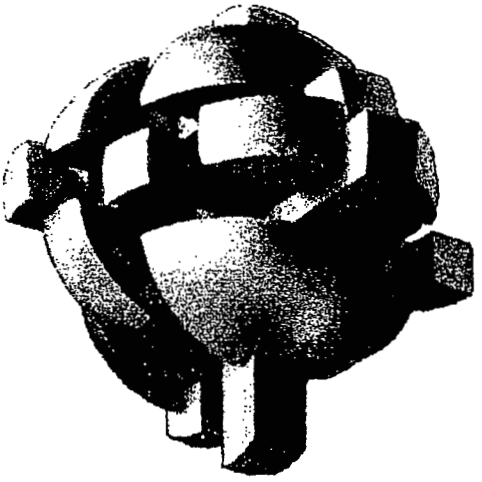
eòlas™



Business Plan

"The only way to successfully predict the future is to invent it."
Alan Kay

Eòlas™ Technologies Incorporated
10 East Ontario Street, Suite 5106
Chicago, IL 60611
voice: (312) 337-8740
fax: (312) 337-8743
email: info@eolas.com

World Wide Web: **http://www.eolas.com**

eòlas™
TECHNOLOGIES
INCORPORATED

BUSINESS PLAN

TABLE OF CONTENTS

eòlas™
TECHNOLOGIES
INCORPORATED

BUSINESS PLAN

1. EXECUTIVE SUMMARY

Eòlas Technologies Incorporated, of Chicago, IL is a new company dedicated to creating and licensing innovative technologies and related products which will enable the World Wide Web to become the preferred environment for all interactive computing applications by the year 2000.

History:

Every place one looks lately there is more and more hype about the "Information Superhighway (or IWAY)." All of the PC trade magazines now have Internet-related stories, and a new host of IWAY-specific magazines, such as Wired, Boardwatch and Internet World have recently become widely popular. Much of this current excitement is a direct result of the development, in 1993, of the NCSA Mosaic graphical interface to the World Wide Web (see Appendix 4a), and the subsequent explosion of commercial interest which it sparked. Mosaic presented users with a simple easy-to-use interface to the Internet, finally making the Net accessible to nontechnical users. The subsequent commercialization of Mosaic, and its successors, such as Netscape, catalyzed an explosion of commercial interest in Web-based technologies.

As good as they are, these standard World Wide Web (WWW) browsers still allowed only a minimal degree of interactivity for the user. Although limited interactivity (in the form of searching and form input capabilities) existed, most users of the WWW still only passively browsed hyperlinked documents. This was largely because of the severe limitations imposed upon WWW developers by the HyperText Markup Language (HTML) and HyperText Transfer Protocol (HTTP) communications standards that the WWW relies upon. Despite these limitations, the rapid commercial acceptance of the Internet has caused many to look at this arena as a platform for the delivery of interactive applications.

Conventional HTML protocols allow Web documents to contain text and "inline" images (which are read from external files, but displayed within the HTML document), as well as links to external files (hypertext links) containing other HTML documents or non-HTML files such as image files, audio files, or animation files. When hypertext links relating to non-HTML files other than inline images are clicked upon, Mosaic downloads the data and then launches external programs to allow those files to be viewed. This type of application launching is similar to the way that the Windows File Manager allows a program to be launched with a particular data file by double-clicking on the data filename in the File Manager window. Mosaic launches the viewer with the relevant data file, but the viewer application runs externally to Mosaic, and control of Mosaic is not returned to the user until the viewer program stops running. There is typically no dynamic communication or sharing of data going on between Mosaic and the viewer program during this process. Those who have used the Web will recognize this as the problem when viewing MPEG files, for example.

Eòlas' innovations were developed by Dr. Michael Doyle, David Martin and Cheong Ang as a solution to these problems, while working at the UCSF academic computing center. They realized that it could be possible to create embeddable interactive application modules that could be treated by Mosaic in a similar manner to the way that it treats inline images. These "inline applications" (later dubbed Weblets) could become part of HTML documents and share data with Mosaic, rather than running as mere external viewers. They further realized that, with this approach, *any* interactive application could be run within HTML documents viewed with an enhanced version of Mosaic, and that these applications could reside on computers that exist anywhere in the world on the Internet. They immediately set out to prove the feasibility of these ideas and, within a period of three months, had

implemented and demonstrated a sophisticated interactive multidimensional data visualization Weblet and an enhanced version of Mosaic that worked smoothly together. Additional demonstration Weblets were soon developed. Further enhancements of Mosaic were soon realized by incorporating Dr. Doyle's patented MetaMAP object-oriented interactive imaging technology, as well.

The University of California (UC) systemwide Office of Technology Transfer quickly recognized the tremendous commercial potential of the Weblet technology and filed a patent application in October of 1994 covering the concept of a Weblet API (WAPI), after spending over 6 months investigating its patentability. In 1994, Martha Luehrmann, the UC licensing office on the case stated openly that UC believes that this patent will eventually be the most profitable intellectual property that the University has ever generated. Recognizing the opportunity, Dr. Doyle left his position at UCSF to guide the commercialization of the technology. He and the other two inventors then immediately founded a company and secured an exclusive option to the patent; and Eòlas was born.

Eòlas' WWW-related technologies have already begun to transform the World Wide Web from a mere net surfing tool to a robust framework for wide area delivery of sophisticated client/server applications. This technology enables the WWW to begin to transform into a complete operating system for the next generation of computer applications. This "Web OS" will eventually make it irrelevant what PC or workstation operating system is running on a user's machine, since *any* application will be accessible through the Web. The timeliness of such a concept has been validated in print by Bob Metcalfe, the inventor of Ethernet, and the Publisher of InfoWorld, in a February 27, 1995 editorial (see Appendix 4b) where he says that he has just come to the "conceptual breakthrough" that the Web may ultimately become a replacement for the operating system.

At the end of Metcalfe's article, he says: "What I can't figure out whether the new Web OS standard will come from IBM, Apple, Microsoft, Oracle, Novell, or some new OS company that we've heard little of." Well, Metcalfe was right on the money, and Eòlas is that new, unheard-of company. What he must not have known at the time is that Eòlas' founders had already developed and demonstrated just such a technology (and also coined the term "Web OS"), and that the University of California had created the opportunity to for them to enforce an Internet-wide standard by securing the intellectual property rights to the fundamental *idea* of using the Web in this way.

After founding Eòlas, Dr. Doyle then set out to take the necessary steps to establish this technology as the *defacto* standard for enhanced interactivity through the Web, and to see that such a standard be accepted long *before* the patent application is acted upon by the US Patent Office. The obvious way to do this was to work towards getting Eòlas' enhancements to be incorporated into a critical mass of WWW browsers, and then to come up with a plan to simulate the market for "plug-in" applications. Since Spyglass, Inc. had acquired the exclusive license to Mosaic from NCSA, the natural path was to try to develop a working relationship with them. The initial fundamental goal is to get as many WWW browsers in the world as possible to support Eòlas' standards. Towards this goal, several important milestones have been reached:

Milestone 1: Eòlas secured exlusive rights to the UC (WAPI) pending patent in July of 1995, and a press release to this effect was issued on August 18th, 1995, gaining widespread coverage in both the trade and general press. The firm has also negotiated terms for an exclusive license to the MetaMAP patent, which will be signed in October of 1995.

Milestone 2: The Weblet API inventors, Doyle, Martin and Ang, have developed a UNIX X-Window version of a WAPI-enabled enhanced Web browser (called WebRouser), based upon enhancements to the award winning NCSA Xmosaic program. This version was released on September 18th, 1995, to the Internet under a free-for-individual-noncommercial-use license, together with three sample Weblet-based applications. During the first two weeks after the release, over 350 companies and educational institutions downloaded versions of the program for Sun, Silicon Graphics, and Linux (Intel-based PC) computing platforms.

Milestone 3 After releasing the patent claims to Spyglass, Inc. (under a nondisclosure agreement), Eòlas quickly reached an agreement-in-principle with them to license the WAPI patent for inclusion in all commercial versions of Mosaic. Negotiation of specific terms is currently underway. Both Eòlas and Spyglass hope to have a formal agreement signed in the fourth quarter of 1995.

Eòlas has received strong expressions of interest from a varied group of large corporations. After the University released a brief technology announcement in August '94, over 100 companies and institutions contacted the UC Office of Technology Transfer wanting more information about obtaining licenses for the technology. Approximately 35 of these potential licensees signed nondisclosure agreements with the University in order to receive more detailed information on the technology. During the 4th quarter of 1995, we will begin an organized campaign to contact these potential clients and to close sales of both licenses and products.

Dr. Doyle has formulated a key technology development plan which will allow the staged release of a series of innovative Weblet-based products that will find ready markets as Web applications enhancements in the near term. and will simultaneously act as building blocks which form the basis for paradigm-shifting applications suites which will literally "take over the desktop" in the longer term.

We are currently in the process of developing a comprehensive marketing and public relations plan with the aim of stimulating market recognition and brand identity for both the WAPI and MetaMAP technologies. This plan will center upon establishing a strong brand identity for the Eòlas embedded-object technology (WAPI), similar to the well, known "Intel Inside" campaign, licensing the technology widely, and marketing and selling innovative products to companies and institutions that wish to capitalize on the popularity of the WWW to expand their market base and/or to solve their enterprise-wide information systems problems.

We have assembled an experienced team of industry leaders, with strong management, marketing and technical abilities, that is prepared to come on board when the company becomes fully capitalized. The company is led by its founder and CEO, Dr. Michael D. Doyle, the former Director of the academic computing center at the University of California, San Francisco, the founder and Chairman of MetaMAP, Inc., and the former Director of the Biomedical Visualization Laboratory at the University of Illinois. The Vice Presidents, Willard Doyle (Sales) and Keith Bates (Marketing), represent a combined total of over 60 years of experience in successful business management, as well as marketing and advertising services for hundreds of fortune 1000 companies. During the 4th quarter of 1995, the company will begin a search for a Chief Operating Officer to work with Dr. Doyle and the vice presidents in managing the rapid growth of the company, and to establish key strategic alliances with major industry players in the software technologies marketplace.

The current capitalization of Eòlas was in the form of an early debt agreement with a Palo Alto venture capital firm (which was repaid in the summer of 1995) and a private offering of Series A preferred stock, representing 7.4% of the company's equity, sold for $490,000. We are planning to further capitalize the company through the sale, in December, 1995, of Series B preferred stock in order to raise an additional $12 million in capital. These funds will be used to support a major marketing and sales campaign, as well as the development of strategically-important Weblet applications and the WebWorks Internet workgroup software package. The funds will also support expansion of Eòlas' marketing and sales oranization, through the second quarter of 1996, when revenues will begin to support the company's cash flow. Although the company has no formal plans to offer stock to the public in the near term, the current excitement in the investment community with respect to Internet-related technology companies makes the prospect of "going public" within the next two to four years attractive indeed.

The projected return on investment of these funds is significant. Early profitability is impressive, since products and services can be brought to market quickly due to the significant early development of the technology while within the University environment, and since the potential market includes *every* developer or provider of WWW-based information or applications. Conservative estimates are that over 100,000 such potential licensees will exist by the end of 1998. Each licensee that we obtain creates a potential ongoing revenue stream, through product upgrades. Although our projections assume only a 30% penetration of this market by the end of 1998, the recurring nature of these revenue streams translates into annual revenues of over $1.9 billion by the start of 1999.

A review of the projected profit and loss statements on page 30 shows that by the end of 1996 Eòlas is expected to have an annual operating profit of over $13,000,000 based on sales of over $45,000,000 in Weblet™ and MetaMAP®-related products and services, even before the Weblet API patent is expected to issue. Assuming that the WAPI patent issues by the start of 1997 and licensing of the patent begins in full force, projected annual profit at the end of 1997 is over $194,000,000, based upon revenues of $471,000,000; and projected profits for 1998 are over $827,000,000, based upon revenues of approximately $1,935,000,000 using conservative operating and expense figures. Of course, various risk factors, such as product competition and legal challenge to the relevant patents, may reduce the company's ability to meet these projected goals.

2. MANAGEMENT TEAM

Chairman & Chief Executive Officer: Michael D. Doyle, Ph.D.

Dr. Doyle founded Eòlas after serving as Director of the UCSF Center for Knowledge Management (the UCSF Academic Computing Center, CKM), in San Francisco. He also is an adjunct member of the faculty of the UCSF Medical School. From 1992 to 1994, he was Vice President for R&D of Muritech Co., based in Boston, MA, and he currently serves as Chairman of MetaMAP, Inc., in Chicago, IL. Prior to his position at UCSF, he served as the Director of the Biomedical Visualization Laboratory at the University of Illinois at Chicago. He received his Ph.D. in Cell and Structural Biology from the University of Illinois at Urbana-Champaign in 1991, and the B.S. degree in Biocommunications Arts from the University of Illinois at Chicago in 1983. He has managed both campus-wide computing service centers and a major technology transfer center at UIC. He has also managed several large production-level programming teams in various applied research projects in the informatics field. He has served as a charter member, and officer, for several professional organizations, including acting as the first president for the Special Interest Group for Biological Computing, at the University of Illinois at Urbana-Champaign, in 1988, and founding, together with his UCSF CKM staff, SIGWEB, the Bay-Area Special Interest Group for the WWW, in 1993. Dr. Doyle currently serves on the Board of Scientific Advisors for the Visible Human Project at the National Library of Medicine and the Scientific Advisory Board for the National Museum of Health and Medicine. He has also served, and continues to serve, as a regular reviewer of computer-related research grants for the National Institutes of Health since 1992. He is a member of Sigma Xi, Mensa, and Phi Kappa Phi.

Chief Operating Officer: To be named

Eòlas has already begun a national search for an acknowledged industry leader with experience in managing rapid-growth companies in the information technology industry, to act as Chief Operating Officer. This individual will work with Dr. Doyle and the vice presidents in managing the expansion of the company, and to establish key strategic alliances with major industry players in the software technologies marketplace. We hope to complete this search so that the new COO can come on board before or during the first quarter of 1996.

Vice President, Sales: Willard H. Doyle

Willard H. Doyle comes to Eòlas from Brand Group, Inc., where he was the founder and President, for more than 15 years. Brand Group was the first firm in the U.S. to specialize in developing corporate and product identity and naming systems. He is also the President of MetaMAP, Inc., which is a small Chicago firm which concentrates on commercialization of the MetaMAP patents in a wide variety of markets. Through Brand Group, Mr. Doyle acted as a consultant in strategic marketing and sales, product planning and development and industrial design for hundreds of Fortune 1000 companies. Mr. Doyle has over 20 years experience in successful business management, new product planning and development, industrial and graphic design, trade show and exhibit design, corporate and product identity, market research (qualitative and quantitative), marketing, sales, business planning and management. He has worked with leading national and international consumer, commercial and industrial companies, government agencies and advertising agencies on hundreds of projects throughout his career. Mr. Doyle holds a B.S. in Industrial Design from Illinois Institute of Technology.

Vice President, Marketing: Keith Bates

Keith Bates is a marketing communications expert with over 20 years of experience providing both counsel and creative services to the nation's software vendor community. From 1970 to 1995 he served as CEO/Creative Director of Keith Bates & Associates, Inc., a high-tech advertising agency supporting the marcom needs of over 125 major software companies. He is a well-known professional in the field of high-tech marketing, a speaker at many industry conferences, contributing author to numerous industry publications and a communications consultant to IBM, Data General, Arthur Andersen, Zenith Data Systems and many others. Bates' strategic planning skills, sensitivity to communications issues, and ability to translate complex ideas and technologies into

easily understood products has put his services in high demand. More recently he has developed a marketing communications methodology known as the Marcom Engine that results in dramatic improvements to marketing productivity. The methodology incorporates elements of both business process reengineering and integrated marketing communications. His career began over 40 years ago with Foote, Cone & Belding, the nation's largest advertising agency, which balances his high-tech experience with an equal number of years in consumer products. In addition to managing his own multimillion dollar ad agency, Bates cofounded a technology oriented public relations firm, a consumer publication distributed worldwide, a PC software company, and an international travel information services company. His schooling has included the Art Institute of Chicago, the American Academy of Art, the University of Illinois, and Northwestern University.

Vice President, Administration and Legal Affairs: Rachelle Tunik, J.D.

Ms. Tunik comes to Eòlas from the University of California, San Francisco. Ms. Tunik specializes in addressing the legal and administrative needs of technology professionals. In addition to working with Dr. Doyle's team at the Center for Knowledge Management, she conducted contract negotiations, developed marketing strategies, and directed a variety of financial and administrative activities. In 1994, she negotiated the first contract successfully approved by the San Francisco Board of Supervisors between San Francisco General Hospital and a health maintenance organization (HMO). Between 1986 and 1989 Ms. Tunik directed two non-profit service oriented businesses, bringing them out of debt and establishing financial and administrative structures for self-supporting expansion. While working for the Federal Government she provided French and Russian translation and designed customer service materials. Her experience also includes work as Assistant Accountant at the multi-million dollar Triangle Nut Company. Ms. Tunik has produced and directed two instructional videos for public television, and participated in numerous fund raising campaigns for public television, the Multiple Sclerosis Foundation and political candidates. Her commitment to community service led to her appointment to a regional Public Library Planning Board and a Mayoral Proclamation of a day in her honor. She has been an active member of the Columbia Alumni Recruitment Committee, providing college counseling and outreach to local high schools since 1990. Ms. Tunik received her undergraduate degree from Columbia College, in New York, and her legal degree from the University of San Francisco.

Director of Engineering: David C. Martin, M.S.

Mr. Martin is an experienced manager of state-of-the-art software development projects. He is also an expert Unix programmer and database designer. As the Director for Innovative Software Systems at University of California's Center for Knowledge Management, he managed a staff of software engineers working on healthcare-related projects in the information sciences. He has gained national recognition for his technical direction and design of a large-scale online scientific journal publishing project. He is named as one of three inventors on the University of California patent application for embedded interactive applications in hypermedia documents. Mr. Martin brings management, software design and production experience from the private sector, where he worked at Sun Microsystems of Mountain View, California, Molecular Simulations, Inc. of Sunnyvale, California and Innovision Corporation of Madison, Wisconsin. Mr. Martin received his B.A. in Interdisciplinary Science from the University of California at Berkeley and his M.S. in Computer Science from the University of Wisconsin at Madison.

Senior Information Scientist: Cheong S. Ang, M.S.

Mr. Ang received his M.S. degree in computer science from the University of Illinois at Chicago in 1993. He did his UIC thesis project while working in Dr. Doyle's Biomedical Visualization Laboratory on interactive medical visualization using distributed parallel processing through (Inter)networked workstations. During this time, he developed a highly portable and modular system for distributed volume visualization (VIS) which later formed the basis for the first Weblet-style embedded program object. After moving to UCSF as a programmer/analyst in the Center for Knowledge Management, he did the primary coding for the demonstration system that embedded interactive applications within WWW HTML documents, viewable through NCSA's MOSAIC client software, modified with his enhancements. He is named as one of three inventors on the University of California patent application that has been submitted for this technology, and is considered a national authority on the development of Internet-based interactive multimedia and visualization systems.

3. THE OPPORTUNITY

This business plan describes two proprietary technologies, the Weblet API (WAPI) and MetaMAP, which drastically expand the ability of companies to use interactive software applications to conduct business over the Internet. Much of this plan relates to commercial applications for the World Wide Web. The WWW was developed by Tim Berners-Lee, at Switzerland's Particle Physics Laboratory (CERN), in 1989 as a way to manage technical documents for groups of geographically-remote collaborators. It was not until 1993, when Marc Andreessen at the University of Illinois' National Center for Supercomputing Applications (NCSA) developed MOSAIC, that things really began to take off. MOSAIC presented people with such an easy-to-use interface to the Internet that it was soon called the "Killer App for the Information Superhighway" (see Appendix 5). Andreessen left NCSA to found a company, originally called Mosaic Communications, Inc., now called Netscape Communications, Inc., to create a commercial competitor to MOSAIC. Despite the great activity in the WWW browser market, current browsers are severely limited in the types of information which they can handle. Eòlas Technologies, Inc., will exploit innovative new technologies (WAPI and MetaMAP) that will overcome these limitations and allow the WWW to greatly expand its marketability.

A few facts will illustrate the speed at which new business opportunities are developing in this area. Although Netscape was founded less than a year and a half ago, it's recent IPO valued the company at over $2 billion for a short time. Several months after Netscape was founded, the University of Illinois' NCSA signed a deal with another company, Spyglass, Inc., to commercialize MOSAIC. To date, Spyglass has licensed over 30 million copies of MOSAIC. Microsoft recently released a version of Spyglass MOSAIC that is now bundled with the latest version of Windows (Windows 95). Financial institutions, such as Bank of America, MasterCard, and Visa are already allied with either Spyglass or Netscape in anticipation of profiting from the vast business opportunities the WWW provides.

3.1. The Weblet Application Programming Interface (WAPI)

While researching ways to implement dynamic three-dimensional real-time imaging through the WWW, the core technology for WAPI was conceived and developed by Michael Doyle, David Martin and Cheong Ang during the summer of 1993 at the University of California, San Francisco's Center for Knowledge Management. The first demonstration, in 1993, allowed real-time manipulation of 3D biomedical image data, driven by a distributed parallel array of powerful computers, embedded within a WWW document, and accessed via a remote low-end workstation running the inventors' enhanced version of NCSA MOSAIC.

This software represents the first time interactive program objects (which have since been coined "Weblet applications") have been embedded within distributed hypermedia documents. The resulting system provides the user of the enhanced MOSAIC client with the ability to interactively control vast remote computational server resources from a low-end (sub $5000) client machine, connected via the Internet. This also allows the creation of compound documents which combine text and graphic elements, as well as fully interactive plug-in applications embedded *inline* (using the jargon of the WWW) into seamless units, with the same look and feel on Unix, Mac and DOS client platforms.

The inventors proceeded to create a standard application programming interface (API) for this technology, now called the Weblet API. This specifies how external applications can both "drive" World Wide Web browsers, such as Mosaic, and "embed" themselves within Web-based documents. Web browsers supporting the Eòlas Weblet API provide users with functionality that seems similar to what Microsoft's object linking and embedding (OLE) API provides to MS Windows users on a single machine, but extends far beyond the limitations of such desktop-bound operating systems to easily link objects and documents over open-standard wide area networks, such as the Internet.

