval times; many of the buildings in the historic, well–preserved old town date from this period.

The city centre is dominated by the "Schloßberg" or "Castle Hill", a dolomite rock formation rising to a height of 473m

Figure 6.21 The inserted inline image.


Netscape: A Short Tour of Graz
File Edit View Go Bookmarks Options Directory Windows Help
Back Forward Home Reload Images Open Print Find Stop
Go To: http://hyperg.iicm.tu-graz.ac.at/graztour
What's New What's Cool Handbook Net Search Net Directory
IDENTIFY OPTIONS SEARCH HELP HOME
Document: Done.

User: kandrews(system)

Introducing Graz

Graz, Austria's **second largest** city and capital of the south–eastern Province of Styria, nestles at the foothills of the Alps astride the river Mur. First settled in the 6th century, Graz was an important bulwark against the invading Turkish armies in mediaeval times; many of the buildings in the historic, well–preserved old town date from this period.

The city centre is dominated by the "Schloßberg" or "Castle Hill", a dolomite rock formation rising to a height of 473m

Figure 6.22 The finished introductory page with three inline images.

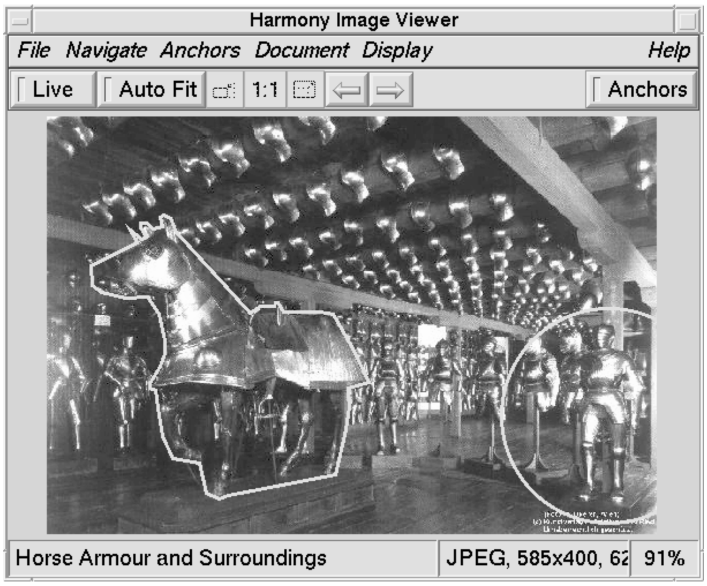

Harmony Image Viewer
File Navigate Anchors Document Display Help
Live
Auto Fit
1:1
Anchors
Horse Armour and Surroundings
JPEG, 585x400, 62
91%

Figure 6.23 Source anchors in an image document.

Creating Links in Image Documents

Anchor regions in image documents are defined in terms of a virtual coordinate space, where (0,0) is the lower left corner of the image and (1,1) the upper right. The image shown in Figure 6.23 has two source anchors (links) defined, a circle and a polygon, whose positions are defined in the `Position` attribute as follows:

```
Circle 0.921321 0.283395 0.177331
Polygon 0.0734733 0.642748, 0.0734733 0.642748, ...
```

Link anchors are defined interactively by selecting an anchor shape (in the Anchors menu) and using the right mouse button to define the region. For rectangular, circular and elliptical anchor shapes, it is sufficient to drag and release the right mouse button to define the anchor region. For polygonal regions, each right mouse click defines a vertex of the polygon and a middle-click completes the polygon. Any freshly marked anchor region can be repositioned by Control-right-dragging the shape. Once the anchor region is satisfactory, the region is defined as a source or destination anchor, and the Link Creator dialogue appears. The procedure now continues in the same way as explained for text anchors.

A destination anchor in an image will cause the image to be panned so that the region of the image containing the anchor is in view. This is illustrated in Figure 6.24, where a link has been followed to a particular region of a large map of Graz.

Creating Links in Film Documents

Anchors in film clips are created by specifying the position and size of an anchor region for a set of key frames. Linear or spline interpolation is used to determine the position of an anchor in between key frames. Anchors may be rectangular, circular, or elliptical in shape. The sequence of screen shots in Figure 6.25 shows a circular link anchor being created to track the helmet of a piece of cavalry armour. The anchor's size and position are first specified for frame 1 of the film, then for frames 15, 30, and 45. The marked anchor region can be played back to check its effect and how it is interpolated between key frames. When a satisfactory result has been achieved, the marked region is defined as either a source or destination anchor using the Anchor menu. In this particular example, an anchor

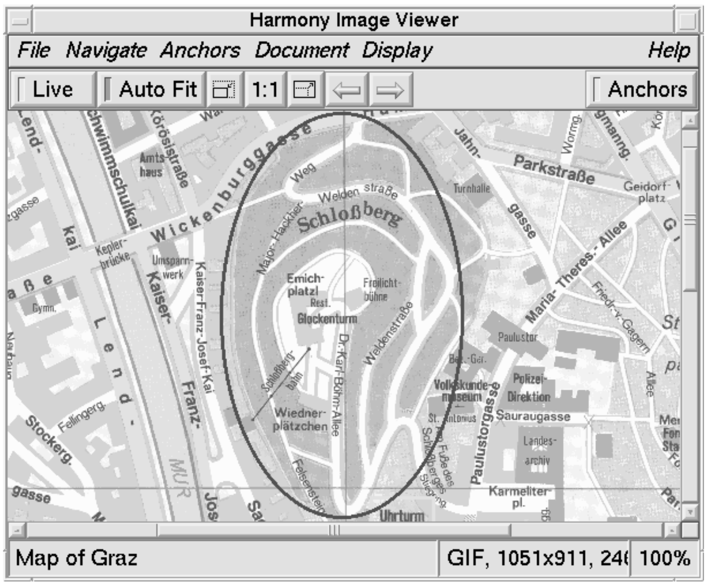


Figure 6.24 Elliptical destination anchor in an image document.

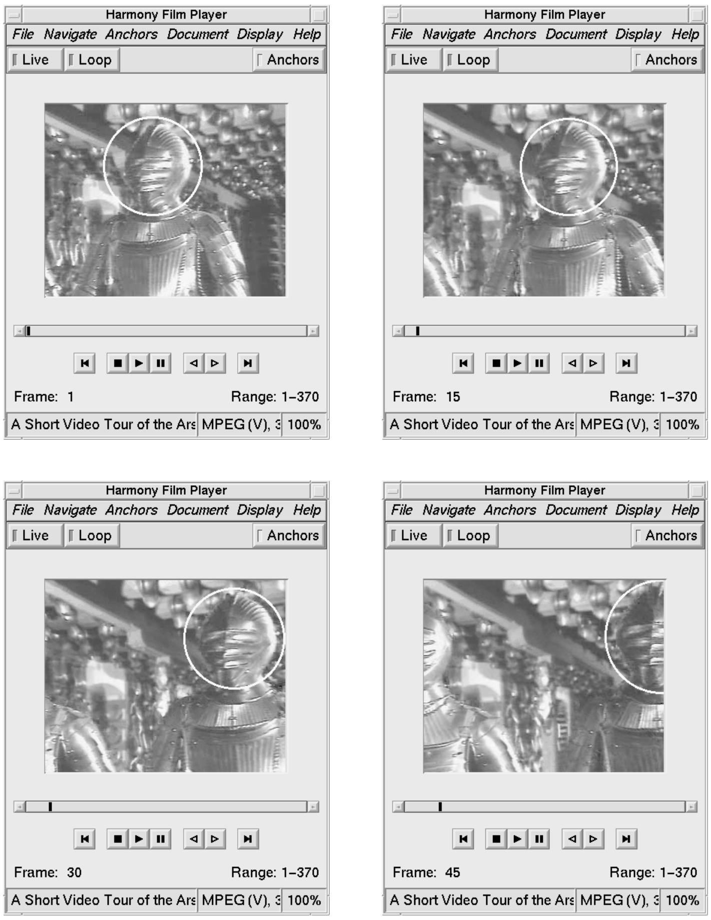
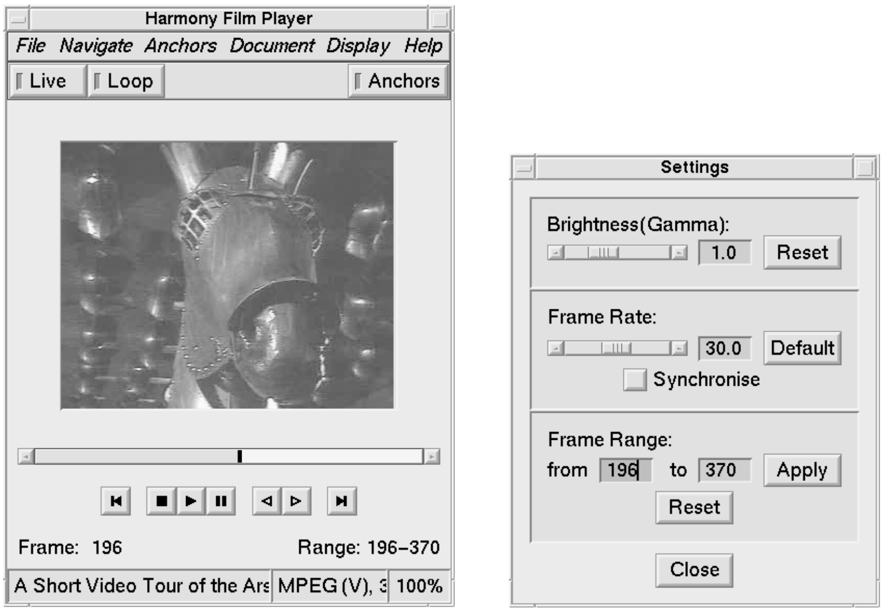


Figure 6.25 Creating an anchor in a film clip.



Frame: 196 Range: 196–370

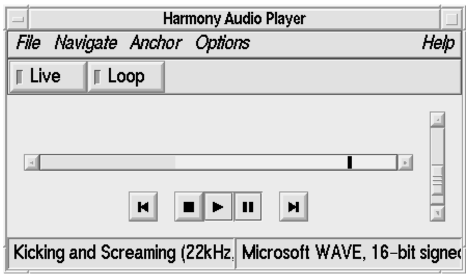
A Short Video Tour of the Ars | MPEG (V), 3 | 100%

Brightness(Gamma): 1.0

Frame Rate: 30.0

Frame Range: from 196 to 370

Figure 6.26 Frame range as a destination anchor.



Kicking and Screaming (22kHz | Microsoft WAVE, 16-bit signe

Figure 6.27 Destination anchor in an audio document.

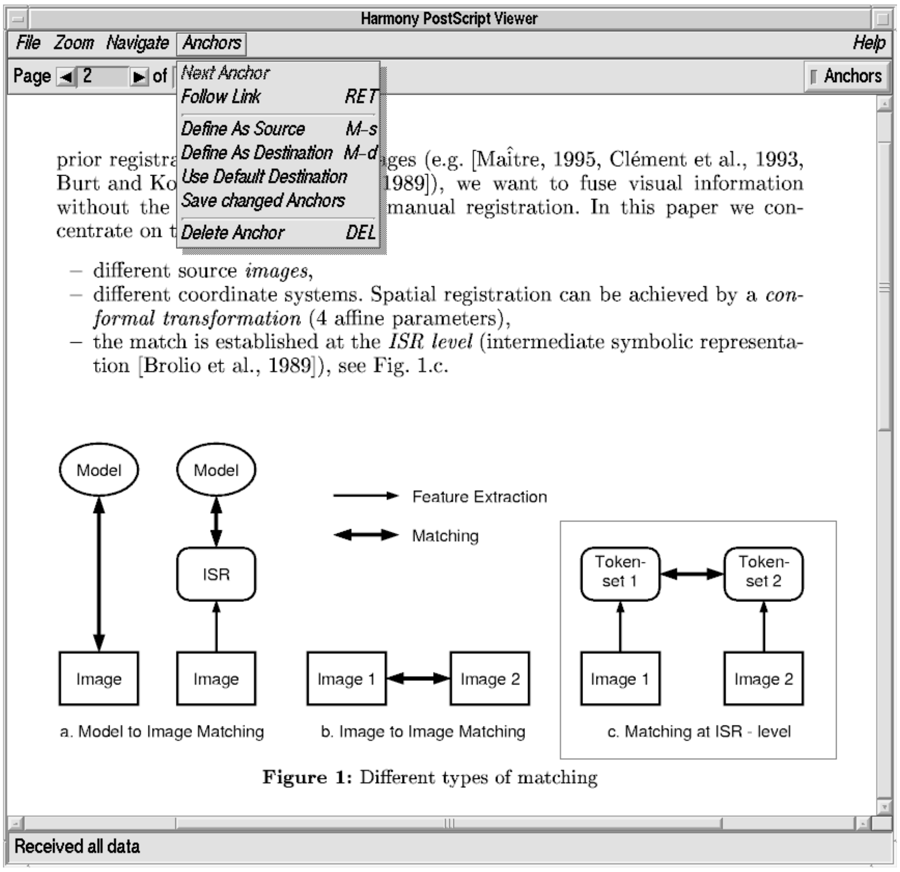
was created with the following `Position` attribute:

```
Circle Linear 4:    1  0.45     0.675781  0.204084,
                   15  0.60625  0.667969  0.201556,
                   30  0.7875   0.695312  0.208885,
                   45  0.99375  0.703125  0.234791
```

A simplified form of anchor, namely a range of frames, can be specified for a destination anchor. Figure 6.26 shows frames 196 to 370 of the Arsenal film clip being defined as the destination of a link. The frame range can be marked approximately by right-dragging across the play bar, or exactly by entering the precise frames in the Settings panel.

Creating Links in Audio Documents

An anchor region in an audio document is defined by a starting and ending position (time) in the audio stream. Currently, only destination anchors are supported, that is, links can be made to particular



prior registra ges (e.g. [Maître, 1995, Clément et al., 1993, Burt and Ko 1989]), we want to fuse visual information without the manual registration. In this paper we concentrate on t

- different source *images*,
- different coordinate systems. Spatial registration can be achieved by a *conformal transformation* (4 affine parameters),
- the match is established at the *ISR level* (intermediate symbolic representation [Brolio et al., 1989]), see Fig. 1.c.

Figure 1: Different types of matching

Received all data

Figure 6.28 Creating a destination anchor in a PostScript document.

regions of an audio document. Figure 6.27 shows the Harmony Audio Player after a link has been followed to the latter part of the document.

The Audio Player must be set up to use NAS (the Network Audio System) [Fulton and Renda, 1994] in order to be able to process link anchors. When the Audio Player is configured to use a native audio play command (which is currently the default due to the lack of a widely installed audio server such as NAS), the audio stream is simply passed on to this external program and not examined further.

Creating Links in PostScript Documents

An anchor region in a PostScript document is a rectangular region defined in terms of PostScript page number and coordinates. A new region is marked by right-dragging the mouse. Figure 6.28 shows a destination anchor being defined in the Harmony PostScript Viewer to cover Figure 1.c of the article. Source anchors are typically highlighted by coloured frames, destination anchors by "lowlighting" the rest of the current page. Anchors in a PostScript document may be repositioned and resized by Control-right-dragging the mouse. Dragging a corner or an edge of an anchor resizes it, dragging the interior repositions it. The repositioned link anchors can then be saved back into the server through the Anchors menu.

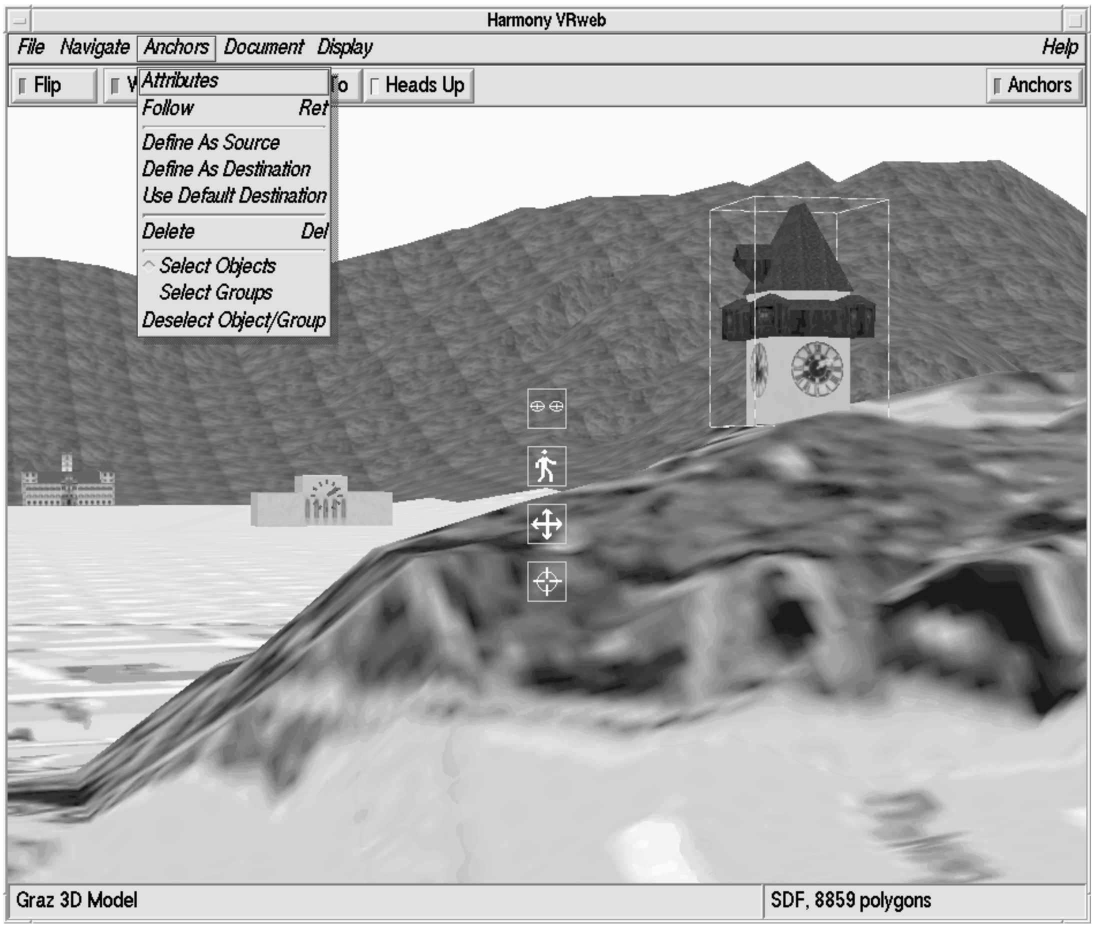


Figure 6.29 Creating a source anchor in a 3d model.

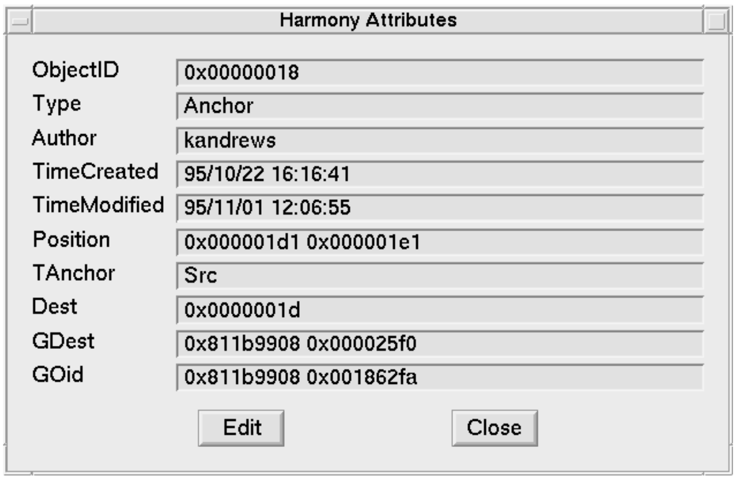
Creating Links in 3D Documents

In a 3d model, source and destination anchors are conceptually distinct. A source anchor can be attached to an arbitrary part of a 3d model. In the Graz model shown in Figure 6.29, a source anchor is being created from the Clock Tower object in the model to more detailed information about the tower. The currently marked object is highlighted by drawing its bounding box.

A destination anchor, on the other hand, is not attached to part of a model, but instead indicates a specific camera position and orientation, that is, a specific view of the model. Destination anchors are defined by navigating to the desired viewing position and then defining the current position as a destination anchor.

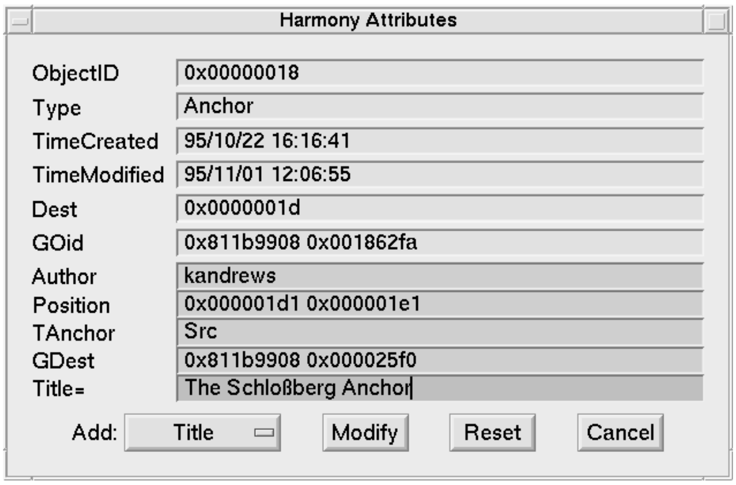
Link Attribute Modification

Since link anchors in Hyper-G are objects in their own right, they may possess attributes such as `Title`, `Author`, and `Rights`. Selecting an anchor in a Harmony document viewer (by left-clicking) and choosing to display its attributes via the Anchors menu brings up Harmony's standard Attributes window (see Section 6.4). The attributes shown in Figure 6.30 are those of the source anchor from the word `Schloßberg` which was created in Section 6.5. The default access rights allow the author of an anchor write access, and all other users read access. In our example, the author of the anchor is `kandrews`, and hence the anchor object's attributes may be edited, as shown in Figure 6.31, where a title has been added to the anchor.



ObjectID	0x00000018
Type	Anchor
Author	kandrews
TimeCreated	95/10/22 16:16:41
TimeModified	95/11/01 12:06:55
Position	0x000001d1 0x000001e1
TAnchor	Src
Dest	0x0000001d
GDest	0x811b9908 0x000025f0
GOid	0x811b9908 0x001862fa

Figure 6.30 Attributes of the `Schloßberg` anchor object.



ObjectID	0x00000018
Type	Anchor
TimeCreated	95/10/22 16:16:41
TimeModified	95/11/01 12:06:55
Dest	0x0000001d
GOid	0x811b9908 0x001862fa
Author	kandrews
Position	0x000001d1 0x000001e1
TAnchor	Src
GDest	0x811b9908 0x000025f0
Title=	The Schloßberg Anchor

Add: Title

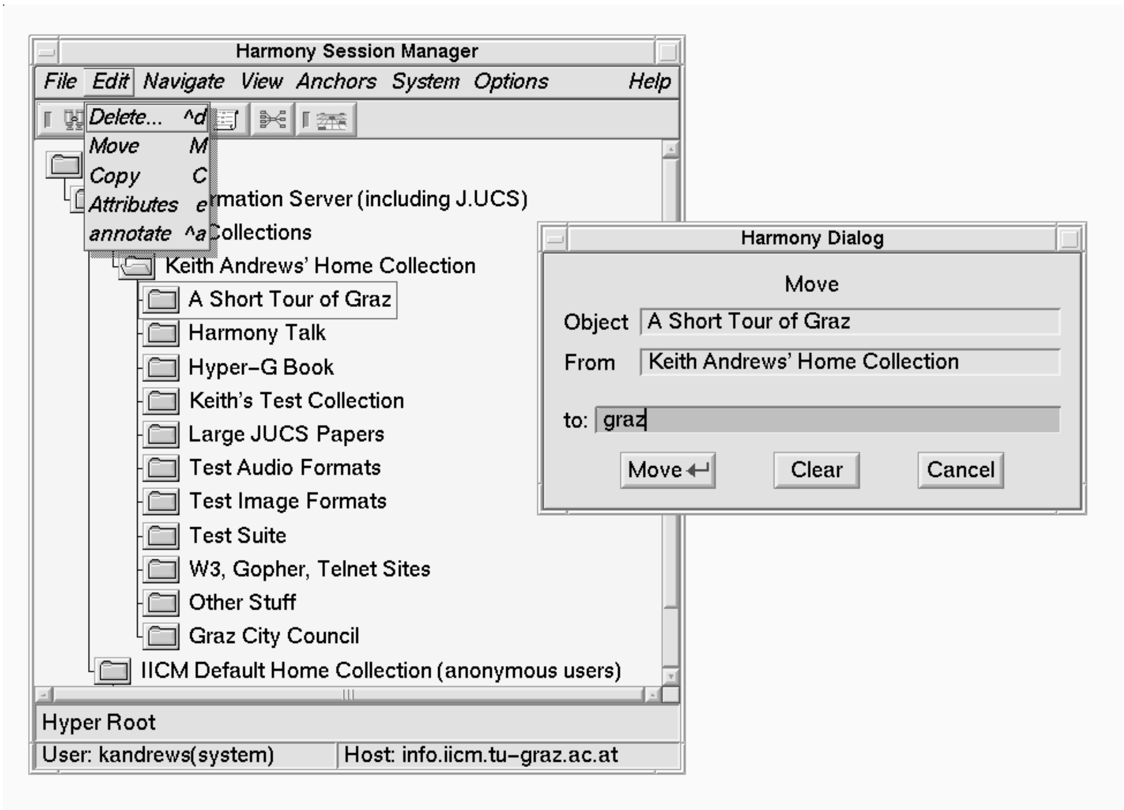
Figure 6.31 Editing the anchor's attributes.

Link Deletion

A link is deleted interactively by selecting its source anchor in the appropriate document viewer (left-clicking) and choosing the `Delete` item from the Anchors menu. In the Harmony Text Viewer, multiple anchors can be selected and then deleted in one operation by shift-left-clicking the anchors. Before actually performing the deletion, Harmony presents a confirmation dialogue box.

6.6 Structure Editing

A prime characteristic of Hyper-G is its ability to structure information into collections and sub-collections in a manner orthogonal to referential hyperlinks. This helps alleviate disorientation by presenting a firm, perceivable structure to users. Harmony's facilities for creating new collections were described in Section 6.2. Naturally, Harmony also has facilities for moving existing documents and collections between collections and for making documents and collections members of multiple parent collections. These functions are currently provided via dialogues, but will soon be available



Object: A Short Tour of Graz

From: Keith Andrews' Home Collection

to: graz

Figure 6.32 Moving a document or collection.

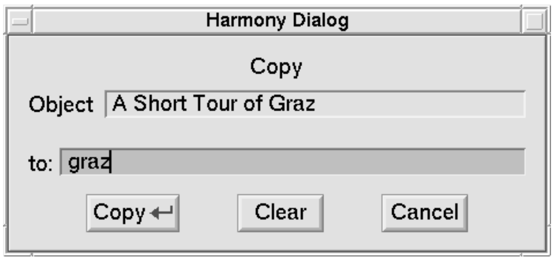
through a drag-and-drop operation directly in the collection browser.

Moving Objects

Assuming that the `A Short Tour of Graz` collection has been completed and is now in a presentable form, the author may want to move it from his home collection into a publicly accessible collection. This is achieved with the `Move` command from the Edit menu, as shown in Figure 6.32. After the move operation, the collection has been removed from the home collection and made a member the collection named `graz`. This would also be the right time to adjust any access rights on the collection and its members so that anonymous users can read its contents.

Copying Objects

The alternative to moving an object is to “copy” it into an additional collection. The object then belongs both to the original collection and to the additional collection, and appears in both locations in the collection hierarchy. The term “copy” is somewhat misleading, because the object is not physically copied, it simply becomes a member of both collections (the same operation in HGTV is called “link”). Perhaps “join” or “insert” would be better names. The dialogue box for the `Copy` operation is shown in Figure 6.33.

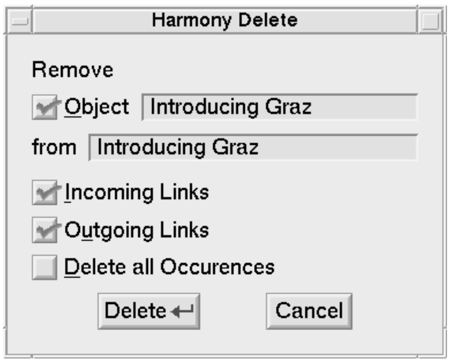


Copy

Object: A Short Tour of Graz

to: graz

Figure 6.33 “Copying” a document or collection.



Remove

☑ Object: Introducing Graz

from: Introducing Graz

☑ Incoming Links

☑ Outgoing Links

□ Delete all Occurences

Figure 6.34 Deleting a document or collection.