This technology also represents the first public demonstration of a working distributed-object-compound-document model using the Internet. The "distributed-object" term refers to the nature of the server-side computation driving

the embedded application objects. In conventional compound-document client/server systems a two-tier structure is standard. The embedded objects in such a system are hosted either by the client machine (first tier) or by a

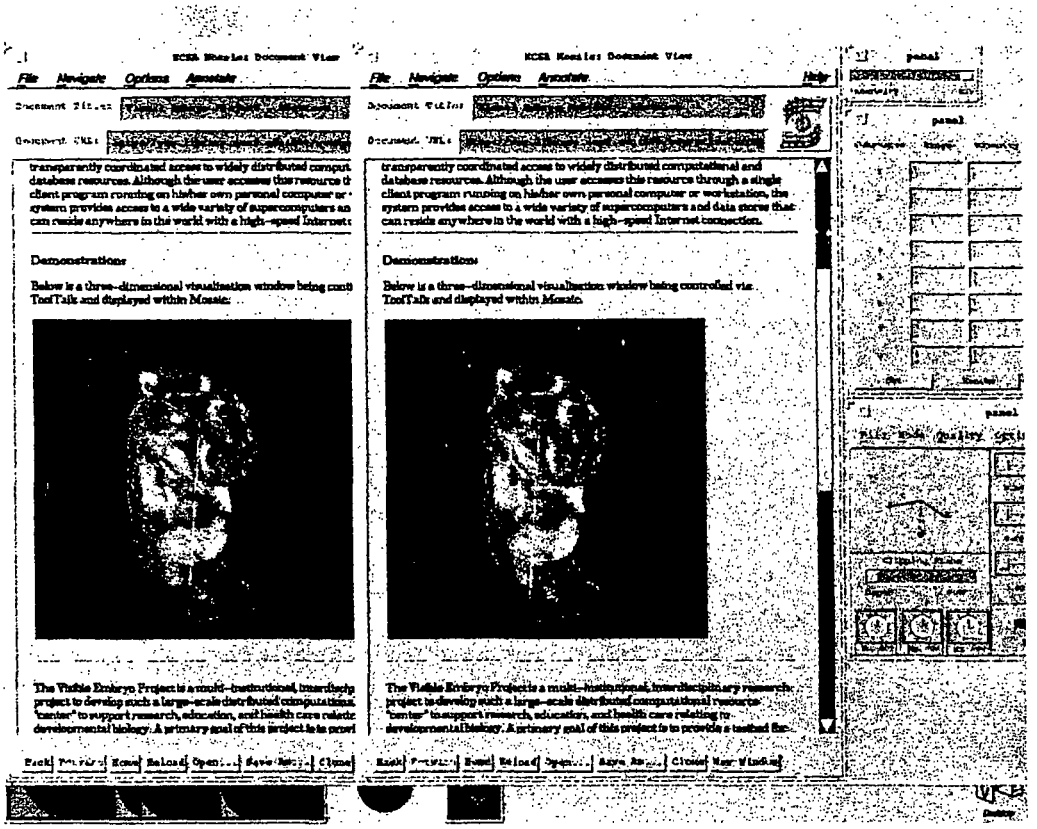


Figure 1: A stereo-pair illustration of the first Weblet™-based application, embedded within an NCSA MOSAIC document, providing realtime interactive control of a volume visualization of a huge 3-dimensional biomedical dataset. The complex calculations necessary for volume rendering of the large dataset were supported by a "farm" of remotely-networked powerful graphics workstations, controllable through Eòlas' founders' enhanced version of NCSA MOSAIC running on a low-end Sun SPARC 2 workstation with only 8 Mbytes of RAM. This technology was developed by Doyle, Martin and Ang at the Center for Knowledge Management at the University of California, San Francisco, and was demonstrated there in November, 1993. Further information can be obtained at http://bubba.afip.mil, via the World Wide Web.

second tier of server computers, where each embedded object runs on a single server machine. A distributed-object system, on the other hand, allows each of the embedded objects to be computed more efficiently by distributing the computational load across a large number of server machines.

The Eòlas Weblet-based technology allows the design of three-tier client/server systems, where server/side logic can be coordinated at the second tier, while computational efficiency can be achieved at the third tier. An additional benefit of the WAPI approach is that the look and feel of the compound document is identical across Unix, Mac and Windows client platforms, and the various machines involved in the system only need to be connected to the Internet in order to intercommunicate.

Just as Microsoft's OLE API has formed the basis for Windows to become the dominant operating environment for the "desktop era" of the late '80s and early '90s, Eòlas' Weblet API will form the basis for the Eòlas Web OS to become the dominant operating environment of the "Internet era" of the late '90s and beyond.

3.2. MetaMAP

The initial motivation for the invention by Dr Doyle of the MetaMAP process in the mid 1980s was his desire to create an interactive hypermedia application, on an Intel 8088 platform, to teach histology to medical students. At the time, hypertext was just starting to be discussed as a potentially viable way to deliver instructional material to students via microcomputers. Hypermedia, which involves integrating several media types (text, images, sound, etc.) with hypertext-style linking, had been suggested but not implemented on PC or Mac platforms, mainly because such sophisticated applications would outstrip the computational power of the slow early PCs and Macs of the time.

Medical students who study histology (the anatomy of cells and tissues) need to learn how to identify particular cell types from microscopic images. Dr. Doyle created a program which allowed a student to point to any arbitrary cell on the computer screen, among the hundreds in a single microscopic image, and have the system correctly identify the cell type for the student, and then provide hyperlinks to find relevant instructional material to present to the student.

The most obvious approach would have been to store polygonal maps of the contours of all of the cells in the image in the computer's memory, and to search through this list of polygons after each mouse click, using geometrical operations to determine whether the point in question was inside the polygon. Such an approach, however, was too inefficient even to run on powerful graphics workstations of that era, much less PCs or Macs.

Dr. Doyle instead developed a method for using the color attributes of the image pixels as a means for indexing object identity in the micrographs. This was done by creating an object-oriented color map, which allowed each pixel in the image to act as an independently-addressable hotspot, allowing individual object classes to "own" certain ranges of color index values. A small look-up table was stored in memory correlating each unique palette "segment" to a particular cell type, or object identity. Then, when a student clicked upon the image with a mouse, the system merely determined the color index under the cursor, and found the corresponding entry in the object identity look-up table in order to describe what the cell type was. This allowed each pixel in the image to act as it's own small hotspot. Hundreds of thousands of such hotspots representing hundreds of different object identities could be incorporated into a single image. Since the identification process was so efficient, and since the object identities were encoded into the image files themselves, Dr. Doyle showed that a very small program running on a very slow PC could allow a student to browse through an enormous database of detailed images with relative ease.

As PCs and Macs have become more powerful, the performance degradation that polygon hotspot encoding imposes has become less and less noticeable for small databases. In fact, polygon hotspot encoding is still the standard technique used by multimedia software developers. The WWW, however, presents new opportunities for MetaMAP encoding. Most of the "home pages" browsable on the Web now include images with hotspots. The current method for defining and interacting with those hotspots is called ISMAP, is based upon polygon hotspot encoding, and is very cumbersome. As is discussed below, the same efficiencies that made the MetaMAP process so efficient on a single platform make it an excellent alternative for the definition of hotspots on Web pages.

3.3. MetaMAP as an improvement to ISMAP

Hotspots on images are the major way designers of WWW home pages currently try to distinguish themselves. The latest and greatest home pages, such as those at www.whitehouse.gov or www.ibm.com, contain large ISMAPped images. The intent is that users will use these images as menus for browsing the remaining pages at those WWW sites. The ISMAP approach for doing this has several problems, however:

- Unlike the hotwords in the text areas of the WWW pages, there is no indication of hotspots as the user's mouse cursor passes over them There is also no display of anchor URLs, therefore the look and feel of the system is different for the user when browsing these ISMAPs than when they interact with hotwords in the text.
- ISMAPs are difficult to implement, requiring complex server-side files and operations.
- There is no capability to create nested hotspots (hotspots surrounded by other hotspots).

- Interaction is inefficient due to the need for geometric operations (polygon queries) to determine the identity of the object selected by each of the user's mouse clicks.
- Since ISMAP object decoding is implemented through scripts that are executed at the remote Web server, local disk-based versions of HTML databases for viewing without network access (such as on CD-ROM) can't be developed. There would be no Web server in such a situation.

MetaMAP-based client-side image maps solve these problems in several ways:

- Anchor URLs are dynamically displayed, and hotspots are highlighted, as the cursor passes over them, so the look and feel are identical for interaction with both image-based and text-based hotspots.
- MetaMAP is easy to implement for the WWW designer, since the image and object names are fully encapsulated. All decoding happens at the browser.
- Since each pixel is an independently-addressable hotspot, unlimited nesting is possible.
- Greater efficiency is provided, since no geometric operations are needed.
- Since MetaMAP-based image maps are decoded entirely by the WWW browser, no HTTP server is needed. Therefore CD-ROM-based HTML applications can be developed for sale to those without adequate network access, still being able to autosense connectivity and capitalize on such access if it exists.

The end result of these improvements is that use of MetaMAP encoding and decoding of interactive hotspots allows a much greater degree of user friendliness for WWW-based online services than the currently-popular ISMAP approach. It also enables merging the best aspects of CD-ROM and Web technologies in products.

3.4. Strength of patents

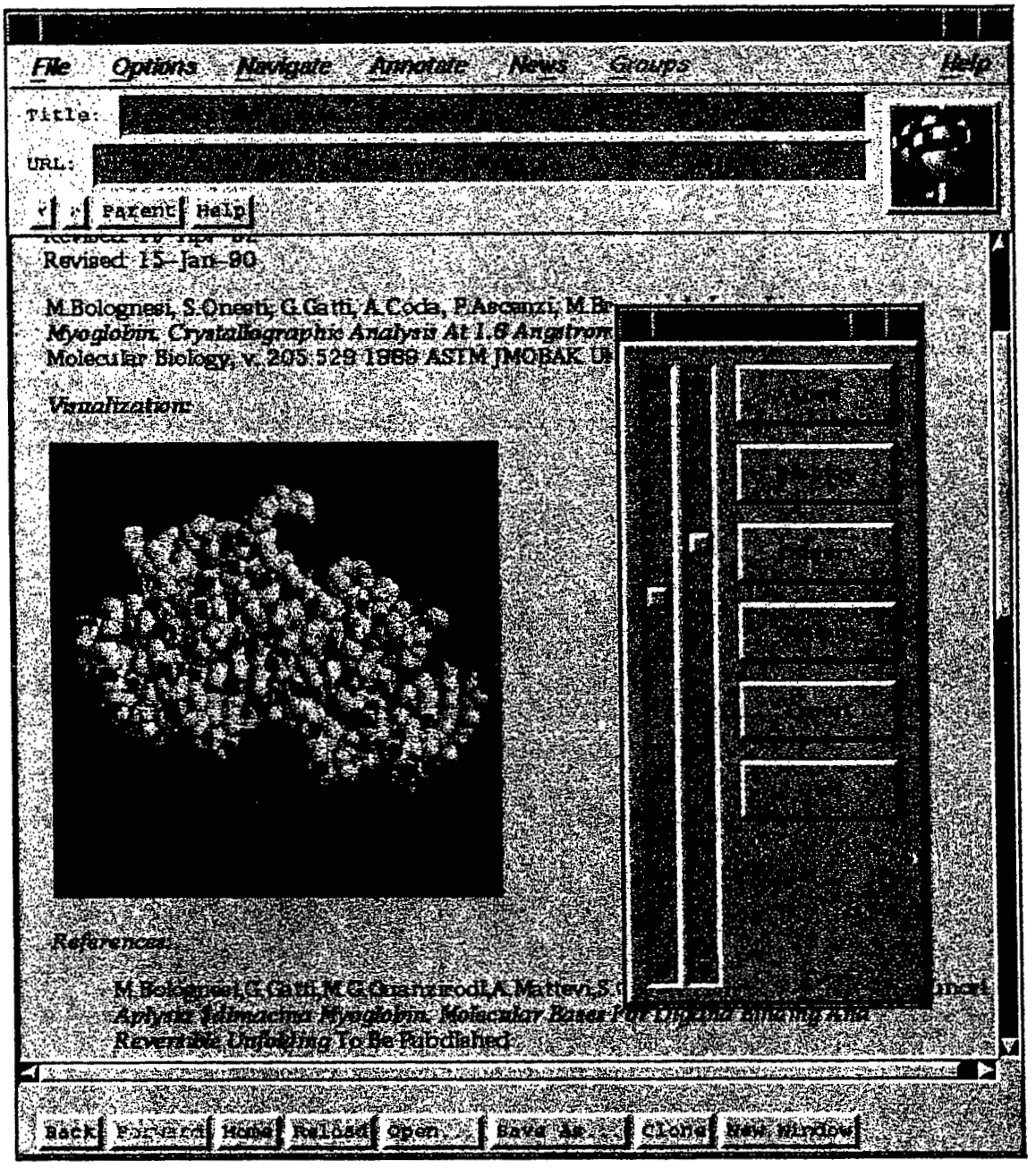
A U.S. patent application was filed by the University of California in October of 1994 for the **Weblet API invention.**

REDACTED

. The MetaMAP invention was applied for in August of 1987, and was subsequently awarded by the U.S. Patent and Trademark Office in July of 1989 (#4,847,604). Patents for **MetaMAP** have also been awarded in both Canada and Australia.

Many Web browser manufacturers, such as Spyglass, Netscape, Sun, AOL/Navisoft, and California Software, are already seeking to implement technology covered by our pending patent claims for the Weblet API (see Metcalfe's article in Appendix 4b). We intend to avoid expensive and time consuming litigation through mutually beneficial licensing and cross licensing. Since we cannot exercise the claims of the pending WAPI patent aggressively until it actually issues as a U.S. Patent, we have already begun to stress to the Web browser manufacturers the advantages of licensing from us now both to gain deeply-discounted licensing terms, and to prevent future liability for themselves and their customers. The fact that Spyglass has already examined the patent claims and remains interested in signing such a license (thereby confirming the validity of our patent claims) will encourage others to do the same. We will price Weblet API developer licenses during the patent-pending phase at deep discounts. In addition, on August 18, 1995, we released the Eòlas WebRouser™ for Unix-based systems, with a a royalty-free-for-individual-noncommercial-use license, together with a few sample Weblets that demonstrate the power of the technology. Within 2 weeks of the press release announcing WebRouser's availablilty from the Eòlas web site, over 350 commercial and educational sites had downloaded and installed the program. Already, a major ARPA-funded educational technologies research consortium, based at George Mason University (http://www.gmu.edu). has decided to use WebRouser as the basis for an online educational system that is expected to be installed on approximately 10,000 personal computers in middle schools and high schools worldwide in Spring of 1996.

4. PRODUCTS, SERVICES & LICENSING



A screen shot of Eòlas WebRouser™ showing the Rasmol demo Weblet in operation. This version of the browser was released over the Internet for free non-commercial use in August of 1995. When the user goes to a Web page with the "x-rasmol" application type specified via the <EMBED> HTML "tag," the associated molecular datafile is downloaded and viewed "inline" using the local Rasmol Weblet-based application.

Although we recognize that a plethora of potential applications areas exist for our technologies, we also recognize that we cannot possibly hope to address all of these markets ourselves. Eòlas' primary business opportunity exists in enabling other companies to use our tools and licenses to address as many of the potential applications areas as possible. We will do this by enabling others to provide high performance Internet applications access via our proprietary enhancements to popular Web interfaces, such as Spyglass' Mosaic, Netscape's Navigator, Sun's HotJava and the like. These enhancements are fundamental and can apply to all who access the Internet for whatever reasons.

Attractive opportunities are seen in two primary areas: products and licensing. In the long term, applications are foreseen in a broad range of markets; consumer, industrial, commercial, government, etc. In the short term, revenue and profit generation is focused on existing and currently emerging markets in which we have experience, existing resources, and/or good contacts.

4.1. Software Products

WebRouser, the first WAPI-enabled World Wide Web browser (available now):

Although it is Eòlas' intention to license the Weblet API broadly to all of the major WebBrowser companies, we also intend to continue to develop and enhance our WebRouser products. This will both allow us to showcase our technological advances in the future and to provide the other browser companies with an incentive to continue to license Eòlas' intellectual properties.

Key Features:

- Runs on Sun, SGI, and Linux Platforms (Mac and Windows due in Q1, 1996)
- Weblet (WAPI) enabled
- Polymap-based client-side image maps
- Document-driven dynamic button bars
- Document-driven menu options
- Also supports all standard features of NCSA/Spyglass Mosaic

Based upon enhancements to NCSA's award-winning Mosaic program, WebRouser features patent-pending technology that drasticaly expands the functionality of Web-based applications, and provides a simple and convenient way to add new features to browser programs through the use of Weblet-based plug-in applications.

The Weblet enhancement allows fully-interactive program objects to be run from within Web pages, through the use of a simple <EMBED> command, or "tag," within the document's text. These Weblet programs become treated by the browser as a part of the Web document, displayed "inline" and controlled by the user in place, without diverting the user's attention from the document itself.

Three demonstration Weblet programs are being distributed with the WebRouser package. These include an "inline" MPEG movie player, a 3D CAD file viewer/manipulator, and a 3D molecular modeling application. For example, when a user visits a Web page that has a URL for a 3D CAD model placed within the Web page through the use of the <EMBED> tag, the browser fetches the CAD file over the network and then launches the CAD viewer weblet on the user's machine. The user sees a live window within the Web page, displaying the fully-rendered 3D model, and a control panel which allows the user to rotate the model and zoom in to see details. When the user then travels from that page to the next destination on the Web, the browser "caches" the Weblet together with the Web page. If the user then hits the "back" button, to return to the most recent site, the Weblet appears again, right where the user left it, having maintained its "state" (rotation position, zoom level, etc) during the time it was cached. ***No other Web browser on the market can do this!***

Many other Weblets are currently under development by Eolas and others, including Weblet-based interpreters for several popular programming languages, such as Safe-Tcl/Tk, PERL, and JAVA. Plans are in the works for the creation of Weblets for both Visual Basic and the GRASP animation language, as well.

Other unique features of WebRouser include client-side image map support, and the ability for the browser's button bar and menu structure to be dynamically modified by simple commands within HTML documents.

Client-side image maps, for example, allow HTML authors to create graphical interfaces to their content that can be distributed on CD ROM, using the same image-map-based front ends as the online versions. Many publishers are attempting to create hybrid CD ROMs that use Web browsers as their front-end, capitalizing on the ability to develop one body of content that can be used both for CD and online distribution. These projects are often stymied by the fact that the image maps that are currently all the rage on the Web usually cannot be used to front-end the CD content.

The current approach on the Web is to use ISMAP-based image maps which require that a remote server decode the hotspots on the image. Since the ISMAP-based image maps are served up by the remote machine, they can't be used to front-end CDROM-based content, where oftentimes a network connection is not available. WebRouser's client-side maps can be loaded directly from the CD ROM, with no network connection required.

Another major advantage of WebRouser is the ability of Web documents to dynamically modify the browser's button bar and menu structure. Most Web designers try to build in some sort of navigation system into their documents, usually at the top of the page. The problem arises when the user scrolls down the page and suddenly the navigation graphical user interface (GUI) is no longer visible. Our <LINK> tag allows the Web document to place a button bar at the top of the screen, as a part of the WebRouser GUI. When the user scrolls down the document, the navigation buttons remain in place. Since the document drives the definition of the buttons' functions, each Web site can have its own "What's New," "What's Cool," etc. button bar pointing to their own content, not to some hard-coded browser company location, such as in other browsers.

Similarly, a new <GROUP> tag allows a Web document to define a new menu option in the WebRouser menu bar, allowing the user to quickly jump to a particular Web page within a large, complex Web site hierarchy.

WebRouser represent a new paradigm in Internet browsers, since its technologies empower Web designers to personalize their Web sites, and to deliver new levels of interactivity via the web. Taken together, these enhancements represent a quantum leap in the ability of Web site designers to build compelling functionality into their Web pages.

Polymap Toolkit (ready for release now):

This is an easy to use Windows 3.x tool for creating blazingly-fast Polymap-based client side imagemaps for the Web. This user-friendly Windows program allows a Web author to import any GIF image and interactively define hotspots that point to specific URL addresses on the Web, or on the user's own disk drive. When done defining hotspots, the author then saves the image as another GIF file. All of the hotspot and URL information associated with the image is saved *in* the image file. All the author then needs to do is use the image in a Web page, just like any other GIF image. When WebRouser (or any other Polymap-aware browser) loads up that Web page, it automatically recognizes that the Polymap-enhanced image is different than the other GIFs, and the user can click on the image to travel to the different Web locations corresponding to the related hotspots.

WAPI Software Development Toolkit (WAPI SDK) (due in Q1 1996):

This inclused everything needed to develop sophisticated Weblet-based applications. The SDK is intended for in-house use, as an MIS product. It is also intended for product managers, to allow the development of prototypes as a step toward justifying their purchase of a product resale license. Many potential licensees will likely buy the SDK in order to investigate the technology, both to find out if they're already infringing, and to adequately evaluate the potential value of the Eolas technologies for their companies.

WAPI SDK Key Features:

- Includes a development **license** and internal use site-licenses, for 5 seats. Larger projects will require a separate license.
- **Documentation** with sample C++ source code and DLLs, for Weblet creation.
- A small group of **sample Weblets**, such as the MPEG player and 3D CAD file viewer with source code, fully explained in the documentation.
- A **Tcl/TK** graphical-scripting-language interpreter Weblet, together with several sample Tcl/TK scripts
- A **JAVA** scripting-language interpreter Weblet, with several sample JAVA scripts

Weblet Applications

"Plug-in" embeddable applications for the Web that will provide 80% of the functionality that 80% of people use 80% of the time, allowing us to bring efficient, compact, and immediately-usable products quickly to market.

We will also market and sell various Weblets that either have already been developed and enhanced (such as the VIS volume visualization application, the MPEG player, the CAD viewer, and the molecular modeling applet) or will be developed by users at educational and research institutions. As time goes on, this portfolio of marketable Weblets is expected to grow rapidly.

Our first Weblet-based applications, aside from those developed for the SDK, will be aimed at allowing MIS-style users, and other legacy-systems users to easily integrate their existing applications within Web-based environments. The first of these, **planned for release in Q1 of 1996**, will include:

- A general-purpose **terminal emulator** Weblet, supporting all of the poular terminal emulation modes. including IBM 3270, VT 100, ANSI BBS, Telnet, etc.
- A **database front-end** Weblet, with support for accessing back-end ODBC- and SQL-based DBMS systems over Internet-standard networks.

Subsequent Weblet development will be aimed at developing key components for suite-style applications, basic modules that can be mixed and matched to fufill various user needs. This will be best exemplified by the WebWorks packages.

WebWorks (AKA the Web OS v. 1.0 & 2.0)

WebWorks will comprise fully integrated family of secure workgroup Weblets for Internet-based enterprise-wide applications. This product will be aimed at a similar audience to that of IBM/Lotus Notes, although enabling far broader capabilities. WebWorks will introduce the concept of a ***peer-to-peer Web,*** and will begin to exploit Eòlas' unique concept of mainstream productivity applications which work largely by manipulation of ***pointers*** to data (the Web's URLs and their URN and URC decendents), rather than manipulating the data themselves. This will drastically increase the functionality and efficiency of group interaction, as well optimize network resource usage.

Version 1.0 (due in Q3 1996) will include:

- **WebManager** - A combined Web browser/editor/server Weblet application (the Web OS "kernel") The functions of the Web browser and HTML editor will be combined with a custom Web server engine, allowing every user on the Web to both access information provided by others and to share one's own information with both individuals and groups in a secure manner. This application will also be embeddable, as a Weblet, enabling the concept of "**subdocuments**," which have been long requested by users of the Web. This will allow the various members of a group to work on separate components of a common document, with immediately updated views of the parent document as a whole.