6.7 Deleting Objects

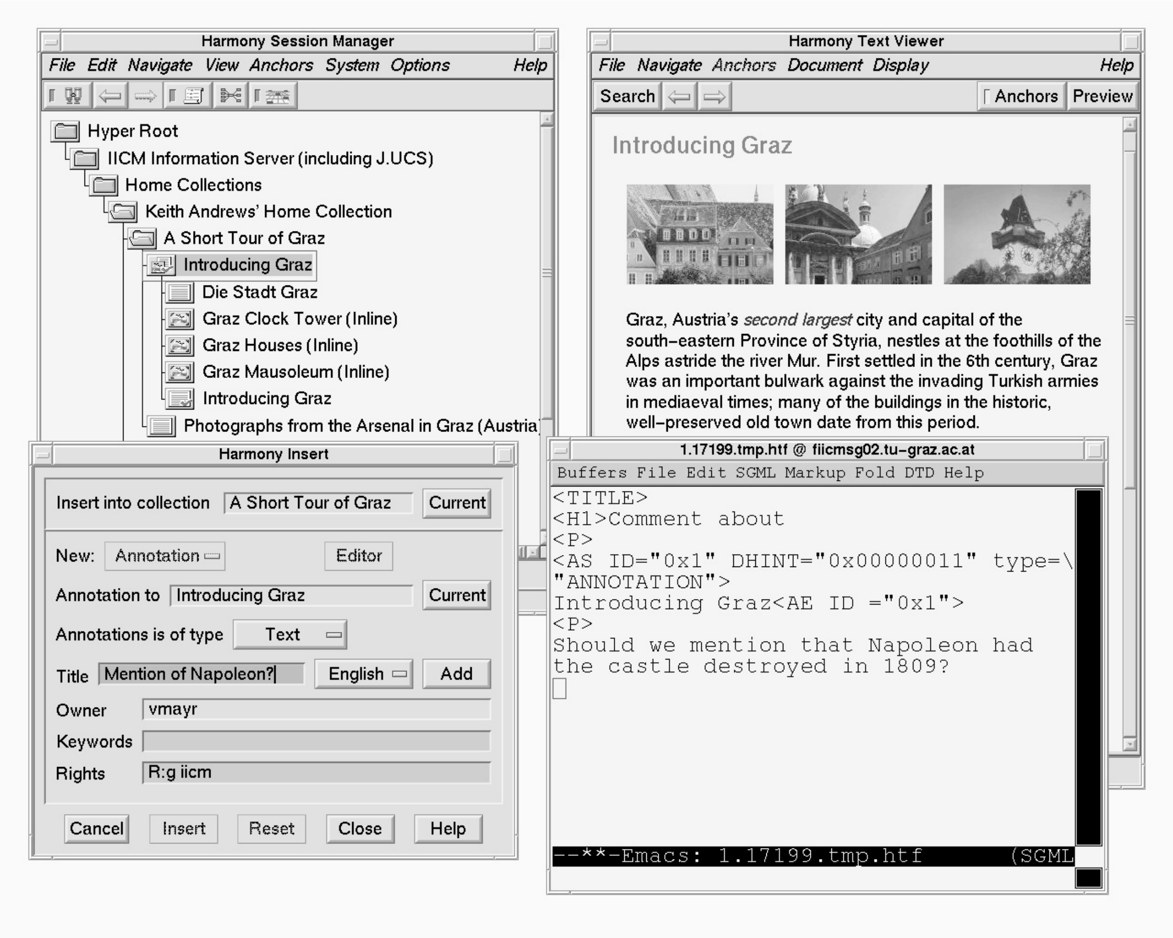
An object can be deleted interactively by selecting it and then activating the `Delete` function from the Edit menu. The Harmony Delete window shown in Figure 6.34 is presented. If the object belongs only to one collection, then it is physically deleted, and by default so are any incoming and outgoing links associated with it. If the object is a member of multiple parent collections, then deleting it simply removes it from the specified collection and has no further consequences. Specifying deletion of all occurrences forces the object’s removal from all parent collections and its physical deletion. Currently, collections can be physically deleted only if they are empty, that is, their members must be deleted first.

6.8 Annotations

Conceptually, an annotation in Hyper-G is really just a special kind of link. Annotations are not restricted to being pieces of text, but can be documents of any type. They are associated with the collection, cluster, or document upon which they comment by a link of type `Annotation`.

Annotations are created using a variant of the Harmony Insert dialogue which was discussed in Section 6.2. To annotate an object, the object is selected and the `Annotate` function in the Harmony Session Manager’s Edit menu is activated. Harmony presents the user with an Insert dialogue pre-configured for inserting an annotation. The annotation may either be uploaded from disk or created afresh with an editor. In the example shown in Figure 6.35, a colleague (Vanessa Mayrhofer, `vmayr`) is editing a new annotation about the introductory cluster `Introducing Graz`.

In the case of a textual annotation, a link back to the document being annotated is pre-inserted into the template text document, as can be seen in Figure 6.36. When the comment or remark is finished, the editor is exited, and a dialogue box asks for confirmation that the annotation should be


Harmony Session Manager
File Edit Navigate View Anchors System Options Help
Harmony Text Viewer
File Navigate Anchors Document Display Help
Search
Anchors
Preview
Harmony Insert
Buffers File Edit SGML Markup Fold DTD Help

Hyper Root
IICM Information Server (including J.UCS)
Home Collections
Keith Andrews' Home Collection
A Short Tour of Graz
Introducing Graz
Die Stadt Graz
Graz Clock Tower (Inline)
Graz Houses (Inline)
Graz Mausoleum (Inline)
Introducing Graz
Photographs from the Arsenal in Graz (Austria)

Introducing Graz

Graz, Austria's *second largest* city and capital of the south–eastern Province of Styria, nestles at the foothills of the Alps astride the river Mur. First settled in the 6th century, Graz was an important bulwark against the invading Turkish armies in mediaeval times; many of the buildings in the historic, well–preserved old town date from this period.

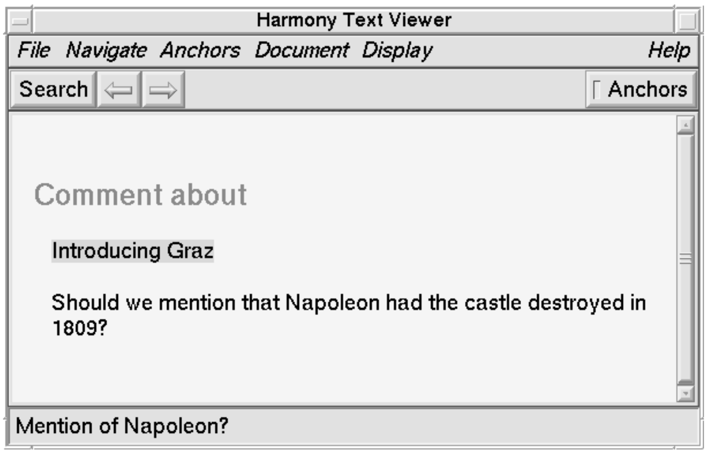
Insert into collection: A Short Tour of Graz [Current]
New: Annotation [Editor]
Annotation to: Introducing Graz [Current]
Annotations is of type: Text
Title: Mention of Napoleon? English [Add]
Owner: vmayr
Keywords:
Rights: R:g iicm
[Cancel] [Insert] [Reset] [Close] [Help]

1.17199.tmp.htf @ fiicmsg02.tu–graz.ac.at

```
<TITLE>
<H1>Comment about
<P>
<AS ID="0x1" DHINT="0x00000011" type=\
"ANNOTATION">
Introducing Graz<AE ID ="0x1">
<P>
Should we mention that Napoleon had
the castle destroyed in 1809?

--**-Emacs: 1.17199.tmp.htf       (SGML
```

Figure 6.35 Creating an annotation to the `Introducing Graz` cluster.



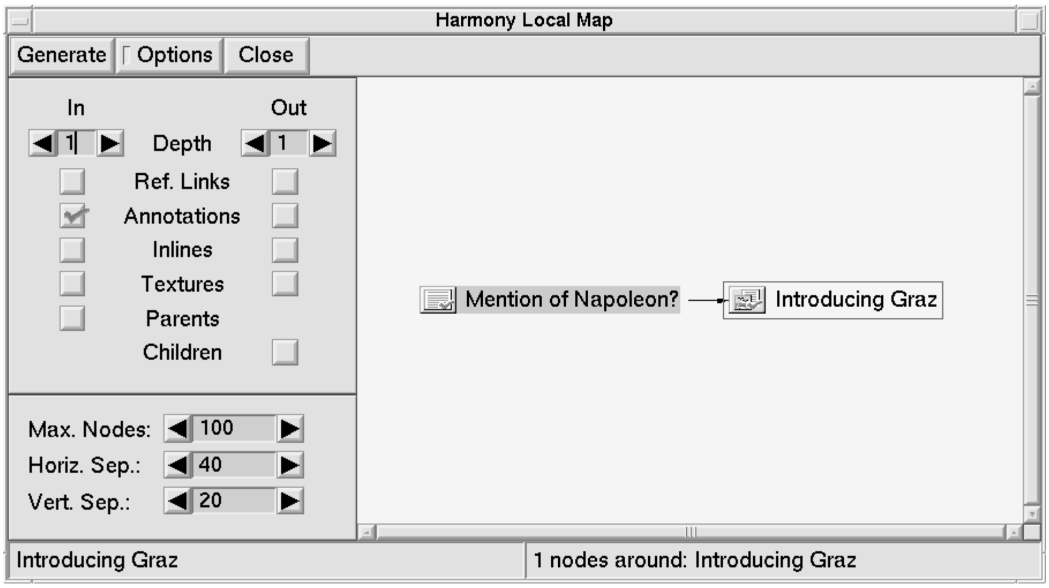
Comment about

Introducing Graz

Should we mention that Napoleon had the castle destroyed in 1809?

Mention of Napoleon?

Figure 6.36 The finished annotation.



In		Out
1	Depth	1
□	Ref. Links	□
☑	Annotations	□
□	Inlines	□
□	Textures	□
□	Parents	
	Children	□

Max. Nodes: 100
Horiz. Sep.: 40
Vert. Sep.: 20

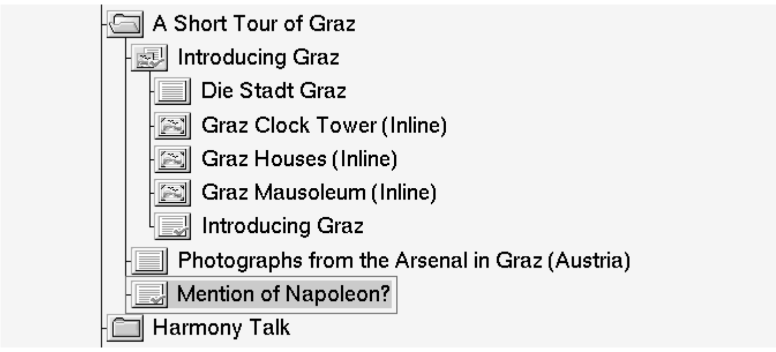
Mention of Napoleon? → Introducing Graz

Introducing Graz

1 nodes around: Introducing Graz

Figure 6.37 The annotation relationship in the Local Map.

inserted into the server. By default, annotations are inserted into the collection to which the annotated object itself belongs, but this can be changed in the Insert dialogue. In the example, a new text document `Mention of Napoleon?` (the annotation) has been inserted into the collection `A Short Tour of Graz` to which the cluster `Introducing Graz` belongs:



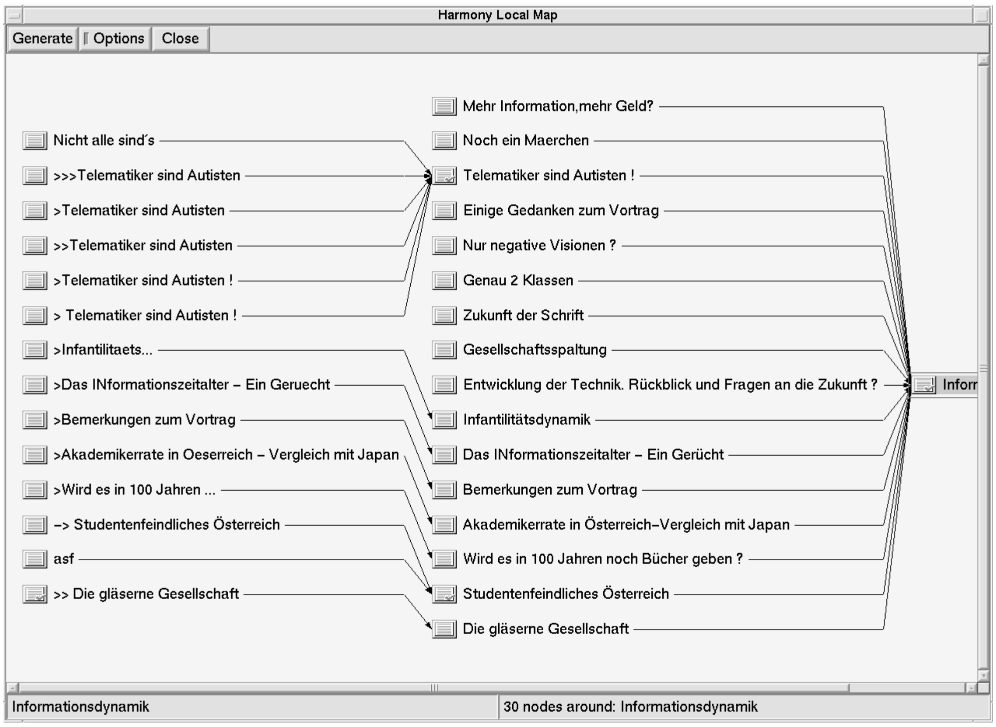
- A Short Tour of Graz
  - Introducing Graz
    - Die Stadt Graz
    - Graz Clock Tower (Inline)
    - Graz Houses (Inline)
    - Graz Mausoleum (Inline)
    - Introducing Graz
  - Photographs from the Arsenal in Graz (Austria)
  - Mention of Napoleon?
- Harmony Talk

Since annotations are a kind of link type, annotation relationships can be comfortably viewed in Harmony's Local Map. Figure 6.37 shows the Local Map for the annotation in our example. By annotating annotations, a structured discussion can evolve. A good example of this is Hermann Maurer's "Social Aspects of Telematics" course at Graz University of Technology, in which students and professors discuss a series of short essays online using pseudonyms. Figure 6.38 shows two levels of the discussion about an essay concerning the dynamics of information.

6.9 Managing Users and User Groups

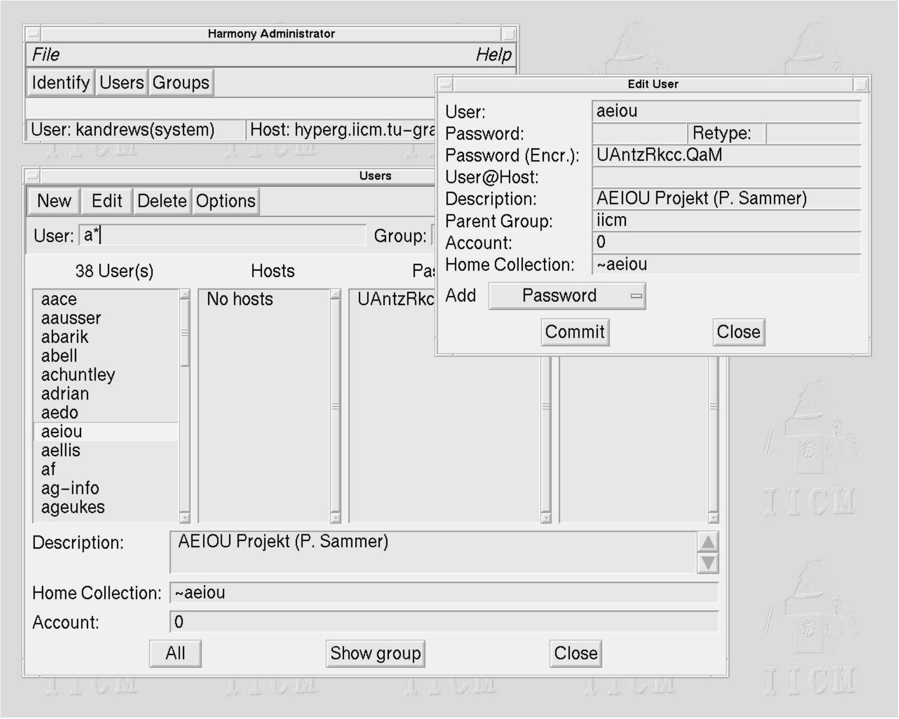
HarAdmin, shown in Figure 6.39 is a graphical administration tool for the interactive creation and management of user accounts and user groups[2] [Windisch, 1995]. Hyper-G users and groups are

[2] HarAdmin was designed by myself and Claudia Windisch and implemented by Claudia under my supervision for her Master's Thesis.


Harmony Local Map
Generate
Options
Close
Mehr Information,mehr Geld?
Nicht alle sind´s
Noch ein Maerchen
>>>Telematiker sind Autisten
Telematiker sind Autisten !
>Telematiker sind Autisten
Einige Gedanken zum Vortrag
>>Telematiker sind Autisten
Nur negative Visionen ?
>Telematiker sind Autisten !
Genau 2 Klassen
> Telematiker sind Autisten !
Zukunft der Schrift
>Infantilitaets...
Gesellschaftsspaltung
>Das INformationszeitalter – Ein Geruecht
Entwicklung der Technik. Rückblick und Fragen an die Zukunft ?
Inforr
>Bemerkungen zum Vortrag
Infantilitätsdynamik
>Akademikerrate in Oesereich – Vergleich mit Japan
Das INformationszeitalter – Ein Gerücht
>Wird es in 100 Jahren ...
Bemerkungen zum Vortrag
–> Studentenfeindliches Österreich
Akademikerrate in Österreich–Vergleich mit Japan
asf
Wird es in 100 Jahren noch Bücher geben ?
>> Die gläserne Gesellschaft
Studentenfeindliches Österreich
Die gläserne Gesellschaft

Informationsdynamik | 30 nodes around: Informationsdynamik

Figure 6.38 Structured discussion using annotations.



User: kandrews(system) | Host: hyperg.iicm.tu–gra

User: a*
Group:

38 User(s) | Hosts | Pa

aace
aausser
abarik
abell
achuntley
adrian
aedo
aeiou
aellis
af
ag–info
ageukes

No hosts

UAntzRkc

Description: AEIOU Projekt (P. Sammer)

Home Collection: ~aeiou

Account: 0

User: aeiou
Password: Retype:
Password (Encr.): UAntzRkcc.QaM
User@Host:
Description: AEIOU Projekt (P. Sammer)
Parent Group: iicm
Account: 0
Home Collection: ~aeiou

Figure 6.39 HarAdmin, the graphical administration tool for user accounts and user groups.

completely separate from UNIX users and groups. The attributes of a User object include a description, (optional) home collection, various passwords, and so forth. The Account attribute embodies a virtual wallet from which amounts can be deducted.

A hierarchical scheme of groups and subgroups can be defined, with groups potentially belonging to multiple parent groups. Hyper-G access rights can be assigned per document or collection on a user or group basis, for example using Harmony's attribute editor.

6.10 Further Reading

Numerous HTML authoring tools are now available for all major platforms, including asWedit [Stochniol, 1996] and Symposia [INRIA/GRIF, 1996] for UNIX. The most popular image manipulation tools for UNIX are XV [Bradley, 1996] and the PBMPLUS utility suite [Poskanzer, 1996]. The best resource for MPEG is Luigi Filippini's online Web site [Filippini, 1996]. The mpegUtil utility [Lougher, 1996] allows MPEG clips to be extracted and concatenated. A good printed source of information on image and video formats is *Graphics File Formats* by Murray and van Ryper [1996]. The Network Audio System is described in [Fulton and Renda, 1994]. The Audio page of the WWW Virtual Library (available at `http://cnidr.org/talks/whyzdist.html`) has numerous links to audio software and samples. The VRML Repository [Eubanks *et al.*, 1996] maintains a list of modeling tools and conversion utilities for VRML.

7 Beholding Cyberspace with VRweb

In this chapter, I present the Harmony 3d Scene Viewer and its successor VRweb. Parts of this chapter were published in *VRweb: A Multi-System VRML Viewer*, Proceedings of the First Annual Symposium on the Virtual Reality Modeling Language (VRML 95), San Diego, California, Dec. 1995 [Pichler *et al.*, 1995] and *Hooking Up 3-Space: Three-Dimensional Models as Fully-Fledged Hypermedia Documents*, in Multimedia, Hypermedia, and Virtual Reality, Springer LNCS 1077, 1996 [Andrews and Pichler, 1996]. The Harmony 3D Scene Viewer was first presented in December 1992 at the European Conference on Hypertext in Milan [Andrews, 1992].

7.1 3D Hypermedia and the Internet

From the start of my work with Hyper-G and Harmony in 1991, and drawing on the work done in Intermedia [Haan *et al.*, 1992], it was clear to me that the storage of hyperlinks in a separate link database would enable *real* hypermedia, i.e. linking from and to all kinds of media. My previous background in computer graphics also made it natural to support 3d models as one of the standard hypermedia document types in Hyper-G and in the Harmony client.

To my knowledge, my work on 3d hypermedia in Hyper-G combining 3d models and hyperlinking across the Internet, was the first such work world-wide. I presented our initial work in this area, the Harmony 3d Scene Viewer [1], as a poster at the European Conference on Hypertext in Milan in December 1992 [Andrews, 1992] (see Figure 7.1); the poster was subsequently also put on display at Xerox PARC. The Harmony 3d Scene Viewer would later become the VRweb browser for the Virtual Reality Modeling Language (VRML) [Pesce, 1995].

The scientific contribution of the Harmony 3d Scene Viewer and later of VRweb is manifold. We demonstrated that 3d models can be integrated into a real hypermedia system as a standard document type, with full hyperlinking capability. The Harmony 3d Scene Viewer embodied in 1993 (!) much of the functionality found in today's VRML browsers: browsing 3d models across the Internet, following hyperlinks to models and other documents, as well as interactive link editing facilities still not widely available in today's VRML browsers.

Today, VRweb is available from some 20 ftp mirror sites world-wide. It has been distributed on CD-ROM in 10 books and magazines, and was demonstrated by Sun Microsystems in the context of a 3d decision support system at their SIGGRAPH'96 booth. The VRweb home page: `http://www.iicm.edu/vrweb` is one of the most requested pages on the IICM server and is accessed around 5000 times a month. Jim Collins said of VRweb in PC Magazine in October 1995:

> *"We found that scenes created with VRweb were a lot less jaggy and appeared more realistic than those created in WebSpace . . . "*

and Howard Wen said in Wired magazine in February 1996:

> *"VRweb is just such a facilitating program. It's a no-fuss VRML viewer that's user-friendly and simple to use. Additionally, because its programming is relatively stable, VRweb serves as a good introduction to VRML browsing."*

[1] The Harmony 3d Scene Viewer was designed by myself and Michael Pichler and implemented by Michael for his Master's Thesis, under my supervision.

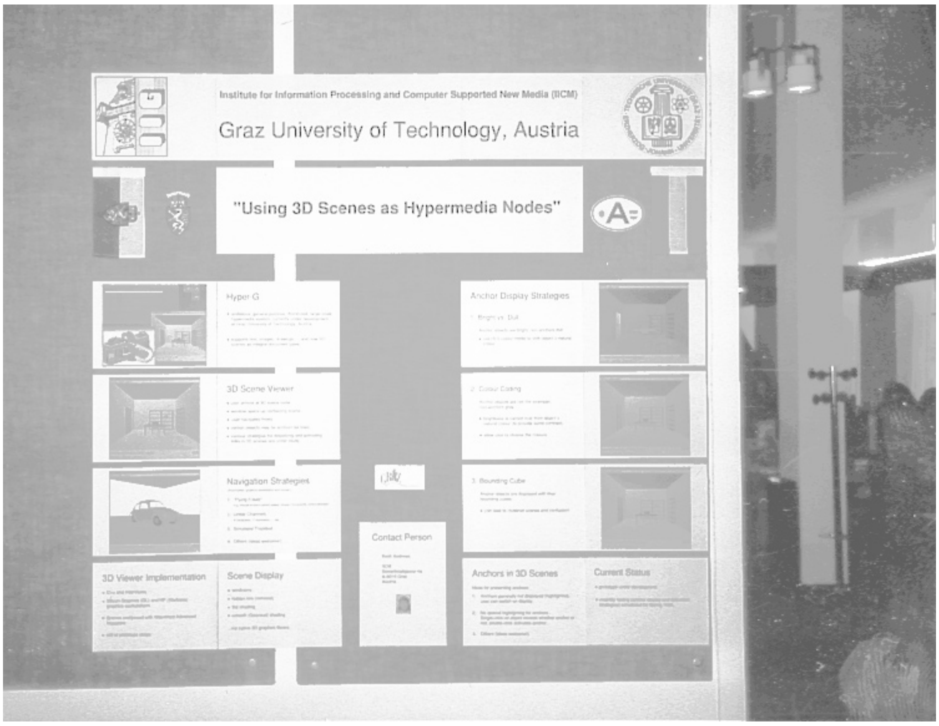

Institute for Information Processing and Computer Supported New Media (IICM)
Graz University of Technology, Austria
"Using 3D Scenes as Hypermedia Nodes"
Hyper-G
Anchor Display Strategies
3D Scene Viewer
Navigation Strategies
Contact Person
3D Viewer Implementation
Scene Display
Anchors in 3D Scenes
Current Status

Figure 7.1 Poster at ECHT'92 in Milan, December 1992: "Using 3D Scenes as Hypermedia Nodes".

7.2 The Evolution of VRweb

For lack of any other standard in 1992, we had to develop our own file format for 3d hypermedia models. We based it on the ascii output from Wavefront's Advanced Visualizer software [Wavefront Technologies, 1991], which we used at the time for modeling. We called this format SDF, for Scene Description Format. The Harmony 3d Scene Viewer at that time and the SDF file format are described in detail in Michael Pichler's Master's Thesis of October 1993 [Pichler, 1993]. I presented a more advanced version of the software at the Virtual Reality Vienna conference in December 1993 [Andrews, 1993a].

At the First World Wide Web conference in Geneva in May 1994, Dave Raggett and Tim Berners-Lee held a Birds of a Feather session to discuss "3d and the Web". This meeting spawned the `www-vrml` mailing list, moderated by Mark Pesce. Initially discussion centered around Dave Raggett's proposal for a high-level scene description language [Raggett, 1994], based on components for indoor and outdoor scenes, tentatively called VRML (Virtual Reality Markup Language). This higher-level approach did not really meet our requirements.

By the end of 1994, however, the focus had shifted to a more low-level geometric modelling language, much more similar to our own SDF format than the original VRML proposal, and VRML was renamed the Virtual Reality *Modeling* Language. Consensus was reached on the mailing list to adopt a subset of the Open Inventor ascii File Format from Silicon Graphics [Wernecke, 1994] as the starting point for VRML 1.0.

At the Third World Wide Web conference in April 1995 in Darmstadt, VRML was one of the "hot topics" and on Developer's Day I was able to announce a joint project [Andrews, 1995b] with the National Center for Supercomputing Applications (NCSA) and the Gopher team at the University of Minnesota to release a freely available VRML browser, named VRweb, based on our original Harmony 3d Scene Viewer. Version 1.0 of the Virtual Reality Modeling Language was finalised on 26th May 1995 [Bell *et al.*, 1995], and clarified as Version 1.0C on 29th January 1996 [Bell *et al.*, 1996a].

```
#VRML V1.0 ascii

Separator {

ShapeHints { vertexOrdering COUNTERCLOCKWISE shapeType SOLID faceType CONVEX }

DEF Mat_pinebrown Material { diffuseColor 0.2576 0.1225 0.0007 }
DEF Mat_bluetest Material { diffuseColor 0.7692 0.6967 0.5908 }
:
DEF Mat_rust Material { diffuseColor 0.1996 0.0567 0.0274 }
DEF Mat_sand Material { diffuseColor 0.2863 0.2367 0.1696 }

PointLight {
color 0.7633 0.7633 0.7633
location -5 2 5
}


TransformSeparator {
DEF Obj_3 Group {
Coordinate3 { point [
-1.13195 1.87823 1.13195,
-1.13195 -1.87823 1.13195,
   :
1.13186 -1.87823 1.13204,
1.13186 1.87823 1.13204]
} # Obj_3 199 vertices

USE Mat_pinebrown
IndexedFaceSet { coordIndex [
9,13,12,8,-1,
   :
25,29,30,26,-1]
}

USE Mat_rust
IndexedFaceSet { coordIndex [
34,33,32,-1,
   :
37,40,38,35,-1]
}
.
.

USE Mat_tower
IndexedFaceSet { coordIndex [
187,188,189,190,-1,
191,192,193,194,-1,
195,196,197,198,-1]
}
} # Obj_3: 141 elements
}

} # end of file
```

Figure 7.2 An example of a typical VRML 1.0 geometry file (shortened for brevity).

7.3 The Virtual Reality Modeling Language (VRML)

The Virtual Reality Modeling Language [Pesce, 1995; Eubanks *et al.*, 1996], or VRML for short, has become *the* standard format for the storage and interchange of 3d models on the Net. VRML is based on the ascii form of Silicon Graphics' Inventor [Wernecke, 1994] file format, an extensive, proven 3d graphics file format which includes a variety of special-purpose nodes. VRML adopted Inventor's node-based hierarchical structure, field syntax, and instancing mechanism, but left out a number of its special-purpose and more esoteric features. VRML defines nodes for geometry (both polygon-based and primitive shapes), text, grouping, multiple levels of detail, transformations (elementary and arbitrary matrices), coordinates, normal data, properties (materials, textures), rendering attributes (bindings and shape hints), light sources (positional, directional, spot lights), and camera definitions (perspective and orthographic).

The example in Figure 7.2 gives an idea of the structure of a typical VRML 1.0 file. The file begins with a standard header line to identify it as a VRML 1.0 ascii file. Then a set of materials are defined using the DEF construct. Light sources are defined (here point sources). Then follows the actual geometry of the model: first a set of 199 vertices is defined, then polygons (IndexedFaceSet) using these vertices. Each polygon is assigned a material with the USE construct.

Other VRML 1.0 constructs include LOD (level of detail), a grouping node allowing finer or courser models to be loaded depending upon the distance from the viewer. WWWInline is a node to include an arbitrary (sub)model given its URL. WWWInline nodes allow a scene to be split into parts which may be fetched on demand. For example, when navigating a large, complex model, components of the model might be fetched only as one approaches them (important on a slow network connection). The WWWAnchor node supports a link anchor attached to an arbitrary component of a model; anchors can be clicked to load a related VRML model or indeed any other type of file. Important for the backward-compatibility of extensions to VRML is the possibility for a node to provide a description of its fields and (optionally) the name of a base node that can be used as its substitute.

VRML 1.0 supports only *static* scene descriptions, only the viewer moves within the model and objects do not have any associated behaviour. Such scripting functionality is the declared goal of VRML 2.0, the specification for which was released on 4th August 1996 [Bell *et al.*, 1996b]. The VRML Architecture Group (VAG) [VAG, 1996], a group of eight technical experts, now coordinates development of the VRML specifications.

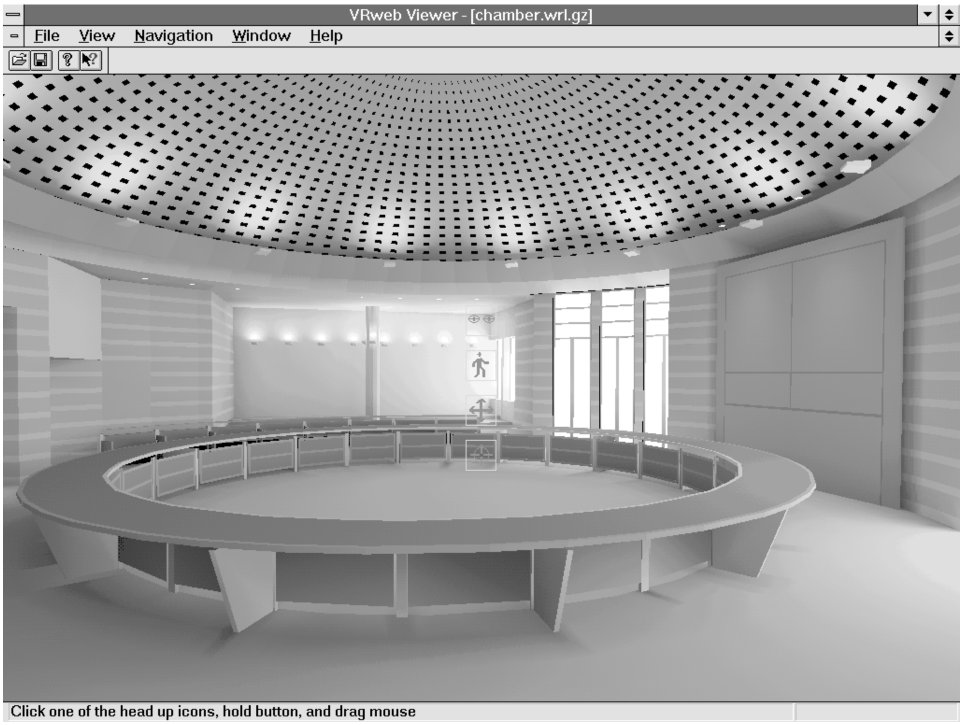
7.4 The VRweb VRML Browser

VRweb began as a joint project between IICM (home of Hyper-G), NCSA (home of Mosaic), and the University of Minnesota (home of Gopher), drawing on several years of experience at the IICM with the Harmony 3D Scene Viewer [Andrews, 1992; Andrews and Pichler, 1996] for Hyper-G, which became the base of VRweb. VRweb is designed to work with both WWW and Hyper-G clients on a variety of platforms. It is currently available for most common UNIX platforms (SGI IRIX, Sun Solaris, Sun OS, DEC Alpha, DEC ULTRIX, HP-UX, IBM AIX, and LINUX) as well as Windows NT, 3.1, and 95. VRweb source code is available free of charge for non-commercial use under the GNU General Public Licence, providing a platform for research and experiment. Unlike other VRML viewers available in source code, VRweb does *not* require additional commercial libraries like Open-Inventor or Motif, it is based entirely on freely available software components (see Section 7.7 for more detail about VRweb's software architecture). Figure 7.3 shows VRweb displaying Lightscape's [Lightscape, 1996] VRML model of the chamber of Jerusalem City Hall. Currently, VRweb supports VRML 1.0, but moves are underway to re-write most of VRweb in Java and add support for VRML 2.0.

7.5 The VRweb User Interface

The VRweb window is divided vertically into four areas: menu bar, tool bar, display area, and status bar. The menu bar provides access to the full functionality of VRweb, the tool bar and additional accelerator keys provide quick access to commonly used functions.

VRweb supports five rendering modes: wire frame, hidden line, flat shading, smooth shading, and texturing; the rendering mode specified in the VRML file can be overridden. A separate rendering mode may be specified for use during interactive navigation. For example, on a display without


VRweb Viewer - [chamber.wrl.gz]
File View Navigation Window Help

Click one of the head up icons, hold button, and drag mouse

Figure 7.3 The VRweb VRML viewer displaying Lightscape's model of Jerusalem City Hall.

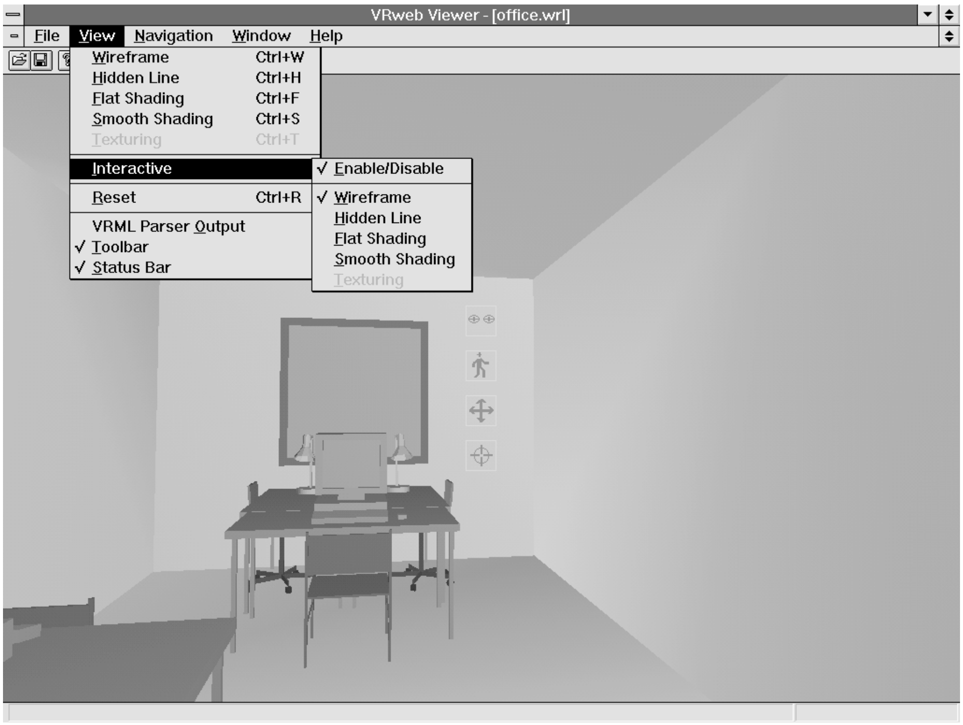
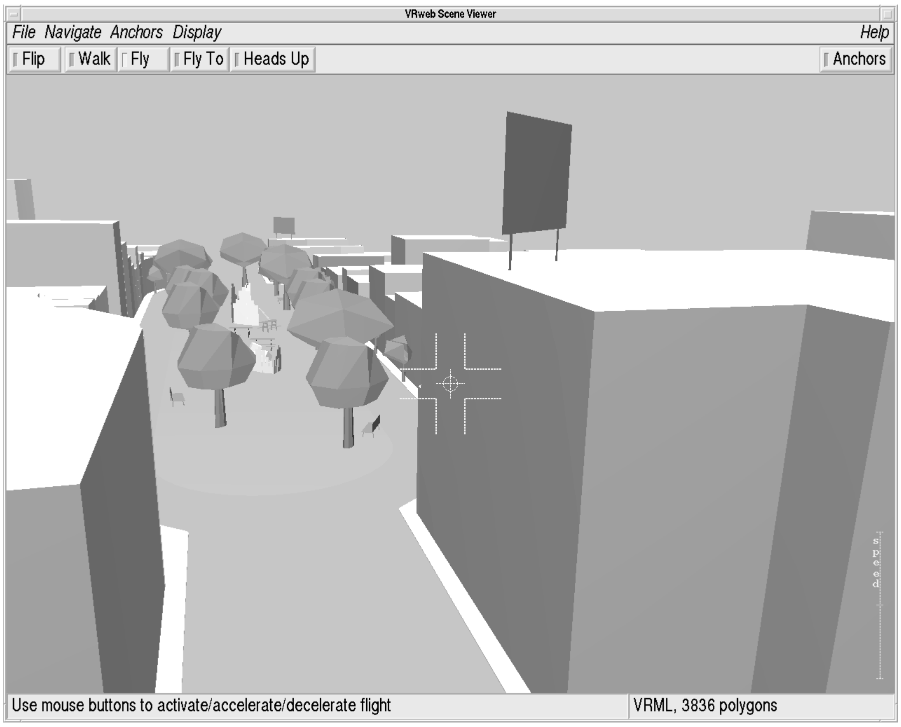


Figure 7.4 VRweb for Windows in Heads-Up navigation mode. Icons are overlaid atop the viewing area for looking, walking, vertical/sideways motion, and point-of-interest navigation. The rendering mode may be set separately for faster interactive navigation.



Use mouse buttons to activate/accelerate/decelerate flight

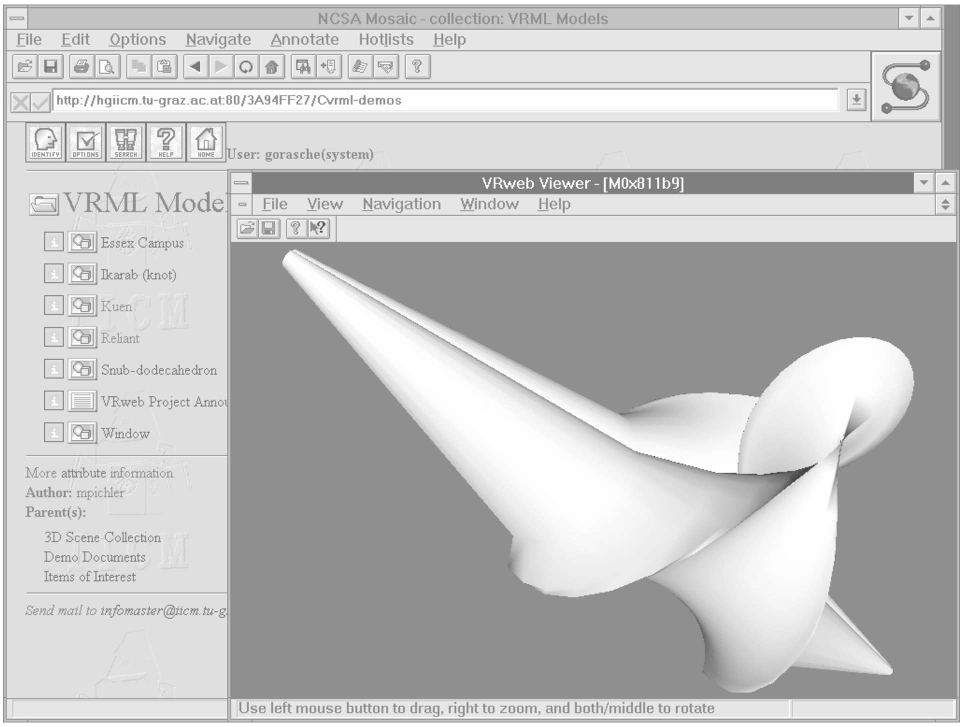
VRML, 3836 polygons

Figure 7.5 VRweb for UNIX in Fly Mode through VirtualSOMA. The speed of flight is overlaid in the bottom right corner of the viewing area, the direction of flight is determined by the position of the cursor relative to the central target crosshairs.

hardware graphics acceleration where frame rates are slow for textured models, it might be advisable to navigate in wire frame mode and see a textured version only in the pauses between interaction.

Navigation in 3d space is complicated by the need to control (at least) six degrees of freedom at the same time. VRweb does not assume the availability of any special 3d input device, such as a spaceball or position/orientation sensors, and provides a number of natural mappings for a standard 2d mouse. There are basically two classes of viewpoint control metaphor, depending on whether users wish to move the model itself or to move themselves through the model. Models of *objects* are typically moved and examined, models of *scenes* are typically navigated through (walking, flying, etc.). To cope with these different needs, VRweb provides five navigation metaphors:

- *Flip* mode is used to examine an object, whilst the viewer remains stationary. The mouse buttons have assignments for translation, rotation, and zooming (the object).
- The *Walk* metaphor is used to stroll through a 3d environment. Natural walking motion (forwards and backwards, possibly veering slightly to left or right) is assigned to the left mouse button. Complementary controls for vertical motion and side-stepping, and for turning the head are assigned to the middle and right mouse buttons respectively.
- *Fly* is like piloting an aircraft. Flight direction is controlled by the position of the mouse cursor relative to the mid-point of the viewing window (denoted by a cross-hair). The left mouse button activates flight, the other two mouse buttons control acceleration and deceleration. The current direction and speed are indicated in an overlay atop the display area.
- *Fly To* mode implements point-of-interest (POI) style navigation [Mackinlay *et al.*, 1990]. Here, the user first selects a point of interest somewhere in the model and is then able to perform controlled, logarithmic motion towards (and away from) the POI, approaching by the same fractional distance in each time step. Optionally, a rotational component can be activated (with the Shift key), which results in a final approach path to the POI head-on along the surface


NCSA Mosaic - collection: VRML Models
File Edit Options Navigate Annotate Hotlists Help
http://hgiicm.tu-graz.ac.at:80/3A94FF27/Cvrml-demos
IDENTIFY OPTIONS SEARCH HELP HOME
User: gorasche(system)
VRML Mode
Essex Campus
Ikarab (knot)
Kuen
Reliant
Snub-dodecahedron
VRweb Project Annou
Window
More attribute information
Author: mpichler
Parent(s):
3D Scene Collection
Demo Documents
Items of Interest
Send mail to infomaster@iicm.tu-g
VRweb Viewer - [M0x811b9]
File View Navigation Window Help

Use left mouse button to drag, right to zoom, and both/middle to rotate

Figure 7.6 VRweb as a helper application with Mosaic for Windows.

normal. This mode is very useful for examining details of a scene and complements other navigation metaphors like Walk or Flip, but is not sufficient as a navigation technique per se.

- *Heads Up* is perhaps the easiest navigation mode for beginners, since its controls are always clearly visible. Icons overlaid across the centre of the viewing window (like a pilot's heads-up display) symbolise the individual navigation tools: eyes to look around, a walking person for walking, crossed arrows for vertical and sideways motion, and a crosshair symbol to activate point-of-interest motion.

Figure 7.4 shows the VRweb user interface in Heads-Up navigation mode. Figure 7.5 shows a user is navigating in Fly mode through VirtualSOMA [Planet9, 1996], the VRML model of the South of Market area of San Francisco. Should users ever get lost, a “Reset View” function is available to restore the initial view of the scene. A “Level View” function makes the current view horizontal. Other functions include opening local files, saving files, setting preferences, changing colours, etc. The current frame rate can be displayed in the status area.

VRweb has optional collision detection and terrain following, so that users do not simply go through walls and floors, but slide along them. An additional viewing light can be switched on for cases when the model is badly lit. Currently in development is extensive editing functionality, which allows users to move, scale, and rotate objects, build simple scenes from prefabricated components, apply textures, and define link anchors interactively. The Hyper-G version of VRweb also supports interactive, point-and-click link creation.

7.6 VRweb and Web Browsers

VRweb can either be used both as a standalone viewer to view local VRML files (e.g. before putting them onto a server) or as a helper application for a standard Web browser. VRweb does not currently support the Netscape Navigator or Microsoft Internet Explorer Plug-In API, but may do so in future.

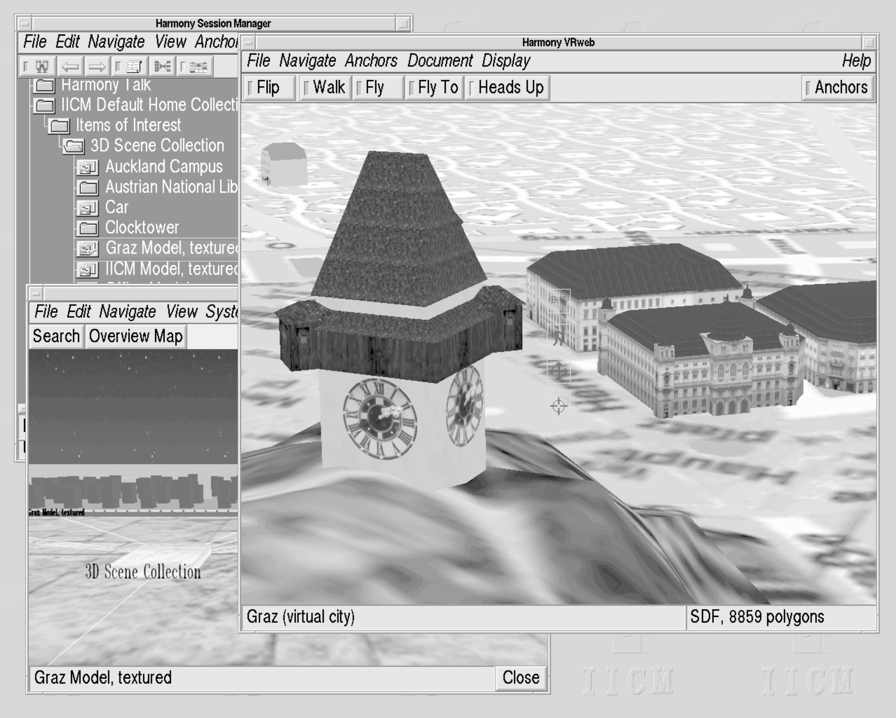


Figure 7.7 VRweb for UNIX and the Harmony client for Hyper-G. The user is navigating through a virtual model of the city of Graz. In the background are the Harmony Session Manager and 3D Information Landscape.

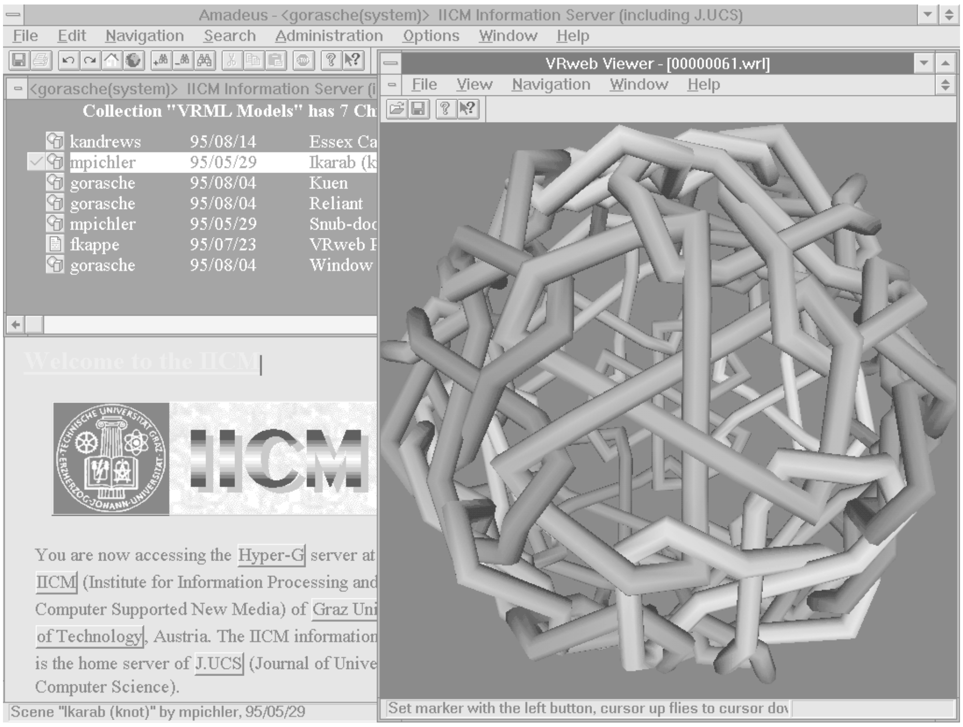
VRweb makes its own HTTP connections to resolve URLs and fetch inline nodes and textures, and directs the Web browser to display non-VRML documents which might be referenced in a WWWAnchor node. Figure 7.6 shows VRweb running with NCSA Mosaic for Windows.

VRweb and Hyper-G Authoring Tools

Harmony VRweb, a version of VRweb for UNIX, functions as a document viewer for Harmony, the Hyper-G client for UNIX/X11. Harmony VRweb speaks Harmony's Document Viewer Protocol to the Harmony Session Manager, which holds a connection to a Hyper-G server. The Session Manager retrieves metadata and hyperlinks directly, documents themselves are retrieved along a direct connection between the document viewer (VRweb in the case of VRML scenes) and the Hyper-G server. Figure 7.7 shows Harmony and VRweb with a virtual tour through through the city of Graz.

A version of VRweb for Windows is also available for Amadeus, the Hyper-G client for Windows. In the Windows environment all communication with the server is done by Amadeus. Inline scenes are handled by Amadeus using asynchronous requests. Link creation is done the same way as for other media types, e.g. text, images, PostScript. Figure 7.8 shows Amadeus and VRweb for Windows.

Even though links in documents on a Hyper-G server are stored in a link database separate from the documents themselves, link anchors contained within VRML files residing on other servers are respected by VRweb, so as to retain compatibility with the rest of the Web. VRML models residing on a Hyper-G server have their links held and managed separately. When the model is requested by a Web browser, the links are merged in on-the-fly.


Amadeus - <gorasche(system)> IICM Information Server (including J.UCS)
File Edit Navigation Search Administration Options Window Help
<gorasche(system)> IICM Information Server (i
VRweb Viewer - [00000061.wrl]
File View Navigation Window Help

Collection "VRML Models" has 7 Ch

kandrews 95/08/14 Essex Ca
mpichler 95/05/29 Ikarab (k
gorasche 95/08/04 Kuen
gorasche 95/08/04 Reliant
mpichler 95/05/29 Snub-doc
fkappe 95/07/23 VRweb F
gorasche 95/08/04 Window

Welcome to the IICM

You are now accessing the Hyper-G server at IICM (Institute for Information Processing and Computer Supported New Media) of Graz Uni of Technology, Austria. The IICM information is the home server of J.UCS (Journal of Unive Computer Science).

Scene "Ikarab (knot)" by mpichler, 95/05/29

Set marker with the left button, cursor up flies to cursor do

Figure 7.8 VRweb for Windows and the Amadeus client for Hyper-G.

7.7 VRweb's Software Architecture

Figure 7.9 shows the basic software architecture of VRweb. The WWW or Hyper-G client communicates via an interface layer with VRweb. The arrows indicate the flow of data and control signals. The VRML data stream is parsed and an internal data structure constructed. Requests for inline scenes and anchor activations are initiated by the link management module. The rendering component manages user input and visualises the data structure. For better device-independence, graphics output is done via an abstract interface to one of several underlying graphics libraries.

To parse the VRML 1.0 input stream, VRweb utilises a heavily modified version of the freely available QvLib parser by Paul Strauss of Silicon Graphics. It supports VRML 1.0 and was extracted from the Inventor library. The parser has object-oriented structure and is implemented in C++. Each node and data type of VRML corresponds to a class, which has generic methods to read the data. Methods for rendering and selecting objects have been added for use with VRweb.

To keep VRweb source code independent from any particular graphics library, an abstract interface layer called GE3D (Graphics Engine for 3D) is placed between the rendering code and the graphics library. GE3D provides functionality at a slightly higher level of abstraction than, say, OpenGL [Neider *et al.*, 1993]. A single GE3D call typically results in several lower-level OpenGL calls. For example, GE3D has single functions for geometric transformations, drawing polyhedra (a set of faces with related properties), defining light sources, etc. Interface functions to window systems, like window mapping and event handling, are kept out of GE3D, so as to achieve a high level of portability. These routines are handled in a platform-specific manner within the application. GE3D is described in appendices to both Michael Pichler and Martin Eyl's theses [Pichler, 1993; Eyl, 1995]. The CGI3D package from the University of Bonn has adopted a similar approach [Fellner, 1996].

GE3D is currently implemented for two graphics libraries: OpenGL and Mesa (an earlier version of GE3D was also ported to Hewlett-Packard's Starbase). OpenGL is the most prominent 3d graphics API, a vendor-neutral standard endorsed by most major suppliers. Its functions are at a relatively low level and designed for hardware acceleration. Its predecessor, GL (now referred to as IrisGL), was Silicon Graphics' original proprietary 3d graphics API. Mesa [Paul, 1996] is a freely

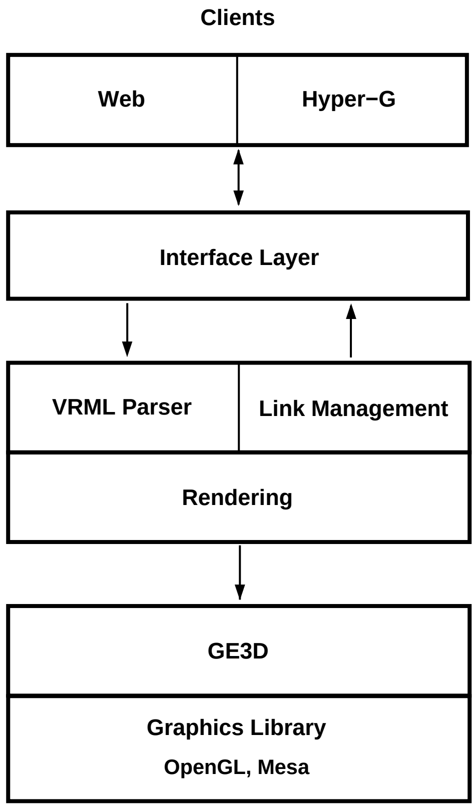


Figure 7.9 VRweb's software architecture.

available graphics library with an API very similar to that of OpenGL. Mesa maps graphics calls to both X Windows and to MS-Windows calls, allowing 3d graphics rendering without any special hardware.

VRweb for UNIX/X11

VRweb evolved from the Harmony 3D Scene Viewer for Hyper-G, which runs under UNIX. The design followed object-oriented principles, and is implemented in C++. The base class is the Scene which provides abstract methods for loading, drawing, picking, link management, and selection. Data management, including parsing and scene graph traversal, is handled by a class dependent on the scene file format.

For VRML scenes, the data structure built by the browser is traversed for drawing. As the structure of VRML is very much tailored to the capabilities of OpenGL, the mapping of VRML primitives to the drawing commands of GE3D is straightforward. Using the Inventor library for this purpose would have been easier, but would also have limited the availability of the viewer to platforms on which Inventor is available. The approach taken allows for the optimisation and tuning of the graphics library to the scope of VRML compared to the more general-purpose Inventor library.