- **WebLock** - security engine
  This Weblet application will use the Pretty Good Privacy public-key encryption system to control access, encryption and digital signature authentication for all information storage and exchange by the other Weblet apps.

- **Webrary** - personal information manager
  This will be an expansion upon the currently-popular concept of a bookmark manager, providing abstracting, annotation, and archiving of a user's information, with easy to use and effective searching capabilities. The embeddable nature of this application will allow innovative new ways for Web documents to act as interfaces to large and complex information systems.

- **WebFile** - Iway file manager
  This will look and feel to the user very much like the file manager in Windows, but with seamless integration of files on the user's own machine and those on remotely-networked systems, allowing Web users to easily browse through both local and remote files.

- **WebMail** - email communications manager
  This will involve a combination of editing and sharing local Web documents, with precise management of encryption, access control and digital signatures. Users will send (using conventional POP mail technology) to message recipients pointers and "tickets" for messages that they have written, with the actual communication of the encrypted message occuring through the Web server, *only when the recipient wants to read a particular message*. The recipient will then have the ability to replicate the message to his/her own system, or just save the pointer/ticker, for future reference. This greatly reduces the amount of redundancy of information currently rampant in email systems, and provides inronclad security for electronic communications.

- **WebGroup** - groupware manager
  This will provide groups of users with something that looks and feels very much like the USENET newsgroup systems currently popular on the Internet, but where pointers to WebMail messages are managed and manipulated, rather than the messages themselves. The WebWorks security engine will allow groups of people working on sensitive projects to share information within the group freely, while protecting the confidentiality of each piece of data.

- **WebGraph** - presentation graphics system
  This Weblet will provide the capabilities commonly found in today's popular presentation graphics systems, such as Power Point and Freelance, but using VRML to allow tight platform-independent integration with other Web-based media.

Each of these Weblets will be, of course, embeddable within Web documents, but the user will be able to run them in stand-alone mode as well. Sophisticated replication and archiving capabilities will allow automated backups, versioning, and content maintenance in the various Web "trees" created by users.

Version 2.0 (due in Q1 1997) will add:

- **Enhancements to the version 1.0 Weblets**, tightening integration and expanding functionality

- **WebBot** - "spider"/macro recorder-player
  This Weblet will combine an internal macro scripting language with a macro recorder to provide users with a tool to allow easy automation of routine tasks in the system. The tool, when combined with the scheduler Weblet app (below) will also enable easy creation of autonomous Web agents, or "spiders" or "knowbots" that go out on the Internet and automatically find information of interest for automatic integration into local projects.

- **WebView** - multi-format media player (2D, 3D, Sound)
  This Weblet will allow the user to add support for new media formats to existing WebWorks-based applications in a modular fashion, without having to reengineer the whole system.

- **WebEdit** - programmer's text editor
  An advanced programmer's text editor Weblet with all the bells and whistles, yet embeddable.

- **WebBase** - relational database system
  A relational database Weblet that allows data elements (records, tables, forms) to be drawn from various locations, either local or remotely-networked.

- **WebSheet** - spreadsheet application
  This Web-aware spreadsheet Weblet will allow individual cells and ranges to be addressed as Web data objects, allowing data and formulas to be shared across the net. This takes the existing paradigm of 3-dimensional spreadsheets one enormous jump further, by creating the concept of the **hyper**-dimensional spreadsheet.

- **WebPlan** - project management module
  This Weblet will provide WebWorks-based workgroup managers to coordinate and track the various components of large and complex WebWorks projects. It will be able to automatically access various components of WebWorks projects to provide instant "snapshots in time" of work in progress and resource utilization.

- **WebTime** - calendaring/scheduler module
  This Weblet provides timekeeping, calendaring, and scheduling functions for other Weblet apps. as well as providing users with a personal time management tool.

Users will purchase WebWorks initially for its unique features. Robust file compatibility with other companies' productivity applications will allow transparent exchange of data with others and easy incorporation of older files into new projects. As time goes on, however, more and more of users' day-to-day computing activities will be able to be accomplished entirely within the WebWorks environment. As this occurs, it will become less and less significant to the user what operating system thay are using, since their applications will look and work identically on Windows, Mac and Unix machines.

The Eòlas Web OS (AKA WebWorks version 3.0, due Q3 1997)

This package will expand upon and enhance the WebWorks suite. It will be available either as an add on to the user's existing operating system, or with its own bundled operating system (probably Linux), obviating the need for the user to purchase Windows 95, 6, 7, etc., for example. Central to the concept of the Web OS will be the ability for the user to access and exploit remotely-networked applications and computational engines, probably on a subscription basis. This will free users from concerns over local computational resources and the varagies of data compatibility, and allow them to concentrate on the tasks at hand. Such features, together with other advanced functional capabilities of the system will make the Web OS the preferred environment for *all* personal computing applications within this decade.

Muse (*to* be released in Q4 1995, demonstrable now)

In addition to our Web-centric applications, we have developed an addictive new MetaMAP-based multimedia game that we feel has the potential to become the game sensation for the Web generation. This program presents the user with what appears to be a simple bit-map painting program, with small animated objects roaming across the surface. As soon as the user selects a color from the palette bar and begins to draw in the window, however, s/he immediately discovers that there is more to this game than meets the eye. As one of the roaming "sensors"

encounters a line drawn by the user, the moving sensor object program explodes with a MIDI musical tone. where the pitch and/or the instrument is determined by the color index of the line drawn by the user. This allows the user to "doodle" in color. The simplicity of the interface, and the appealing sounds produced even by random scribbles make using Muse a meditative, almost hypnotic, experience. If the Muse window is minimized, the music continues to play in the background, as the user works with other applications.

Once a pleasing image is produced, the rusulting image can be saved to disk as a normal GIF image, so groups of Muse devotees can share "scores" with each other. GIF images produced by other graphics applications can be imported into the program, and played by the user. Putting Muse-generated image on a Web page, and configuring any Web browser to use Muse as a helper application allows large collections of Muse scores to be shared across the Internet. We intend to release a Weblet version of Muse in the near future, that will allow playback *within* Web pages. We intend to sell muse as a shareware product, initially, then to find an OEM to sell a more robust commercial version. Another potential market for the program is as a screen saver. We expect that Muse will become an almost certain hit, and produce significant revenues for Eòlas. The fact that it is based on the patented MetaMAP process means that the game cannot be "cloned" by competitors, without opening their companies to serious liability for patent infringement damages..

4.1.1 Product Pricing Model

A free-for-noncommercial-use version of WebRouser was released on the Internet in August in order to build grass-roots support for our technology. Commercial versions of WebRouser will be sold through OEM channels. Since this product is based upon enhancements to Spyglass/NCSA Mosaic, a royalty is due Spyglass for each copy of WebRouser sold. Spyglass' pricing is by numbers of end-user products to be sold : 10,000 units at $8 per unit; 25,000 units at $5 per unit; and 100,000 units at $2 per unit. Their philosophy, which we agree with, is to encourage the customers to make larger initial commitments, in order to secure lower per unit prices. We will therefore follow a similar schedule for WebRouser pricing:

WebRouser OEM Sales:

Units (min. commitment)	Unit Price	Total
10,000	$ 15.00	$150,000
25,000	$ 10.00	$250,000
100,000	$ 4.00	$400,000

Direct-Sales Products:

Product	Price
WAPI SDK	$ 200.00
Weblets	$ 20.00
WebWorks	$ 200.00
Muse	$ 30.00

4.2. Licensing Program

Once the Weblet API patent is approved, OEM licensing will generate a significant share of Eòlas' revenues. Pending approval of the patent, we are pursuing methods to establish WAPI as a "must have" technology. We will

do this through several strategic cooperative agreements with major Web browser companies and other software applications developers. In order to provide an incentive to early licensees, we will deeply discount (by as much as 50%) licenses during the patent-pending phase. Another incentive to prospective licensees will be to minimize their risk for future liability under the patent. Finally, the functional advantages of the technology and the appeal of widespread compatibility and interoperability of software products for users of our API will be perhaps the most powerful reason for purchasing licenses.

There will be two major categories of WAPI licensee. The first will be the Web browser manufacturers, such as Netscape, Spyglass, Sun, AOL/Navisoft, Compuserve/Spry, and IBM. This category of customer will license the WAPI browser-side technology. We intend to provide these licenses either free or at minimal cost, obligating these manufacturers only to support our API in their products, and to conspicuously mark their products as such. The browser-side licenses will specifically ***not*** cover Weblet applications.

Over time, widespread support in browsers will establish our API as a *defacto* industry standard. We intend to build Eòlas around revenue streams generated by the server-side (Weblet) technology, through both product sales and licensing the WAPI others. To this end, the second category of licensees will include both software developers and online information providers who wish to capitalize on the WAPI framework both to sell products and services relating to the Web, and to develop Web-base applications for internal company-wide use. Two types of licenses will be sold : internal use licenses (which are not for resale) and licenses for development of products for resale. Simlar to the strategy for WebRouser pricing, larger initial commitments by customers will garner lower per unit prices.

4.3.1. ***Internal-use site licenses:***

- For development of Weblet-based applications for use within companies and institutions, but not for resale outside of the licensing entity.

Pricing (during patent-pending phase):

Seats	Price per seat	Total $
1	20.00	20.00
5	15.00	75.00
25	10.00	250.00
100	8.00	800.00
1000	5.00	5,000.00
10,000	4.00	40,000.00

4.3.2. ***Resale Licenses:***

- For development of Weblet-based applications for resale, based upon bundles of unit licenses

Pricing (during patent-pending phase):

Qty. of Units	Unit Price	Total $	
1000	8.00	8,000.00	
10,000	5.00	50,000.00	
50,000	2.00	100,000.00	
200,000	1.00	200,000.00	
500,000	0.80	400,000.00	
2,000,000 or more	0.50	1,000,000.00	(or more)

After the WAPI patent issues, these prices will be uniformly raised by 50%. This fact will be advertised widely to encourage early licensing by customers.

5. MARKET ANALYSIS

Our marketing plan is currently in development. What follows under Market Analysis and Marketing Strategy is not yet complete at the finest levels of detail, but the general thrust of the approach descibed below is accurate.

Introduction

A visit to the "Web Crawler" URL index via Netscape's WWW home page last Spring showed approximately 1.53 million unique URLs currently present on the WWW. Among these sites are over 2500 verified commercial WWW servers. This figure is increasing by 50% per quarter, and the rate is accelerating rapidly. The January '95 issue of Boardwatch magazine reported that there were then at least 3,600,000 verifiable users of the full access to the Internet. This figure was instantly doubled as AOL, Compuserve and Prodigy rolled out full WWW access by May of 1995.

These figures, while large, represent only the tip of the iceberg. Spyglass reported in January '95 that it had then already licensed 16 million copies of MOSAIC. Spyglass MOSAIC is now bundled with Microsoft Windows '95. Microsoft projects that 30 million copies of Windows will sell in 1995 alone. Add to that another 10 million users of IBM's OS/2 Warp, which also includes a Spyglass-licensed Web browser. Netscape, Inc. reports 4 million users of its Navigator WWW browser. By the end of 1995, virtually all of these browsers will either support our WAPI technology or will likely be infringing the pending patent, when it issues, through something that implements functionally-equivalent features to those covered by the claims in the patent.

It's reasonable to expect that 10 million people will be regularly using the WWW by 1996, accessing at least 5000-10,000 commercial servers. A conservative estimate for the end of 1997 would be 30,000 servers, and at least 100,000 by the end of 1998. We anticipate that 20% of these server operators will license the WAPI technology in 1997, and 30% by 1998. Our licensing revenue projections are based upon these assumptions.

5.1 Market Definition

The key point in defining the major market for Eolas' technology is the WWW and companies that have (or will have) active involvement with client/server development for Web-based environments - either with tools, applications or distribution of content. This market is based on a universal desire for improved information transfer and communications via the Web.

Needs and Wants

Users (client side) of the WWW (and other networks) are looking for affordable and easy-to-use enhancements of the ability to interact with larger, more complex applications. Developers are seeking ways to provide these enhancements. Developers are also seeking ***defacto*** standards for application and content delivery across the Web. Several competing industry leaders are currently fighting with each other to impose their own standards.

The explosive recent growth of client and server products for the Web has resulted in the immediate need for the enhanced capabilities that Eòlas can provide. Our technologies remove the current limits on network interactivity and open whole new dimensions of opportunities for delivering more powerful, efficient and cost-effective applications on the Web.

Currently, Eolas is unique in its ability to provide workable solutions to the needs for standards, power, efficiency and cost-effectiveness in WWW applications, due to the fundamental nature of our patented technologies.

Growth and Trends

The stability of this market segment is volatile, but rapid growth is projected by industry forcasters, based on WWW browser performance and the growth in the number of commercial Web sites over the past two years. Saturation of the market is not forseen, except in common projections such as: "everyone will be on the Web by the year 2000". It is also too early to develop quantitative data on the WWW market size. The reaction of Wall

Street to the recent IPOs of the two leading browser companies, Netscape & Spyglass, indicates that the tremendous potential of the Web is widely recognized.

The entire WWW is experiencing rapid early growth, including quickly expanding numbers of users (private and commercial), servers, URLs, advertisers (home pages), and related products and services. Industry analysts agree that there will be dynamic growth for at least the next decade.

The foundation for the explosion in popularity of the Web was the development of fundamental products including simple easy to use browsers, servers, applications and development tools, and ever-widening access. Recent growth has been based on improvements to the fundamental products which enable enhanced interactivity and performance. Referenced sources agree that the major trend for the next two years will be: "the ability to embed small aplets in Web documents", "development of distributed or shared resources" and "potential use of the WWW as an operating system;"

5.2 Our Customers

CONSUMER SEGMENTS:

Because clarifying consumer segmentation is a long term goal for the fledgling WWW and interactive media industries, (and for Eòlas), this section of the plan will be further developed during 1996. For now it will be passed over in favor of non-consumer, or organization segments which offer more immediate promise.

We recognize the importance, however, of having a clear picture of consumers. Accordingly, beginning during our mid-term stage, we will identify and track consumer segments by means of original and secondary-source market research. We will develop relevant data on consumer "demographics" (geographic location, age, sex, education, income and other vital statistics), and on "Psychographics", "Sociographics" and "Synchrographics" (awareness, attitudes, motivations, preferences, lifestyles, needs, interests, etc.). These will be used to guide R&D, product testing, media selection and creative strategies, and the like.

ORGANIZATION SEGMENTS:

We have selected several market segements for immediate attention. Selections are based on the size, state of WWW sophistication, current interest, and ability (theirs and ours) to move quickly into significant programs incorporating Eòlas' technology.

Primary market segments include:

Browser companies
Applications tools developers
Software OEMs
Corporate MIS departments

We will sell these market segments licenses for application development, the WAPI SDK, as well as various general-purpose Weblet-based applications.

Secondary market segments include, in part:

Internet Marketing and Advertising
Transaction Security Providers – EDI, EFT
Enterprise-wide Client/Server Vertical Market Applications Developers
Teleconferencing and Groupware Systems Integrators
Software Trial-Use and Sales Systems / Online Catalogs

CAD and Industrial Design Applications and Service providers
Educational publishers
Virtual Reality applications and services providers

... in effect, any business or institution that desires to use the Web to do business ...

Our primary customers include...

Applications Developers
Applications Tool Developers
Content Providers

...developing Web-based applications...

For sale to end user
For sale to resellers (OEMs, VARs)
For internal use

Our most typical non-consumer customers are medium to large size organizations (and some trend-setting smaller ones) who (want to) develop software tools, applications and/or content for distribution on the WWW. This includes a broad range of of organizations in almost every conceivable field : consumer, commercial, industrial and institutional.

Our primary customers are likely to be familiar with the technicalities of developing and marketing applications for the Web. They will readily understand and accept our licenses, products and services, provided that we make them convenient, affordable and functionally superior.

Our target prospects include all those (inside and outside our target organizations) who initiate the inquiries for our offerings, who influence the decision(s) to buy them, who decide which licenses, products or services to buy, and who permit the purchases to be made.

The principal buying motive is to aquire enhanced WWW capabilities at affordable cost. It is easy to understand why. The Web is seen as the greatest advancement in communications since the printing press, but with few of the costs and limitations of printing or other current forms of communications. Organizations throughout the world are eager to take advantage of the Web's power and incredibly low cost for a multitude of applications, and are buying enhancements as fast as developers can provide them. Our technology represents not just enhancements, but a quantum leap to a whole new dimension of capabilities, with lower costs and greater efficiencies. Our main job is to get our potential customers to understand this.

There are a variety of complimentary products already in use by our customers (such as Web browsers, scripting languages, and other tools and applications) that will work with our technology. We see these as launching pads for our technology and as a tremendous help in compelling customers to acquire and use our licenses, products and services.

The marketplace is just beginning to become aware of Eòlas. Recent articles in the trade and general press have identified us as a start up company with a patent pending for a technology that can be the foundation for the WWW of the future. Our customers' attitudes are malleable and ready to be formed. Publishers clearly recognize the general nature and potential impact of our technology. The general public does not yet know much about Eòlas, but are very interested as a result of our recent national press coverage. We will attempt to capitalize on this interest in every conceivable way. Through a series of carefully orchestrated steps (product releases, advertising, promotion, partnering efforts, and the like), we will build Eòlas' market identity and image as a well managed, competitive, innovative, technically proficient and ethical industry leader who can provide the WWW enhancements that the market wants and needs.

DECISION MAKER AND INFLUENCERS:

In addition to industry and organization data, we are working to develop an extensive profile database of key people inside and outside the industry who initiate the inquiries, influence the buying decision(s), decide what to buy, and authorize purchases. Our target customers within the organization include CEOs, CFOs, CIOs, VPs Sales/Marketing, MIS Mgrs., API Development Mgrs., Product Mgrs., Tech Services. Profiles will include useful data, for example:

(CORPORATE EXECUTIVE)
Title:
Power:
Viewpoint:
Position:
Emotional Influences:
Practical Influences:
Education:
Limitations:

PROSPECTS LIST

Eòlas has received strong early expressions of interest from a varied group of large corporations. After the University released a brief technology announcement in August '94, over 100 companies and institutions contacted the UC Office of Technology Transfer wanting more information about obtaining licenses for the technology. Approximately 35 of these potential licensees signed nondisclosure agreements with the University in order to receive more detailed information on the technology. Hundreds of other high potential prospects have emerged in a variety of ways, eg., in response to recent press. In addition, we have identified a list of several hundred very likely prospects from commercial and institutional segments that we consider to be ready-to-buy now or in the near future.

In the third and fourth quarters of 1995, Eòlas will contact these high potential prospects directly to inform them further about our products and technologies and to generate leads and sales.

CUSTOMER CASE HISTORIES

We will maintain a program of documenting customer case histories and their selection, applications and successes with our licenses, products and services. This will become an valuable sales tool, beginning with the first few entries.

GEOGRAPHIC SEGMENTATION:

We will concentrate initially on markets in the United States and Canada in order to effeciently use our limited existing distribution channels. We are, however, planning to take advantage of lucrative international markets as well. The global access nature of the WWW will facilitate reaching foreign customers.

There are ready markets worldwide for our licenses, products and services, including major western and eastern European countries, the Far East, Austrailia, South Africa, etc. And, many of our U.S. customers have extensive. multi-national operations. Many of the companies that have already contacted us are in Europe and Japan. Accordingly, rapid expansion into international markets will be part of our planning. Because we have extensive contacts overseas, markets may be assessed sooner than would be normal in a startup company. (Contacts are

already well established in Johannesburg, London, Paris and Stuttgart.) Research through IDG Communications International Marketing Services is expected to indicate the potential for international licensing and sales.

5.3 Competition

Competitive Analysis

Eòlas currently conducts an ongoing research effort to track alternative solutions developed by others to the problems of embeding objects in distributed documents, in order to keep aware of the nature and extent of potential competition and infringers.

Competition is defined here as any alternate resource a prospective buyer could turn to to enhance the interactivity of Web-based applications, thereby potentially denying Eòlas revenue. Competition doesn't have to be from duplicate product offerings although Sun with its "applets" and other company/products that appear somewhat similar to unsophisticated users (probably partially infringing on Eòlas patent rights) need to be considered as potential threats. Also, our definition of competition is not limited to the Internet market alone. Other potential markets for competition include alternate forms of interactive communications, such as interactive TV, videotapes, kiosks, seminars, videoconferencing, presentation graphics diskettes, diskads. It is conceivable that these, or related, technologies could also pose potential, admittedly probably remote, threats to Eòlas' revenue stream.

In our defense, Eòlas' technology offers exceptional performance and low cost in a wide range of applications and across all platforms, especially with regard to adapting ***existing*** legacy applications to the Web. We currently have no competitors that can make a similar claim.

Competitive threats today come from:

- companies seeking to dominate WWW market segments (Netscape, Sun, etc.), and/or establish their own *defacto* standards,
- companies who refuse to license or are waiting until the patent issues before deciding to license, possible emerging/new technologies,
- other communications industries,
- foreign governments and companies (esp. Asia-Pacific basin) who refuse to defend U.S. patents/copyrights

Eòlas' technology performs in virtually all situations that our customers are likely to need. The ability to provide enhanced WWW performance with full capability on any platform and with any type of application is unique to our technology and our research indicates that our approach is superior to anything else on the market today.

In all comparisons, APIs based on Eòlas' technology will provide more features and have superior performance than competing products from other companies. In most cases, the number of differences will be substantial. A complete technical comparison will be prepared and made available as each Eòlas API is developed.

Licenses

Our primary source of revenues will be licenses to others to use our technology. The major source of competition would come from alternative/superior technologies. None are known at present. Competition will also come from infringers: API developers who refuse to purchase licenses, thus requiring costly legal action. We expect to minimize this situation and discourage infringers by obtaining a favorable decision in a court action against a carefully selected "model" infringer.

The only way that licensed APIs and non-licensed APIs compete is on basic functionality, features and price. APIs that do not incorporate Eòlas' specific technology will not provide the same capabilities in many situations where application performance and speed is a major motivation for the customer's purchase decision, such as sophisticated multimedia systems, games, virtual reality, large-scale databases, teleconferencing, etc.

Products

Eolas' will develop certain products, such as browsers, APIs and SDKs, that will compete with similar products from other producers. Our products will, however, have several important competitive advantages. They will primarily emphasize and promote the use of our technology. This will distinguish them from non-licensed competitors. Licensed competitors will be paying fees to Eòlas, which will give our products a slight price advantage. APIs incorporating our technology can be priced competitively, as our licensing fees will not add significantly to end-user unit prices. Detailed competitive analysis and comparisons will be developed and made available as each of our products is developed.