Atop this layer, one class is responsible for user interaction. For the X11 version of VRweb, the InterViews [Linton *et al.*, 1989] toolkit and a library of widgets built on top of it are used to provide menus and other user interface elements. All events, like keystrokes, mouse clicks and movements, are handled and result in function activation or are mapped onto elementary navigational functions provided by the Camera class, depending on the current navigation metaphor. This modular approach allows for a relatively simple port to another window system or platform, as was done for MS-Windows with Visual C++.

The interface to WWW and Hyper-G clients is encapsulated in a separate module. Although they vary in their functionality and capabilities, the different clients are accessed via a common abstract interface.

VRweb for Windows

The design goal of the Microsoft Windows version of VRweb was to have a single source tree and one executable for each Windows platform (Windows 3.x, Windows 95, Windows NT on Intel, Alpha, PowerPC, and MIPS). Therefore, VRweb for Windows is based on Microsoft's WIN32s 32-bit call interface. Using WIN32s has a number of other advantages over the 16-bit Windows programming interface, such as its flat memory model, higher stability, and better device performance.

To render the three dimensional scenes we used OpenGL at a low level and the GE3D library at a higher level, thus staying compatible with the UNIX versions of VRweb. Depending on the system capabilities, either original Microsoft OpenGL (for Windows NT and Windows 95, possibly with hardware acceleration) and/or the public domain library Mesa (software-only rendering) may be used. The Mesa library was enhanced by an additional rendering mode for the Windows driver using RGB colours and a 256 colour palette with index addressing. This mode uses the WING library which was designed especially for fast bitmap display.

The architecture of VRweb for Windows is based on the Windows Multiple Document Interface (MDI) and the Microsoft Foundation Classes (MFC). This makes integration into other environments very easy. The VRweb viewer is fully integrated into Amadeus, the Hyper-G client for Windows. It will also interface with Mosaic and the Gopher client. A stand-alone version is available using the same code but not providing any link or inline facilities. Hooks for these functions are provided in the source code, so anyone can implement them for a proprietary target platform. The user interface is similar to that of VRweb for X11, but has Windows look and feel.

7.8 Beholding Cyberspace with VRweb

Until the advent of the Harmony 3D Scene Viewer, the Harmony Information Landscape, and later the introduction of VRML browsers, cyberspace was essentially flat. The integration of 3d models into the Web opens up possibilities both for transporting inherently three-dimensional information across the Web and for visualising the structure of cyberspace itself.

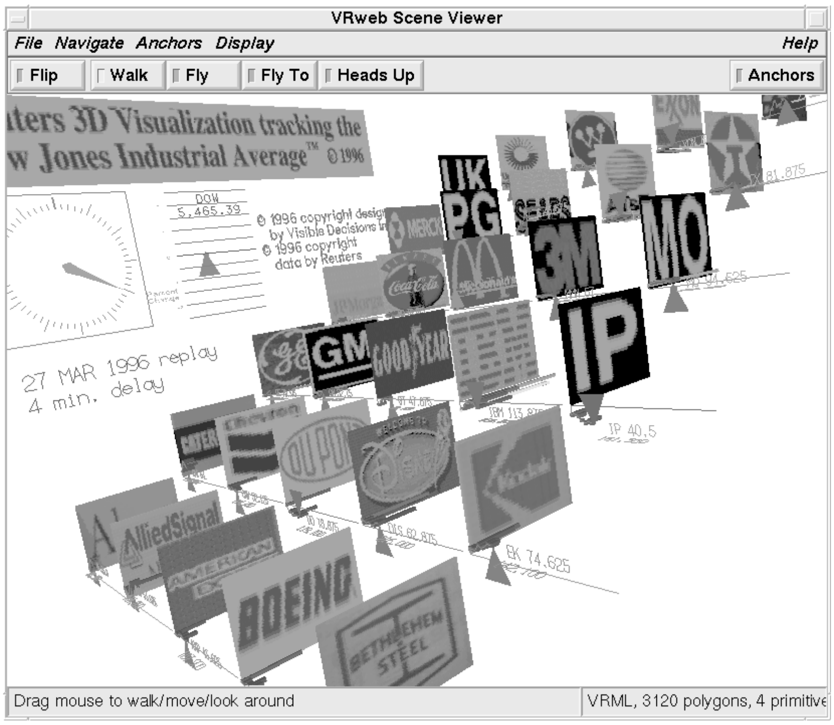
Figure 7.10 shows VRweb displaying a dynamically generated 3d visualisation of the Dow Jones Industrial Average, from Visible Decisions' web site [VDI, 1996]. The stocks incorporate links to more detailed information and the latest trade prices and volumes.

Figure 7.11 illustrates a structural visualisation of a Hyper-G server similar to that of the Harmony Information Landscape (described in Chapter 8), except that, in this case, it is a VRML model of the structure displayed in VRweb, rather than the software drawing the structure directly using OpenGL.

7.9 Further Reading

Information on the Virtual Reality Modeling Language (VRML) is available at the VRML Repository [Eubanks *et al.*, 1996] or in Mark Pesce's book [Pesce, 1995]. The VRML specifications are available from the VRML Architecture Group [VAG, 1996].

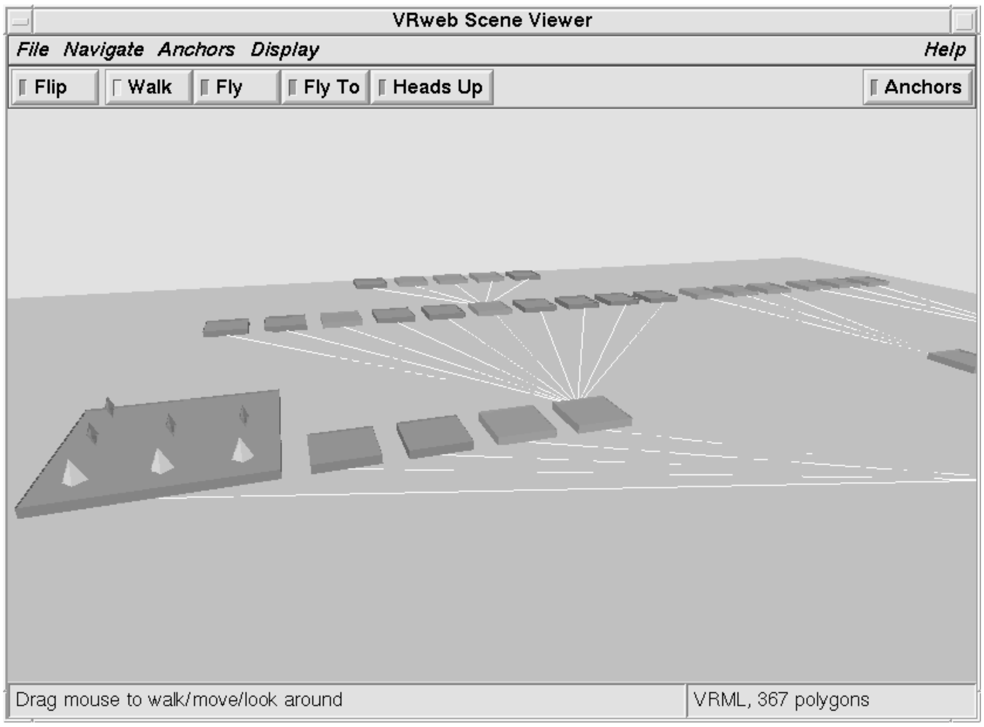
A very comprehensive text on computer graphics is [Foley *et al.*, 1990], a more tutorial introduction is given in [Hill, 1990]. The OpenGL programming interface is covered in two books [Neider *et al.*, 1993; OpenGL Architecture Review Board, 1992].


VRweb Scene Viewer
File Navigate Anchors Display
Help
Flip
Walk
Fly
Fly To
Heads Up
Anchors
ters 3D Visualization tracking the
w Jones Industrial Average™ ©1996
DOW
5,465.39
© 1996 copyright design by Visible Decisions Inc
© 1996 copyright data by Reuters
27 MAR 1996 replay
4 min. delay
IP 40.5
EK 74.625
IBM 113.8
BOEING
BETHLEHEM STEEL
GM
3M
MO
IP

Drag mouse to walk/move/look around

VRML, 3120 polygons, 4 primitive

Figure 7.10 Dow Jones Visualisation: A dynamic 3d visualisation of the Dow Jones Industrial Average displayed in VRweb. Screen shot of a VRML model generated by Visible Decisions Inc.



Drag mouse to walk/move/look around

VRML, 367 polygons

Figure 7.11 Information landscape in VRML: visualising the structure of a Hyper-G server by generating VRML.

8 Visualising Cyberspace in Harmony

This chapter discusses information visualisation in general and the information visualisation facilities of Harmony in particular. Parts of this chapter are based on material published in *Visualising Cyberspace: Information Visualisation in the Harmony Internet Browser*, Proceedings of the First IEEE Symposium on Information Visualization (InfoVis'95), Atlanta, Georgia, Oct. 1995 [Andrews, 1995a], and *Towards Rich Information Landscapes for Visualising Structured Web Spaces*, Proceedings of the 2nd IEEE Symposium on Information Visualization (InfoVis'96), San Francisco, Oct. 1996 [Andrews *et al.*, 1996].

8.1 Visualising Information Spaces

As the structure of information spaces grows more complex and the sheer amount of information available to us becomes staggering, techniques are needed to help us grasp the structure, scope, and extent of information. This is the domain of the emerging field of information visualisation. Presenting information visually utilises the human brain's remarkable perceptual ability to scan, recognise, recall, and detect changes in visual images and hence increases the bandwidth of information.

In my capacity as project leader for the Harmony client and authoring tool for Hyper-G, I decided to implement two kinds of visualisation tool for Hyper-G spaces in Harmony. The Harmony Local Map is a dynamic, two-dimensional structure map. It visualises the local relationships of a particular document or collection. Usually these are standard referential hyperlink relationships, but the Local Map can also filter links by type, for example inline images, annotations, or 3d textures. An innovation in the Local Map is to visualise the parent/child collection membership relationships of an object, particularly when multiple parents are present and the standard tree-like collection browser view becomes messy. The Harmony Local Map is described in Section 8.7.

The Harmony Information Landscape presents a three-dimensional landscape view of the collection structure of a Hyper-G Web server, similar to the way FSN [Tesler and Strasnick, 1992] visualises a file system. While FSN has fixed icons and constrains the user to face forward, the Harmony Information Landscape has configurable document icons, and allows the user to freely navigate around the space. A significant innovation of the Harmony Information Landscape is the *combined* display of both hierarchical structuring and hyperlink relationships in *one* visualisation, utilising the three available dimensions to compactly present both kinds of structure. FSN and the recent patent granted to SGI relating to information landscapes [Strasnick and Tesler, 1996] are described in Section 8.4. The Harmony Information Landscape itself is described in Section 8.8.

Both the Local Map and the Information Landscape provide location feedback, the contextual orientation mechanism covered in Chapter 5. Other kinds of visualisation are currently under development, both for Harmony and for generic web browsers in the form of Java applets, and are mentioned in Chapter 9.

8.2 Types of Information Visualisation

I distinguish between two fundamental kinds of visualisation of information spaces:

- *Hand-crafted* visualisations are constructed manually by a designer or end-user using appropriate tools.

- *Automatically generated* visualisations are composed on-the-fly by the underlying system according to a set of pre-supplied layout rules and components.

Hand-crafted Visualisations

Hand-crafted visualisations typically come in three varieties:

- *Schematic overview diagrams* are an abstract representation of the structure of information. For example, a sketch of the structure of course materials in an English literature course, or an overview map of a Web site. Schematic overview diagrams are carefully constructed for a very specific application by a designer using appropriate tools.
- *Geographic metaphors* are often employed in situations where the information space lends itself well to such mapping. Examples include access to tourist information via a country or city map, network traffic statistics overlaid on a topology map, etc.
- An *information workspace* provides tools for users themselves to organise, group, and structure (pointers to) information resources.

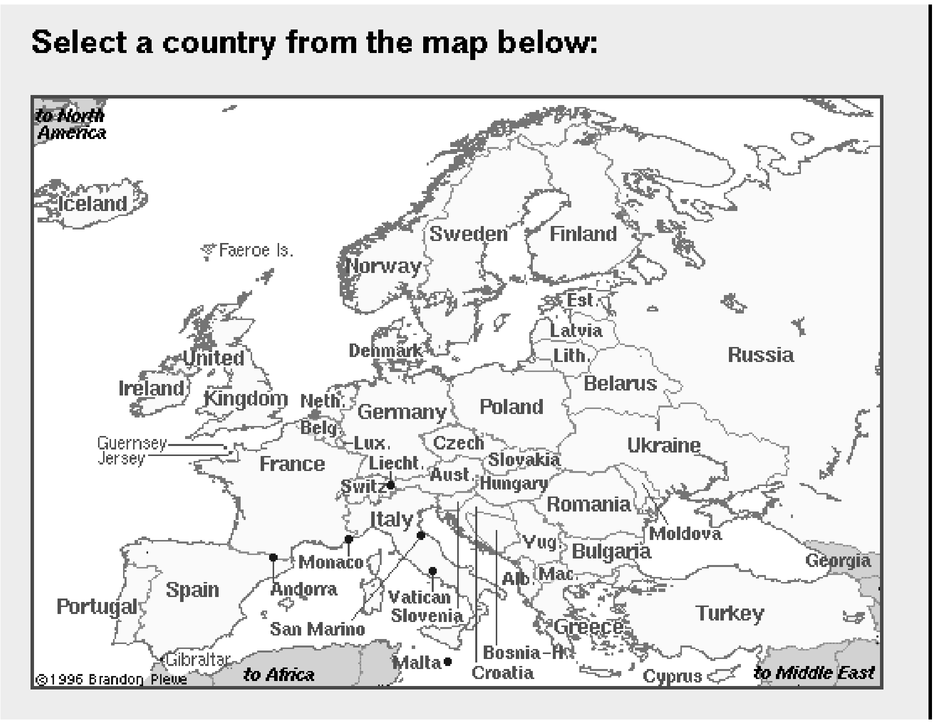
Automatically Generated Visualisations

Automatically generated information visualisations also fall into three categories:

- *Direct mapping* of attribute fields and values in the information to dimensions and attributes of the visualisation [Mackinlay, 1986; Domik, 1991]. For example, mapping a person's age to the X dimension, salary to the Y dimension, sex to the colour of icon, and job type to the shape of icon.
- *Statistical methods* from the field of information retrieval are used to analyse document collections and generate proximity measures, document clusters, and pseudo-hierarchies which can then be visualised.
- *Explicit structure* is supported by several information systems. The World Wide Web has explicit hyperlinks between documents, Gopher spaces have explicit hierarchical structure. The inherent structure of an information space can, of course, also be visualised in numerous ways, for example local link maps of hyperlinks, or 3d landscapes of hierarchical structure.

Automatically generated visualisations often employ abstract spatial metaphors. One issue affecting such visualisations is that of dynamic restructuring, for example to take into account changes in the information base or in response to user queries or filters. In some cases this may be desirable, but in many cases an inherent advantage of spatial visualisations, that users can incrementally build a mental map of places and locations and navigate by visual memory, will be lost when the information space continually changes its form. At the very least, such changes should be made via smooth transitions as far as possible so as to take advantage human visual perception and object constancy [Robertson *et al.*, 1993].

Other issues affecting both hand-crafted and automatically generated visualisations include the choice of visual representations and navigational metaphors. What kind of visual representations are best? Two or three-dimensional icons of books for text documents, framed pictures for images, radios emitting audio clips, TV sets for video, etc. Should planar or truly 3d structure maps be used, indoor or outdoor scenes? In 3d visualisations, typical navigational metaphors include walking, flying, driving, teleportation, and magic carpets. However, why not bring the information to the user, rather than having the user navigate to the information? Indeed, the use of user-configurable workspaces, where users can place and organise frequently accessed information (or handles to it) might reduce the need for navigation in the first place.



Select a country from the map below:

Figure 8.1 Virtual Tourist II: a map-based interface to local and regional tourist information.

A neutral file format like the Virtual Reality Modeling Language (VRML) [Pesce, 1995] for 3d models, allows the distinction between hand-crafted and automatically generated visualisations to be blurred somewhat as far as final presentation is concerned, since both hand-crafted and automatically generated VRML files can be displayed by one and the same viewer.

Ben Shneiderman has a simple guiding principle for information visualisation, the lesson he abstracted from a series of projects, which he calls the *Visual Information Seeking Mantra* [Shneiderman, 1996]:

> *"Overview first, zoom and filter, then details-on-demand"*

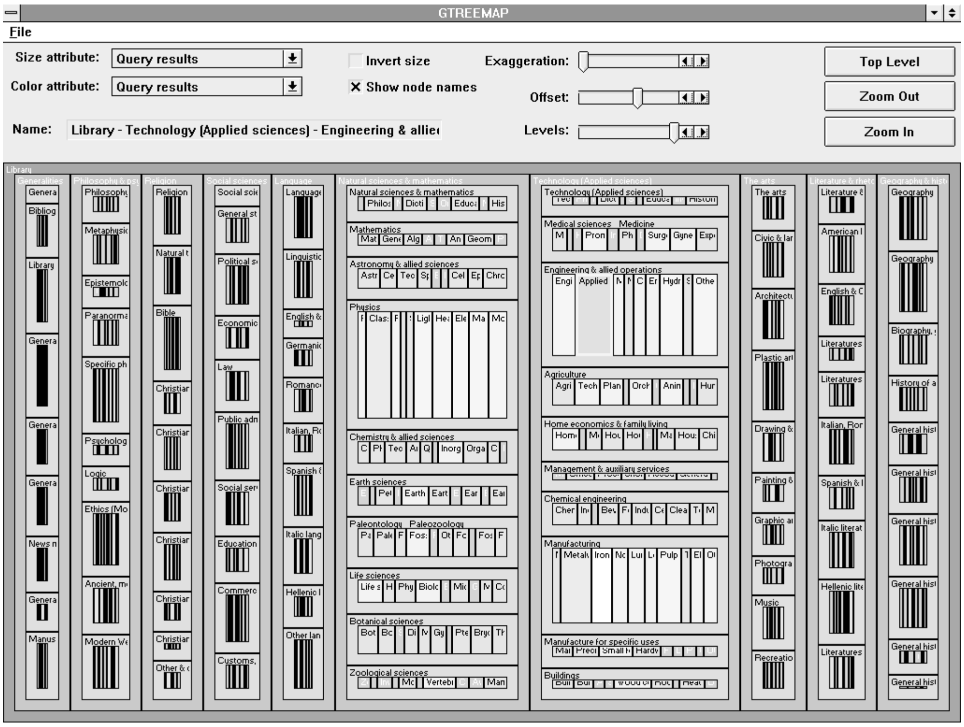
Expanding on this, Ben produced a taxonomy of information visualisation by task and by data type.

8.3 Examples of Information Visualisation

Figure 8.1 shows a good example of a hand-crafted visualisation employing a geographic metaphor: the clickable world and regional maps provided as part of the Virtual Tourist II, a map-based interface to City.Net which is a library of local and regional tourist information.

The Xerox Information Visualizer [Robertson *et al.*, 1993; Card *et al.*, 1991] is an early example of an information workspace. Users can interactively explore workspaces modeled as three-dimensional rooms. Particular data sets (nodes) are visualised in rooms of their own designed for specific tasks, and doors lead from one room to another in a manner similar to hypermedia links.

Within the 3D/Rooms metaphor, the Information Visualizer provides automatically-generated, three-dimensional representations of linear and hierarchically structured information. Linear information, such as chronologically ordered information, is pasted on to a virtual *perspective wall* from left to right [Mackinlay *et al.*, 1991]. The wall has a large front section, and left and right sides which tail off into the background. Information can be slid along the wall to bring it into focus on the front



Size attribute: Query results

Color attribute: Query results

☐ Invert size

☒ Show node names

Name: Library - Technology (Applied sciences) - Engineering & allie

Figure 8.2 Tree Map: visualising large hierarchies by successive horizontal and vertical partitioning. This tree map illustrates the Dewey Decimal Classification System. Copyright ©1996 Ben Shneiderman, University of Maryland, used with permission.

section, whilst retaining context to the left and right. The information can also be stretched or shrunk along the wall.

Hierarchical information, such as part of a file system or a company hierarchy chart, can be visualised as a *cone tree* [Robertson *et al.*, 1991], a uniform 3d cone of information branches, like the one in Figure 8.13. The tree can be rotated to bring interesting parts to the front and pruned to remove non-relevant information.

Tree maps [Johnson and Shneiderman, 1991] are a two-dimensional space-filling visualisation technique for hierarchically structured information. As can be seen in Figure 8.2, the available screen space is successively partitioned horizontally and then vertically, such that the area of each rectangle reflects the size (in bytes, or in number of members) of the corresponding subtree.

Another automatically-generated approach to visualising large hierarchies is to use hyperbolic geometry [Lamping *et al.*, 1995]. *Hyperbolic maps* always display the entire tree, but focussed on one specific part. Elements too far down the hierarchy are essentially too small to be seen. Figure 8.3 shows a hyperbolic map of the Xerox PARC web site.

Numerous hypermedia systems have provided *2d link maps* for hyperlink structures. A good overview can be found in [Utting and Yankelovich, 1989].

SemNet [Fairchild *et al.*, 1988] was an exploratory system which represented knowledge bases as directed graphs in three dimensions. Labeled rectangles (nodes) were connected by lines or arcs. The 3d layout has the advantage over 2d layouts that the nodes of an arbitrary graph can be positioned such that no arcs intersect. Several techniques for positioning nodes were explored: random, multi-dimensional scaling, heuristics, and manual editing. Clustering techniques and fisheye views were also implemented.

VizNet [Fairchild *et al.*, 1993] also uses a cone tree representation for hierarchical information, but provides an additional spherical representation for associative relationships (local map). The current node is located (say) at the north pole, nodes similar to it are strung along lines of longitude.

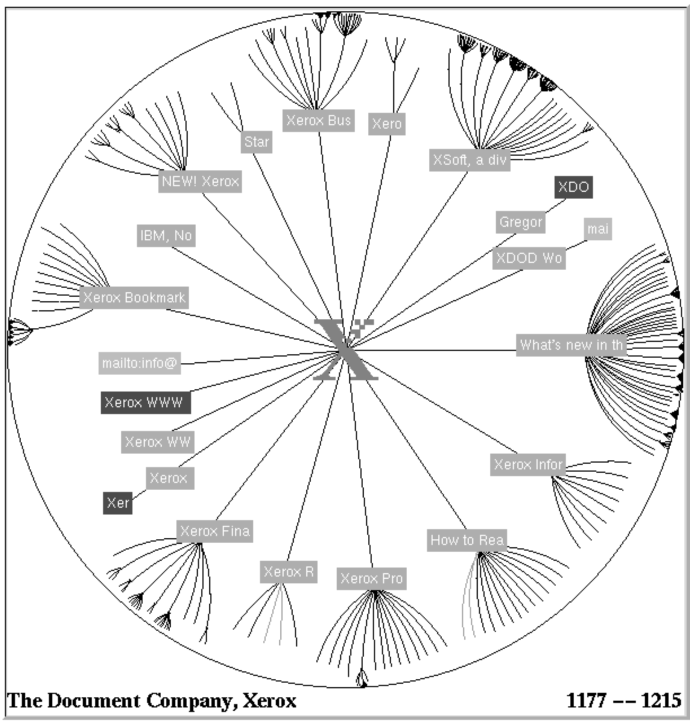


Figure 8.3 Hyperbolic Map: mapping a large hierarchy to the surface or a sphere using hyperbolic geometry. Copyright ©1996 Xerox PARC, used with permission.

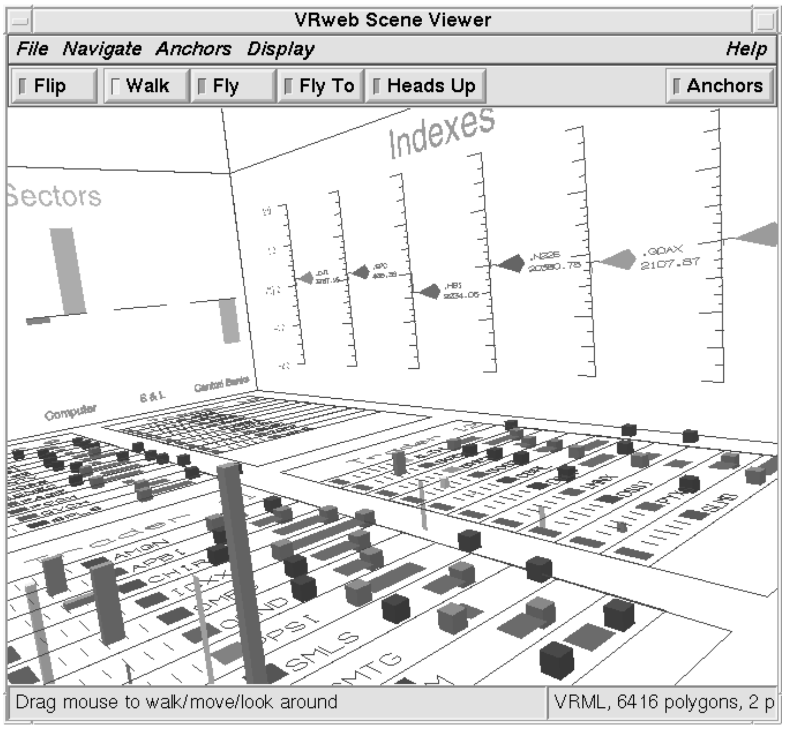


Figure 8.4 Discovery Head Trader: A dynamic visualisation of the trading floor at a brokerage house. Individual traders' desks are spread across the floor, the walls display summaries of performance and market conditions. Screen shot of a VRML model by Visible Decisions Inc.

Lower level objects are displayed on lower level spheres (like peeling away layers of an onion).

Serra et al [Serra *et al.*, 1991] discuss the use of 3d object hierarchies with attached multimedia documents. Each component in the 3d object hierarchy (part-of relationships) may be combined into a *concept node* with text, image, and video documents. Links may be made from these text, image, and video documents to other concept nodes. However, there is no support for arbitrary links from (parts of) the 3d object as such, and the 3d object hierarchy itself forms the entire extent of the hyperstructure.

Smith and Wilson [Smith and Wilson, 1993] described a prototype system based on HyperCard and Virtus Walkthrough (a 3d visualisation system), in the context of an academic departmental information system. They enabled users to interactively explore a 3d model of the department: when they approached within a certain distance of a source anchor, it automatically triggered to display a corresponding text document.

Visible Decisions Inc. [VDI, 1996] specialises in generating dynamic information visualisations in VRML, which can be displayed by any VRML browser. The example shown in Figure 8.4 is Head Trader, a dynamic visualisation of the trading floor at a brokerage house. Individual traders' desks are spread across the floor, the walls display summaries of performance and market conditions.

8.4 FSN and SGI's Information Landscape Patent

The File System Navigator (FSN, or "Fusion") written by Joel Tesler and Steve Strasnick at Silicon Graphics (SGI) [Tesler and Strasnick, 1992] visualises a UNIX file system as an information landscape. Directories are represented by blocks laid out on a plane, their height representing the cumulative size of the contained files. Smaller blocks atop the directory blocks represent files in the directory (their size also mapped to their height), as shown in Figure 8.5. Users can "fly" over the landscape, taking it in as a whole, or swoop down to a specific directory. Clicking on the arc to a subdirectory results in an invisible hand grabbing you and leading you through space to that subdirectory. Clicking on a file block brings a virtual spotlight to bear on that block, as illustrated in Figure 8.6. Double-clicking opens the file for editing, etc.

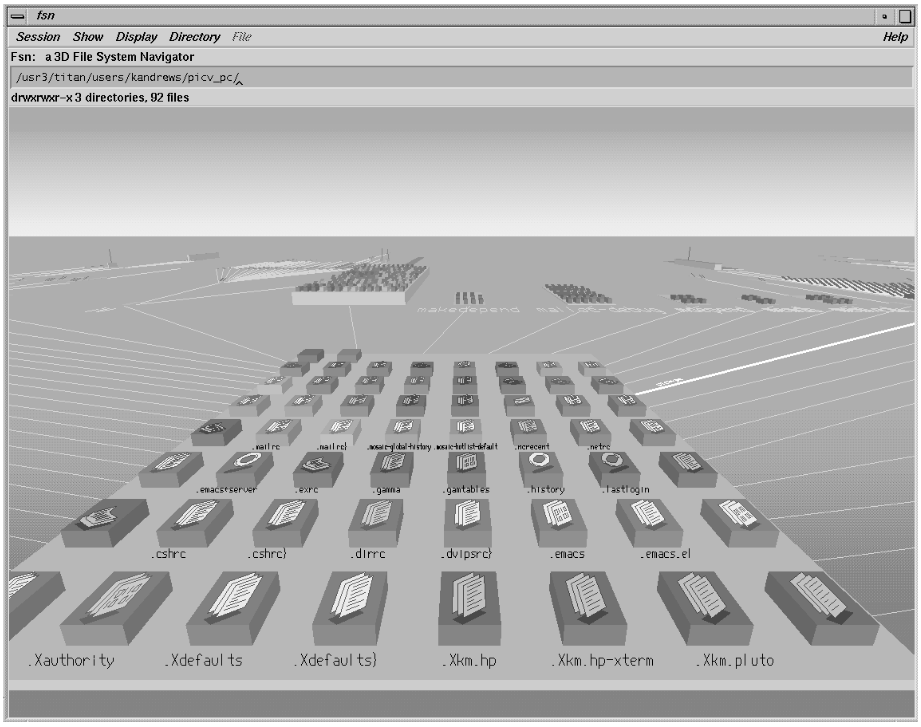
United States patent number 5,528,735 entitled "Method and Apparatus for Displaying Data within a Three-Dimensional Information Landscape", filed on 23rd March 1993, was granted to SGI on 18th June 1996 [Strasnick and Tesler, 1996]. FSN is cited along with a sales database as an example of a preferred embodiment of the patent. Of the 30 individual claims, 16 cover computer-implemented methods. In summary, the claims relate to:

- Laying out objects within a 3d display landscape.
- Compressing and expanding the fan out of the tree based on viewpoint, so as to fit more objects in the display.
- Displaying an overview with a positional cursor.
- Navigation through the 3d display landscape.

Note, however, that the patent only applies in the United States of America. A second US patent application entitled "Method and Apparatus for Navigation within a Three-Dimensional Information Landscape" is pending.

8.5 Visualising Cyberspace

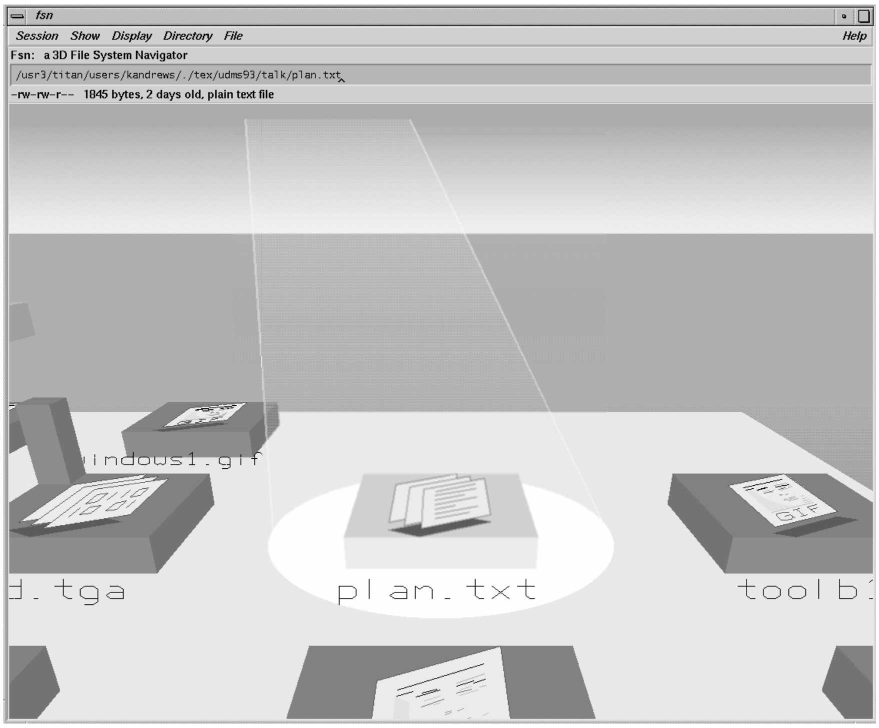
Information visualisation techniques are increasingly being applied to Internet information systems such as Gopher [McCahill and Anklesaria, 1995] and the World Wide Web (WWW) [Berners-Lee *et al.*, 1994].



/usr3/titan/users/kandrews/picv_pc/

drwxrwxr-x 3 directories, 92 files

Figure 8.5 FSN: The File System Navigator uses a landscape metaphor to visualise the hierarchical structure of a file system.



/usr3/titan/users/kandrews/./tex/udms93/talk/plan.txt

-rw-rw-r-- 1845 bytes, 2 days old, plain text file

Figure 8.6 FSN: A selected file is brought under a virtual spotlight.

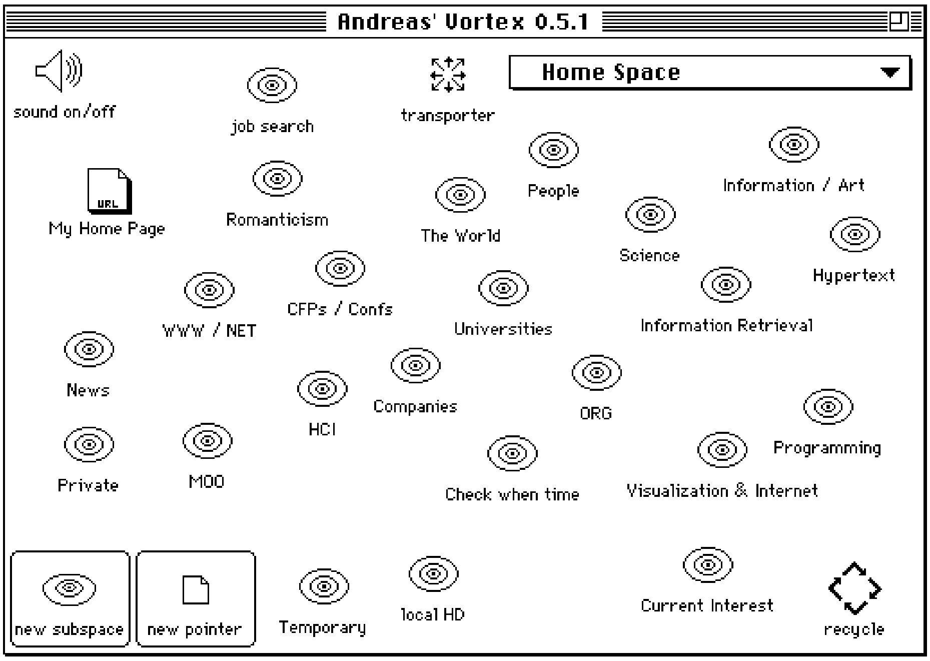


Figure 8.7 The Vortex: A personal workspace for organising pointers to World Wide Web resources. Copyright ©1996 Andreas Dieberger, used with permission.

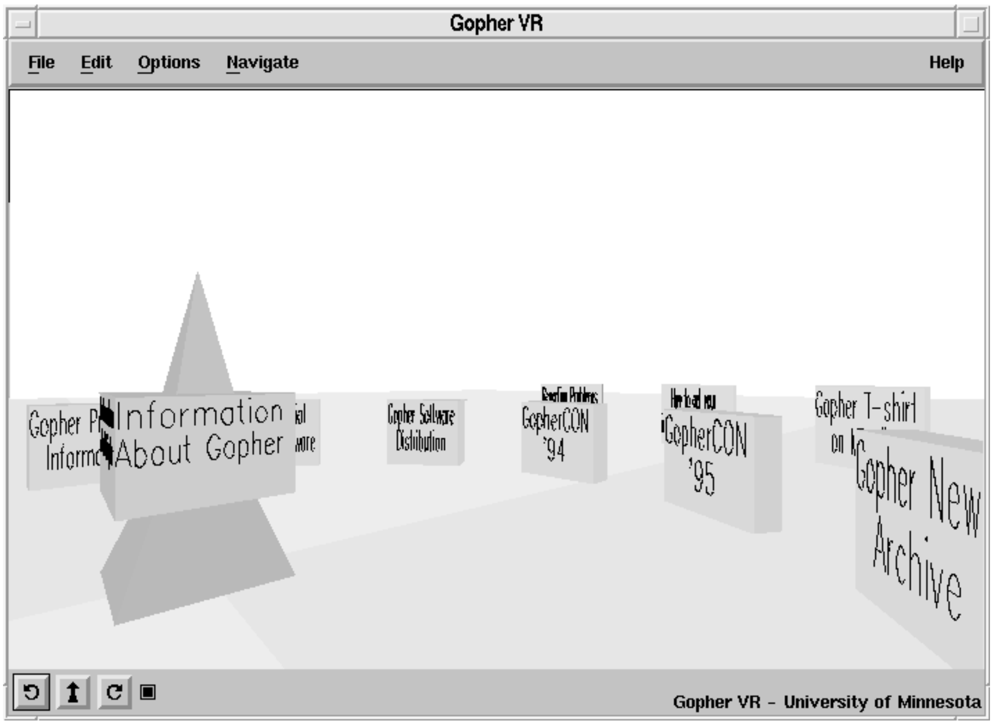


Figure 8.8 GopherVR: A landscape visualisation of the hierarchical menu structure of a Gopher space. The entries of a menu are arranged around a central pillar representing the parent.

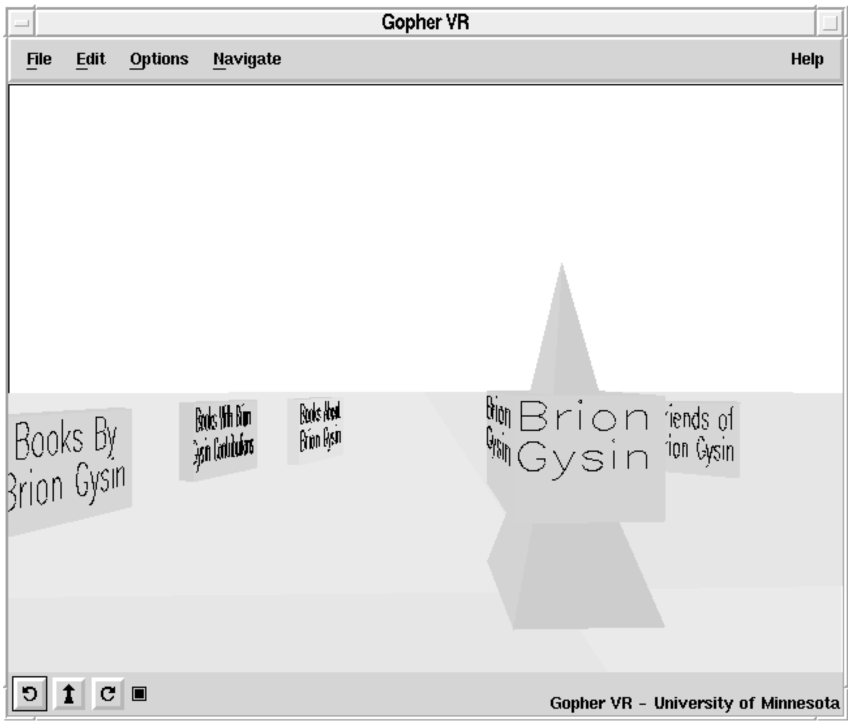


Figure 8.9 GODOT: Spatial coordinates are assigned as attributes to Gopher+ menu entries and are visualised in GopherVR.

The Vortex, by Andreas Dieberger, shown in Figure 8.7, is a personal workspace for pointers to World Wide Web resources, like an extended bookmark list. Hierarchical structuring tools are provided for grouping pointers into subspaces. Pointers and subspaces can also be arranged on screen by proximity. Clicking on a resource activates a Web browser to access the resource.

The WebBook/Web Forager [Card *et al.*, 1996] from Xerox PARC is a three-dimensional personal workspace for pointers to World Wide Web resources. Web pages can be retrieved and organised as leaves of books according to their content, books can be placed on shelves, etc.

GopherVR [McCahill and Erickson, 1995] is an experimental 3d spatial interface to the hierarchically structured information on a Gopher server. As shown in Figure 8.8, the entries of each menu level are arranged around a central pillar representing the parent. Clicking on the central pillar navigates one level up in the menu tree, clicking on a radial entry navigates down one level and visualises the members of that menu. GopherVR includes some nice “personalised” touches to make users feel more comfortable: users can jump up in the air for an overview, and land with a slight bump! GODOT, an extension to GopherVR utilises the attribute metadata of a Gopher+ [Anklesaria *et al.*, 1993] server to assign positional coordinates to the members of a menu [Iacovou and McCahill, 1995]. The creator of a Gopher+ space can position and group members of a menu according to arbitrary criteria such as their similarity. Figure 8.9 shows an example for the author Brion Gysin.

HyperSpace [Wood *et al.*, 1996], from the University of Birmingham, uses a simulated annealing technique to visualise the link structure of a WWW space. Spheres representing nodes are initially placed randomly in 3d space and connected by lines representing links between them. Nodes repel each other, whilst links provide an attractive force. The initially chaotic mesh self-organises itself such that highly inter-related (i.e. inter-linked) documents are pulled together and clustered in space, as shown in Figure 8.10. New documents can be fetched and their links added to the visualisation.

WebView from the University of Washington [Linden, 1996] implements a 2d simulated annealing algorithm in Java. From a given URL, local links are accumulated and laid out on a 2d plane, as shown in Figure 8.11.

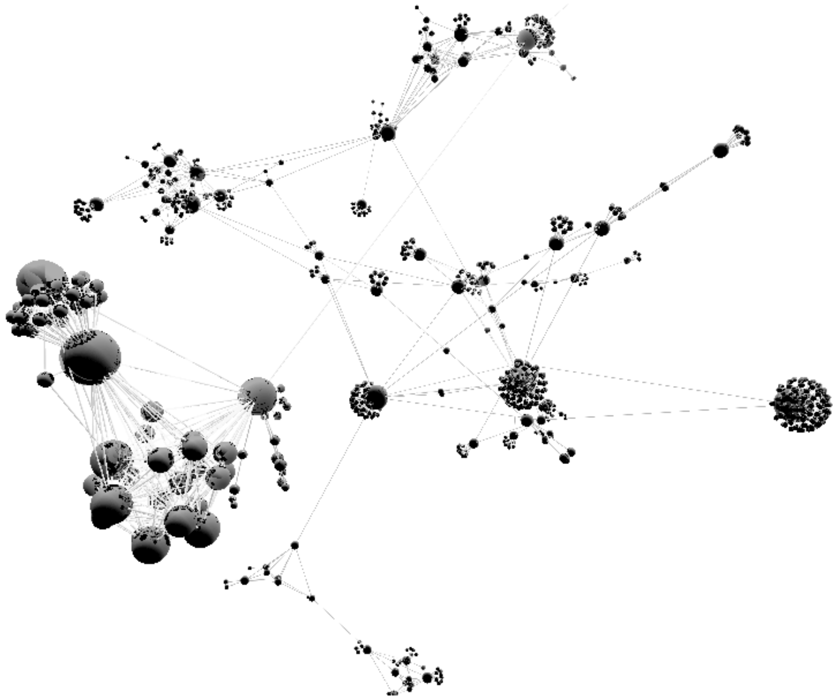
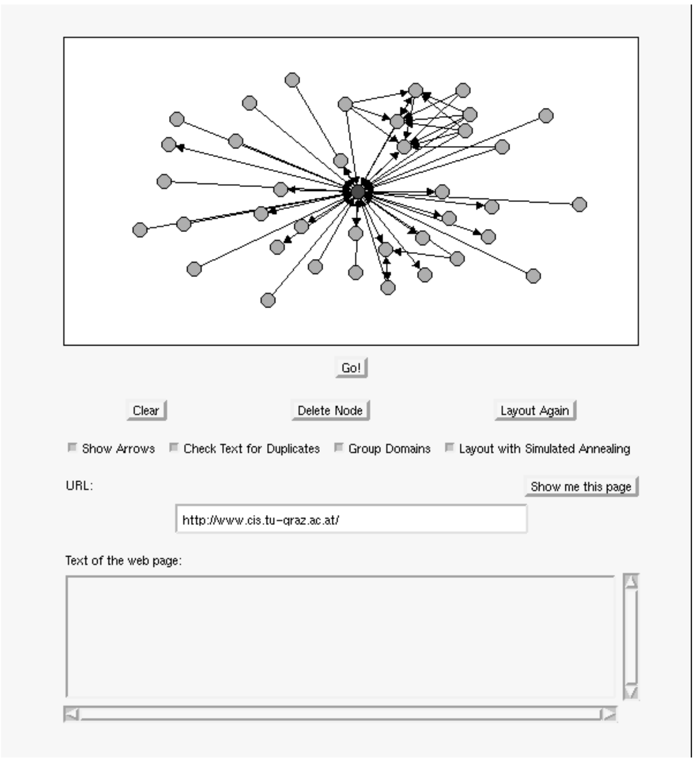


Figure 8.10 HyperSpace: a self-organising 3d map of documents and links in a WWW space based on simulated annealing. Copyright ©1995 Andy Wood, University of Birmingham, used with permission.



□ Show Arrows □ Check Text for Duplicates □ Group Domains □ Layout with Simulated Annealing

URL: http://www.cis.tu-graz.ac.at/

Text of the web page:

Figure 8.11 WebView: Java-based simulated annealing visualisation of WWW space. This is a map of Graz University of Technology Computer Centre’s web site.

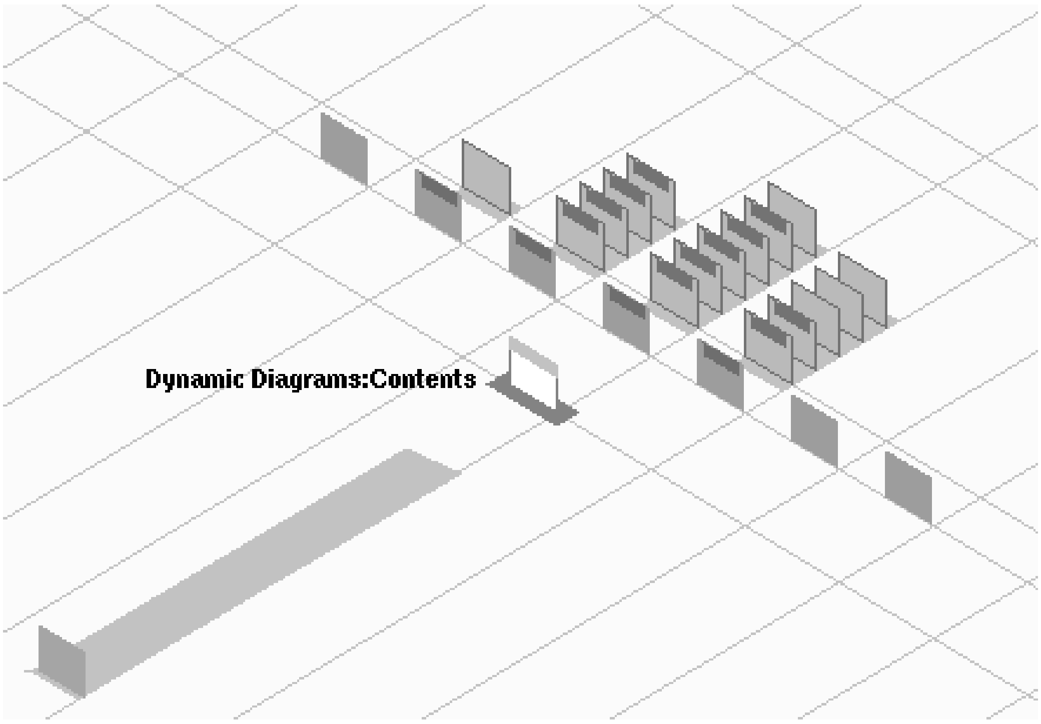


Figure 8.12 MAPA: a Java-based visualisation of a hierarchical map of WWW space. The Web space here is the Dynamic Diagrams web site.

MAPA from Dynamic Diagrams [Kahn, 1996] gathers link data from a Web site, and assembles a synthetic hierarchy reflecting the editorial intention of the authors. The hierarchy is stored in a database and can be manually fine-tuned. A Java applet is used to display the hierarchical structure from within a Web browser, as depicted in Figure 8.12.

The Navigational View Builder [Mukherjea and Foley, 1995] uses a combination of link structure analysis and manually assigned attributes (such as Author, Author-Type, Directory, Topic, and SubTopic) to cluster documents and form synthetic hierarchies. These then form the basis for visualisations such as the cone tree diagram shown in Figure 8.13.

Taking an entirely different approach, Tim Bray used information contained in Open Text's robot-generated index of about 1.5 million Web documents to generate dynamic visualisations in VRML of the biggest and most visible sites on the Web at the time [Bray, 1996], as shown in Figure 8.14. Characteristics used to drive the visualisation include: the number of links pointing to a site (its *visibility*), the number of links pointing from a site (its *luminosity*), and the number of pages a site contains (its *size*).

The Internet Foyer, by the Communications Research Group (CRG) at the University of Nottingham [Benford *et al.*, 1996], tracks the position of users on the CRG web site and creates a navigable, populated visualisation of the CRG home pages using the DIVE Collaborative Virtual Environment (CVE) platform [DIVE, 1996]. Nodes are modeled as spheres and hyperlinks as 3d arrows. Visitors to the site are represented as 3d avatars. Figure 8.15 shows a view of (other) visitors in the foyer, gathered around the particular documents they are accessing. Visitors using a CVE client are able to freely navigate the virtual foyer and interact (talk) with other CVE visitors. Visitors using a web browser see a series of flat projections of the foyer and a list of current visitors.

A particularly innovative feature is the integration of a real and virtual foyer in a "mixed reality" application. As can be seen in Figure 8.16, the virtual foyer is projected onto the wall of a real physical foyer. At the same time, a camera transmits live video images from the physical foyer, which are represented as a texture map in the virtual environment. A microphone and speakers in the

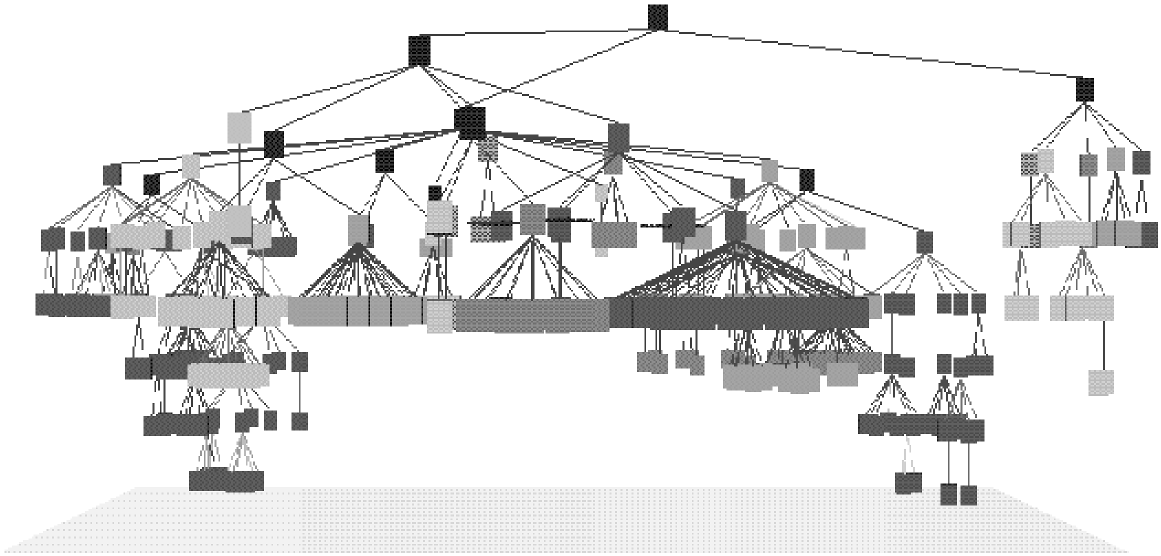


Figure 8.13 The Navigational View Builder: A synthetic cone-tree style hierarchy map of a web site generated by a combination of structure and attribute analysis. Copyright ©1995 Sougata Mukherjea, Georgia Institute of Technology, used with permission.

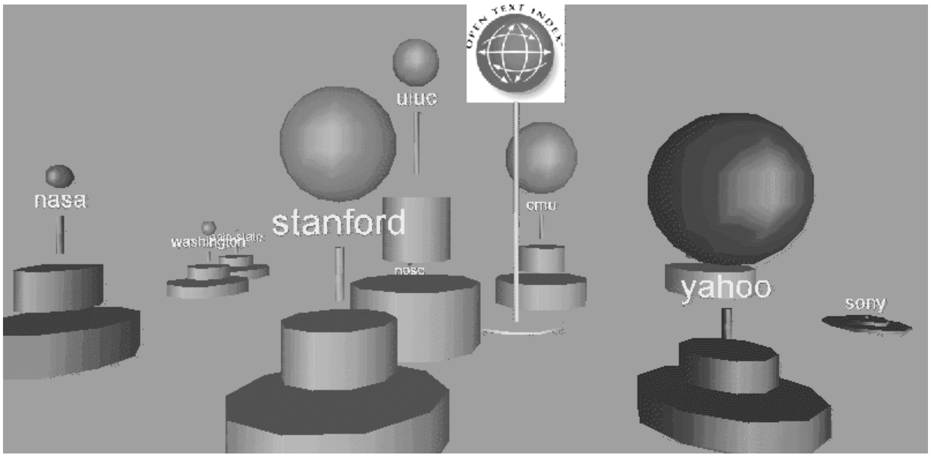


Figure 8.14 A visualisation of the biggest and most visible sites on the Web, using data in the Open Text robot-generated index of the Web. Copyright ©1996 Tim Bray, Open Text Corporation, used with permission.

physical foyer complete the link-up between physical and virtual spaces. One of the consequences of this set-up is that members of the CRG can *literally* watch as visitors browse their web pages!

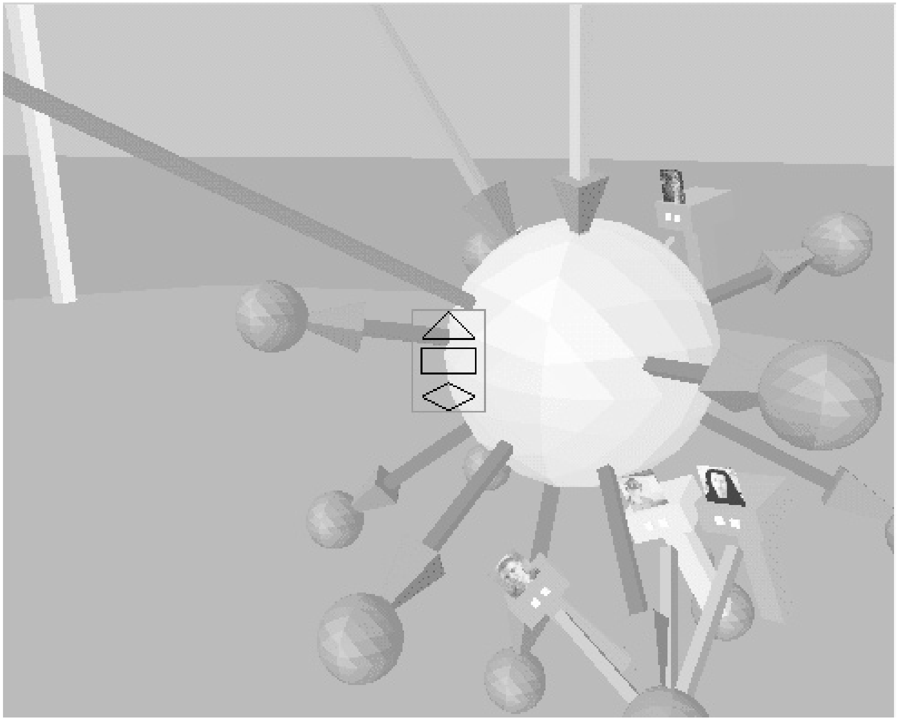


Figure 8.15 The Internet Foyer. A navigable, populated visualisation of a Web site, showing visitors to the site as 3d avatars. Copyright ©1996 Communications Research Group, University of Nottingham, used with permission.