Services

Eolas will provide (in certain special circumstances) custom development services as a consultant or independent developer. This will typically involve competitive bid situations against other independent developers/consultants. Our advantages will be that we will concentrate in areas where we have exceptional expertise (eg., scientific publications, medical visualization, databases); we will emphasize our technology, and we will select areas which are strategically beneficial to Eòlas. All such activities that we envision at present will have a significant **R&D** dimension which will contribute to **future Eolas patents and products**.

Industry and Consumer Market Research

We have already begun, and will continue, an extensive effort to aquire strategic market data on our target market segments, leading organizations and prospects within the organizations.

Our management staff includes two members (W. Doyle and K. Bates) who have extensive experience with both primary and secondary-source market research (qualitative and quantitative; in business-to-business, consumer. industrial and institutional markets). We will continue to be committed to conducting quality market research and our ability to assure appropriate design, analysis and cost.

5.4 Eòlas' Strengths and Weaknesses

STRENGTHS

In terms of company strengths, EOLAS' products/services have several distinct advantages over the "competition". First, they are based on a marked advancement in technology. Second, the underlying technology is covered by a broad pending patent. Third, we have the unique ability to offer managers of MIS departments the ability to adapt their existing legacy systems to the WWW. Fourth, by concentrating on APIs, rather than particular programming languages, we can allow users to use the best application development tools at their disposal to develop high performance software. Finally, our tools and APIs greatly simplify the task of integrating applications with the WWW, enabling production efficiency, ease of implementation for the Web application producer.

In marketing, our most powerful assets are 1) a unique, proprietary technology with large, ready markets and 2) strong marketing and sales management with years of experience and success in launching software products and services.

In pricing, our strengths include 1) a broad ranging market so that we will not be limited to pricing driven by a few, powerful customers, 2) the ability to profit from high volume and low initial prices, and 3) the flexibility to easily revise pricing strategies to maximize profitability based on the lack of direct competition.

We can differentiate our offering from the competition in terms of both performance and technology. At the present time there are no direct competitors for the full breadth of applications areas we can address.

WEAKNESSES

Any weaknesses inherent in our products or technologies are unknown at this writing.

MARKETING WEAKNESSES:

Among Eòlas' apparent weakness are the ususal lack of resources typical in startup companies. Specifically, our only notable marketplace disadvantages are brand awareness/product recognition, lack of collateral, lack of documentation, and a generally-negative attitude towards software patents among rank-and-file programmers. The fact that the patent is currently pending. and has not yet been awarded by the Patent and Trademark Office may cause customers to adopt a "wait-and-see" attitude, risking higher royalty rates and potential legal damages against the chance that the patent will not issue.

CORPORATE WEAKNESSES:

Our only apparent corporate shortcomings at this time focus on the need for further financing and the pending nature of the Weblet API patent.

5.6 Risks

Various risk factors, such as product competition and legal challenge to the relevant patents, may reduce the company's ability to meet our projected goals. We must keep alert to competitive activities and be ready to react tactically, if necessary. A "Red Flag" will go up within Eòlas if competitive prices are cut to a point at which we no longer can profitably compete, and we will attempt to take immediate action to correct the situation. Our pricing allows for considerable flexibility in order to counteract this. Also, we have budgeted a "cushion"into our cost estimates which would allow us to withstand a limited squeeze action by competitors. We will also be ready to employ sales promotions for tactical purposes, such as to communicate to the various unlicensed Web browser companies' potential customers that they are putting their own companies at risk for potential future liability if they were to develop and use applications based upon unlicensed Web browsers.

5.7 Opportunities

Based on existing conditions, introduced in the above analysis, it is apparent that our projected potential can be achieved with a comfortable margin for error. While much of our analysis is based on educated assumptions, our competitive position, market strategy and management team give us the ability to win big in this dynamic and lucrative marketplace. The preceding market analysis indicates that our upside potential justifies the risks with a reasonable margin of error. Our identified markets constitute a viable base, which we can develop cost-effectively via agressive sales, distribution and communications strategies.

6. MARKETING STRATEGY

Eòlas' marketing strategy is to gain market acceptance as a standard through licensing our API to industry leaders, and then to leverage the power of our technology in promoting the concept of Eòlas as the preferred provider of tools and applications that enable interactive applications on the WWW.

We intend to establish Eòlas' technology as the standard for enhanced WWW interaction across a wide range of application categories.

Eòlas and its technology are already widely recognized in technical circles, due largely to publications by the inventors through UCSF, and through the existence, since August, 1994, of its WWW site (http://www.eolas.com). Dr. Doyle has also made dozens of personal presentations at conferences, trade shows and privately to a variety of organizations. The uniqueness and potential power of the WAPI technology is readily perceived and acknowledged by the technically savvy. We are currently developing a full blown marketing/PR/advertising program aimed at raising awareness of Eolas and its technology, products and services broadly among both technical and non-technical audiences.

Positioning Eòlas as the leader in Web-enabling interactive technology is a common theme throughout our market communications planning. The fundamental nature and unique advantages of our APIs and products (technical. quality, and performance) can be exploited to arrive at a winning position in the consumer's mind. We want customers to see our technology and products as the platform on which the Web will be transformed into a new and more powerful means for doing everyday work. We want people to change their view of the Web as a utility that they can access from their personal computer to seeing the Web as a kind of "virtual computer" whose resources they can access from anywhere, even with inexpensive communications devices. We will conduct periodic market research surveys to determine awareness and attitudes towards towards Eòlas, its products, and technologies. The results of market research will guide refinement and redirection of our marketing comminications and development of products and services.

We intend to establish Eòlas' technology as the standard for enhanced Web-based interaction across a wide range of application categories. Long term, this position will lead to Eolas' products and technologies becoming the driving force behing the transition of the Web into a fully-funtional distributed operating system. In the short term, we will establish a strong presence for Eòlas via licensing, products and services which enable others to develop innovative applications using our standards.

Also, short term, we will market a number of our own products that are unique and several that compete to some degree with existing existing products. It is not our primary intention to compete with existing products such as WWW browsers, servers, API's, tools and the like. Our competing products, eg. enhanced browsers, will be to demonstrate the power of Eolas' technology. While potentially profitable, our competing products will be offered mainly to accelerate interest and broad acceptance. Our primary revenues and profits will come from licensing and from products and services that support our ultimate position objective as the provider of the enabling technology for the WEB OS.

Our primary position, therefore, is in direct competition with Microsoft (Windows), IBM (OS/2) and others who are currently marketing operating systems and products for individual personal computers. Eòlas intends to be the leader in establishing nothing less than a complete paradigm shift an Internet-based universal operating system.

6.1 Reposition The Competition

The WWW is in the early stages of development. Companies such as Netscape and Sun, that provided early incremental enhancements in functionality, are seen as leaders. We can reposition our competitors by emphasising the technical superiority of our products, and the unique functionality that our technology enables.

So many opportunities exist for applications based on Eolas' technologies that we cannot begin to fill these markets ourselves. Eòlas' primary business opportunity is to enable as many other companies as possible to use our tools and licenses to develop as many applications as possible. We will facilitate providing high performance Internet applications by licensing our proprietary enhancements to WWW interfaces (browsers) which are provided by ourselves and other companies including Mosaic, Spyglass, Netscape and the like. These enhancements are fundamental and can apply to all who access the Internet for whatever reasons and on whatever platform.

We see attractive revenue opportunities in three primary areas: licensing, products and services. In the long term, many applications are foreseen in a broad range of markets; consumer, industrial, commercial, government, etc. In the short term, revenue and profit generation is focused on existing and currently emerging markets, especially the exploding markets for WWW servers, browsers, scripting languages, utilities, tools and the like. Following is a brief synopsis of our current view of the best short term opportunities.

6.2 Product Strategy

Free Technology Licensing:
Eolas' general policy concerning licensing is to encourage widespread development and exploratory use by distributing free-for-individual-**noncommercial**-use licenses for its products. As we begin to release these products, we will strive to maintain this policy wherever possible. We do this with the intent to stimulate commercial applications on the Internet without constraining the drive that made the Net what it is today.

Spyglass, Netscape, Sun and IBM

Immediately following the finalization of the browser-side license from Eòlas to Spyglass, we will proceed to negotiate arrangements with Netscape, IBM, and the other major browser companies, to have those companies incorporate the Weblet API into their WWW browsers. Although these agreements are not intended to result directly in significant revenue for Eòlas, they will serve to make Eòlas' WAPI a defacto standard for Internet compound document representation and interprocess communication. We also intend to purchase a license from Sun for the source code and redistribution rights to their Java language interpreter. This will allow us to quickly develop and release a Weblet-based Java interpreter. Our Java Weblet application will allow *any* Web browser developer to easily add support for the Java language merely by adding support for our API to their products, and then bundling our Java Weblet with their product. This will put Sun into the difficult position of having one of two choices: 1) to license our patent and add WAPI support to their HotJava Web browser (and to advise all of their Java licensees to follow suit) or 2) to face the disturbing (to Sun) prospect of having the Eòlas Java Weblet represent the ***only*** liability free means for developers to exploit the capabilities of this powerful programming language.

Marketing Activities

Initial licensing deals will be located and negotiated by the management team, working from the initial list of contacts provided by the University, as well as the list of contacts which resulted from the two Eòlas press releases. When additional equity funding comes into Eòlas in December, a sales and marketing team will be hired and several marketing and sales activities will occur. These include retaining a PR firm, developing and placing strategic national ads, participating in select trade shows, instituting a nationally-advertised Weblet programming contest, and methodically following up on sales leads generated. A complete marketing communications plan is now under development and will be available shortly.

Partners Support Program

Each of our licensees will receive complimentary membership in the Eòlas Partners Support Program. This will entitle them to free technical support, development tools, and software tools updates for the lives of their licenses.

New Product Development Strategy:

We will leverage existing freeware/cheaply licensed source code for the development of demonstration Weblet applications. There will be a direct relationship between research projects and new product development. As stated above, commercial rights to individual Weblets developed under consulting contracts will be maintained by Eòlas, while using them as building blocks for creating proprietary systems for our clients. These Weblets will then be marketed as products in their own right.

We will port existing imaging Weblets to Mac and Windows platforms. Additional applications, requiring between six and nine months for design and fine-tuning, will be developed from scratch in key areas for vertical markets. Additional strategic cross-licensing will be pursued to use/implement Eòlas intellectual property for servers and applications, coupled with royalties to Eòlas, for other applications as appropriate (i.e. where internal development is not cost effective or strategic). Research funded through federal grants will be carried on and aimed at developing new marketable intellectual properties for Eòlas.

7. FINANCIAL

See the attached 4-year financial projection spreadsheets in Appendix 7 for more details.

7.1 Sources and Applications of Funds

Seed capital in the amount of $151,000 was raised during the first three months of Eòlas' existence, largely in the form of debt. A subsequent private offering of Series-A prefereed stock raised $490,000, based upon a pre-money company valuation of approximately $6.6 million..

The third phase of equity fundraising will be held at the beginning of December, 1995, where we intend to sell class B preferred stock in order to raise an additional $12 million in capital. These funds will be used to allow the company to drastically expand the marketing and sales efforts, as well as to support the development of strategically-important Weblet applications and the WebWorks workgroup software package. These funds will also cover other operating expenses through the second quarter of 1996, when sales will begin to sustain the company's cash flow.

7.2 Assumptions and Projections:

This plan assumes that the first office action from the U.S. Patent and Trademark Office, expected by early Spring 1996, will indicate that at least some of the 43 claims in the patent application will be granted, and that the patent itself will be granted by Jan 1, 1997. We therefore won't begin **aggressive** marketing of licenses until late Spring 1996. Therefore projected license income starts in August '96, although it actually might begin much sooner, and rises consistently through 1998. During this patent-pending period, the Eòlas will concentrate on product sales and agreements with strategic partners to build revenues as well as a long list of "happy licensees." This plan also makes the assumption that 20% of potential infringers will purchase licenses in 1997, and 30% will do so in 1998. If we succeed in establishing our technology as a *defacto* industry standard, then these estimates for market penetration may be very low.

Eòlas				
Income Statement Projection (1,000's)				
1995-1996				
	1995	% of Rev	1996	% of Rev
Revenue	**60**		**45,220**	
Expenses				
Cost of Revenue	$3	5%	$4,930	11%
Engineering & Tech Support	$78	130%	$3,656	8%
Sales & Marketing	$85	142%	$11,747	26%
G&A	$490	817%	$2,514	6%
Total Expenses	**$656**	1094%	**$22,846**	51%
Operating Income	**($596)**	-994%	**$22,374**	49%
Net Interest Income (expense)	$13		$11	
Provision for taxes @40%	$0		$8,950	
Net Income	**($609)**	-1015%	**$13,413**	30%
Net Profit Margin	**-1015%**		**30%**	
Total Employees	18		50	
Revenue per Employee	$3		$904	

Profit and Loss Statement, 1995-1996

Eòlas				
Income Statement Projection (1,000's)				
1997-1998				
	1997	% of Rev	1998	% of Rev
Revenue	**471,730**		**1,935,692**	
Expenses				
Cost of Revenue	$55,459	12%	$227,435	12%
Engineering & Tech Support	$16,166	3%	$42,646	2%
Sales & Marketing	$68,404	15%	$269,686	14%
G&A	$7,336	2%	$17,001	1%
Total Expenses	**$147,365**	31%	**$556,767**	29%
Operating Income	**$324,365**	69%	**$1,378,925**	71%
Net Interest Income (expense)	$0		$0	
Provision for taxes @40%	$129,746		$551,570	
Net Income	**$194,619**	41%	**$827,355**	43%
Net Profit Margin	**41%**		**43%**	
Total Employees	165		330	
Revenue per Employee	$2,859		$5,866	

Profit and Loss Statement, 1997-1998

The projected return on investment of these funds is significant. Early profitability is impressive, since products and services can be brought to market quickly due to the significant early development of the technology while within the University environment, and since the potential market includes *every* developer or provider of WWW-based information or applications. Conservative estimates are that over 100,000 such potential licensees will exist by the end of 1998. Each licensee acquired represents a potentially recurring revenue stream. Although our projections assume only a 30% penetration of this market by the end of 1998, the recurring nature of these revenue streams translates into annual revenues of over $750 million by the start of 1999.

A review of the projected profit and loss statements shows that by the end of 1996 Eòlas will have an annual operating profit of over $13,400,000 based on sales of approximately $45,220,000 in Weblet and MetaMAP-related products and services, even before the WAPI patent is expected to issue.

Assuming that the WAPI patent issues by the start of 1997 and licensing of the patent begins in full force, projected annual profits at the end of 1997 are over $194,600,000, based upon revenues of $471,730,000; and projected profits for 1998 are over $827,000,000, based upon revenues of $1,935,692,000, using conservative operating and expense figures.

8. APPENDICES

1) MEAPI/DHOE specification, supported in WebRouser

2) WAPI patentability report

3) MetaMAP patent

4) Eòlas Press Releases

5) Press Articles

a) "The (Second Phase of the) Revolution Has Begun," Wired, October, 1994

b) "At Demo 95 it became clear the Web is our next-generation OS," by Bob Metcalfe, InfoWorld, 2/27/95

c) "Patent War Pending Over 'Applets', Interactive Week, 8/28/95

d) "Eòlas Technologies Granted Exclusive Licensing Rights to Pending Web Applet Use," Comm. Week. 9/28/95

e) "Has Microsoft Reached its Peak?", Associated Press, 9/27/95

6) Research Papers/Proposals by Eòlas Founders

a) "Integrated Control of Distributed Volume Visualization through the World Wide Web," IEEE Vis '94

b) "Polymap: A Versatile Client-Side Image Map for the Web," 1995 International World Wide Web Conf.

c) Summary of 1994 research collaboration between Dr. Doyle and the NCSA Mosaic development team (Netscape's founders), documenting that they were aware of our technology in 1993

7) Detailed Financial Projections

a) 1995-1996

b) 1997-1998

Appendix 1: MEAPI/DHOE Specifications

Parent

Multiple Embedded Applications API (MEAPI)

Distributed Hypermedia Object Embedding (DHOE) is a patent-pending protocol which enables hypermedia browsers with EMBED implementation to view/manipulate the hypermedia objects embedded in HTML documents as inline embedded objects. For more information about DHOE, please refer to "Integrated Control of Distributed Volume Visualization Through the World-Wide-Web", published in IEEE Visualization '94 conference proceeding.

MEAPI is a simple application interface for Eòlas' DHOE-capable WebRouser (or any other similarly-capable browser) to communicate with an external application through the DHOE protocol. MEAPI provides the external application (DHOE server) with a way to coordinate data display as a hypermedia browser (DHOE client). We have only included the most fundamental/important messages in the current version:

Messages from DHOE server->DHOE client (External App -> Hypermedia Browser)

(a) Server updating: XtNrefreshNotify
(b) Server ready: XtNpanelStartNotify
(c) Server exiting: XtNpanelExitNotify

Messages from DHOE client->DHOE server (Hypermedia Browser -> External App)

(a) DHOE area shown: XtNmapNotify
(b) DHOE area hidden: XtNunmapNotify
(c) DHOE area distroyed: XtNexitNotify
(d) DHOE area button down: XtNbuttonDown
(e) DHOE area button up: XtNbuttonUp
(f) DHOE area button move: XtNbuttonMove
(g) DHOE area key down: XtNkeyDown
(h) DHOE area key up: XtNkeyUp

Interface Functions

The above messages are defined in protocol_lib.h, which is necessary to be included in your program.

The DHOE fundamental functions are provided in protocol_lib.c, and are listed below:

(1) void handle_client_msg(Widget w, caddr_t client_data, XEvent *event)
A function called back by XtAddEventHandler when it sees a message from the DHOE client (the Hypermedia Browser). Your program (DHOE server) should call

XtAddEventHandler(Widget app_shell, NoEventMask, True, handle_client_msg, 102);

to register this function with Xt. Here handle_client_msg will be called with parameters w=app_shell, client_data=102, and event pointing to an XEvent structure generated by Xt when it sees the message. app_shell is usually the application shell returned by XtInitialize, or XtAppInitialize, or XtVaAppInitialize.

(2) void register_client(Widget w, Display *remote_display);
To register your program with the DHOE client.

(3) void register_client_msg_callback(char *msg, void (*function_ptr)());
To register a function to be called back when Xt sees a string that matches msg. This function may appear any where in your program. You do not need to handle the XtNmapNotify/XtunmapNotify pair for DHOE servers deiconify/iconify when they receive these messages. You will need to specify a "quit" function to shutdown your application gracefully on XtNexitNotify. Button/key message handling are optional. You may call get_mouse(int *x, int *y) to get the mouse coordinates in the button handling functions, and get_keysym(KeySym *keysym) in the key handling functions. Keysym's is defined by X11 (in keysymdef.h) for cross-platform compatibility.

(4) void send_client_msg(char *msg, Display *remote_display, Window remote_window);
To send a message with value msg to the DHOE client at a display=remote_display and has an X Windows ID of remote_window. The remote_display and remote_window will have to be provided. This function may appear any where in the program after register_client.

Here is a simplified sample program outline:

```
...
#include "protocol_lib.h"

...

/* X-way to define resources and parse the cmdline args */
/* xmosaic-2.4v gives the embedded window information through these args */
typedef struct{
        int     win;
        int     pixmap;
        int     pixmap_width;
        int     pixmap_height;
        char    *datafile;
} ApplicationData, *ApplicationDataPtr;

static XtResource myResources[] = {
        {"win", "Win", XtRInt, sizeof(int),
         XtOffset(ApplicationDataPtr, win), XtRImmediate, 0},
        {"pixmap", "Pixmap", XtRInt, sizeof(int),
         XtOffset(ApplicationDataPtr, pixmap), XtRImmediate, 0},
        {"pixmap_width", "Pixmap_width", XtRInt, sizeof(int),
         XtOffset(ApplicationDataPtr, pixmap_width), XtRImmediate, 400},
        {"pixmap_height", "Pixmap_height", XtRInt, sizeof(int),
         XtOffset(ApplicationDataPtr, pixmap_height), XtRImmediate, 400},
        {"datafile", "Datafile", XtRString, sizeof(char*),
         XtOffset(ApplicationDataPtr, datafile), XtRImmediate, NULL},
};

static XrmOptionDescRec myOptions[] = {
```

```
        {"-win", "*win", XrmoptionSepArg, 0},
        {"-pixmap", "*pixmap", XrmoptionSepArg, 0},
        {"-pixmap_width", "*pixmap_width", XrmoptionSepArg, 0},
        {"-pixmap_height", "*pixmap_height", XrmoptionSepArg, 0},
        {"-datafile", "*datafile", XrmoptionSepArg, NULL},
};

ApplicationData myAppData;

void myDraw()
{
    /* do your drawing... */
    ...

    /* if you draw into your own drawables (myPixmap in this case) */
    if (myAppData.win) {
        /* copy from myPixmap to the "shared" pixmap */
        XCopyArea(display, myPixmap, myAppData.pixmap, myGC, 0, 0, WIN_WIDTH, WIN_HE
        /* tell Mosaic to update the drawing window */
        send_client_msg(XtNrefreshNotify, display, myAppData.win);
    }
}

void myQuit()

{
        /* tell Mosaic you are exiting... */
        if (myAppData.win)
                send_client_msg(XtNpanelExitNotify, display, myAppData.win);

        /* Motif way of exiting */
        XtCloseDisplay(XtDisplay(any widget));

        exit(1);
}

...

main()
{
    Widget app_shell;
    ...

    /* XtInitialize does XOpenDisplay, as well as creates a toplevel widget */
    app_shell = XtInitialize("wt", "Wt", myOptions, XtNumber(myOptions), &argc, argv
    ...

    /* This func fill up myAppData with the user specified values/default values */
    /* We get the embedded window's info this way */
    XtGetApplicationResources(app_shell, &myAppData, myResources, XtNumber(myResourc
    ...

    /* if we have an external window to display the image... */
    if (myAppData.win) {
        XtAddEventHandler(app_shell, NoEventMask, True, handle_client_msg, NULL);
        register_client(app_shell, display);

        /* register the func to be called when Mosaic exit */
        register_client_msg_callback(XtNexitNotify, myQuit);

        /* tell Mosaic you have started fine */
```

```
        send_client_msg(XtNpanelStartNotify, display, myAppData.win);
    }

    ...
    XtMainLoop(); /* Motif's event loop */
}
```

Parent

Distributed Hypermedia Object Embedding (DHOE)

1.0 Introduction

DHOE is a patent-pending mechanism to do object embedding in a distributed hypermedia world. It utilizes hyperlinks to achieve the effect of embedding of hypermedia objects residing anywhere on the Internet in hypertext documents.

With DHOE, hypermedia browsers no longer have to have built-in capabilities to decode all types of hypermedia data, nor do they have to depend on gateways to translate data into acceptable forms. DHOE-capable browsers would match the type of the hypermedia data with the potential external application capable of handling it. The preferred type/application matching database is the MIME database, but other databases are acceptable.

2.0 DHOE protocol

DHOE protocol defines the communication mechanism between a DHOE client (e.g. a hypermedia browser) and a DHOE server (e.g. an external application that handles DXF datafiles).

DHOE protocol consists of message strings. A predefined fundamental set of message strings include

DHOEserverUpdate : Tells a client to update data
DHOEserverReady : Tells a client the server is ready
DHOEserverExit : Tells a client the server is exiting
DHOEserverConfigureWin : Tells a client to resize/reposition the DHOE window

DHOEclientAreaShown : Tells the server the DHOE area is exposed
DHOEclientAreaHidden : Tells the server the DHOE area is being hidden
DHOEclientAreaDestroy : Tells the server the DHOE area is being destroyed
DHOEbuttonDown : Sends mouse pointer coordinates to the server on button down
DHOEbuttonUp : Sends mouse pointer coordinates to the server on button up
DHOEbuttonMove : Sends mouse pointer coordinates to the server on button move
DHOEkeyDown : Sends the corresponding keysym to the server on key down
DHOEkeyUp : Sends the corresponding keysym to the server on key up

The above are the most basic messages required for DHOE. Other messages may be defined.

2.1 Requirements

DHOE is currently implemented for X-Windows systems. Windows and Mac versions will be built atop

DDE/OLE and Apple Events respectively.

3.0 Programming with DHOE

Since DHOE is based on windowing systems' communication protocols, it is event-driven, and it utilizes callbacks extensively. Programming with DHOE involves initializing DHOE by installing a message-handling function, registering the DHOE client with the DHOE server, and registering various callbacks with their corresponding messages. The DHOE client/server may at any time after client/server registration, send messages to each other. The messages are character strings, and may be followed by different types of data. DHOE also supports buffer sharing (i.e. bitmaps and pixmaps) between DHOE clients and DHOE servers.

3.1 Why DHOE?

Programmers may be using X, DDE/OLE, or Apple Events directly, bypassing DHOE completely to achieve the same goal. In this case, however, they will have to build their own DHOE-like protocol, and deal with different programming conventions to do the same things on different platforms. DHOE provides an application interface that simplifies interclient communication programming in windowing systems, or more specifically, distributed hypermedia object embedding. For a rough guideline of DHOE programming, please refer to "Multiple Embedded Applications API (MEAPI)".

4.0 Extensibility

Extending DHOE is as simple as extending the DHOE message set. DHOE will soon be supporting communication of most of the mouse and keyboard events between DHOE clients and DHOE servers. In the near future, DHOE will be extended to support the common set of windowing messages of X, Microsoft Windows, and Macintoch.

Bates Numbers

E 006266 - E 006272

Not Used

Appendix 3: MetaMAP® Patent

United States Patent [19]

Doyle

[11] Patent Number: 4,847,604

[45] Date of Patent: Jul. 11, 1989

[54] METHOD AND APPARATUS FOR IDENTIFYING FEATURES OF AN IMAGE ON A VIDEO DISPLAY

[76] Inventor: Michael D. Doyle, 511 W. Oregon, Urbana, Ill. 61801

[21] Appl. No.: 90,112

[22] Filed: Aug. 27, 1987

[51] Int. Cl.[4] G09G 1/16

[52] U.S. Cl. 340/706; 340/703; 340/709; 340/799

[58] Field of Search 340/701, 703, 706, 707, 340/709, 710, 721, 723, 798, 799; 364/518, 521, 188, 189, 190; 434/322, 323; 273/DIG. 28

[56] References Cited

U.S. PATENT DOCUMENTS

4,074,254	2/1978	Belser et al.	340/799
4,200,867	4/1980	Hill	340/703
4,203,107	5/1980	Lovercheck	340/799
4,249,172	2/1981	Watkins et al.	340/799
4,303,912	12/1981	Stafford et al.	340/703
4,395,707	7/1983	Satrapa	340/703
4,414,636	11/1983	Ueda et al.	364/526
4,439,759	3/1984	Fleming et al.	340/703
4,441,104	4/1984	Finney, II	340/724
4,451,824	5/1984	Thayer et al.	340/720
4,471,465	9/1984	Mayer et al.	364/900
4,481,529	11/1984	Kerling	358/30
4,484,187	11/1984	Brown et al.	340/703
4,488,245	12/1984	Dalke et al.	364/526
4,517,654	5/1985	Carmean	364/521
4,520,454	5/1985	Dufour et al.	364/900
4,521,014	6/1985	Sitrick	273/DIG. 28
4,524,421	6/1985	Searby et al.	364/521
4,570,217	2/1986	Allen et al.	364/188
4,574,277	3/1986	Krause et al.	340/703
4,580,134	4/1986	Campbell et al.	340/703
4,583,186	4/1986	Davis et al.	364/526
4,586,036	4/1986	Thomason et al.	340/720
4,600,918	7/1986	Belisomi et al.	340/711
4,616,220	10/1986	Grunewald et al.	340/747
4,620,289	10/1986	Chauvel	364/521
4,648,028	3/1987	DeKlotz et al.	364/188
4,648,046	3/1987	Copenhaver et al.	364/518
4,648,050	3/1987	Yamagami	364/526
4,673,930	6/1987	Bujalski et al.	340/703
4,675,666	6/1987	Peterson	340/721
4,710,806	12/1987	Iwai et al.	340/703

OTHER PUBLICATIONS

Alan Borning, "Thinglab-A Constraint-Oriented Simulation Laboratory," Stan-CS-79-746, Ch. 2, pp. 14-37 (1979).

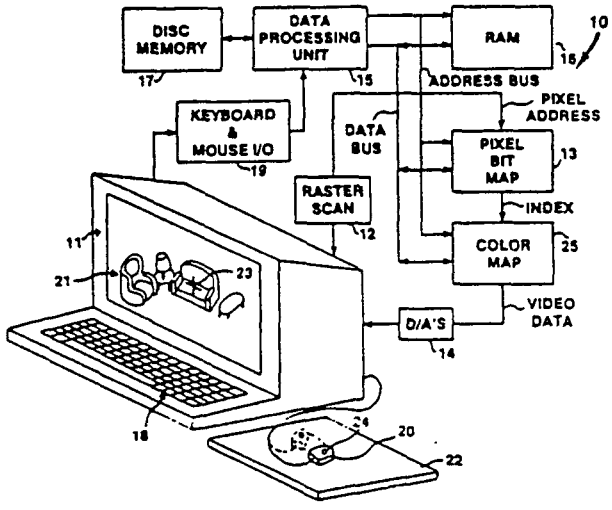
Steve Ciarcia, "High-Resolution Sprite-Oriented Color Graphics," *Byte*, pp. 57-70, 72, 76, 78, 80 (Aug. 1982).

Primary Examiner—John W. Caldwell, Sr.
Assistant Examiner—Jeffery A. Brier
Attorney, Agent, or Firm—Richard C. Auchterlonie

[57] ABSTRACT

A computer graphic interface allows a user to obtain descriptive information concerning a feature of a displayed image by pointing to the location of the feature. Conversely, the user may enter descriptive textual information, and the locations of responsive features are indicated. The data processing and memory storage requirements are minimized by encoding information about the image as a pixel bit map, and a color map in which the addresses or indices of the color map are correlated with the addresses or pointers to strings of descriptive information. Each color map address corresponds to a predefined set of features and descriptive information about those features. Since the pixel bit map defines a color map address for each location on the image, suitable programming of the color map can insure proper correlation of descriptive information with corresponding locations on the image. The correlation between color map addresses and the descriptive information about the features is represented most compactly by arranging or sorting the entries in the color map so that there is a correspondence between each predefined feature and a continuous range of color map addresses. Therefore, for a specified color map address, the corresponding set of features and their pointers can be found by comparing the specified color map address to the limits of the color map address ranges for the various features.

27 Claims, 3 Drawing Sheets



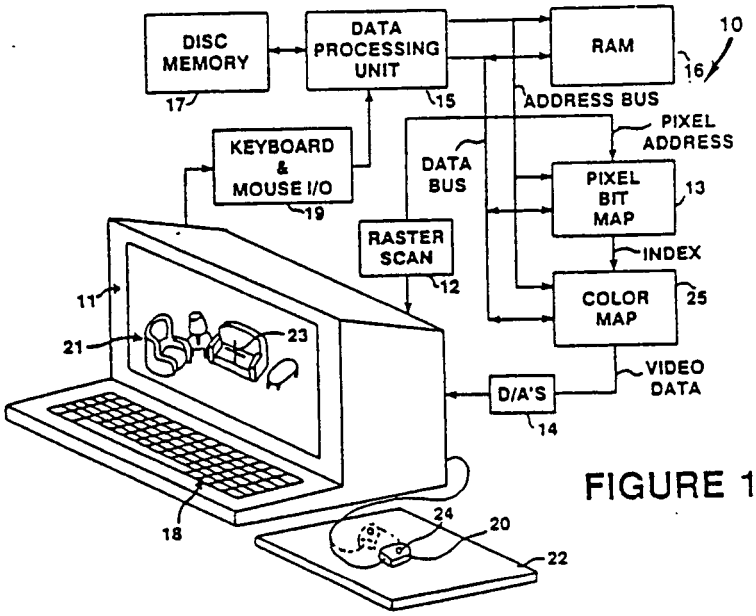


FIGURE 1

25

COLOR MAP

INDEX	RED	GREEN	BLUE	
0	0	0	0	SEG. 1
1	0	0	31	
⋮	⋮	⋮	⋮	
31	255	255	255	
32	0	0	0	SEG. 2
33	0	255	0	
⋮	⋮	⋮	⋮	
63	255	255	255	
64	0	0	0	SEG. 3
65	128	0	0	
⋮	⋮	⋮	⋮	
95	255	255	255	
⋮	⋮	⋮	⋮	⋮

27

TEXT POINTER TABLE (RAM)

INDEX	POINTER
31	STRING 1
63	STRING 2
95	STRING 3
⋮	⋮

26

POINTER	TEXT STRINGS (RAM)
STRING 1	C
	H
	A
	I
	R
STRING 2	L
	A
	M
	P
STRING 3	S
	O
	F
	A
	⋮

FIGURE 2

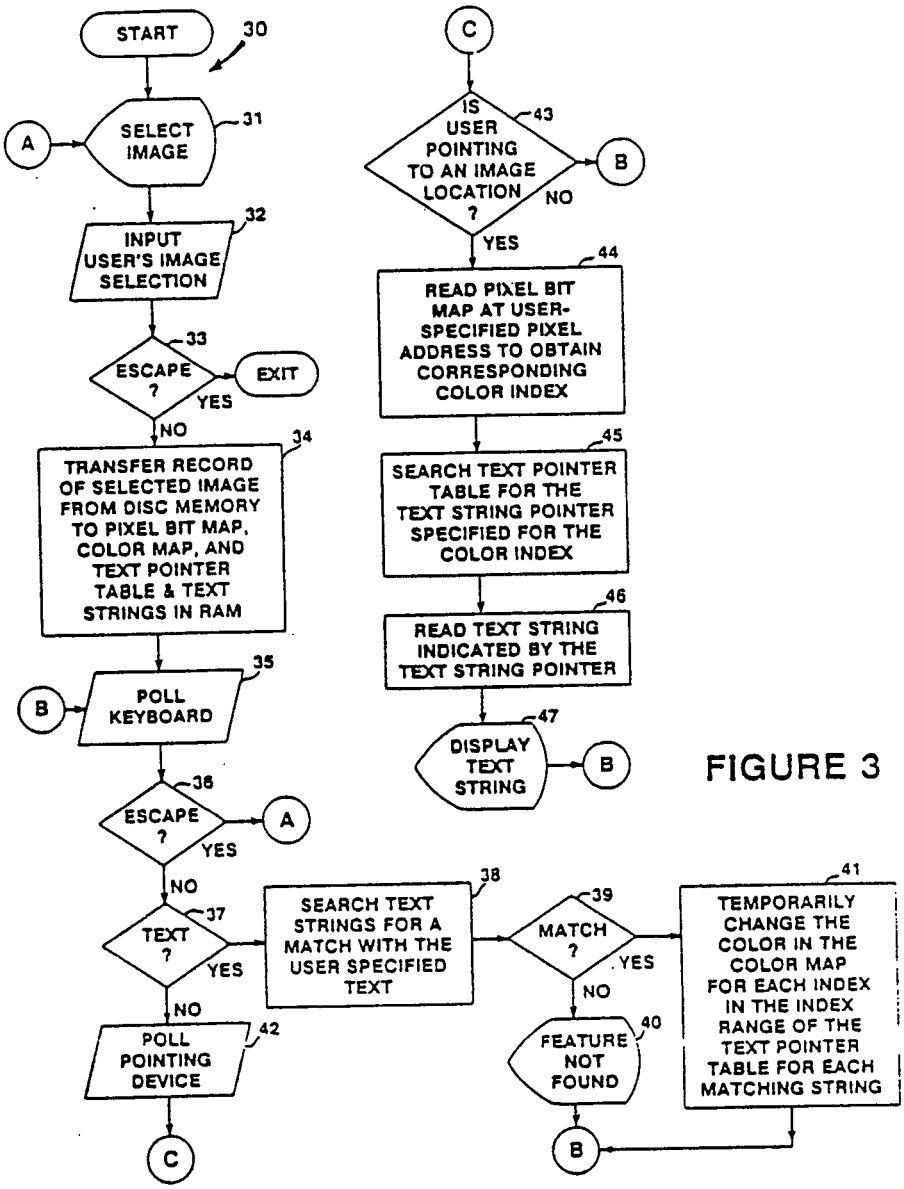


FIGURE 3

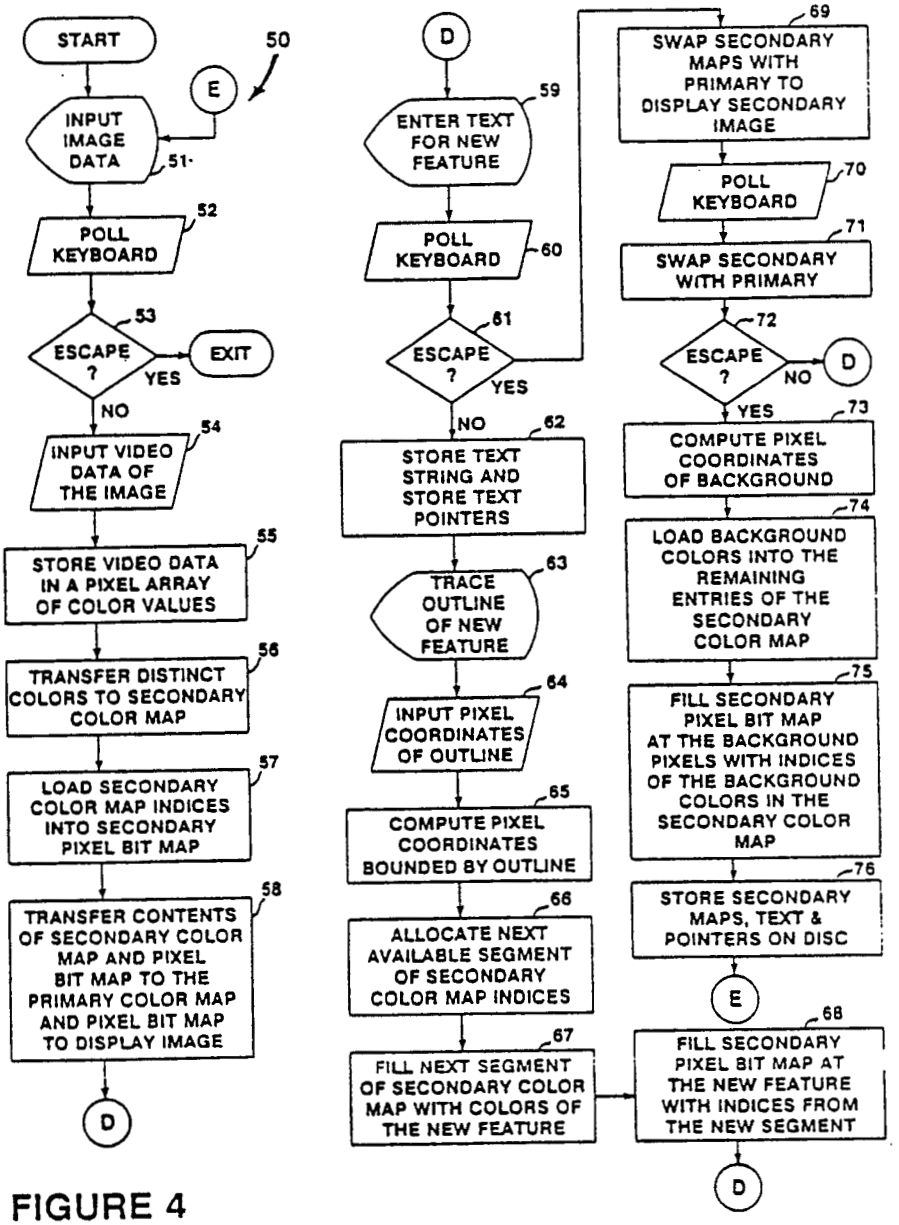


FIGURE 4

METHOD AND APPARATUS FOR IDENTIFYING FEATURES OF AN IMAGE ON A VIDEO DISPLAY

BACKGROUND OF THE INVENTION

1. Field Of The Invention

The present invention relates generally to video displays, and more particularly to display systems in which an image is displayed to enable a user to recognize distinct features. Specifically, the present invention is directed to the problem of correlating the features of a displayed image with information which further identifies or describes the features.

2. Description Of The Related Art

Systems for providing high-quality digital images on video display screens are well known. The most common is known as a bit mapped raster scan display system. Such a system includes a memory which stores a set of bits corresponding to each pixel location on the display screen, and the information in the memory is read out and displayed in synchronism with the raster scan. The memory including the set of bits for the pixels is known as the pixel bit map.

For displaying high quality images having a variety of gray shades or colors, the set of bits for each pixel may specify the gray shade or color of the pixel. About 24 bits of resolution—eight bits for each of three primary colors—are required for reproducing the entire spectrum of chromatocity and luminance within the resolving ability of the human visual system. In order to reduce the memory requirements for obtaining high color resolution, it is well known to provide a color look-up table or "color map" which is indexed by the set of bits for each pixel. In this case the color map need store only once each color that is actually found in the image being displayed.

The use of a color map has many advantages in addition to the reduction of the memory requirements for the pixel bit map. As described in Hill U.S. Pat. No. 4,200,867, the color map can be used as a palette allowing the video display user to electronically compose or "paint" video images. For mass storage of digital images or band limited transmission between video systems, the color map can be used to reduce the required bit transmission rate and bit storage capacity. In this regard, the color map can also be used in connection with other data compression techniques, such as in the color block coding scheme disclosed in Campbell et al., U.S. Pat. No. 4,580,134.

The use of a color map further permits all pixels having the color identified by a single index to be modified simultaneously without disturbing the pixel bit map. The modification can be done for one color at a time by storing the color map in a read-write or random access memory, and writing a new color value to the memory address corresponding to the index of the color to be changed. Fleming et al. U.S. Pat. No. 4,439,759, for example, describes the use of a data processor for providing color blinking by means of a linked list of multiple processes. For example, using this technique a ball may appear to bounce across an image, a river may appear to flow, or stars may appear to twinkle.

As described further in Brown et al. U.S. Pat. No. 4,484,187, it is possible to change all of the colors simultaneously by using a color map having a plurality of segments for the range of indices specified by the pixel bit map, and by using a data processor to select which segment is to be addressed by the pixel bit map. In other words, the data processor substitutes one color map segment for another color map segment in order to change simultaneously a plurality of colors. The memory capacity of the color map must be increased in this case, but in practice the substitution is easily performed by using a memory circuit having at least one additional address select line which receives an output of the data processor instead of the pixel bit map. Brown et al. also discloses that the substitution of one properly programmed color map segment for another can instantly provide different priority for various images stored in different planes of bit map memory.

In order to display moving objects which may obscure a background or obscure each other, it is common to provide a bit map memory which has, for each pixel, a respective set of bits for each object or the background. Each set of respective bits is stored in a respective "bit plane" for the bit map memory. The respective sets of bits for each pixel are compared to each other in a priority decoder which has an output which controls the select input of a multiplexer which selects one of the sets of bits for display. The priority decoder, for example, operates the multiplexer to select the set of bits which represents the largest binary number, or selects the set of bits which represents a non-zero binary number from the bit plane assigned the highest priority. In the latter case, the binary number zero represents a transparent area in the bit plane. See, for example, Ciarcia, "High-Resolution Sprite-Oriented Color Graphics," Byte, August 1982, pp. 57-80.

The priority of an object can also be encoded as an attribute separate from the color map index. Peterson U.S. Pat. No. 4,675,666, for example, describes a video display system in which a user can display "true" objects by individual pixels in a "bit-plane mode," or fixed objects in a "list mode." The fixed objects in a list mode are defined in "image tables" which contain their pixel-by-pixel description. A list mode screen memory is arranged as a "display list" of pointers to entries in the image tables, which include "attributes" of the images. These attributes allow each individual occurrence of an image to be altered, for example, by underlining or flashing. By using a suitable attribute list and a suitable priority decoding circuit, Peterson says that the color of a "true" object overlapping a fixed object of lesser priority may be modified such that it appears that the true object has entered the shadow of the fixed object.

Brown et al. U.S. Pat. No. 4,484,187 discloses that the priority decoding function can be fully programmable by providing a unique color map address for each combination of bits from the respective sets of bits from the various bit map planes. Specifically, the output from the memory for one bit plane is used as one portion of the color map address and the output from the memory for another bit plane is used as another portion of the color map address. Therefore, the collisions or overlap between objects in different bit planes provide unique color map addresses, and the color map can be programmed at these addresses to provide any desired color for the points of collisions or overlap. FIG. 6 in Brown et al. shows ranges of color map addresses assigned to respective objects in a specific example.

A color map display system allowing rapid shifting of subpictures and rapid changing of the subpictures or the overall picture (such as having invisible information suddenly appear on the display screen) is described in Allen et al. U.S. Pat. No. 4,570,217. As described in Col.

4, line 50 to Col. 5, line 17, the screen is broken down into a plurality of zones, each zone providing up to sixteen different colors selectable for each pixel in the zone. The 16 colors from each zone are selected from one of four color palettes. Each color palette in turn selects its colors from up to 512 separate colors. Typically, a zone comprises eighty pixels of graphic information and thus each of those eighty pixels can be selected to have any one of the zone colors. It is said that by use of the color palette technique in association with each of the plurality of zones, the graphics can present complicated displays. Also, a high level graphics language is described for facilitating the design and configuration of a process control system. The graphics language has static and dynamic commands for facilitating graphics display update on a real time basis.

The ability to recognize colors in a real-world digital video image is important for geological and agricultural assessments, military reconnaissance, city planning and land use, and coastal oceanography. Dalke et al. U.S. Pat. No. 4,488,245 discloses a computer and video display system which is said to be suitable for such color recognition applications. The color picture is digitized and represented as three eight-bit words for each pixel in a 1024 × 1024 pixel array image. The first value is an approximation of luminance, while the second and third values represent variations of chromaticity or chrominance. Pattern recognition techniques are used to determine if a given color value lies within a specified three-dimensional color subspace. A high-speed digital processing technique is said to enable the real time modification of any portion of the color image within the display. A weighting function is disclosed to enable the operator to modify or insert colors.