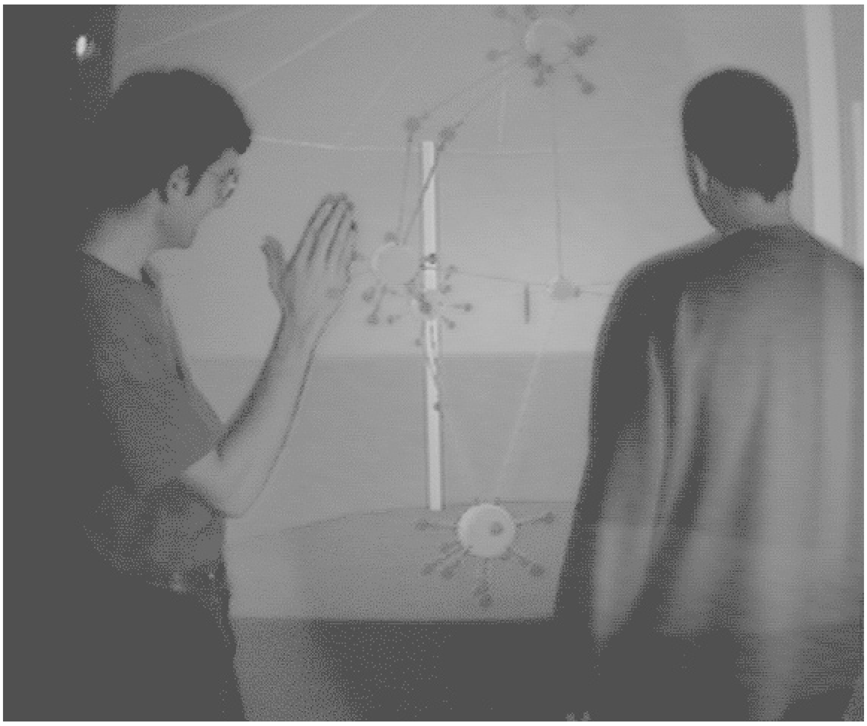


Figure 8.16 The Internet Foyer is a “mixed reality” application, integrating a real and a virtual foyer. Copyright ©1996 Communications Research Group, University of Nottingham, used with permission.

8.6 Hyper-G's Rich Infrastructure for Visualisation

Three strategies have traditionally been employed to help users find their way in Internet information spaces:

- A-priori organisation of information into hierarchies, like the table of contents of a book (and the distinguishing feature of Gopher).
- Search facilities, like the index of a book (and typical of WAIS).
- Hyperlinks, like cross-references in a book (and embodied in the World-Wide Web).

The scope for visualisation of Gopher, WAIS, and WWW spaces is severely restricted by the limitations of their respective data models. Visualisations, insofar as they have been provided at all, have necessarily treated each separately. Worse still there is little standardised meta-information, such as document type, size, owner, importance, etc., which can be bound to visual attributes of a visualisation.

Gopher supports a-priori organisation of information into hierarchies, and the most recent Gopher protocol, Gopher+ [Anklesaria *et al.*, 1993], supports both rich metadata attributes and HTML documents (and hence to some degree hyperlinks), but Gopher+ is not widely implemented.

Looking specifically at the World-Wide Web, we find that on standard WWW servers, the hyperlink is the only structuring facility available, leading to the "World Wide Spaghetti Bowl" (to quote Robert Cailliau). Hierarchical structures can be simulated using menus of links, but such structural links are not readily distinguishable from standard associative links.

Moreover, most WWW servers do not make hyperlinks available separately, but store them embedded within documents, making the determination of back-links both difficult and time-consuming. Similarly, search facilities in WWW are not standardised. Search engines are typically bolted on to a standard Web server and their interfaces vary widely. To make things worse, the WWW currently provides only a handful of standardised meta-data attributes such as document size, type/format, and modification time[1].

Thus, although techniques from information visualisation can be used to present graphical overviews of WWW information spaces, visualisations of the WWW today are, in general, limited to maps of one-way hyperlinks, due to the lack of bidirectional links, additional structuring facilities, and standardised meta-information. Or as I put it in my talk [Andrews, 1996g] at the CHI'96 panel on *Visualising the Internet* [CHI96, 1996]: WWW lacks the "meat" to visualise.

Hyper-G, as described in Chapter 4, supports a-priori organisation of information into hierarchies, provides rich sets of standardised meta-data, and has a standard representation for search results. In addition, Hyper-G uses a separate link database for hyperlink facilities, has typed links, and has fully integrated search facilities. Only with such rich supporting infrastructure is it possible to generate tightly-coupled, multi-faceted visualisations of information spaces. This leads to "Keith's Recipe for Visualisation":

(1) Provide the "meat":
    - Hierarchical structuring in collections.
    - Standardised, integrated search facilities.
    - Hyperlinks in separate link database.
    - Rich metadata for documents, links, and collections.

(2) Cook *tightly-coupled* visualisations of the first three.

(3) Flavour with attribute metadata.

[1] There is a proposed mechanism (the META tag) to include non-standard attributes in HTML documents. However, the availability and naming of such tags varies widely from server to server.

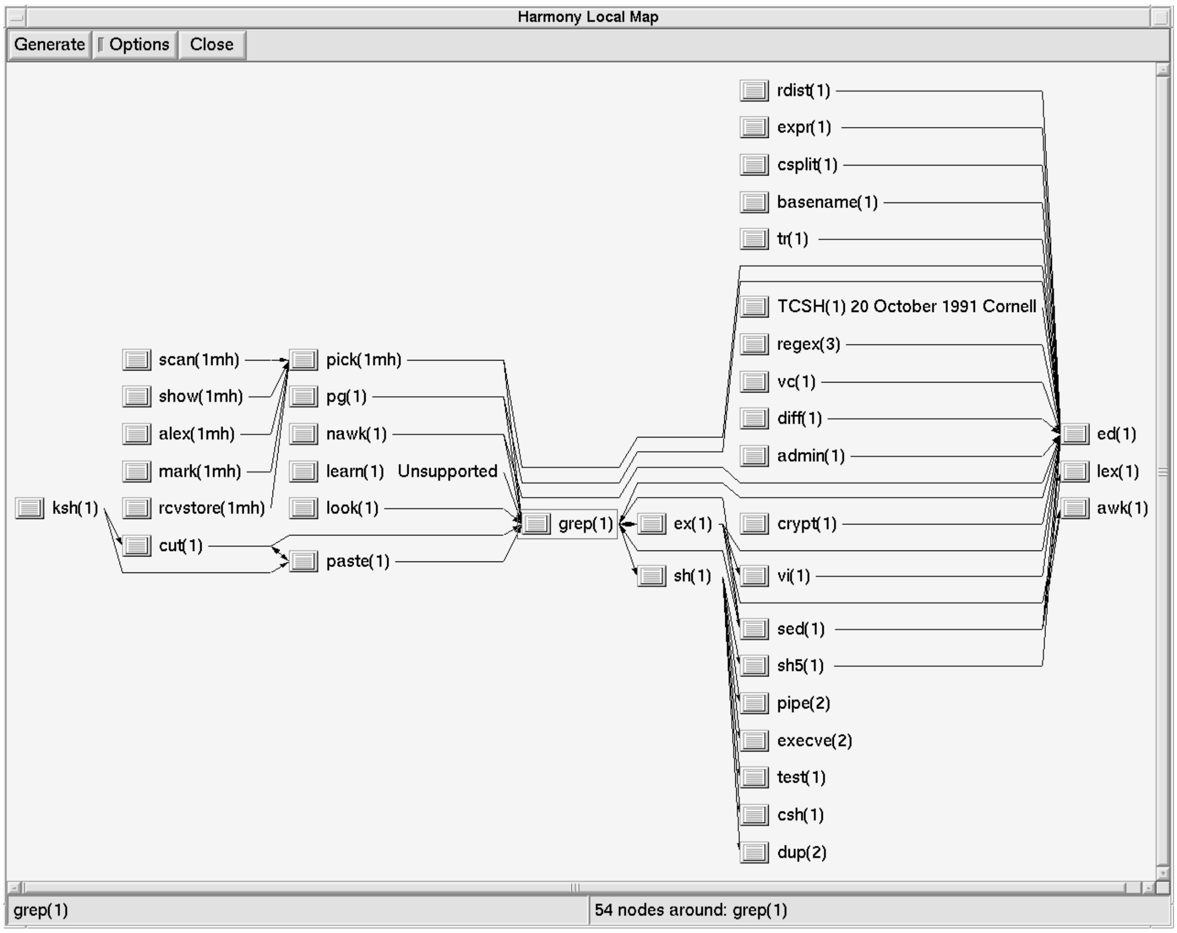


Figure 8.17 The Harmony Local Map, showing UNIX manual pages within two links of the `grep` manual page.

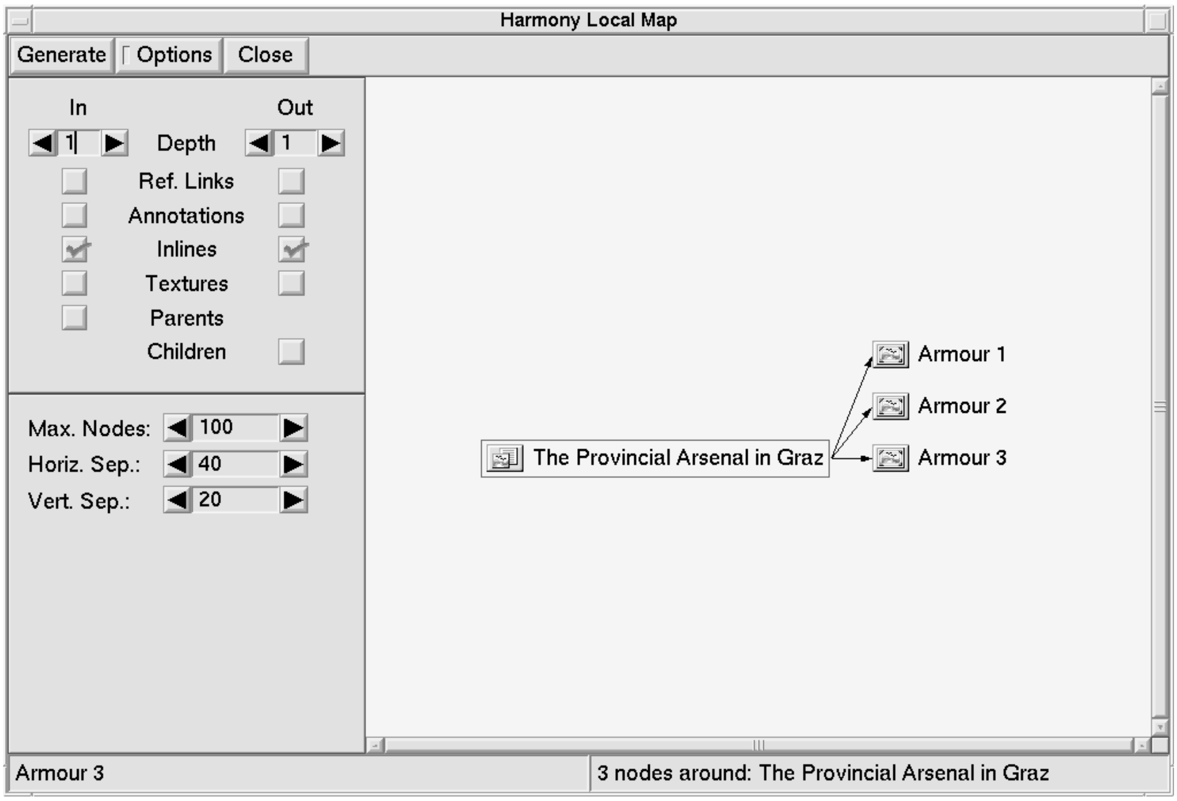
8.7 The Harmony Local Map

Link maps have been attempted in numerous hypermedia systems. The accepted wisdom is that it is near impossible to visualise the entire space of hyperlink relationships for more than a few dozen links, and that local views around a particular document are very much more manageable and practical to generate [Utting and Yankelovich, 1989].

Harmony's Local Map is a graphical visualisation of relationships between objects in Hyper-G. By default, the Local Map generates a visualisation of an object's immediate (referential) hyperlink relationships, i.e. other documents and objects one link away (both forward and backward) from the current object, as was shown in Figure 5.21. Hyper-G's links are stored in a separate link database and hence can be followed both forwards and backwards. Multiple levels of incoming and outgoing links can also be visualised, as can be seen in Figure 8.17.

Since links in Hyper-G are typed, the Local Map can filter links by type, for example: annotations, inline images, and textures. Furthermore, the Harmony Local Map is able to represent parent–child collection membership relationships, as an alternative representation to that of the collection browser. This is extremely useful when objects have more than one parent collection, as we shall see shortly.

In Hyper-G, hyperlinks always emanate from documents, but may point to either documents, collections or clusters. Since collections and clusters do not themselves have hyperlinks, the links of the collection head and the cluster members are visualised in the local map. Multiple links between the same objects are collapsed down to a single edge in the local map. The map can be regenerated



In		Out
1	Depth	1
□	Ref. Links	□
□	Annotations	□
☑	Inlines	☑
□	Textures	□
□	Parents	
	Children	□

Max. Nodes: 100
Horiz. Sep.: 40
Vert. Sep.: 20

Armour 3

3 nodes around: The Provincial Arsenal in Graz

Figure 8.18 Local map of inline image relationships.

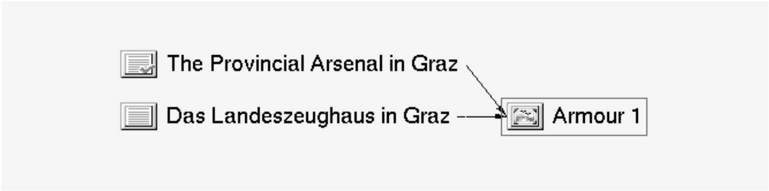


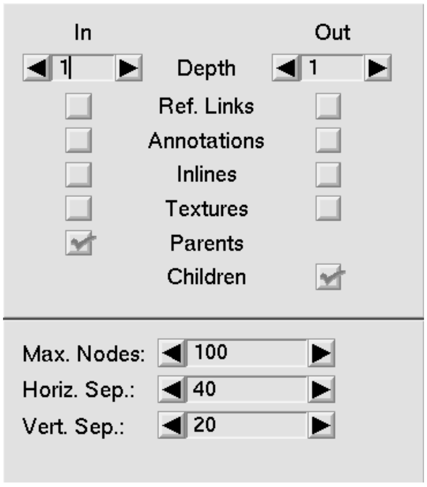
Figure 8.19 Using the local map to determine where an inline image is used.

around any object by selecting (single-left-clicking) the object and issuing a Generate command, providing an alternate form of link navigation.

Double-left-clicking an object activates that object, for example retrieving and displaying a document in the appropriate viewer. Single-clicking selects an object and activates *location feedback*, displaying the object's location within the collection structure in the Session Manager's collection browser (see Section 5.5).

The Local Map can be used to visualise other kinds of relationships as well as referential hyperlinks. Figure 8.18 shows a map of the inline images used in the cluster about the Graz Arsenal (see Figure 5.5). Imagine you were considering editing the image "Armour 1" or replacing the image with a newer photograph, but wanted to know whether "Armour 1" is used elsewhere. Selecting "Armour 1" and regenerating the Local Map shows the places where the image is used as an inline, in this case only in the English and German versions of the Arsenal text, as shown in Figure 8.19.

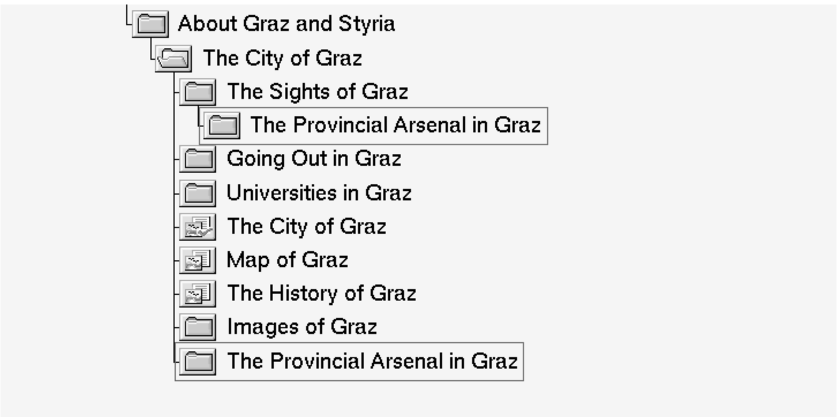
The Harmony Local Map has numerous options and settings, which can be configured in the Options panel. These include the number and kind of incoming and outgoing relationships to be visualised, the maximum number of objects in the map, and the spacing to be used in the layout.



In		Out
◀ 1 ▶	Depth	◀ 1 ▶
□	Ref. Links	□
□	Annotations	□
□	Inlines	□
□	Textures	□
☑	Parents	
	Children	☑

Max. Nodes: ◀ 100 ▶
Horiz. Sep.: ◀ 40 ▶
Vert. Sep.: ◀ 20 ▶

Figure 8.20 The Local Map Options Panel set for parent/child relationships.



- About Graz and Styria
  - The City of Graz
    - The Sights of Graz
      - The Provincial Arsenal in Graz
    - Going Out in Graz
    - Universities in Graz
    - The City of Graz
    - Map of Graz
    - The History of Graz
    - Images of Graz
    - The Provincial Arsenal in Graz

Figure 8.21 Multiple paths in the Collection Browser.

Figure 8.20 shows the Options panel with settings to display parent–child membership relationships.

Since Hyper-G collections, clusters and documents may belong to multiple parent collections, the collection browser's hierarchical display is sometimes confusing, as multiple paths have to be

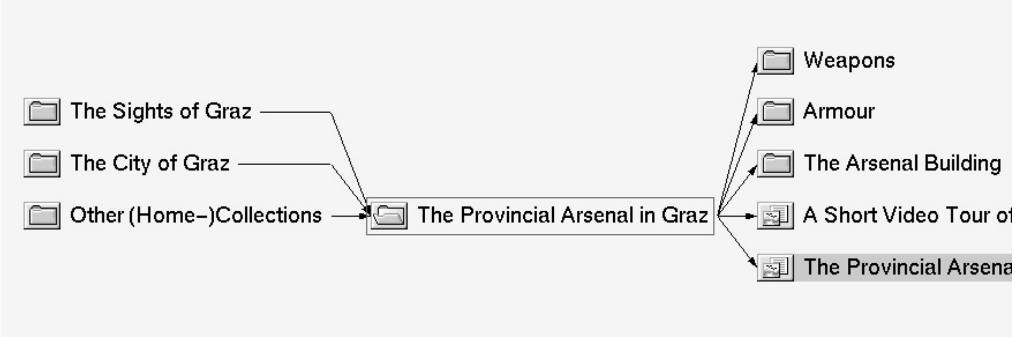


Figure 8.22 Local map of parent–child relationships for the collection “The Provincial Arsenal in Graz”.

opened up to objects which belong to multiple parents. For example, in Figure 8.21, the collection "The Provincial Arsenal in Graz" belongs to both the collection "the City of Graz" and to its child "The Sights of Graz." In such cases, the Local Map can be used effectively to display parent–child membership relationships, in this case generating the map shown in Figure 8.22.

As a further helpful aid, both the Harmony Session Manager and the various document viewers have in-built shortcuts for generating local maps. For example, the document viewers have menu items to generate a local map of References, Annotations, and Parents for the currently displayed document. The Harmony Text Viewer has a menu item to generate a map of the current text's inline images, the Harmony VRweb VRML viewer has menu shortcuts to generate maps of a model's inline models and texture images, and so on.

8.8 The Harmony Information Landscape

The Harmony Information Landscape is an interactive, three-dimensional visualisation of the Hyper-G collection structure, essentially a three-dimensional alternative to the traditional two-dimensional collection browser view shown in Figure 5.3. The Harmony Information Landscape is described in detail in [Eyl, 1995; Wolf, 1996][2]. As illustrated in Figure 8.23 and Figure 8.24, the collection hierarchy is mapped out onto a plane, similar to the approach used by FSN to visualise a file system (FSN is described in Section 8.4). Flat blocks or pedestals represent collections and clusters, simple abstract icons atop these represent individual documents. The standard default mapping of document attributes to visual representation is to encode document type through the shape of the corresponding 3d icon, document size as the size of the icon, and use colour either to reinforce the document type or to convey the age of documents.

Users can fly over the landscape looking for salient features. The mouse buttons are used to control the speed and direction of motion. Collections can be opened and closed and documents accessed through mouse clicks, menu commands, and keyboard shortcuts. As more collections are opened, the landscape fans out to accommodate them. The landscape's Overview Map (top right of Figure 8.24) provides both a sense of global context and orientation and an alternative form of navigation. Numerous settings can be configured interactively, including the colour coding used for collections and documents, the logarithmic scaling factor applied to the size of the icons, and the relative sensitivity of the mouse.

Much of the functionality of the Harmony Session Manager (see Section 5.3) is also available in the Information Landscape. Objects can be selected by single-left-clicking and activated by double-left-clicking. Middle-clicking a collection or document symbol flies the user gently to down to that object. Right-clicking brings up Harmony's attribute window to display the object's metadata.

The Information Landscape and the collection browser are synchronised, so that changes in one are automatically reflected in the other. Single-clicking an object in a search result list or in Harmony's Local Map activates location feedback in both the collection browser and the information landscape: the path(s) to the object are opened up in both, giving users visual feedback about the context of an object's use, its type, and its size, before any decision on whether to retrieve it. Location feedback in the landscape also results in the user being taken by the hand and gently flown to the selected object.

Unlike in FSN, where navigation is constrained to forward and backward motion, users are able to navigate freely within the landscape, and may turn around and look back to view the parents of a set of objects, as shown in Figure 8.25. The textual labels in the display are rotated and tilted as necessary to match the user's viewing position and angle.

A significant innovation of the Harmony Information Landscape is the *combined* display of both hierarchical structuring and hyperlink relationships in *one* visualisation, as depicted in Figure 8.26 for the image document "Map of Graz". Documents having links to this document are displayed beneath the ground plane. Documents or collections reachable from the image are displayed above the ground

[2] The Harmony Information Landscape was implemented by Martin Eyl and Peter Wolf, under my supervision, in the course of their Master's Theses.

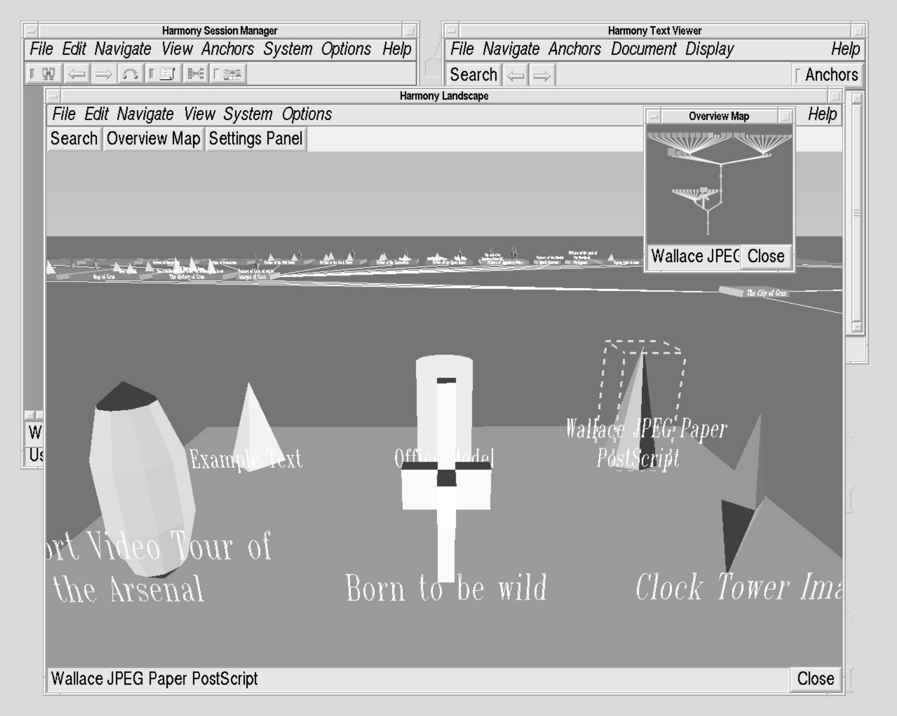

Harmony Session Manager
File Edit Navigate View Anchors System Options Help
Harmony Text Viewer
File Navigate Anchors Document Display Help
Search
Anchors
Harmony Landscape
File Edit Navigate View System Options
Search
Overview Map
Settings Panel
Overview Map
Help
Wallace JPEG
Close
The City of Graz
Example Text
Office Model
Wallace JPEG Paper PostScript
ort Video Tour of the Arsenal
Born to be wild
Clock Tower Ima
Wallace JPEG Paper PostScript
Close

Figure 8.23 The Harmony Information Landscape. Blocks on the plane represent collections and clusters. The default style represents various types of document with simple 3d icons.

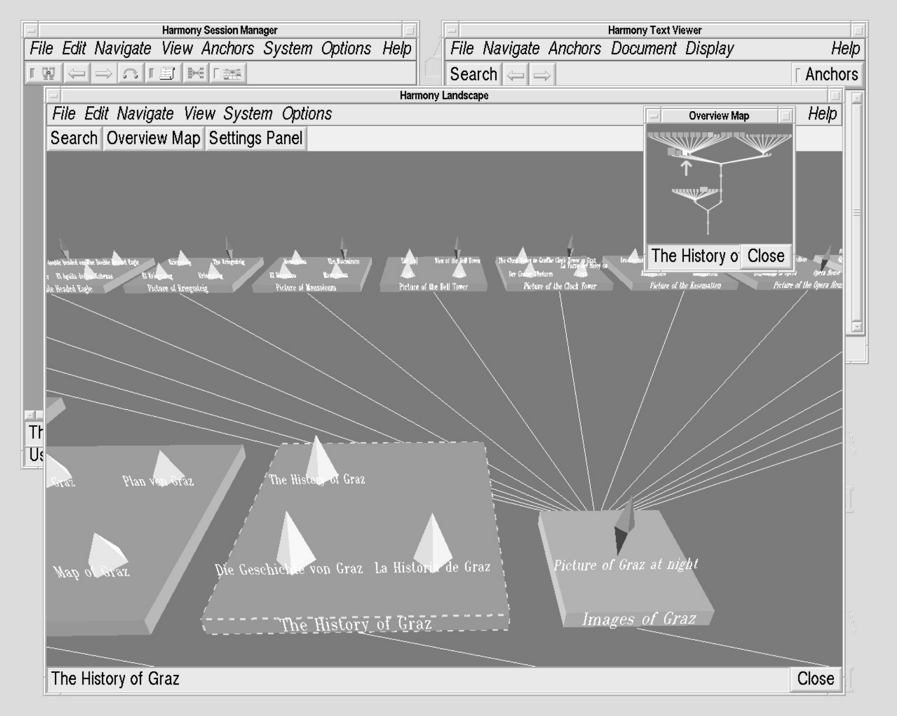


Figure 8.24 Document clusters containing text and image documents. Note the overview window in the top right, the arrow indicating the user's position and orientation.

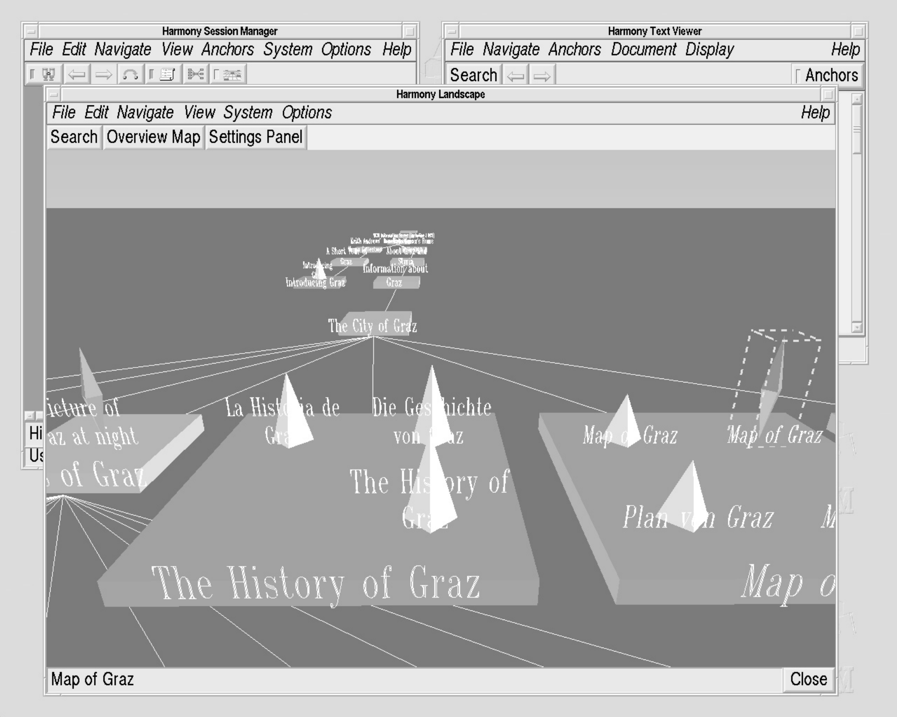


Figure 8.25 Looking back to view the parents of a collection.