In Col. 15 lines 12–21, it is said that in operation a threshold circuit is used to generate a mask to enable the operator to determine if, in fact, a color selection circuit has correctly identified all of the objects having the targeted colors; the mask is automatically superimposed over all of the areas on the color display having the same color value as the pixels selected during the sampling process. In Col. 15, lines 55 to 58, it is said that if desired, additional circuitry may be incorporated in the data processing unit to enable the operator to identify certain areas by their physical coordinates using a cursor, and a light pen or stylus.

In simulation programs using the techniques of artificial intelligence, it has become common to provide a user interface which displays a schematic diagram of a system under consideration, and which permits the user to graphically select or modify components of the schematic diagram by manipulating a pointing device. Further, upon such selection or modification, certain textual attributes of the object will be automatically modified if necessary and displayed in a window area adjacent to the schematic diagram. In such programs, the coordinates of the cursor are compared to coordinates defined for the components of the schematic diagram, in order to determine whether the user has selected a component. See, for example, Borning, "Thinglab—A Constraint-Oriented Simulation Laboratory", Chapter 2, Stanford Computer Science Department Report STAN-CS-79-746 pp. 14–37 (July 1979).

SUMMARY OF THE INVENTION

The primary object of the present invention is to provide an improved method of identifying predefined features of an image on a video display.

A specific object of the invention is to provide an improved method of operating a video display system for providing descriptive information responsive to user identification of the physical location of a perceived feature of a displayed image. A related object is to provide such a system with the capability of receiving a description from the user, and indicating the physical locations of predefined features which are responsive to the description.

Another object of the invention is to provide an efficient method of storing digitized video images along with descriptive information about various features of the images and information about the locations of the features in the images.

Briefly, the advantages of the present invention stem from encoding information about a video image as a pixel bit map and a color map in which the addresses or indices of the color map are correlated with the addresses or pointers to strings of descriptive information about predefined features of the video image. In other words, each color map address corresponds to a predefined set of features and descriptive information about those features. Since the pixel bit map defines a color map address for each physical location on the image, suitable programming of the color map can insure proper correlation of descriptive information with corresponding physical locations on the image. The correlation between color map addresses and the descriptive information about the features is represented most compactly by arranging or sorting the entries in the color map so that there is a correspondence between each predefined feature and a continuous range of color map addresses. Therefore, for a user specified physical location, the corresponding color map address can be found by addressing the pixel bit map, and the corresponding set of features and pointers can be found by comparing the corresponding color map address to the limits of the color map address ranges for the various features, for example, by a binary tree search or by a table look-up procedure.

Conversely, given a user specified description, the descriptive information for the various features can be searched to determine whether there is a correspondence with the user specified description, in order to find the features responsive to the user specified description. Next the address limits of the color map memory are found which correspond to responsive features. The physical locations of the features on the displayed image are then indicated, for example, by temporary modification of the color map entries at each address within the address limits of the responsive features.

It should be apparent that separate pixel bit maps are not required for indexing both the color map and a pointer file in order to correlate each pixel with one or more strings of descriptive information. Moreover, in most real-world applications, the features of interest are highly correlated with ranges of color in the real-world images. Therefore, it is possible to use a single pixel bit map or pixel plane of common indices for representing the video image and also for correlating descriptive information with the pixels, without requiring the memory capacity of the color map to be substantially increased, and without loss of perceived image quality. By assigning or sorting the entries of the color map so that predefined features correspond to respective continuous ranges of color map addresses, the memory space for storing the pointers is reduced. Therefore, digitized video images are capable of being efficiently

stored along with descriptive information about various features of the images and information about the locations of the features in the images.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects and advantages of the invention will become apparent upon reading the following detailed description and upon reference to the drawings in which:

FIG. 1 is a schematic diagram of a video display system incorporating the present invention;

FIG. 2 is a diagram illustrating the organization and contents of a color map, a text pointer table, and certain text strings used to represent and describe the image and various features of the image displayed as shown in FIG. 1;

FIG. 3 is a flowchart of the procedure executed by a data processing unit in the system of FIG. 1; and

FIG. 4 is flowchart of the procedure executed by the data processing unit for encoding images and textual information describing their features in the format shown in FIG. 2.

While the invention is susceptible to various modifications and alternative forms, a specific embodiment thereof has been shown by way of example in the drawings and will herein b described in detail. It should be understood, however, that it is not intended to limit the invention to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the spirit and scope of the invention as defined by the appended claims.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Turning now to FIG. 1, there is shown a schematic diagram generally designated 10 of a video display system incorporating the present invention. As is conventional, the video display system includes a color video display 11 for displaying text and images. The video display preferably includes a raster-scanned color cathode ray tube as a display device, although the invention is applicable to other video display devices such as liquid crystal, plasma, and electro-luminescent displays, and the invention is also applicable to other kinds of scanning methods. A most suitable display device is, for example, an RGB analog video monitor sold by Sony Corporation.

In addition to the cathode ray tube and its analog circuits, the video display 11 includes raster scanning circuits 12 which generate pixel addresses in synchronism with the scanning of the cathode ray tube, a pixel bit map or memory 13 which stores information about the image intensity to be generated by the cathode ray tube at each pixel, and a digital-to-analog converter 14 responsive to video data for generating analog signals which control the electron guns in the cathode ray tube. Conventionally, the raster-scan circuits 12, the pixel bit map 13, and the digital-to-analog converters 14 are included on circuits that are in close proximity to a data processing unit 15 such as a microprocessor, and a certain amount of random access memory 16 which is connected to the data processing unit via address and data buses. The pixel bit map 13 is typically included in a memory that is multiplexed or time shared between the raster-scanning circuits 12 and the data processing unit 15. Typically the data processing unit 15 changes the data in the pixel bit map 13 during the vertical retrace of the raster-scanning, so that the entire image displayed by the display 11 appears to change instantaneously.

For permanently storing large amounts of data, and also for providing input of large amounts of data, the video display system 10 includes disc memory 17 such as a floppy magnetic disc drive, a hard magnetic disc drive, or an optical disc such as a CD-ROM drive.

For receiving text input from a user, the video display system 10 has a keyboard 18 which is connected to the data processing unit 15 through an interface circuit 19.

The video display system 10 further includes a manually-operated pointing device 20 for receiving an indication of a specified location on the image 21 displayed by the video display

As shown in FIG. 1, the pointing device is a "mouse" which the user manually rolls over a flat surface 22. The data processing unit 15 is conventionally programmed to display a cursor 23 at a particular pixel on the image 21, and the data processing unit is further programmed to move the cursor up and down or left and right in response to the user's movement of the mouse 20 forward and back or left to right. The user moves the mouse 20 until the cursor 23 is aligned with a desired pixel location, and the user then specifies the desired pixel location by depressing a push button switch 24 on the mouse 20. It should be understood, however, that the kind of pointing device used by the operator is of no consequence to the practicing of the present invention, since a wide variety of other kinds of pointing devices can be used, such as a "track ball," a "joy stick," or a "light pen." All of these devices enable the user to specify a desired location on the image 21 displayed by the video display 11.

As described above, the video display system uses components which are found in a wide variety of programmable video display terminals and microcomputers. For practicing the invention, the video display system 10 further includes a color map 25 which is used in conjunction with the pixel bit map 13 to specify video data fed to the digital-to-analog converter 14 of the video display system. Color maps have provided a wide variety of advantages in video display terminals, as should be apparent from the detailed discussion above of the background of the present invention. In short, the color map 25 is used to represent the colors in the image independent of where those colors appear in the image Specifically, the color map is a kind of memory which specifies the video data for each of a plurality of indices, and the pixel bit map 13 specifies one of these indices for each of the pixels in the image 21.

Suitable raster-scan circuits 12, pixel bit map 13, color map 25, and digital-to-analog converters 14 are found in commercially available video graphic circuit boards for personal computers. For practicing the invention, the inventor has used a "190 9 Revolution Board" manufactured and sold by the #9 Computer Corp., 725 Concord Avenue, Cambridge, Mass. This video graphics circuit board was inserted in an IBM-XT personal computer. Such a video graphics circuit board has a pixel bit map with 8 bit words for storing an index ranging from 0 to 255, and sufficient memory capacity for a pixel matrix array of 512 columns by 484 rows comprising the image 21. Moreover, the color map 25 specifies 24 bits of video data for each index, including 3 bytes of 8 bits each which specify the intensity of the red, green and blue primary colors.

The video display system of the present invention allows the user to obtain descriptive textual information concerning a feature of the displayed image 21 by using the pointing device 20 to point to a location of the feature. Conversely, the user may use the keyboard 18 to enter descriptive textual information, and the locations of responsive features are indicated by the video display 11. As shown in FIG. 1, the image 21 includes four distinct features which are recognized as a chair, a lamp, a sofa, and a table. It should be readily appreciated, however, that the present invention has great practical utility in connection with the identification of features occurring in complex real-world images such as the images that are analyzed for military reconnaissance or the remote satellite sensing of agricultural crops or land areas for mineral exploration. The inventor has found the present invention especially useful for identifying predetermined features of medical images and, in particular, tissue cross-sections. A collection of tissue cross-section images in the form of a "histology atlas" would be of great benefit to doctors and medical students as an aid in teaching and diagnosis.

Turning now to FIG. 2, there is shown a schematic diagram illustrating a specific organization of the color map 25 and certain data structures in the random access memory 16 which illustrate the method of the present invention.

In accordance with an important aspect of the invention, the color map 25 includes at least one entry for each different color included in the image 21, and includes additional entries for each color which occurs both inside and outside a predefined feature of the image. Therefore, a unique entry in the color map is provided for each color included in a distinct combination of features. Therefore, each index has associated with it a predefined set of features and a certain color.

For the predefined features in the image 21 shown in FIG. 1, there is no overlap between the features. Therefore, the above condition can be satisfied when the entries of the color map are preferably segmented into a continuous numerical range of indices for each of the predefined features. In general, this can be done so long as the features are either disjoint or related as sets and subsets.

Since each index corresponds to a particular combination of features as well as a color, the particular feature associated with a specified pixel location can be found by reading the index from the pixel bit map and then finding the particular combination of features associated with the index.

In accordance with an important aspect of the present invention, the memory of the video display system 10 further stores data describing the features associated with each index. Therefore, given the index of a user specified pixel location, the combination of features at that pixel location can be determined by reading from memory the descriptive data for the features associated with the index. In particular, as shown in FIG. 2 the random access memory 16 stores text strings 26 including a particular string for each feature. Also, the random access memory 16 stores a text pointer table 27 which specifies the text strings associated with each index. As further shown in FIG. 2, the text pointer table is very compact due to the fact that the text pointer table need merely store the numerical limits of each range of indices for the segment associated with each feature. Moreover, the text pointer table is relatively compact because the indices are associated with pointers to the text strings rather than the strings themselves.

By inspection of the contents of the color map 25, the text strings 26, and the text pointer table 27, it should be apparent that the first segment of the color map is addressed by indices ranging from 0 to 31 and it stores colors for the chair. The second segment of the color map stores colors at indices 32 to 63 for the lamp. Similarly, the third segment stores colors at indices 64 to 95 for displaying the sofa.

As a specific example, the user may operate the mouse 20 to move the cursor to a pixel on the chair having a light blue color represented by the red-green-blue video data triple of zero, zero, 31. The data processing unit 15 may operate the interface circuits 19 to receive the pixel address of that pixel. By addressing the pixel bit map with that pixel address, the data processing unit will read an index value of 1. Then, by comparing that index value to the indices in the text pointer table 27, the data processing unit will decide that it falls within the range of 0 to 31 specified for the pointer "STRING1." By addressing the text strings 26 starting at the pointer STRING1 and continuing up to but not including the next string pointer STRING2, the computer may read and display the text "CHAIR" which describes the chair.

Conversely, the user may enter the word "CHAIR" via the keyboard. The data processing unit 15 may obtain the word chair from the interface circuits 19, and search the text strings 27 for a match. The computer will then determine that the text entered by the user matches the text string starting at the address of "STRING1" and continuing up to but not including the pointer "STRING2." The computer may then search the text pointer table for the occurrence of these pointers, and thereby determine that the feature associated with the first text string is specified for indices ranging from 0 to 31. Then, by temporarily changing the video data stored in the color map 25 at the indices 0 to 31, the video display 11 will temporarily change the color of the chair to indicate to the user that the chair is a feature of the image responsive to the text that the user had entered on the keyboard 18.

Turning now to FIG. 3, there is shown a flowchart generally designated 30 of a control procedure for performing the operations just described. In the first step 31, the data processing unit 15 displays to the user a message requesting the user to select an image, it being understood that the disc memory 17 includes a plurality of images for selection by the user. In step 32 the data processing unit 15 operates the interface circuits 19 to input the user's image selection which is to be entered via the keyboard 18. If the user presses the "escape" key on the keyboard 18, then in step 33 the data processing unit recognizes that the user desires to terminate or exit from the control procedure. Otherwise, the user's response should be information identifying a particular image to be viewed. Therefore, in step 34, the data processing unit transfers the record of the selected image from the disc memory 17 to the pixel bit map 13, the color map 25, and preassigned memory locations in the random access memory 16 for the text pointer table 27 and the text strings 26.

To determine the specific identification operation desired by the user, in step 35 the data processing unit polls the keyboard by operating the interface circuits 19. As tested in step 36, if the user hits the "escape" key, execution jumps to step 31 to enable the user to select a

new image. Otherwise, if text is received from the user, as tested in step 37, then in step 38 the data processing unit searches the text strings 26 for a match with the user specified text. If a match does not occur, as tested in step 39, then in step 40 a message is displayed to the user to indicate that a responsive feature cannot be found. Otherwise, the result of the match is a list of string pointers to the responsive text. In step 41, the data processing unit searches the text pointer table to determine the ranges of indices of the features corresponding to the string pointers. To display the responsive features to the user, the data processing unit temporarily changes the color in the color map for each index in the index range of the text pointer table specified for each matching string. This could be done a number of times to repetitively "flash" the responsive features.

If in step 37 it was found that the user had not entered text, then in step 42 the data processing unit operates the interface circuits 19 to poll the pointing device 20. In step 43, the data processing unit tests whether the user is pointing to an image location. If so, then in step 44 the pixel bit map 13 is read at the user specified pixel address to obtain a corresponding color index. Next, in step 45, the data processing unit searches the text pointer table for the text string pointer specified for the color index. One specific way of searching the text pointer table is to successively compare the index to each of the indices specified in the table. This method is illustrated by the computer program appended to the present specification. Once the text string pointer is obtained, then in step 46 the data processing unit reads the text string indicated by the text string pointer, and finally in step 47 the data processing unit displays the text string to the user. Execution then jumps back to step 35 to perform additional identification operations, if desired by the user.

Turning now to FIG. 4, there is shown a flowchart generally designated 50 of a control procedure for encoding an image to obtain the stored data in the format illustrated in FIG. 2. In the first step 51, the user is told to input image data. In order to terminate the program when desired by the user, in step 52 the keyboard is polled and in step 53 execution terminates if the user hits the "escape" key. Otherwise, the user hits the keyboard when the video data for an image is available. The video data, for example, is provided by a video camera (not shown), an analog-to-digital converter (not shown), and a frame buffer (not shown). In step 55 this video data is stored in the random access memory 16 in a pixel array of color values.

In order to display the raw image, in step 56 the distinct colors of the video data are transferred to a secondary color map area in random access memory 16. Also, in step 57 the secondary color map indices are loaded into a secondary pixel bit map set aside in the random access memory. In other words, the secondary pixel bit map and the secondary color map are set aside in order to construct a color mapped representation of the video image. In step 58, the color mapped video image is displayed by transferring the contents of the secondary color map and the secondary pixel bit map to the "primary" color map 25 and the "primary" pixel bit map 13.

In order to encode information about predefined features of the image, in step 59 a message is displayed telling the user to enter text describing a new feature. In step 60, the keyboard is polled to either receive the text or to receive an escape which would indicate that no more features are to be defined. This condition is tested in step 61. If text is entered, then in step 62 the text string is stored at the next available text string addresses, and the associated string pointers at the beginning of the string and just after the end of the string are stored in the text pointer table 27. Then, in step 63, the user is told to trace the outline of a new feature. To do this, the user operates the mouse 20. In step 64, the data processing unit inputs the pixel coordinates of the outline, and in step 65 computes the pixel coordinates bounded by the outline; these pixel coordinates have the same row addresses as the pixel coordinates of the outline and for each row address they include all of the column addresses within the column addresses of the pixels in the outline. Therefore, the location of the new feature has been specified by all of the pixels included in that feature. It should be noted, however, that a single feature may be comprised of entirely separate regions of pixels, in which case the user should outline each region included in the feature.

Next, in step 66, the next available segment of the secondary color map indices is allocated or set aside to receive the colors of the new feature. In step 67, this new segment of the secondary color map is filled with the colors of the feature. Similarly, in step 68, the secondary pixel bit map at the pixel addresses of the feature are filled with the indices from the new segment of the secondary color map memory. Execution then jumps to step 59 in order to define a new feature. Alternatively, the user may hit the "escape" key on the keyboard, so that in step 69 the new image being built up in the secondary bit map and the secondary color map will be swapped with the contents of the primary memories and therefore displayed to the user. This interactive displaying could be useful for permitting the user to decide how many color map memory addresses should be allocated to each feature. The user could, for example, be provided with the option of interrupting the control procedure and going back and allocating more color map memory addresses to the previously defined feature, or even for taking some color map memory addresses from some previously entered features and reallocating them to the feature that was just defined. In these cases of scarce color map memory, the user could make available additional memory addresses by eliminating certain colors from the features.

In step 70 the data processing unit polls the keyboard to determine whether the user is ready to go on and define another feature or to terminate the program. In step 71 the data processing unit first responds by swapping the contents of the secondary pixel bit map and the secondary color map with the contents of the primary pixel bit map and the primary color map. Then, in step 72, execution jumps back to step 59 to define another feature unless the user has entered an "escape". If so, then all of the features have been defined for the image being encoded. However, at this time the secondary pixel bit map and the secondary color map do not include the information for defining the background areas of the image which do not include any defined features.

In order to provide information about the background of the image, in step 73 the pixel coordinates of the background are computed or identified so that in step 74 the background colors can be loaded into the remaining entries of the secondary color map. Also, the pixel coordinates of the background are computed or identified so that in step 75 the secondary pixel bit map is filled at the background pixel addresses with indices

of the background colors in the secondary color map. After this is done, the secondary pixel bit map and the secondary color map include the entire encoded image. Therefore, in step 76, the secondary maps, text and pointers can be stored in the disc memory 17 for use by the identification procedure of FIG. 3.

One advantage of segmenting the color map memory is that the pixel bit map for a feature and the color map segment for the feature are severable from the composite encoded image. Therefore, it is possible to independently compose or remove, edit, and reinsert the video data for a feature of the composite encoded image.

In view of the above, there has been described a video display terminal that allows a user to obtain descriptive information concerning a feature of a displayed image by pointing to the location of the feature. Conversely, the user may enter descriptive textual information, and the locations of responsive features are visually indicated. The digitized video images are efficiently stored along with textual information about various features of the images and information about the locations of the features in the images. In particular, the information about the video image is encoded as a pixel bit map and a color map in which the addresses or indices of the color map are correlated with the addresses or pointers to descriptive information about predefined features of the video image. The correlation between color map addresses and the descriptive information about the features is represented most compactly by arranging or sorting the entries in the colored map so that there is a correspondence between each predefined feature and a continuous range of color map addresses. Due to the high correlation in real-world images between color and features of interest, the memory capacity of the color map need not be substantially increased to provide the correlation between the physical locations of the features and the descriptive information about the features.

Although the invention has been described in connection with a specific embodiment that correlates textual information with the features of an image, the present invention is applicable to correlating other kinds of information about the defined features such as additional pictorial information for showing the selected features in greater detail, or predefined computer programs or subroutines which relate to the defined features. It should be apparent to one of ordinary programming skill that the specific embodiment of FIGS. 2 and 3 could be readily modified to provide these capabilities by substituting for the text strings either strings of pixel information or strings of computer program steps

To provide an "histology atlas," it would be desirable to use a pointer table (27 in FIG. 2) having a column of pixel string pointers as well as a column of text string pointers. When the user selects a feature of the displayed image, a magnified view of the selected feature could be displayed along with the textual information describing the selected feature. Alternatively, a mouse (20 in FIG. 1) could be used having a pair of push-button switches, and either a magnified view of the selected feature, or the textual information about the selected feature, could be displayed depending upon whether the user activates one or the other of the push-button switches.

Strings of computer program steps could be useful for providing the user with a menu of choices about the selected feature. These menu choices could select data input and control options as well as data display options. Upon selection of a feature, the control procedure of FIG. 3 in step 46 would read the specified program string pointer from the string pointer table, transfer that pointer to the operand of an upcoming "jump" or "jump to subroutine" instruction, and then execute the "jump" or "jump to subroutine" instruction to thereby pass execution to the computer program steps specified for the selected feature.

In general, computer program steps specified for predefined features would permit the program flow to be altered in any desired fashion, including alteration of the original procedure for displaying the image, and execution of data input and control procedures related to the selected feature. This would be most useful for hierarchical operation of an interactive computer terminal in which the strings of computer program steps specified for certain high-level features would include steps to interrupt the display of an original image, recursively display a magnified image of the selected feature, and redisplay the original image upon returning from the interrupt. Therefore, a user could successively select more detailed features until reaching the highest available level of magnification representing a very specific feature or control function.

```
/*  ATLAS.1C  Copyright 1987 Mike Doyle  */

#include "stdio.h"
#define ESCAPE 27

main()

{
        static char device[] = "halonine.dev";
        static char pfile[] = "atlas.pal";
        static char *strucref[] = { "jgstruc1.txt",
                                    "jgstruc2.txt",
                                    "jgstruc3.txt",
                                    "jgstruc4.txt",
                                    "jgstruc5.txt",
                                    "jgstruc6.txt",
                                    "jgstruc7.txt",
```

```
"jgstruc8.txt",
"smstruc1.txt",
"smstruc2.txt",
"smstruc3.txt",
"smstruc4.txt",
"smstruc5.txt",
"smstruc6.txt",
"smstruc7.txt",
"smstruc8.txt",
"tmstruc1.txt",
"tmstruc2.txt",
"tmstruc3.txt",
"tmstruc4.txt",
"tmstruc5.txt",
"tmstruc6.txt",
"tmstruc7.txt",
"tmstruc8.txt",
"urstruc1.txt",
"urstruc2.txt",
"urstruc3.txt",
"urstruc4.txt",
"urstruc5.txt",
"urstruc6.txt",
"urstruc7.txt",
"urstruc8.txt") ;


char disc[21];
char *picname = "jga00000.pic";  /* picture file name                    */
int picnum = 0;
int mode = 0;
int x, y, c, index, error, cx, cy, w;
int one = 1;
int zero = 0;
int maxcolor, xmax, ymax, height, width, sw;
setdev( device );          /* set up #9 board and gread picture     */
startgraphics( &mode );
pread( pfile, disc );
gread( picname );

inqdrange( &xmax, &ymax );
inqcrange( &maxcolor );
height = 10;
width = 10;

setlocator( &one, &one );              /* set up locator            */
inithcur( &height, &width, &maxcolor );
x = xmax / 2;
y = ymax / 2;
orglocator( &x, &y );
movhcurabs ( &x, &y );
do {
        do {    readlocator( &x, &y, &sw ); /* read locator position */
                movhcurabs( &x, &y );
                 if ( kbhit() )            /* button or ESC pushed  */
                  c = getch();
           } while ((( sw & 128 ) == 0 ) && c != ESCAPE );

     movhcurabs( &x, &y);
```

```
            if (( sw & 4 ) == 4 ) {
               cx = x - 1; cy = y - 1;
               inqclr( &cx, &cy, &index );

               if ( 0 <= index && index <= 31 ) {  /* correlates color index*/
                w = ( picnum * 8 ) + 0;            /* to strucref and sends */
                type( strucref[ w ] ); }           /* text filename to type */
               else if ( 32 <= index && index <= 63 ) { /* function          */
                w = ( picnum * 8 ) + 1;
                type( strucref[ w ] ); }
               else if ( 64 <= index && index <= 95 ) {
                w = ( picnum * 8 ) + 2;
                type( strucref[ w ] ); }
               else if ( 96 <= index && index <= 127 ) {
                w = ( picnum * 8 ) + 3;
                type( strucref[ w ] ); }
               else if ( 128 <= index && index <= 159 ) {
                w = ( picnum * 8 ) + 4;
                type( strucref[ w ] ); }
               else if ( 160 <= index && index <= 191 ) {
                w = ( picnum * 8 ) + 5;
                type( strucref[ w ] ); }
               else if ( 192 <= index && index <= 223 ) {
                w = ( picnum * 8 ) + 6;
                type( strucref[ w ] ); }
               else if ( 224 <= index && index <= 255 ) {
                w = ( picnum * 8 ) + 7;
                type( strucref[ w ] ); }

            }  else continue;
         } while ( c != ESCAPE );                  /* end infinite for loop */

         setlocator( &zero, &zero );
         closegraphics();
         exit( 0 );

} /* end main */

type( fn )                                  /* function to use DOS TYPE command */
char *fn;
{
         char result[20];
         char *cmd;
         cmd = "type ";

         strcpy( result, cmd );
         strcat( result, fn );

         system( "cls" );
         system( result );

} /* end type */
```

What is claimed is:

1. A method of operating a video display system of the kind having memory for storing data, a data processor, and a video display including a video data input and means for displaying to said user an image having at certain image locations intensities responsive to video data received at said video data input, and a user operated input means for providing an indication of a location on said image, wherein said memory includes means for storing said video data in an indexed data structure having data elements associated with respective indices having different values, said method comprising the steps of:

storing in said memory a first map specifying video

data for each of a plurality of indices,

storing in said memory a second map specifying one of said indices for each of said image locations,

storing in said memory respective additional data about predefined features of said image, said features occurring at certain of said image locations, said second map specifying for said features respective ones of said indices at said certain of said image locations, said respective additional data being specified for said respective ones of said indices,

displaying said image to said user by reading said second map to obtain the indices specified for said image locations, reading said first map to obtain the video data specified for the indices read from the second map, and transmitting the specified video data to the video data input of said video display,

operating said input means to receive an indication of a location on said image at which one of said features occur,

reading said second map to obtain the index specified for the indicated location, and

reading and displaying to said user the respective additional data for said index.

2. The method as claimed in claim 1, wherein said means for displaying also displays chromaticities responsive to the video data received at said video data input.

3. The method as claimed in claim 2, wherein said video display includes a raster-scanned color cathode ray tube.

4. The method as claimed in claim 1, wherein said input means includes a light pen sensitive to the intensity displayed by the video display at a location selected by manual positioning of the light pen.

5. The method as claimed in claim 1, wherein said input means includes means for manually commanding movement of a cursor displayed at a certain image location, and means for receiving a command for accepting the cursor location as the indicated location on said image.

6. The method as claimed in claim 5, wherein said input device is a mouse.

7. The method as claimed in claim 1, wherein said certain image locations are arranged as a matrix of pixels.

8. The method as claimed in claim 1, wherein said respective additional data are stored as respective lists, and a set of pointers to said lists, said pointers being indexed by said respective ones of said indices.

9. The method as claimed in claim 8, wherein said respective ones of said indices for each feature are included in continuous numerical ranges, said pointers are stored with respective numerical range limits, and the respective additional data are read for said index by comparing said index to the respective numerical ranges to determine the respective pointer.

10. The method as claimed in claim 1, wherein said respective additional data comprise descriptive text strings for the features.

11. The method as claimed in claim 1, wherein said respective additional data comprise video data for displaying the features with increased magnification.

12. The method as claimed in claim 1, wherein said step of reading and displaying comprises the step of executing respective computer programs predefined for the predefined features of said image.

13. A method of operating a video display system having a data processor, a color video display including means responsive to video data for specifying the color which is displayed at respective pixel locations on an image, a manually operated input means for specifying a pixel location, and a memory including means for storing data in an indexed data structure having data elements associated with respective indices having different values, said memory storing data including a color map specifying video data for each of a plurality of indices, a pixel bit map specifying one of said indices for each of said pixel locations thereby defining the content of said image, said image having a plurality of predefined features, said features occurring at certain of said pixel locations, said pixel bit map specifying for each of said features respective ones of said indices at said certain of said pixel locations, said memory also storing respective strings of descriptive text for said features, said text strings being specified for said respective ones of said indices, said method comprising the steps of:

operating said input means to receive an indication of a specified pixel location,

reading said pixel bit map to obtain the index specified for the specified pixel location,

reading the text string specified for said index specified for the specified pixel location, and

displaying the text string specified for the index specified for the specified pixel location.

14. The method as claimed in claim 13 wherein said text strings are specified for said respective ones of said indices by respective text string pointers stored in said memory, and specified for said respective ones of said indices, and wherein said step of reading the text string comprises the steps of reading the pointer specified for the index specified for the specified pixel location, and read in the text string indicated by the pointer so read.

15. The method as claimed in claim 14, wherein said respective ones of said indices for each feature are included in continuous numerical ranges, said pointers are stored in said memory along with limits of the respective numerical ranges, and said step of reading the pointer includes the step of comparing the numerical range limits to the index specified for the specified pixel location.

16. A video display system for displaying an image to a user and permitting the user to point to predefined features of the image and obtain descriptive information about the specified features, said video display system comprising, in combination:

a color video display for displaying text and images, said video display including means responsive to video data for specifying the color which is displayed at respective pixel locations on said image,

manually-operated input means for enabling the user to point to a specified pixel location,

a memory including means for storing data in an indexed data structure having data elements associated with respective indices having different values, said memory storing data including a color may specifying video data for each of a plurality of indices, and a pixel bit map specifying one of said indices for each of said pixel locations thereby defining the content of said image, said image having a plurality of predefined features, said features occurring at certain of said pixel locations, said pixel bit map specifying for each of said features respective ones of said indices at said certain of said

pixel locations, said memory also storing respective strings of descriptive information for said features, said string being specified for said respective ones of said indices,

means responsive to said input means for reading said pixel bit map to obtain the index specified for the specified pixel location,

means for reading the string specified for said index specified for the specified pixel location, and

means for operating the video display in response to the string so read.

17. The video display system as claimed in claim 16, wherein said respective ones of said indices for each feature are included in continuous numerical ranges, the limits of said numerical ranges are stored in said memory, and said means for reading the string includes means for comparing the limits to said index specified for the specified pixel location to determine the continuous numerical range including said index and thereby determine the feature indicated by the user, and means for reading the string which describes the indicated feature.

18. The video display system as claimed in claim 16, further comprising means for receiving descriptive information from the user, means for comparing the information received from the user to the descriptive strings to select at least one descriptive string which is responsive to the descriptive information received from the user, means for reading the memory to determine the set of indices for which the selected string is specified, and means for changing the video data in said color map that are specified for the indices in said set of indices, to thereby provide a means for visually indicating to the user the predefined features in the image which are responsive to the descriptive information received from the user.

19. The video display system as claimed in claim 16, wherein said strings are strings of text, and said means for operating the video display includes means for displaying the string so read.

20. The video display system as claimed in claim 16, wherein said strings are strings of pixel information defining their respective features with increased magnification and resolution, and said means for operating the video display includes means for displaying the string of pixel information so read.

21. The video display system as claimed in claim 16, wherein said strings are strings of computer program steps defining respective computer programs relating to the predefined features and said means for operating the video display includes means for executing the string of computer program steps so read.

22. A method of operating a video display system having a data processor, a color video display including means responsive to video data for specifying the color displayed at respective pixel locations on an image, means for receiving descriptive text form the user, and a memory including means for storing data in an indexed data structure having data elements associated with respective indices having different values, said memory storing data including a color map specifying video data for each of a plurality of indices, a pixel bit map specifying one of said indices for each of said pixel locations thereby defining the content of said image, said image having a plurality of predefined features, said features occurring at certain of said pixel locations, said pixel bit map specifying for each of said features respective ones of said indices at said certain of said pixel locations, said memory also storing respective strings of descriptive information for said features, said strings being specified for said respective ones of said indices,

said method comprising the steps of:

operating said means for receiving to receive descriptive text from the user,

comparing the text received from the user to the descriptive strings to select at least one of the descriptive strings which is responsive to the descriptive text received from the user,

reading the memory to determine the set of indices for which the selected string is specified, and

changing the video data in said color map that are specified for the indices in said set of indices, to thereby visually indicate to the user the predefined features in the image which are responsive to the descriptive text received form the user.

23. A method of operating a data processor having a memory in order to encode and store in said memory video data representing a color image together with descriptive text about certain predefined features of said image and the locations of said features in the image, said image being subdivided into pixels at locations on said image, said features comprising certain predefined sets of said pixels, said memory including means for storing data in an indexed data structure having data elements associated with respective indices having different values, said method comprising the steps of:

storing in said memory a color map including at least one entry for each different color to be included in said image, and additional entries for each color occurring both inside and outside a predefined feature so that a unique entry is provided for each color included in a distinct combination of features, the color map entries being addressable by respective indices, each index thereby having associated with it a predefined set of features and a certain color,

storing in said memory data describing the features associated with each index, and

storing in said memory a pixel bit may specifying for each pixel the index for the combination of features including that pixel and addressing the image color at the pixel location.

24. The method as claimed in claim 23, wherein said data describing the features associated with each index comprises data indicating the set of features associated with each index, and data describing each feature.

25. The method as claimed in claim 24, wherein the data describing each feature includes a string of descriptive text.

26. The method as claimed in claim 24, wherein the indices for each feature form a consecutive numerical range of indices, and wherein the data indicating the set of features associated with each index include limits of the numerical range of the indices for each feature.

27. The method as claimed in claim 23, further comprising the steps of storing in memory the video data indicating the color of the image at each pixel location, and for each feature, storing in memory the set of pixel locations to be included in the feature and allocating a consecutive numerical range of indices to the feature.

* * * * *

Appendix 4: Eòlas Press Releases

eòlas™
TECHNOLOGIES
INCORPORATED

PRESS RELEASE

EOLAS ACQUIRES COMMERCIAL RIGHTS TO KEY WORLD WIDE WEB PATENT

8/21/95 CHICAGO: Eolas Technologies Inc. announced today that it has completed a licensing agreement with the University of California for the exclusive rights to a pending patent covering the use of embedded program objects, or "applets," within World Wide Web documents.

Also covered is the use of any algorithm which implements dynamic bi-directional communications between Web browsers and external applications.

This development will have a major impact on the ability of Internet content providers to exploit the expanding interactive capabilities of the Web to gain advantage in the highly competitive online market.

Currently, various combinations of embedded applets and software development APIs (application development interfaces) are major features of Web browsers from Netscape, Spyglass, Microsoft, AOL/Navisoft, NeXT, and Sun Microsystems (especially Sun's new Java language. A quote from the current Forbes ASAP states "Browsers and servers may come and go, but Sun's breakthrough Java language, OR SOMETHING LIKE IT, will be the key to a truly interactive Internet..."). Talks have been going on for several months between Eolas and several of these companies regarding both the licensing of the underlying technology and associated products.

The licensed technology was invented in 1993 by a team led by Eolas CEO, Dr. Michael Doyle, a UCSF faculty member and past Director of the university's academic computing center. Prior to joining UCSF, Dr. Doyle was Director of the Biomedical Visualization Lab at the University of Illinois at Chicago. He received his Ph.D. from the University of Illinois at Urbana-Champaign, where he was active in the area of scientific informatics and collaborated with several members of National Center for Supercomputing Applications, the birthplace of Mosaic.

According to Dr. Doyle,"We recognized early on that the Web could be expanded beyond the limits of plain vanilla HTML document browsing to become an all-encompassing environment for interactive applications. We then developed an enhanced version of the recently-announced NCSA Mosaic program that added technology which enabled Web documents to contain fully-interactive "inline" program objects, called Weblets (by Eolas), which one could manipulate in place using the enhanced Mosaic program."

The first Weblet created was an interactive 3D medical visualization application which employed a three-tier distributed object architecture over the Internet to allow a "farm" of powerful remote computers to generate images of internal human anatomy in response to the Mosaic user's interactive commands, all from within Mosaic. This allowed a user with nothing but a low-end networked workstation and the Eolas browser to transparently access supercomputer-level power and interactively look inside an MRI scan of the human body which was embedded within a Web page.

The Eolas technology will soon be available for licensing. Information and demonstrations are available at the Eolas World Wide Web home page (http://www.eolas.com). Further information can be obtained by sending email to info@eolas.com.

---

10 East Ontario, Suite 5106, Chicago, IL, 60611 phone 312/337-8740 fax 312/337-8743
email: info@eolas.com World Wide Web: http://www.eolas.com

eòlas™
TECHNOLOGIES
INCORPORATED

P R E S S R E L E A S E

Eòlas Releases WebRouser via the Internet

9/18/95 Chicago: Eolas Technologies announced today that it has released its WebRouser(TM) applet enabled World Wide Web browser, royalty-free for individual non-commercial use. Versions are currently available for Sun, SGI and Linux platforms, with Windows and Mac versions to follow in first quarter '96. The application, and sample Weblets can be downloaded via the Web at http://www.eolas.com/eolas/webrouse/. Based upon enhancements to NCSA's award-winning Mosaic program, WebRouser features patent-pending technology that drastically expands the functionality of Web-based applications, and provides a simple and convenient way to add new features to browser programs through the use of plug-in applications, called Weblets(TM).

The Weblet enhancement allows fully-interactive program objects to be run from within Web pages, through the use of a simple <EMBED> command within the document's text. These Weblet programs become treated by the browser as a part of the Web document, displayed "inline" and controlled by the user in place, without diverting the user's attention from the document itself.

Three demonstration Weblet programs are being distributed with the WebRouser package. These include an "inline" MPEG movie player, a 3D CAD file viewer/manipulator, and a 3D molecular modeling application.

For example, when a user visits a Web page that has a URL for a 3D CAD model placed within the Web page through the use of the <EMBED> tag, the browser fetches the CAD file over the network and then launches the CAD viewer weblet on the user's machine. The user sees a live window within the Web page, displaying the fully-rendered 3D model, and a control panel which allows the user to rotate the model and zoom in to see details. When the user then travels from that page to the next destination on the Web, the browser "caches" the Weblet together with the Web page. If the user then hits the "back" button, to return to the most recent site, the Weblet appears again, right where the user left it, having maintained its "state" (rotation position, zoom level, etc) during the time it was cached.

Many other Weblets are currently under development by Eolas and others, including Weblet-based interpreters for several popular programming languages, such as Safe-Tcl/Tk, PERL, and the GRASP animation language. Plans are in the works for the creation of both JAVA and Visual Basic Weblets as well.

Other unique features of WebRouser include client-side image map support, and the ability for the browser's button bar and menu structure to be dynamically modified by simple commands within HTML documents.

Client-side image maps, for example, allow HTML authors to create graphical interfaces to their content that can be distributed on CD ROM, using the same image-map-based front ends as the online versions. Many publishers are attempting to create hybrid CD ROMs that use Web browsers as their front-end, capitalizing on the ability to develop one body of content that can be used both for CD and online distribution. These projects are often stymied by the fact that the image maps that are currently all the rage on the Web cannot be used to front-end the CD content.

The current approach on the Web is to use ISMAP-based image maps that require that a remote server decode the hotspots on the image. Since the ISMAP-based image maps are served up by the remote machine, they can't be used to front-end CDROM-based content, where oftentimes a network connection is not available. WebRouser's client-side maps can be loaded directly from the CD ROM, with no network connection required.

Another major advantage of WebRouser is the ability of Web documents to dynamically modify the browser's button bar and menu structure. According to Eolas CEO, Mike Doyle, "Most Web designers try to build in some sort of navigation system into their documents, usually at the top of the page. The problem arises when the user scrolls down the page and suddenly the navigation GUI is no longer visible. WebRouser's <LINK> command allows the Web document to place a button bar at the top of the screen, as a part of the WebRouser GUI. When the user scrolls down the document, the navigation buttons remain in place. Since the document drives the definition of the buttons' functions, each Web site can have its own Netscape-style "What's New," "What's Cool," etc. button bar pointing to their own content, not to some hard-coded browser company location, such as in other browsers."

Similarly, a <GROUP> command allows a Web document to define a new menu option in the WebRouser menu bar, allowing the user to quickly jump to a particular Web page within a large, complex Web site hierarchy.

"This represent a new paradigm, since these technologies empower Web designers to personalize their Web sites, and to deliver new levels of interactivity via the web. Taken together, these enhancements represent a quantum leap in the ability of Web site designers to build compelling functionality into their Web pages."

Eolas also announced the launching of their commercial licensing program for both WebRouser and the development of Weblet-based commercial applications. Further information can be found at the Eolas WWW site (http://www.eolas.com), or by calling (312-337-8740), faxing (312-337-8743), or emailing (info@eolas.com) Eolas directly. Eolas(TM) and Weblets(TM) are trademarks of Eolas Technologies Incorporated. Other trademarks mentioned are property of other companies.

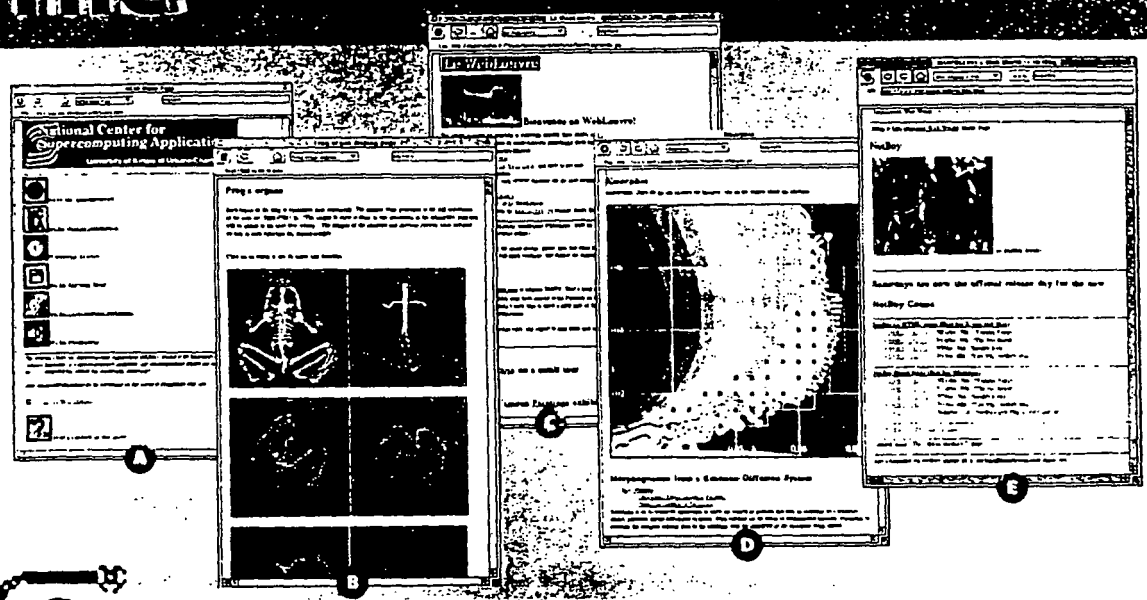
10 East Ontario, Suite 5106, Chicago, IL, 60611 phone 312/337-8740 fax: 312/337-8743
email: info@eolas.com World Wide Web: http://www.eolas.com

Appendix 5a: Article: "The (Second Phase of the) Revolution Has Begun," Wired, October, 1994

Don't look now, but Prodigy, AOL, and CompuServe are all suddenly obsolete –

and Mosaic is well on its way to becoming the world's standard interface

The (Second Phase of the)



When it comes to smashing a paradigm, pleasure is not the most important thing. It is the *only* thing.

If this sounds wrong, consider Mosaic. Mosaic is the celebrated graphical "browser" that allows users to travel through the world of electronic information using a point-and-click interface. Mosaic's charming appearance encourages users to load their own documents onto the Net, including color photos, sound bites, video clips, and hypertext "links" to other documents. By following the links – click, and the linked document appears – you can travel through the online world along paths of whim and intuition.

Mosaic is not the most direct way to find online information. Nor is it the most powerful. It is merely the most *pleasurable* way, and in the 18 months since it was released, Mosaic has incited a rush of excitement and commercial energy unprecedented in the history of the Net.

Intense efforts to enhance Mosaic and similar browsers are underway at research institutes around the world. At least six companies are gearing up to sell commercial versions of Mosaic. In April 1994, Jim Clark, founder of Silicon Graphics Inc., helped spur the frenzy, creating the Mosaic Communications Corporation and hiring a half dozen of the most experienced Mosaic developers away from the National Center for Supercomputing Applications (NCSA), where Mosaic was born (see "Why I Dig Mosaic," page 120). Two months later, Digital Equipment Corporation announced plans to ship a version of Mosaic (enhanced by Spyglass Inc.) with every machine it sells. Rumors have circulated that Microsoft was secretly licensing Mosaic to incorporate it into Windows. (Microsoft says only that it is "considering" a Mosaic license.) Jim Clark's partner at Mosaic Communications, a 23-year-old University of Illinois graduate named Marc Andreessen, will tell you with a straight face that he expects Mosaic Communications's Mosaic to become the world's standard interface to electronic information.

Long-frustrated dreams of computer liberation – of a universal library, of instantaneous self-publishing, of electronic documents smart enough to answer a reader's questions – are taking advantage of Mosaic to batter once more at the gates of popular consciousness. This time, it looks like they might break through. Mosaic is clumsy but extraordinarily fun. With Mosaic, the online world appears to be a vast, interconnected universe of information. You can enter at any point and begin to wander; no Internet addresses or keyboard commands are necessary. The complex methods of extracting information from the Net are hidden from sight. Almost every person who uses it feels the impulse to add some content of his or her own. Since Mosaic first appeared, according to the NCSA, Net traffic devoted to hypermedia browsing has increased ten-thousandfold.