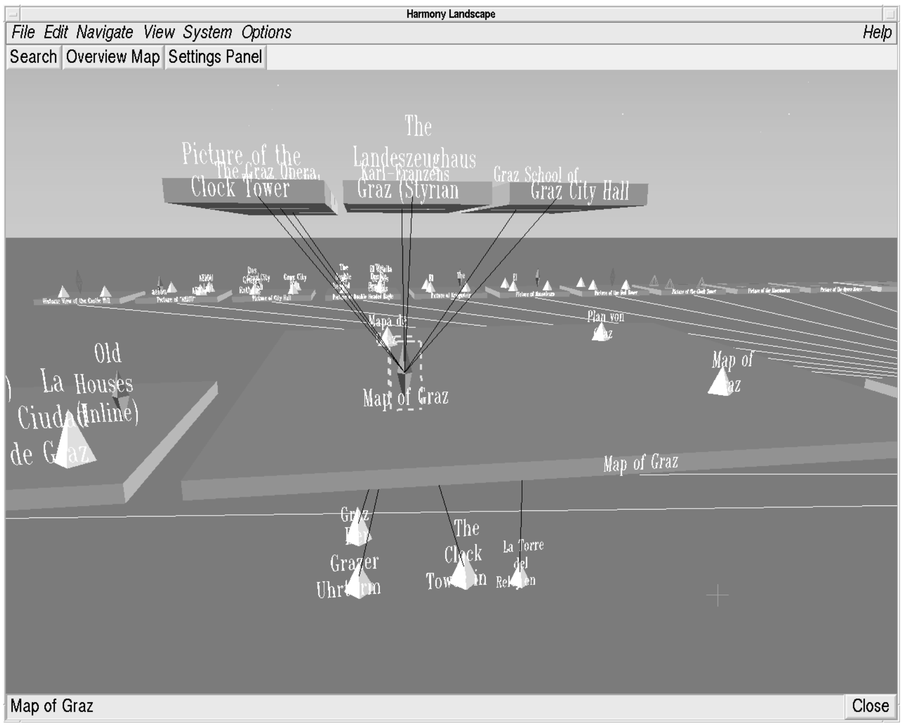


Figure 8.26 Superimposing hyperlink relationships upon the structure map. Links to and from the image "Map of Graz" have been selected for display. Documents having links to this image are displayed on the lower plane. Documents or collections reachable from the image are displayed in the upper plane.

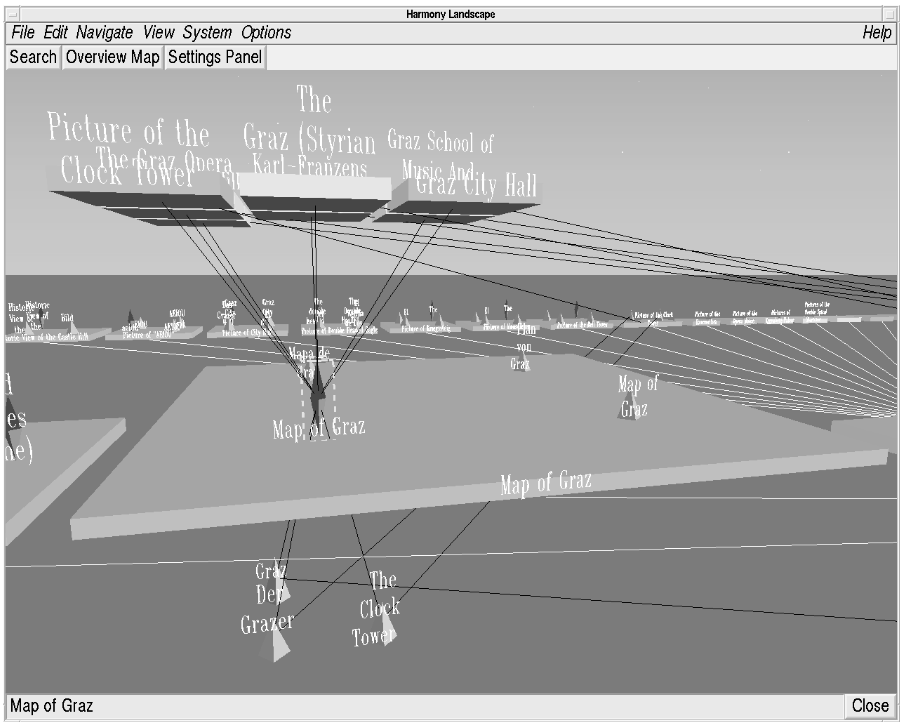


Figure 8.27 Arcs linking objects to their actual locations in the collection structure. Clicking an arc flies the user gently down the path to the appropriate location.

plane. This combined visualisation, rather like having a Harmony Local Map overlaid in the vertical plane orthogonal to the landscape, utilises all three available dimensions to compactly present both kinds of structure.

In analogy to Harmony's Local Map, multiple levels of incoming and outgoing links may be displyed and the map is not limited to standard referential hyperlinks, but can selectively show only inline images, model textures, annotations, etc. It can also provide an orthogonal view of multiple parents, in cases where objects belong to more than one parent collection.

To provide further contextual feedback, a control button shows where the linked documents (in the vertical plane) are actually located in the collection structure, by opening up paths to their position in the main plane and drawing arcs to them, as can be seen in Figure 8.27. Clicking such an arc flies the user gently down the path to the appropriate location.

To achieve good performance across a wide range of platforms, the landscape is flat-shaded and documents are represented by simple abstract icons by default. However, unlike FSN, the icons are configurable and numerous other styles are available. Figure 8.28 shows the 3d icon chooser, through which users can select icons to represent individual document types. Similar choosers are available to select colours, fonts, line widths, etc.

Figure 8.29 illustrates the use of textured information landscapes. Thumbnail images pasted to nearby document icons to give an indication of their content. At the moment, a generic image is used, but in future a specific thumbnail image could be generated from an image document, or from the first (or a representative) frame of a film clip. Explicitly assigned keywords could be placed next to or mapped onto a document's icon. For text documents, representative keywords could be automatically extracted from the document, possibly in conjunction with the server's full text indexing. Currently, real-time texturing is only available for high-end graphics workstations, but as graphics hardware with texture support becomes more widely available, this technology should also migrate down to desktop PCs.

As Kevin Lynch points out in his seminal work "The Image of the City" [Lynch, 1960], *land-*

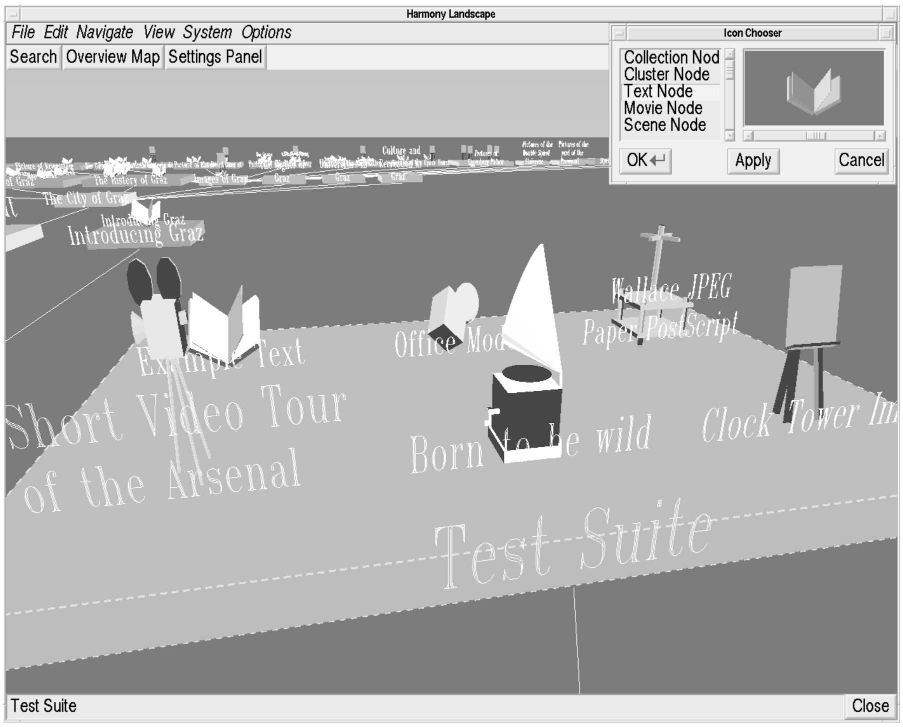


Figure 8.28 A geometrically more complicated set of document icons and the 3d icon chooser (top right).

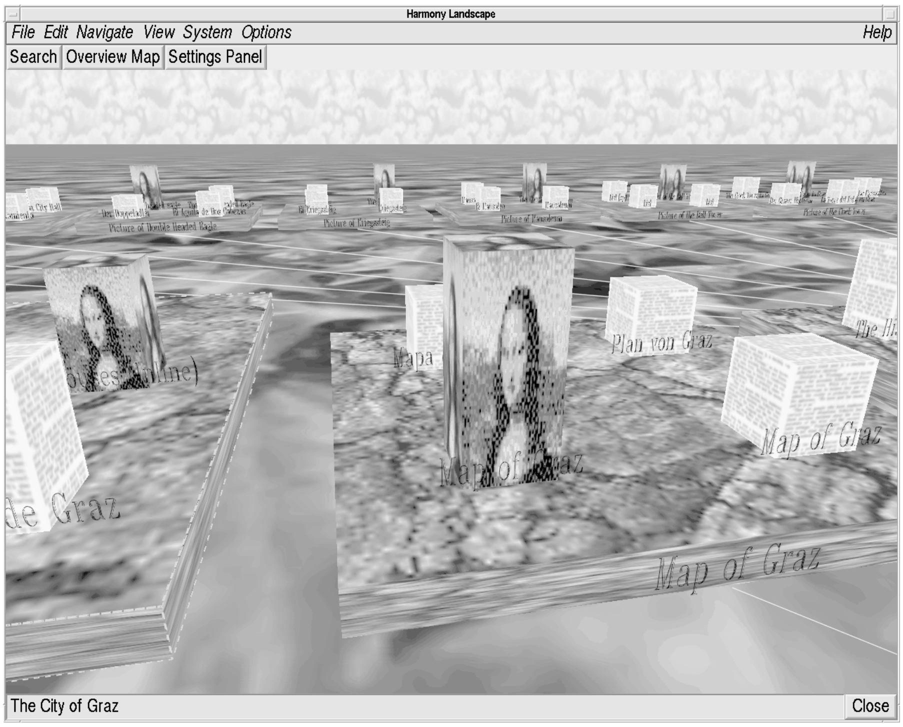


Figure 8.29 Textured information landscape. Thumbnail images pasted to nearby document icons give an indication of their content. “Landmark” 3d icons representing whole collections will be visible from afar.

marks play an important role when people navigate a city. A future extension to the Harmony Information Landscape will allow authors to assign a "landmark" 3d icon to a collection, representing the collection as a whole and visible from afar. For example a model of the Eiffel Tower might be assigned to represent a collection about Paris.

As with VRweb, Harmony's 3d viewer described in Chapter 7, the Information Landscape comes in two versions: a software-only version using the Mesa 3d library which runs on any UNIX platform under X Windows and an OpenGL version which can take advantage of OpenGL hardware acceleration, if available.

8.9 Further Reading

Two good overviews of the field of information visualisation are *Visualization's New Tack: Making Sense of Information* [Gershon and Eick, 1995b] and *Computer Graphics and Visualization in the Global Information Infrastructure* [Gershon and Brown, 1996].

The proceedings of the IEEE symposia on information visualisation [Gershon and Eick, 1995a; Gershon, 1996] are a source of detailed information on specific approaches.

An online starting point is Peter Young's survey of three-dimensional information visualisation [Young, 1996].

9 Outlook

There is a saying that "an Internet year is like a dog year": there are seven of them for each calendar year, such is the pace of innovation and development. In this chapter, I take a look at the direction of current developments in the fields of Internet information systems, the World Wide Web, and information visualisation, I describe work in progress which I am directing, and outline some ideas for future research.

9.1 General Trends

According to Jakob Nielsen's analysis [Nielsen, 1996], 1993 was the Web year of glorified home pages, 1994 the year of content, 1995 the year of *focussed* content, and 1996 is the year of community-building, as web sites struggle to retain their audience. What will be the main Web trends in 1997?

My guess is that 1997 will be the year of the applet. Java is well on the way to displacing C++ as *the* application programming language, and to establishing itself as *the* downloadable, Internet application (applet) language. Java development tools, compilers, and virtual machine implementations are progressing in leaps and bounds. Most Java applets around the Web today are of the tinker-toy variety, but I believe we will see more and more powerful applets, capable of real, useful work gracing web sites in 1997. Further ahead, Java "servlets", or agents uploaded to a web server, will be able to garner information on your behalf.

By autumn 1996 we will see a number of VRML 2.0 browser implementations become available for all the major platforms, cementing VRML as *the* 3d modeling language. We might even see some real, useful virtual worlds and applications modeled in VRML!

Microsoft is investing huge amounts of money in Internet products and development, and will continue to catch up on Netscape. I expect to see the market shares of Microsoft Internet Explorer and of Web servers running on Windows NT increase sharply. Apache will reinforce its position as the free server of choice, as NCSA winds down development of its httpd server. Web server development is moving fast from a research field to an engineering task. Check out the Netcraft Web Server Survey [Netcraft, 1996] for the latest server statistics.

The trend in web tools will be towards information management tools. As smaller sites, "powered by vi", graduate to become larger sites, webmasters cannot fail to realise the necessity for such tools (can they?).

As for the Internet itself, there are no signs of a slow-down in the increase in numbers of hosts and users. The new top-level domains [IANA, 1996] will relax pressure for `.com` domain names, but I expect IP address space will remain tight as IPng is slow in gaining acceptance.

Finally, I expect the emerging field of information visualisation to grow and crystalise further, especially since the Web is a vast information space just longing to be visualised.

9.2 Work in Progress

Of the work in progress which I am currently supervising, a suite of visualisation tools for Hyper-G, written as Java applets, is probably the most strategically important. Using Java, visualisation (and also editing, structuring, and management) tools can be delivered to end-users' Web browsers without them having to explicitly install any additional software. In contrast, to gain the benefits of the advanced navigation and visualisation tools in Harmony, users must first fetch and install the Harmony distribution package. To use Java visualisation applets, they must do nothing more than wait until the applets have been downloaded into the browser (which admittedly may take a non-negligible amount of time, depending on the size of the applet). Writing these tools in Java also means that they are platform-independent and can be incorporated easily into a (future) Hyper-G client application written in Java. The visualisation applets currently being implemented include:

- A collection browser for navigating and editing Hyper-G's collection structure.
- A local map to view the link and collection relationships around a specific object.
- A tree map view of the collection structure.
- A hyperbolic view of the collection structure.
- 2d and 3d views of search result sets, using clustering techniques from the field of information retrieval.

We are also working on a new version of the VRweb VRML browser supporting VRML 2.0. To support the scripting facilities in VRML 2.0, it makes most sense to support Java as the scripting language and incrementally re-write part or all of VRweb in Java:

(1) We are starting with the user interface code, since that is the least portable part of the C++ version.

(2) In parallel, we are writing a VRML 2.0 parser in Java (from scratch).

(3) Next, we will port the rest of the program logic to Java.

(4) Finally, once Java's 3d API is available, we will move the 3d output as well.

The (current) performance of Java is certainly a concern, and we may need to keep some native code for performance reasons, but the advantages of a VRML browser written entirely in Java, once Java's 3d API is ready and the interpreters and compilers are optimised, are persuasive:

- platform-independence,
- compilable as a native application for better performance, and
- downloadable as an applet when required.

As VRML models get larger and more ambitious and as people try to model whole cities, for example, information management of complex models is starting to become an issue. Here, I see an ideal application for Hyper-G's information management tools. By extracting texture links, inline model links, and referential hyperlinks from a VRML model as it is inserted into a Hyper-G server (in the same way as for HTML text documents) and placing them in Hyper-G's link database for management, large complex VRML models become much easier to handle. We are implementing exactly this facility at the moment.

Other work in progress includes:

- A home space builder and editing functionality for VRweb. Users can construct their own 3d workspace complete with desk, shelves, and filing cabinets, within which to organise pointers to resources.
- An improved search interface for Harmony and a visualisation of search result sets.
- An improved version of the Harmony Audio player, with full hyperlink support, and a graphical display.

- An instrumented version of Harmony which can log user interface actions for later analysis. We will ask a sample set of Harmony users to participate in an experiment and track their actions over an extended period. Among other things, I hope that the experimental data will provide objective evidence to help answer a question which has long interested me: do users prefer to navigate by searching, by following hyperlinks, or by traversing a (collection) hierarchy?

One of the next projects I intend to undertake is a 3d visualisation of the structure of Hyper-G servers based on generating VRML on the fly and/or using the promised Java3D API. The idea again is to bring the visualisation inside the Web browser.

9.3 Ideas for Future Work

The Harmony Information Landscape provides rich ground for further research and experimentation. As mentioned in Section 8.8, authors might be allowed to assign a “landmark” 3d icon to a collection, representing the collection as a whole and visible from afar. For example, a model of the Eiffel Tower might be assigned to represent a collection about Paris.

It would be desirable for users to be able to interactively configure the binding between information attributes and visualisation characteristics in the Harmony Information Landscape: for example mapping age or price, rather than size in bytes, to the size of the corresponding document icon.

FSN [Tesler and Strasnick, 1992] allows users to set spatial bookmarks, like stakes in the ground, to which users can return at will in a single operation. These are equivalent to viewpoints in VRML and should also probably be explored in the Harmony Information Landscape. An additional feature would be the recording of a “flight path” for later playback; a guided tour which one could also interrupt and resume at any point.

An untapped source of information are usage statistics. Currently, the Hyper-G server logs accesses in log files, but a minor modification to the server would enable a usage count to be kept for each document and collection. This in turn would allow a document or collection’s *popularity*, i.e. number of accesses, to become a further candidate for mapping to visualisation characteristics. This idea could also be extended to hyperlink and collection navigation – popular links and paths become the main roads of a visualisation, less popular links are shown as side roads.

One could also consider assigning fixed coordinates to collections and visualising them in 3d space, like having virtual real estate. This would allow users to incrementally build a mental map of places and locations and navigate by visual memory. However, real estate management becomes a hard issue as soon as fixed coordinates are introduced, especially for very large and very dynamic information spaces. A slightly different approach would be to allow users to group and position resources in a landscape themselves, in a kind of personal 3d workspace similar to the home space builder described in Section 9.2 above.

The Hyper-G server is session-oriented and hence tracks which documents or collections particular users are accessing. In other words, the Hyper-G server knows where users are and what they are doing at any point of time. A promising use of this would be to visualise (other) users visiting nearby collections in the information landscape, similar to the Internet Foyer [Benford *et al.*, 1996] described in Section 8.5. Visitors to the same collection could be allowed to communicate with one another, through chat windows or audio channels, enabling virtual meeting places.

Taking this idea further, I see research promise in the area of collaborative shared spaces across the Net, where virtual communities and project teams can meet and do work. These spaces might, but do not have to, have a three-dimensional rendering. As an application scenario, consider a virtual project meeting where physically distributed team members can see each other’s avatars, can talk to one another, and can see and work on shared objects and documents.

Or as another example, consider two (or more) market analysts meeting in a space like the Discovery Head Trader shown in Figure 8.4. One analyst turns a dial and says to the other something like, “Look, if I increase the investment in this branch, we can expect to see returns over here doubling...”. There are already a number of efforts in the direction of collaborative shared spaces, including NCSA’s Habanero project [NCSA, 1996a], Black Sun’s CyberHub [Black Sun, 1996], Worlds Inc.’s Active Worlds and Gamma [Worlds Inc., 1996], and the University of Bonn’s MRTspace [Fellner and Jucknath, 1996].

Finally, a possible enhancement to Hyper-G which could bring rich returns would be to add support for workflow processes. In the workplace environment, a document typically undergoes many revisions, reviews, and changes of status in the course of its life. Formalising support for this kind of process in Hyper-G would increase its attractiveness, particularly in the corporate intranet market.

10 Concluding Remarks

In this thesis I have presented my research work of the past five years in the field of navigation, construction, and visualisation of hypermedia on the Internet – the browsing, building, and beholding of cyberspace.

Having begun the thesis with an introduction to the Internet and its services and information systems, I gave an overview of hypermedia systems. I then described the Hyper-G second-generation Web server, which provides the supporting infrastructure and serves as the platform upon which to base my work.

Chapters 5, 6, 7, and 8 of the thesis described my original work, embodied in the Harmony client and authoring tool for Hyper-G, for which I was project leader until the summer of 1995. The scientific and technical contribution of this work includes advanced browsing and navigational facilities, remote management and authoring tools, the first 3d Internet hypermedia browser, and two innovative information visualisation tools.

The thesis concluded with an analysis of current trends, an outline of work currently in progress, and some ideas for future research.

Bibliography

Adobe (1996). *Adobe SiteMill*. `http://www.adobe.com/prodindex/sitemill/main.html`.

Adobe Systems Inc. (1990). *PostScript Language Reference Manual*, 2^{nd} edition. Addison-Wesley, Reading, Mass.

AltaVista (1996). *AltaVista*. `http://www.altavista.digital.com/`.

Andreessen, M. (1993). *NCSA Mosaic Technical Summary*. `ftp://ftp.ncsa.uiuc.edu/Mosaic/Papers/mosaic.ps.Z`.

Andrews, K. (1992). *Using 3D Scenes as Hypermedia Nodes*. Poster at the ACM European Conference on Hypertext (ECHT'92), Milan, Italy.

Andrews, K. (1993a). *Constructing Cyberspace: Virtual Reality and Hypermedia*. Presented at Virtual Reality Vienna '93. `ftp://ftp.iicm.edu/pub/Hyper-G/papers`.

Andrews, K. (1993b). *ECHT'92 Conference Report*. *SIGLINK Newsletter*, **2**(1).

Andrews, K. (1994). *Spatial Metaphors for Information Systems.* Workshop on Spatial User Interface Metaphors in Hypermedia Systems, ACM European Conference on Hypermedia Technology (ECHT'94), Edinburgh, Scotland. `http://www.gatech.edu/lcc/idt/Faculty/andreas_dieberger/ECHT94.WS.Andrews.html`.

Andrews, K. (1995a). *Visualising Cyberspace: Information Visualisation in the Harmony Internet Browser*. In *Proc. of First IEEE Symposium on Information Visualization (InfoVis'95)*, pages 97–104, Atlanta, Georgia.

Andrews, K. (1995b). *VRweb Project Announcement (Press Release).* `http://www.iicm.edu/vrweb-press-announce01.html`.

Andrews, K. (1996a). *Authoring with Harmony*. In H. Maurer, editor, *HyperWave: The Next Generation Web Solution*, Chapter 16, pages 345–376. Addison-Wesley. `http://www.iicm.edu/hgbook`.

Andrews, K. (1996b). *Hypermedia Systems*. In H. Maurer, editor, *HyperWave: The Next Generation Web Solution*, Chapter 3, pages 19–29. Addison-Wesley. `http://www.iicm.edu/hgbook`.

Andrews, K. (1996c). *HyperWave: The Next Generation Web Server*. HyperWave course slides. `http://www.iicm.edu/hgcourse`.

Andrews, K. (1996d). *Information Systems and the Internet*. In H. Maurer, editor, *HyperWave: The Next Generation Web Solution*, Chapter 2, pages 9–18. Addison-Wesley. `http://www.iicm.edu/hgbook`.

Andrews, K. (1996e). *Installing and Configuring Harmony*. In H. Maurer, editor, *HyperWave: The Next Generation Web Solution*, Chapter 16, pages 235–242. Addison-Wesley. `http://www.iicm.edu/hgbook`.

Andrews, K. (1996f). *Using Harmony*. In H. Maurer, editor, *HyperWave: The Next Generation Web Solution*, Chapter 15, pages 193–234. Addison-Wesley. `http://www.iicm.edu/hgbook`.

Andrews, K. (1996g). *Visualising the Internet: Where's the Meat?* Panel Presentation, CHI'96, Vancouver, Canada.

Andrews, K. and Kappe, F. (1993). *Strait-Jacketing Authors: User Interface Consistency in Large-Scale Hypermedia Systems*. In *Proc. Hypermedia '93, Zürich, Switzerland*, pages 130–137, Berlin. Springer.

Andrews, K. and Kappe, F. (1994). *Soaring Through Hyperspace: A Snapshot of Hyper-G and its Harmony Client.* In W. Herzner and F. Kappe, editors, *Proc. of Eurographics Symposium on*

Multimedia/Hypermedia in Open Distributed Environments, pages 181–191, Graz, Austria. Springer.

Andrews, K. and Kappe, F. (1995). *Using the Hyper-G Authoring Environment to Create Web Information*. In *Poster Proceedings, Third International World-Wide Web Conference, WWW'95, Darmstadt, Germany*, pages 99–102.

Andrews, K. and Pichler, M. (1994). *Hooking Up 3-Space: Three-Dimensional Models as Fully-Fledged Hypermedia Documents*. In *Proc. of East-West International Conference on Multimedia, Hypermedia, and Virtual Reality (MHVR'94)*, pages 11–18, Moscow, Russia.

Andrews, K. and Pichler, M. (1996). *Hooking Up 3-Space: Three-Dimensional Models as Fully-Fledged Hypermedia Documents*. In *Multimedia, Hypermedia, and Virtual Reality*, number 1077 in LNCS, pages 28–44. Springer.

Andrews, K., Kappe, F. and Schipflinger, J. (1994a). *Harmony: A Tool for Navigating Through Deep Hyperspace*. In *Proc. of ED-MEDIA 94,* page 627, Vancouver, Canada. AACE.

Andrews, K., Maurer, H. and Scherbakov, N. (1994b). *Object-Oriented Data Modeling in Hypermedia: A New Data Model*. In *Proc. of ED-MEDIA 94,* page 703, Vancouver, Canada. AACE. Poster and Demo.

Andrews, K., Nedoumov, A. and Scherbakov, N. (1995a). *Embedding Courseware Into Internet: Problems and Solutions*. In *Proc. of ED-MEDIA 95*, pages 75–80, Graz, Austria. AACE. `http://www.iicm.edu/edmedia_papers_ps`.

Andrews, K., Kappe, F. and Maurer, H. (1995b). *Hyper-G and Harmony: Towards the Next Generation of Networked Information Technology*. In *Formal Demonstration, CHI'95 Conference Companion*, pages 33–34, Denver, Colorado. `http://www.acm.org/sigchi/chi95/Electronic/documnts/demos/ka_bdy.htm`.

Andrews, K., Kappe, F. and Maurer, H. (1995c). *Hyper-G: Towards the Next Generation of Network Information Technology*. *Journal of Universal Computer Science*, **1**(4), 206–220. Special Issue: Selected Proceedings of the Workshop on Distributed Multimedia Systems, Graz, Austria, Nov. 1994. `http://www.iicm.edu/jucs`.

Andrews, K., Kappe, F. and Maurer, H. (1995d). *Hyper-G: Towards the Next Generation of Network Information Technology*. *Journal of Universal Computer Science*, **1**(4), 206–220. Special Issue: Selected Proceedings of the Workshop on Distributed Multimedia Systems, Graz, Austria, Nov. 1994. `http://www.iicm.edu/jucs`.

Andrews, K., Kappe, F., Maurer, H. and Schmaranz, K. (1995e). *On Second Generation Network Hypermedia Systems*. In *Proc. of ED-MEDIA 95*, pages 69–74, Graz, Austria. AACE. `http://www.iicm.edu/edmedia_papers_ps`.

Andrews, K., Maurer, H. and Scherbakov, N. (1995f). *Semantic Modeling of Object-Oriented Hypermedia Databases*. In *Proc. of Fifth International Conference on Human-Computer Interaction (EWHCI'95)*, pages 121–134, Moscow, Russia. Springer LNCS 1015.

Andrews, K., Kappe, F. and Maurer, H. (1995g). *Serving Information to the Web with Hyper-G*. *Computer Networks and ISDN Systems*, **27**(6), 919–926. Proc. Third International World-Wide Web Conference, WWW'95, Darmstadt, Germany. `http://www.igd.fhg.de/www/www95/proceedings/papers/105/hgw3.html`.

Andrews, K., Pichler, M. and Wolf, P. (1996). *Towards Rich Information Landscapes for Visualising Structured Web Spaces*. In *Proc. of 2nd IEEE Symposium on Information Visualization (InfoVis'96),* San Francisco, CA.

Anklesaria, F., Lindner, P., McCahill, M., Torrey, D., Johnson, D. and Alberti, B. (1993). *Gopher+: Upward Compatible Enhancements to the Internet Gopher Protocol*. `ftp://boombox.micro.umn.edu/pub/gopher/gopher_protocol/Gopher+/Gopher+.txt`.

Apache (1996). *The Apache HTTP Server Project*. `http://www.apache.org/`.

Asymetrix Corporation (1989). *Using ToolBook. A Guide to Building and Working with Books.* Asymetrix Corporation, Bellevue, Washington.

Aubrey, D. (1996). *Hyper-G, the new networking hypermedia system, adds new depth to Web* *searches*. *Computer Shopper*, pages 579–588. `http://www.zdnet.com/cshopper/`.

Begeman, M. L. and Conklin, J. (1988). *The Right Tool for the Job*. *Byte*, **13**(10), 255–266.

Bell, G., Parisi, A. and Pesce, M. (1995). *The Virtual Reality Modeling Language – Version 1.0 Specification*. `http://vrml.wired.com/vrml.tech/vrml10-3.html`.

Bell, G., Parisi, A. and Pesce, M. (1996a). *The Virtual Reality Modeling Language – Version 1.0C Specification*. `http://vag.vrml.org/vrml10c.html`.

Bell, G., Carey, R. and Marrin, C. (1996b). *The Virtual Reality Modeling Language Specification, Version 2.0*. `http://vag.vrml.org/VRML2.0/FINAL/`.

Benford, S., Brown, C., Reynard, G. and Greenhalgh, C. (1996). *The Internet Foyer*. University of Nottingham. `http://www.crg.cs.nott.ac.uk/Foyer/foyer.html`.

Berk, E. and Devlin, J., editors (1991). *Hypertext/Hypermedia Handbook*. Software Engineering Series. McGraw-Hill, New York.

Berners-Lee, T. and Conolly, D. (1993). *HyperText Markup Language (HTML)*. `http://www.w3.org/pub/WWW/Markup/Markup.html`.

Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H. F. and Secret, A. (1994). *The World-Wide Web*. *Communications of the ACM*, **37**(8), 76–82.

Black Sun (1996). *CyberHub*. Black Sun Interactive, Inc. `http://www.blacksun.com/`.

Bradley, J. (1996). *XV*. `ftp://ftp.cis.upenn.edu/pub/xv/`.

Bray, T. (1996). *Measuring the Web*. *Computer Networks and ISDN Systems*, **28**(7–11), 993–1005. Proc. Fifth International World-Wide Web Conference, WWW5, Paris, France. `http://www5conf.inria.fr/fich_html/papers/P9/Overview.html`.

Brown, P. J. (1992). *UNIX Guide: lessons from ten years' development*. In *Proc. ECHT'92*, pages 63–70, Milan, Italy. ACM Press.

Buford-Koegel, J., editor (1994). *Multimedia Systems*. ACM Press.

Bush, V. (1945). *As We May Think*. *The Atlantic Monthly*, **176**(1), 101–108. Reprinted in "CD-ROM — The New Papyrus", Microsoft Press, 1986. `http://www.isg.sfu.ca/~duchier/misc/vbush/vbush.shtml`.

Cailliau, R. (1995). *About WWW*. *Journal of Universal Computer Science*, **1**(4), 221–231. Special Issue: Selected Proceedings of the Workshop on Distributed Multimedia Systems, Graz, Austria, Nov. 1994. `http://www.iicm.edu/jucs`.

Canter, D., Rivers, R. and Storrs, G. (1985). *Characterizing User Navigation Through Complex Data Structures*. *Behaviour and Information Technology*, **4**(2), 93–102.

Card, S. K., Robertson, G. G. and Mackinlay, J. D. (1991). *The Information Visualiser, An Information Workspace*. In *Proc. CHI'91: Human Factors in Computing Systems*, pages 181–188, New Orleans, Louisiana. ACM.

Card, S. K., Robertson, G. G. and York, W. (1996). *The WebBook and the Web Forager: An Information Workspace for the World Wide Web*. In *Proc. CHI'96: Human Factors in Computing Systems*, pages 111–117, Vancouver, Canada. ACM.

CHI96 (1996). *Visualising the Internet: Putting the User in the Driver's Seat*. In *Panel, CHI'96 Conference Companion*, pages 163–164, Vancouver, Canada.

CNIDR (1994). *freeWAIS 0.3*. Clearinghouse for Networked Information Discovery and Retrieval (CNIDR). `ftp://ftp.cnidr.org/pub/NIDR.tools/freeWAIS-0.3.tar.gz`.

Conklin, J. (1987). *Hypertext: An Introduction and Survey*. *IEEE Computer*, **20**(9), 17–41.

Conklin, J. and Begeman, M. L. (1988). *gIBIS: A Hypertext Tool for Exploratory Policy Discussion*. *ACM Transactions on Office Information Systems*, **6**(4), 303–331.

Dalitz, W. and Heyer, G. (1996). *HyperWave: The New Generation Internet Information System Based on Hyper-G Technology*. dpunkt. `http://www.dpunkt.de/produkte/Hyperwave.html`.

December, J. (1996). *Internet Tools Summary*. `http://www.rpi.edu/Internet/`

Guides/decemj/itools/internet-tools.html.

Dewan, P. (1993). *A Survey of Applications of CSCW Including Some in Educational Settings*. In *Proc. ED-MEDIA 93, Orlando, Florida*, pages 147–152, Charlottesville, VA. AACE.

DIVE (1996). *The Distributed Interactive Virtual Environment*. Swedish Institute of Computer Science (SICS). http://www.sics.se/dive/.

Domik, G. (1991). *The Role of Visualization in Understanding Data*. In H. Maurer, editor, *New Results and New Trends in Computer Science*, pages 91–107, Graz, Austria. Springer LNCS 555.

Duncan, R. (1990). *An introduction to building applications with ToolBook*. *PC Magazine*, **9**(21), 493–497.

Edwards, D. M. and Hardman, L. (1989). *Lost in Hyperspace: Cognitive Mapping and Navigation in a Hypertext Environment*. In R. McAleese, editor, *Hypertext: Theory into Practice*, pages 105–125. Blackwell Scientific Publications, Ablex.

EFF (1996). *Internet Guide*. Available from Electronic Frontier Foundation at http://www.eff.org or http://www.iicm.edu/internet_guide.

Emtage, A. and Deutsch, P. (1992). *Archie – An Electronic Directory Service for the Internet*. In *USENIX Winter 1992 Technical Conference Proceedings*, pages 93–110, San Francisco, CA. USENIX Association.

Engelbart, D. C. (1988). *The Augmented Knowledge Workshop*. In A. Goldberg, editor, *A History of Personal Workstations*, pages 187–236. ACM Press.

Eubanks, C., McKeon, R., Nadeau, D. and Moreland, J. (1996). *VRML Repository*. San Diego Supercomputer Center. http://www.sdsc.edu/vrml/.

Eyl, M. (1995). *The Harmony Information Landscape: Interactive, Three-Dimensional Navigation Through an Information Space*. Master's thesis, Graz University of Technology, Austria. ftp://ftp.iicm.edu/pub/Hyper-G/doc/eyl.ps.gz.

Fairchild, K. M., Poltrock, S. E. and Furnas, G. W. (1988). *SemNet: Three-Dimensional Representations of Large Knowledge Bases*. In R. Guindon, editor, *Cognitive Science and its Applications for Human-Computer Interaction*, pages 201–233. Lawrence Erlbaum, Hillsdale, New Jersey.

Fairchild, K. M., Serra, L., Hern, N., Hai, L. B. and Leong, A. T. (1993). *Dynamic FishEye Information Visualizations*. In R. A. Earnshaw, M. A. Gigante, and H. Jones, editors, *Virtual Reality Systems*, pages 161–177. Academic Press.

Fellner, D. (1996). *3D-CGI*. http://hyperg.cs.uni-bonn.de/CompGraph.ResearchProjects.CGI-3D.

Fellner, D. W. and Jucknath, O. (1996). *MRTspace: Multi-User 3D Environments using VRML*. In *Proc. WebNet'96, San Francisco, California*.

Filippini, L. (1996). *Moving Picture Expert Group MPEG Home Page*. http://www.crs4.it/HTML/LUIGI/MPEG/.

Flohr, U. (1995). *Hyper-G Organizes the Web*. *BYTE*, pages 59–64. http://www.byte.com/art/9511/sec5/art4.htm.

Flynn, P. (1995). *The World Wide Web Handbook*. International Thomson Computer Press, London.

Foley, J. D., van Dam, A., Feiner, S. K. and Hughes, J. F. (1990). *Computer Graphics: Principles and Practice*, second edition. Addison-Wesley, Reading, Massachusetts.

Foss, C. (1988). *Effective Browsing in Hypertext Systems*. In *Proc. RIAO'88*, pages 82–98, Cambridge, Massachusetts. MIT.

Fulton, J. and Renda, G. (1994). *The Network Audio System*. ftp://ftp.x.org/contrib/audio/nas/xcon94paper.ps.gz.

Gershon, N., editor (1996). *Proc. of Second IEEE Symposium on Information Visualization (InfoVis'96)*, San Francisco, California. IEEE Computer Society Press.

Gershon, N. and Brown, J. R. (1996). *Computer Graphics and Visualization in the Global Information Infrastructure*. *IEEE Computer Graphics and Applications*, **16**(2), 60–75.

Gershon, N. and Eick, S. G., editors (1995a). *Proc. of First IEEE Symposium on Information Visualization (InfoVis'95)*, Atlanta, Georgia. IEEE Computer Society Press.

Gershon, N. and Eick, S. G. (1995b). *Visualization's New Tack: Making Sense of Information*. *IEEE Spectrum*, pages 38–56.

Gibson, W. (1984). *Neuromancer*. ACE Books, Berkley Publishing Group, New York.

Goodman, D. (1987). *The Complete HyperCard Handbook*. Bantam, New York.

Haan, B. J., Kahn, P., Riley, V. A., Coombs, J. H. and Meyrowitz, N. K. (1992). *IRIS Hypermedia Services*. *Communications of the ACM*, **35**(1), 36–51.

Halasz, F. G. (1988). *Reflections on NoteCards: Seven Issues for the Next Generation of Hypermedia Systems*. *Communications of the ACM*, **31**(7), 836–852.

Hall, W., Davis, H. and Hutchings, G. (1996). *Rethinking Hypermedia: The Microcosm Approach*. Kluwer.

Hill, F. S. (1990). *Computer Graphics*. MacMillan.

Iacovou, N. and McCahill, M. P. (1995). *GODOT: GopherVR Organized Directories of Titles*. In *Proc. of CIKM '95 Workshop on New Paradigms in Information Visualization and Manipulation*, Baltimore, Maryland. ACM. `http://www.cs.umbc.edu/˜cikm/1995/npiv/`.

IANA (1996). *New Registries and Domains*. `http://www.iana.org/iana/registries.html`.

INRIA/GRIF (1996). *Symposia*. `http://symposia.inria.fr/`.

ISOC (1996). *Internet Society*. `http://info.isoc.org/`.

JavaSoft (1996). *JavaSoft*. `http://java.sun.com/`.

Johnson, B. and Shneiderman, B. (1991). *Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures*. In *Proc. IEEE Visualization '91, San Diego, California*, pages 284–291, Los Alamitos, California. IEEE Computer Society Press.

Kahle, B., Morris, H., Davis, F., Tiene, K., Hart, C. and Palmer, R. (1992). *Wide Area Information Servers: An Executive Information System for Unstructured Files*. *Electronic Networking: Research, Applications and Policy*, **2**(1), 59–68.

Kahn, P. (1996). *MAPA Mapping Web Sites*. Dynamic Diagrams. `http://DynamicDiagrams.com/services_map_main.html`.

Kappe, F. (1991). *Aspects of a Modern Multi-Media Information System*. Ph.D. thesis, Graz University of Technology, Austria. `ftp://ftp.iicm.edu/pub/Hyper-G/papers/report308.ps.gz`.

Kappe, F. (1993). *Hyper-G: A Distributed Hypermedia System*. In B. Leiner, editor, *Proc. INET '93, San Francisco, California*, pages DCC–1–DCC–9. Internet Society.

Kappe, F. (1995a). *Hyper-G Interchange Format (HIF): Version 1.0*. `http://www.iicm.edu/hif`.

Kappe, F. (1995b). *Hyper-G Text Format (HTF): Version 2.12*. Technical report, IICM, Graz University of Technology. `ftp://ftp.iicm.edu/pub/Hyper-G/papers/htf-212.ps.gz`.

Kappe, F. (1995c). *A Scalable Architecture for Maintaining Referential Integrity in Distributed Information Systems*. *Journal of Universal Computer Science*, **1**(2), 84–104. Available at `http://www.iicm.edu/jucs`.

Kappe, F. and Maurer, H. (1993). *From Hypertext to Active Communication/Information Systems*. IIG Report 363, IIG, Graz University of Technology, Austria. `ftp://ftp.iicm.edu/pub/Hyper-G/doc/`.

Kappe, F. and Pani, G. (1994). *Hyper-G Client/Server Protocol (HG-CSP): Version 7.05*. Technical report, IICM, Graz University of Technology. `ftp://ftp.iicm.edu/`

pub/Hyper-G/papers/HG-CSP-7.05.ps.gz.

Kappe, F., Andrews, K., Faschingbauer, J., Gaisbauer, M., Pichler, M. and Schipflinger, J. (1994). *Hyper-G: A New Tool for Distributed Hypermedia*. In *Proc. Distributed Multimedia Systems and Applications*, pages 209–214, Honolulu, Hawaii. IASTED/ISSM, ACTA Press, ISBN: 0-88986-194-3.

Krol, E. (1994). *The Whole Internet: User's Guide and Catalog*, second edition. O'Reilly & Associates.

Laky, G. (1993). *Usability Evaluation ausgesuchter Interface Parameter eines grošen Hypermediasystems anhand eines ToolBook Prototyps*. Master's thesis, Graz University of Technology, A-8010 Graz, Austria. In German.

Lamping, J., Rao, R. and Pirolli, P. (1995). *A Focus+Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies*. In *Proc. CHI'95, Denver, Colorado*, pages 401–408. ACM.

Leberl, F., Andrews, K. and Helmut Mülner (1993). *Graphical, Visual, and Immersive Interaction with Geographic Data*. In *Proc. UDMS'93, Vienna, Austria*, pages 1–10, A-1010 Vienna, Austria. ADV.

Levitt, J. (1996). *Managing Your Web Site*. *InformationWeek*. http://techweb.cmp.com/iw/567/67olweb.htm.

Lightscape (1996). *Lightscape Visualization System*. Lightscape Technologies, Inc. http://www.lightscape.com/.

Linden, G. (1996). *WebView: A new way to search and view pages on the World Wide Web!* University of Washington. http://www.cs.washington.edu/homes/glinden/WebView/WebView.html.

Lindner, P. (1995). *Using Gopher with the World-Wide-Web*. gopher://boombox.micro.umn.edu/00/gopher/Gopher_Conference_95/Papers/WebbedGopher.

Linton, M. A., Vlissides, J. M. and Calder, P. R. (1989). *Composing User Interfaces with InterViews*. *IEEE Computer*, **22**(2), 8–22.

Lippman, A. (1980). *Movie-Maps: An Application of the Optical Videodisc to Computer Graphics*. *Computer Graphics*, **14**(3), 32–42.

Lougher, P. (1996). *mpegUtil*. http://www.comp.lancs.ac.uk/computing/users/phillip/mpegUtil.html.

Lycos (1996). *Lycos, Inc.* http://www.lycos.com/.

Lynch, D. C. and Rose, M. T., editors (1993). *Internet System Handbook*. Addison-Wesley.

Lynch, K. (1960). *The Image of the City*. MIT Press, Cambridge, Massachusetts.

Mackinlay, J. D. (1986). *Automating the Design of Graphical Presentations of Relational Information*. *ACM Transactions on Graphics*, **5**(2), 110–141.

Mackinlay, J. D., Card, S. K. and Robertson, G. G. (1990). *Rapid Controlled Movement Through a Virtual 3D Workspace*. In *Proc. SIGGRAPH '90, Dallas, Texas*, pages 171–176, New York. ACM.

Mackinlay, J. D., Robertson, G. G. and Card, S. K. (1991). *The Perspective Wall: Detail and Context Smoothly Integrated*. In *Proc. CHI'91, New Orleans, Louisiana*, pages 173–179, New York. ACM.

Maurer, H., editor (1996). *HyperWave: The Next Generation Web Solution*. Addison-Wesley. http://www.iicm.edu/hgbook.

Maurer, H. and Scherbakov, N. (1996). *Multimedia Authoring for Presentation and Education. The Official Guide to HM Card*. Addison-Wesley.

Maurer, H. and Schmaranz, K. (1994). *J.UCS — The Next Generation in Electronic Journal Publishing*. *Computer Networks for Research in Europe*, **26**, S63–S69. Supplement to Vol. 26 of Computer Networks and ISDN Systems.

Maurer, H., Philpott, A. and Sherbakov, N. (1994a). *Hypermedia Systems without Links*. *Journal of Microcomputer Applications*, **17**(4).

Maurer, H., Scherbakov, N., Andrews, K. and Parthasarathy, S. (1994b). *Object-oriented modelling of hyperstructure: overcoming the static link deficiency*. *Information and Software Technology*, **36**(6), 315–322.

Maurer, H., Scherbakov, N. and Andrews, K. (1996). *Browsing Hypermedia Composites: An Algebraic Approach*. In *Proc. of WebNet'96*, San Francisco, CA.

Mayrhofer, V. and Andrews, K. (1996). *Harmony User Guide: Version 1.4*. `ftp://ftp.iicm.edu/pub/Hyper-G/papers/hug.ps.gz`.

McAleese, R. (1989). *Navigation and Browsing in Hypertext*. In R. McAleese, editor, *Hypertext: Theory into Practice*, pages 6–44. Blackwell Scientific Publications, Ablex.

McCahill, M. P. and Anklesaria, F. X. (1995). *Evolution of Internet Gopher*. *Journal of Universal Computer Science*, **1**(4), 235–246. Selected Proceedings of the Workshop on Distributed Multimedia Systems, Graz, Austria, Nov. 1994. `http://www.iicm.edu/jucs`.

McCahill, M. P. and Erickson, T. (1995). *Design for a 3D Spatial User Interface for Internet Gopher*. In *Proc. of ED-MEDIA 95*, pages 39–44, Graz, Austria. AACE. `http://www.iicm.edu/edmedia_papers_ps`.

Meckler (1996). *Mecklermedia Internet World*. `http://www.iw.com/`.

Medienhaus Styria (1996). *Kleine Online*. `http://www.styria.co.at/kleine`.

Memex (1996). *Memex and Beyond Web Site*. `http://www.cs.brown.edu/memex/home.html`.

Merit NIC (1996). *NSFNET Backbone Statistics*. `ftp://nic.merit.edu/nsfnet/statistics`.

Microsoft (1996). *Internet Information Server*. `http://www.microsoft.com/industry/sap/iis/iis.htm`.

Microsoft (1996). *Microsoft FrontPage*. `http://www.microsoft.com/msoffice/frontpage/`.

Microsoft (1996). *Microsoft Internet Explorer*. `http://www.microsoft.com/ie/ie.htm`.

Microsoft (1995). *Microsoft Unveils Wide-Ranging Commitment to Internet*. `http://www.microsoft.com/corpinfo/press/1995/95dec/iswpr4.htm`.

Mukherjea, S. and Foley, J. D. (1995). *Visualizing the World-Wide Web with the Navigational View Builder*. *Computer Networks and ISDN Systems*, **27**(6), 1075–1087. Proc. Third International World-Wide Web Conference, WWW'95, Darmstadt, Germany. `http://www.igd.fhg.de/www/www95/proceedings/papers/44/mukh/mukh.html`.

Murray, J. D. and van Ryper, W. (1996). *Encyclopedia of Graphics File Formats*, second edition. O'Reilly & Associates.

Musciano, C. and Kennedy, B. (1996). *HTML: The Definitive Guide*. O'Reilly & Associates.

NCSA (1996a). *NCSA Habanero Project Overview*. `http://www.ncsa.uiuc.edu/SDG/Software/Habanero/`.

NCSA (1996b). *The NCSA HTTPd Home Page*. `http://hoohoo.ncsa.uiuc.edu/`.

Neider, J., Davis, T. and Woo, M. (1993). *OpenGL Programming Guide*. Addison-Wesley.

Nelson, T. H. (1965). *A File Structure for the Complex, the Changing, and the Indeterminate*. In *Proc.* 20th *ACM National Conference*, pages 84–100.

Nelson, T. H. (1987). *Computer Lib/Dream Machines*, revised edition. Tempus/Microsoft Press.

Nelson, T. H. (1993). *Literary Machines*, 93.1 edition. Mindful Press, 3020 Bridgeway Suite 295, Sausalito CA 94965. Available from `http://www.eastgate.com/products/Cat_Books.html`.

Netcraft (1996). *The Netcraft Web Server Survey*. `http://www.netcraft.co.uk/Survey/`.

Netscape (1996). *Netscape Communications Corporation*. `http://home.netscape.com/`.

Netscape (1996). *Netscape LiveWire*. `http://home.netscape.com/comprod/products/tools/livewire_datasheet.html`.

Netscape (1996). *Netscape Navigator*. `http://home.netscape.com/comprod/products/navigator/`.

Network Wizards (1996). *Internet Domain Survey*. `http://www.nw.com/zone/WWW/top.html`.

Nielsen, J. (1995). *Multimedia and Hypertext: The Internet and Beyond*. Academic Press, San Diego, California.

Nielsen, J. (1996). *Relationships on the Web*. The Alert Box. `http://www.sun.com/-960101/columns/alertbox/`.

Niemeyer, P. and Peck, J. (1996). *Exploring Java*. O'Reilly & Associates. `http://www.ora.com/catalog/expjava/`.

OpenGL Architecture Review Board (1992). *OpenGL Reference Manual*. Addison-Wesley.

Orasche, G., Pichler, M. and Andrews, K. (1995). *Viewing VRML Scenes with VRweb*. In D. Fellner, editor, *Modeling – Virtual Worlds – Distributed Graphics, Proc. of MVD'95*, pages 179–189, Bonn, Germany. infix, D-53757 Sankt Augustin. `http://hyperg.cs.uni-bonn.de/GI411.events.contrib.ws1195.Orasche`.

Paul, B. (1996). *The Mesa 3-D Graphics Library*. `http://www.ssec.wisc.edu/~brianp/Mesa.html`, Space Science and Engineering Center, University of Wisconsin.

Pesce, M. (1995). *VRML: Browsing and Building Cyberspace*. New Riders/Macmillan.

Pichler, M. (1993). *Interactive Browsing of 3D Scenes in Hypermedia: The Hyper-G 3D Viewer*. Master's thesis, Graz University of Technology, Austria. `ftp://ftp.iicm.edu/pub/Hyper-G/doc/pichler1.ps.gz`.

Pichler, M., Orasche, G., Andrews, K., Grossman, E. and McCahill, M. (1995). *VRweb: A Multi-System VRML Viewer*. In *Proc. First Annual Symposium on the Virtual Reality Modeling Language (VRML '95)*, pages 77–85, San Diego, California.

PLACE (1996). *The PLACE Language*. `http://www.iicm.edu/wm_doc/place`.

Planet9 (1996). *VirtualSOMA*. Planet 9 Studios. `http://www.hyperion.com/planet9/vrsoma.htm`.

Poskanzer, J. (1996). *PBMPLUS: Portable Bitmap Utilities*. `ftp://ftp.x.org/`.

Quittner, J. (1994). *Billions Registered*. *Wired*, **2**(10). `http://www.hotwired.com/wired/2.10/departments/electrosphere/mcdonalds.html`.

Raggett, D. (1994). *Extending WWW to Support Platform Independent Virtual Reality*. In *Proc. INET'94*, pages 242–1–6, Prague. Internet Society. `http://info.isoc.org/isoc/whatis/conferences/inet/94/proceedings/242.ps`.

Reisinger, B. and Faschingbauer, J. (1996). *The Hyper-G Gateway Interface (HGI)*. `http://www.iicm.edu/HGI`.

Robertson, G. G., Mackinlay, J. D. and Card, S. K. (1991). *Cone Trees: Animated 3D Visualizations of Hierarchical Information*. In *Proc. CHI'91, New Orleans, Louisiana*, pages 189–194, New York. ACM.

Robertson, G. G., Card, S. K. and Mackinlay, J. D. (1993). *Information Visualization Using 3D Interactive Animation*. *Communications of the ACM*, **36**(4), 56–71.

Salton, G. (1989). *Automatic Text Processing*. Series in Computer Science. Addison-Wesley, Reading, MA.

Sams (1995). *The Internet Unleashed*, 2^{nd} edition. Sams.net Publishing.

Savetz, K. (1996). *The Unofficial Internet Book List*. `http://www.northcoast.com/savetz/booklist/`.

Sengstack, J. (1996). *Hyper-G: The Next Generation Web Server?* *New Media*, pages 19–20. `http://www.hyperstand.com/`.

Serra, L., Chua, T.-S. and Teh, W.-S. (1991). *A Model for Integrating Multimedia Information Around 3D Graphics Hierarchies*. *The Visual Computer*, **7**(5-6), 326–343.

Shneiderman, B. (1996). *The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations*. In *Proc. 1996 IEEE Symposium on Visual Languages*, Boulder, Colorado. IEEE Computer Society. `ftp://ftp.cs.umd.edu/pub/papers/papers/3665/3665.ps.Z`.

SIGLINK (1996). *ACM SIGLINK Home Page*. `http://www.acm.org/siglink/`.

Smith, P. A. and Wilson, J. R. (1993). *Navigating in Hypertext Through Virtual Environments*. *Applied Ergonomics*, **24**(4), 271–278. Butterworth-Heinemann Ltd.

Stochniol, A. (1996). *asWedit*. `ftp://sunsite.doc.ic.ac.uk/packages/www/asWedit/`.

Strasnick, S. L. and Tesler, J. D. (1996). *Method and Apparatus for Displaying Data within a Three-Dimensional Information Landscape*. Silicon Graphics, Inc. US Patent number 5,528,735. Filed 23rd March 1993, granted 18th June 1996. Available from the US Patent and Trademark Office `http://www.uspto.gov/`.

Stroustrup, B. (1991). *The C++ Programming Language*, second edition. Addison-Wesley, Reading, Massachusetts.

Stubenrauch, R., Kappe, F. and Andrews, K. (1993). *Large Hypermedia Systems: The End of the Authoring Era*. In *Proc. ED-MEDIA 93, Orlando, Florida*, pages 495–502, Charlottesville, VA. AACE.

Sun Microsystems (1996). *World Cup USA '94*. `http://sunsite.doc.ic.ac.uk/wc94/`.

Tesler, J. D. and Strasnick, S. L. (1992). *FSN: The 3D File System Navigator*. Silicon Graphics, Inc. `ftp://sgi.sgi.com/sgi/fsn`.

Utting, K. and Yankelovich, N. (1989). *Context and Orientation in Hypermedia Networks*. *ACM Transactions on Information Systems*, **7**(1), 58–84.

VAG (1996). *VRML Architecture Group*. `http://vag.vrml.org/`.

van Dam, A. (1988). *Hypertext '87 Keynote Address*. *Communications of the ACM*, **31**(7), 887–895.

VDI (1996). *Discovery*. Visible Decisions, Inc. `http://www.vdi.com/`.

W3C (1996). *The World Wide Web Consortium (W3C)*. `http://www.w3.org/`.

Wavefront Technologies (1991). *Advanced Visualiser User's Guide*. Wavefront Technologies, Inc., Santa Barbara, California.

WebCompare (1996). *Web Servers Feature Chart*. `http://webcompare.iworld.com/`.

Weiss, S. and Schwartz, M., editors (1987). *Proc. Hypertext'87*, Chapel Hill, North Carolina. ACM Press.

Weiss-Fersko, H. (1991). *Guide. (OWL International Inc.'s Guide 3.01 — Software Review)*. *PC Magazine*, **10**(10), 257–259.

Wernecke, J. (1994). *The Inventor Mentor*. Addison-Wesley.

Williams, G. (1987). *HyperCard*. *Byte*, **12**(14), 109–117.

Windisch, C. (1995). *HarAdmin: A Graphical Tool for Hyper-G Server Administration*. Master's thesis, Graz University of Technology, Austria. `ftp://ftp.iicm.edu/pub/Hyper-G/doc/windisch.ps.gz`.

Wired (1996). *Wired Magazine*. Wired Ventures. `http://www.hotwired.com/wired/`.

Wolf, P. (1996). *Three-Dimensional Information Visualisation: The Harmony Information Landscape*. Master's thesis, Graz University of Technology, Austria. `ftp://ftp.iicm.edu/pub/Hyper-G/doc/wolf.ps.gz`.

Wood, A., Drew, N., Beale, R. and Hendley, B. (1996). *HyperSpace: Web Browsing with Visualisation*. University of Birmingham. `http://www.cs.bham.ac.uk/~amw/hyperspace/`.

Worlds Inc. (1996). *Worlds Chat, Alpha World, Gamma*. Worlds, Inc. `http://www.worlds.net/`.

Yahoo (1996). *Yahoo!* `http://www.yahoo.com/`.

Yankelovich, N., Haan, B. J., Meyrowitz, N. K. and Drucker, S. M. (1988). *Intermedia: The Concept and the Construction of a Seamless Information Environment*. *IEEE Computer*, **21**(1), 81–96.

Young, P. (1996). *Three Dimensional Information Visualisation*. University of Durham. `http://www.dur.ac.uk/˜dcs3py/pages/work/Documents/IV-Survey/`.