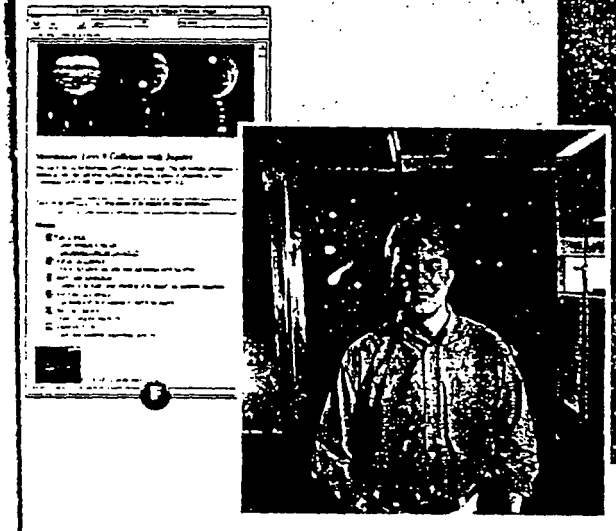
Looks Count

Ironically, the ingenious network that you see with Mosaic has been around for several

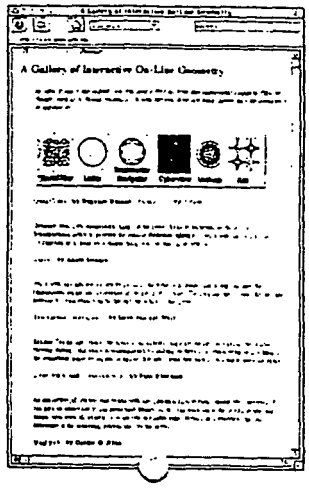
Gary Wolf reports on everything you ever wanted to know about Mosaic.

to electronic information.

Revolution Has Begun



In January 1993, Marc Andreessen (left) released the first version of the NCSA's Mosaic browser. By year's end, browsers were being downloaded at an average rate of more than a thousand per day. By mid-1994, active Web pages (like those shown here) grew from 50 to more than 1,500.



years. It is called the World Wide Web, and it was developed by a group of programmers at the European Particle Physics Laboratory (more commonly known by its old French acronym, CERN, for Conseil Européen pour la Recherche Nucléaire) led by Oxford graduate Tim Berners-Lee. Berners-Lee and his colleagues faced the problem of creating a unified hypertext network for high-energy physicists working in a diverse international environment. They came up with a stunning solution. Rather than attempt to impose standards on the hardware or software, they defined standards for the data. They also created a universal addressing system. Using a relatively simple set of commands, World Wide Web users can turn their documents into hypertext: insert the proper bit of code, and a word becomes a link; insert a different bit of code, and a sentence becomes a headline or begins a new paragraph. With the new addressing system, nearly any Net document – text, picture, sound, or video – can be retrieved and viewed on the World Wide Web.

The beauty of this approach is that it allows maximum openness and flexibility. All World Wide Web documents are similar, but every World Wide Web reader, or browser, can be different. From the smallest laptop to the most outrageous supercomputer, nearly every machine can hook into the Web. The Web, despite its sophisticated hypertext capabilities, is as catholic as the Net itself. All you need for exploring is a browser.

This, of course, is where Mosaic comes in. The first World Wide Web documents and browsers were functional but off-putting. They were not point-and-click. They did not have colors or images. But the Web was free, and as Tim Berners-Lee and other Web developers enriched the standard for structuring data, programmers around the world began to enrich the browsers. One of these programmers was Marc Andreessen, who was working for the NCSA in Urbana-Champaign, Illinois. In January 1993, Andreessen released a version of his new, handsome, point-and-click graphical browser for the Web, designed to run on Unix machines. In August, Andreessen and his co-workers at the center released free versions for Macintosh and Windows. In December, a long story about the Web and Mosaic appeared in *The New York Times*. And by the year's end, browsers were being downloaded from the NCSA at an average rate of more than a thousand per day.

Some programmers active in the World Wide Web community resent all the attention Mosaic has received. They know that the real heart of the World Wide Web is the data standard and the addressing system. They argue that any bozo – or at least any sufficiently talented bozo – can write a browser. "A guy on our project wrote a browser in a week," says one unimpressed programmer at the Massachusetts Institute of Technology, whose name I withhold out of

---

Gary Wolf (gwolf@igc.apc.org) writes regularly for Wired. He and Michael Stein are the authors of Aether Madness: An Off-Beat Guide to the Online World *(Peachpit Press '94).*

Why Jim Clark Loves Mosaic

When *Wired* last spoke with Jim Clark, in the fall of 1993 for a profile of Silicon Graphics Inc. (issue 2.01, page 116), he was chair of the red-hot company that continues to dominate 3-D interactive computer graphics. But Clark was clearly unhappy with how he'd been treated at his own company, and during the course of a lengthy interview, he let his hair down, revealing that he almost resigned when SGI executives, including company president Ed McCracken, were resistant to his ideas for bringing the company into the future.

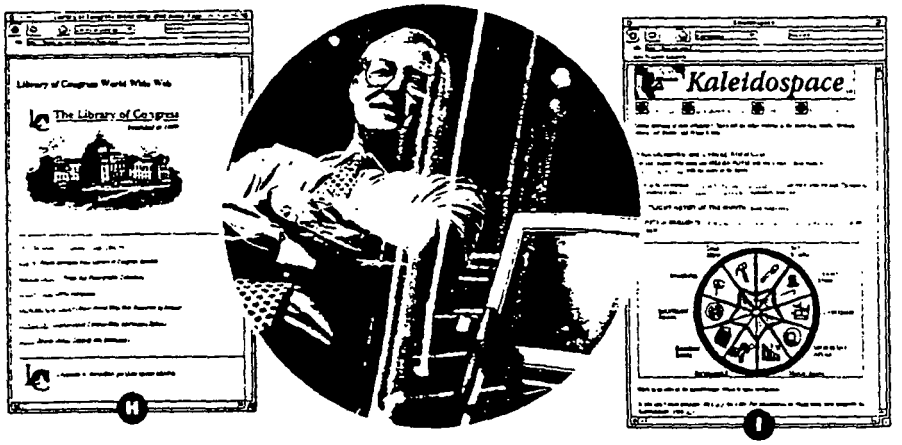
What he didn't say then was that while it appeared that SGI was adapting to Clark's view of the future, a future in which SGI technology would fuel interactive-TV networks and Nintendo videogames, he was viewed less like a visionary than as an eccentric uncle – tolerated, but not taken very seriously. "I was not happy there for about four years," Clark now says. "It was too much of a struggle to get anything done."

So in a February 1994 SGI board of directors meeting, Clark handed in his resignation. Within weeks, he was talking vaguely about a software company that would develop graphical user interfaces. During one conversation with *Wired* more than a month before he made public his new company, his response to a casual question about his thoughts on Mosaic elicited a telling reply: "Why are you asking me that?" He suddenly sounded wary, almost paranoid.

In April, Clark announced that he had formed Mosaic Communications Corporation – which would be producing a commercial version of Mosaic with new and improved features – with Mosaic author Marc Andreessen.

Michael Goldberg met with Clark at his new offices in Mountain View, California. After years of frustration at SGI, he seemed content.

Wired: Six months ago, you were chair of the board of Silicon Graphics; now you're heading a new software company, Mosaic Communications Corporation. How did that come about?



After leaving Silicon Graphics, Jim Clark wanted to get into the interactive-television business, but wasn't sure where fire would next strike. With Mosaic, Clark (left) thinks he has found the spark.

sympathy for the administrator of his e-mail account.

Other Web wizards agree. "Mosaic is about to get a boot up the backside," says an experienced developer at CERN in Geneva. "There are an awful lot of good browsers coming out. Mosaic isn't the only one."

These kibitzers are correct: Mosaic isn't the only one. And yet Mosaic is the one that did the trick. The Web statistics tell the tale plainly. The explosion of interest in the World Wide Web began as soon as Andreessen's first Mosaic browser appeared. At that time, in January 1993, there were 50 known Web servers. By October, there were more than 500. By June 1994, there were 1,500.

The secret of Mosaic's success is no mystery. When you browse with Mosaic, you see a series of well-proportioned "pages," with neat headlines and full-color images. You can fiddle with the screen to suit your own preferences. (I like grayish-purple text, with links in blue.) You can mark your progress forward and back in the Web, and make a "hotlist" of places you visit often. On the Macintosh version, which I use, you move up and down the page in the conventional fashion, using a scroll bar on your right.

Mosaic may not be a work of technical genius, but it is hard to stop using. Every day, interesting new hypermedia documents appear. Andreessen and other developers claim there are already at least a million copies of Mosaic on computers around the world.

At the same time, it's hard not to sympathize with the naysayers' irritation. Mosaic illustrates an axiom that many brain-workers find dismaying: looks count. But advocates of hypertext have been struggling to realize their dreams for years without success, and the shadow of disappointment that surrounds the names of earlier hypertext projects – such as Ted Nelson's Xanadu or Bill Atkinson's Hypercard (both of which represented a set of highly interesting ideas about interconnected information) – contrasts sharply with what Mosaic has achieved.

This aesthetically pleasing browser has begun a revolution in the way we experience knowledge. In the world of the Web, knowledge is not something you produce, but something you participate in. A document isn't a self-sufficient individual creation, but a perspective, or collection of perspectives, on the entire Web.

This may sound abstract, but with Mosaic on your screen, it is suddenly, strikingly concrete. All the documents in the Web are within reach. What path will you take to get to them? What path will you mark for others to take?

Going Commercial

Although the NCSA versions of Mosaic are still free, a number of for-profit software companies have purchased nonexclusive licenses to sell and support it. The exception is Jim Clark's Mosaic Communications, which, rather than license the source code, simply hired a half dozen programmers away from the NCSA in order to reengineer a Mosaic-like browser of its own. To license Mosaic, as of July 1994, the NCSA is charging an initial fee of US$100,000 plus $5

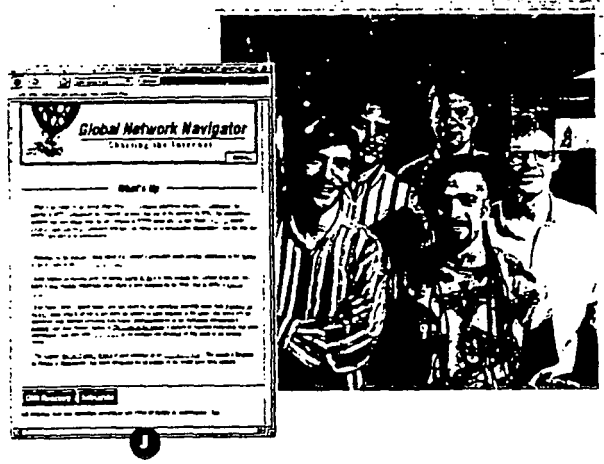
Jim Clark: A lot of the things that I was trying to get done at SGI were going as rapidly as a company that size was going to be able to push them. I wasn't happy with the pace.

What was wrong?

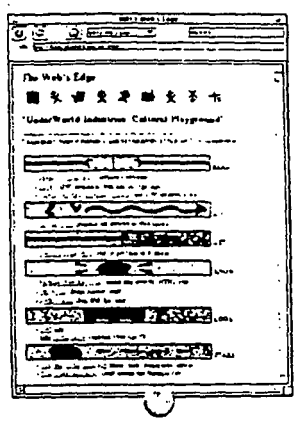
SGI isn't pushing volume into the marketplace. Volume is going to set the standard and it may be already too late. My challenge to the company is make that technology available at prices that are competitive with PCs. It's doable; just do it. That's what I used to tell them, but no one ever got it. If I wanted to right now, I could just blow SGI to smithereens. I'd never do it. But all that's got to happen is somebody takes the basic graphics library – there's no patent on it, I had that technology, parts of it, before SGI ever started – and builds an absolutely killer chip. It does take some knowledge, and – fortunately – for the company – most of that knowledge is held within SGI. But just build it, design it. Design it for ultra-high volume, strike a deal with Bill Gates. Say: "You embed this set of graphics calls into your operating system, and I will sell this chip set, just like Intel sells the X86. We have a balanced CPU and graphics system; you support it in your software. And there would be no reason to buy these high-end expensive things that SGI sells." But I'm not going to do that. And I don't think anybody else has either the knowledge or the ability to muster what would be required. But it could be done. That's their most serious vulnerability. It's an excellent group of people but they – and people at every other company – begin to define themselves by what they have been doing, not by what they can do.

That seems to be human nature. It's comfortable.

Something has gone wrong when you stop valuing the person that tries to set a little bit of farsightedness. I didn't feel valued there. I feel incredibly happy to be out of that place. It was too much of a struggle to get anything done. And that's too bad. It's a sign of bad things when a company 154 ►



Clark and Andreessen were going to try to pick off the NCSA Mosaic team (left) one at a time, but then they realized it was only a matter of time until someone else did. They grabbed the key developers in one day of hotel-room meetings.



each for any number of copies. Licensees are encouraged to enhance Mosaic and resell it to consumers. In June, Fujitsu announced a Japanese Mosaic priced at ¥5,000 (about US$50). SPRY Inc., in partnership with O'Reilly and Associates, a San Francisco Bay area publishing company, plans to have a shrink-wrapped, user-friendly product called "Internet in a Box," including Mosaic, on the shelves by fall.

Jeff Stockett has other plans. He is one of the owners of Quadralay Corporation, a Mosaic licensee in Austin, Texas, that is retooling the browser slightly and repackaging it as an online customer support and service system. Quadralay has also announced a consumer version of Mosaic for Windows, officially priced at US$249. Stockett admits that Mosaic is not the last word in browsers. "There may be something that comes tomorrow that transcends anything we have seen thus far," he says. "Sometimes I think that Mosaic may be the VisiCalc of the '90s."

president of The Internet Company, a technical services firm in Cambridge, Massachusetts. Raisch insists that the current network is simply not ready for Mosaic and estimates that because graphical Web browsers

neurs like Stockett know that VisiCalc – a simple spreadsheet program for personal computers – eventually morphed into Lotus 1-2-3 and helped launch a complete transformation in business computing. Mosaic's impact

Mosaic is a graphical browser for the Web. Say what?

The World Wide Web (aka WWW, the Web) is a unified "information space" that consists of hypertext documents and links between documents.

Hypertext is a word coined by Ted Nelson to describe a seamless world of information, in which any part of any document can be linked to any part of any other document.

A Web browser is a computer program that retrieves and interprets documents on the World Wide Web. Mosaic is a browser that offers a graphical user interface, but not all browsers do. Lynx, for instance, is a popular text-only browser.

HTML (HyperText Markup Language) is the high-level programming language in which World Wide Web documents are written.

A SLIP (Serial Line Internet Protocol) connection provides a way for hosts and networks to link into the Internet via phone lines.

The URL (Uniform Resource Locator) is the address of a document on the World Wide Web.

Maybe, but, then again, maybe not. "When VisiCalc first came out, it could run on every damned 8086 in the known universe, with nothing added, nothing extra," writes Rob Raisch,

like Mosaic require a high-speed connection to the Internet, they can run on no more than 2 percent of all currently Internet-connected machines.

On the other hand, entrepre-

could be even larger, for its potential market includes not only businesses, but every individual who wants access to electronic information. Consumers whose Web browsers choke on

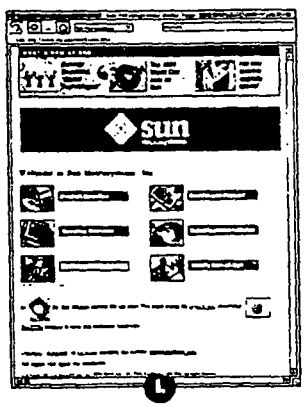
Last night Mosaic blew my mind. It was not the underlying technical elegance of the browser, for Mosaic functions lurchingly, with many gasps and wheezes. Images traveling through the Net don't appear quickly, even when they flow through a 56-KByte line. But Mosaic blew my mind nonetheless. With seamless grace, it brought me in contact with information that I didn't know I wanted to know.

I launched Mosaic for a prosaic reason: to track down some details about the World Wide Web on the pages at CERN in Geneva. But I typed the address incorrectly – or had copied it down wrong – and I soon found myself wandering aimlessly along the interwoven strands of the Web, listlessly clicking on links, circling in the near vicinity of CERN (not geographically, of course, but along vectors of association), hoping in a rather lame way to hit on the document I was looking for. Finally, I found myself standing on the NCSA demo page Ⓐ, much as tourists wandering through the complex alleys of an old city will, when their energy runs out, eventually walk along with the flow of traffic and find themselves in one of the main intersections or town squares.

Many documents are linked into the NCSA demo page, which is full of links leading out into the Web. I scanned down the lines of gray text and selected a blue link that had nothing to do with my official mission: "An experiment in hypermedia publishing: excerpts and audio from a book reading by author Paul Kafka of his novel *LOVE Enter*," it said. This, I hoped, would be a nice breather.

Upon entering the page, I was immediately distracted by another link, a quiet alcove halfway down that read poetry archive. I wanted to see the poetry archive. I clicked. "Unable to connect to remote host," Mosaic responded. I was peeved. The door was locked! I clicked a link at the bottom of the screen, where the name of the author of the page was listed: Paul Mende. After a minute of waiting (not unusual), Mende's picture

I clicked on Benjamin's Home Page, and a beaming, gap-toothed 3-year-old filled my screen. The top of the screen announced that his research interests were, "Sand. Also music, boats, playing outside."



Ⓑ

the incoming data are likely to join the clamor for a better network with higher bandwidth. Whether the Net can answer this demand – technically, commercially, and socially – remains to be seen. "If people continue to sell Mosaic as the easy way to market to more than 25 million willing Internet consumers, we are heading for a 'marketing crash' of immense proportions," Rob Raisch warns.

One thing is clear: with the commercialization of Mosaic, the global network of hypertext is no longer just a very cool idea. It is now a global competition. The second phase of the revolution is about to begin.

The Man and the Myth

I first meet Marc Andreessen, accompanied by his publicist Rosanne Siino, in the fifth-floor reception area of Mosaic Communications's Silicon Valley headquarters. As Andreessen gets a glass of water from the nearby kitchen, he takes approving notice of the stash of Oreo cookies in the cabinet. We retire to a conference room, which is bare except for a table, chairs, and a large jar of M&Ms.

Andreessen mentions that at night, when the office is full and the tables are littered with pizza boxes, Mosaic Communications doesn't seem all that different from the environment back at the NCSA. But this afternoon, the comparison seems forced. Other than Siino and a receptionist, there is nobody else in the office. The air-circulation system is humming. The setting is quiet and corporate. A little way into the interview, Andreessen removes his dress shirt and answers the rest of my questions in a white T-shirt. This gesture, combined with cautious answers to my questions, leaves the impression of a man doing battle against the businesslike backdrop – and losing.

Two years ago, Andreessen was one of a handful of programmers who were taking an interest in Tim Berners-Lee's research on the World Wide Web. To Andreessen, who says he majored in computer science because electrical engineering was too much work, the lack of an easy-to-use graphical interface for the Web was a glaring omission.

"There was this huge hole in the world," says Andreessen, "because a network existed with all these people hooked up to it, and the software was 10 years behind the hardware. This is typical of the personal computer industry today," he continues. "Perhaps because of people like me." Andreessen argues that people who write software are often people who, like him, are daunted by building hardware. Therefore the machines outstrip our capacity to use them.

When Andreessen's first Mosaic release at the beginning of 1993 seemed to strike a chord with Web users, other developers joined in the effort. Chris Wilson, 24, who now works for SPRY Inc., went to work on a Windows version. The center retained ownership of the software but made it available free for individual use. As Mosaic spread through the Internet, Wilson could see problems looming. It was tricky to load and operate, and users around the world began besieging the NCSA with demands for help. "The center was just getting swamped," says Wilson. "They were hiring new people as

appeared. He was smiling and young, with bushy brown hair and a large mustache. His page listed his research interests: "String Theory, Quantum Chromdynamics." Then came a section called Odds and Ends, under which were listed New Fiction and Readings, Benjamin's Home Page, and "local docs." What were the local docs? Who was Benjamin?

Before finding out, I glanced at the rest of the document, and it was then that I began to experience the vertigo of Net travel. On the lower parts of the page were abstracts of Paul's scientific papers, some co-authored with Benjamin Grinstein. "High energy string collisions in a compact space," was one of the titles. This meant nothing to me, of course. But, having sought a respite in poetry, it was dizzying to have wandered into the company of a physicist.

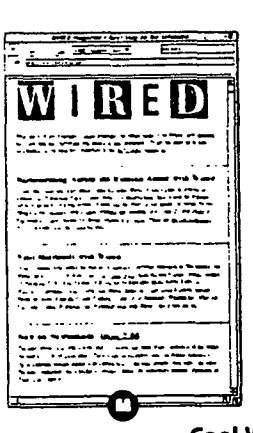
It was a type of voyeurism, yes, but it was less like peeking into a person's window and more like dropping in on a small seminar with a cloak of invisibility.

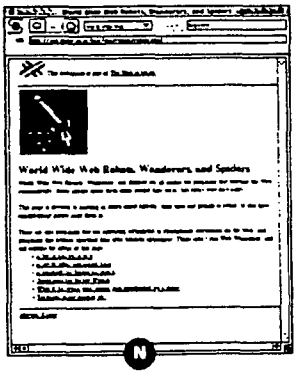
One thing it was not like: it was not like being in a library. The whole experience gave an intense illusion, not of information, but of *personality*. I had been treating the ether as a kind of data repository, and I suddenly found myself in the confines of a scientist's study, complete with family pictures.

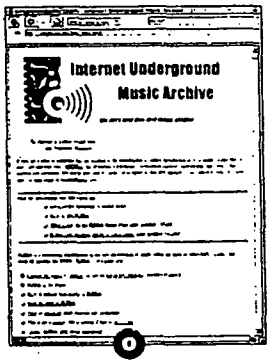
When I clicked on the link titled Benjamin's Home Page, I found that it did not belong to Benjamin Grinstein, Paul's scientific co-author, but rather to Benjamin Mende, his son, a beaming, gap-toothed 3-year-old, who announced at the top of the page that his research interests were, "Sand. Also music, boats, playing outside."

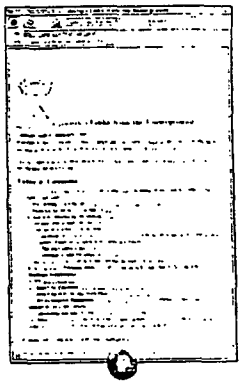
"Playing outside" was a link to a picture that filled my 21-inch screen. Benjamin was sitting on the grass in a hooded sweatshirt, wearing corduroy booties, laughing.

It was late. I'd been in Paul Mende's life for an hour. I turned the computer off. It was not until this morning that I remembered I had never made it back to CERN.
– *Gary Wolf*









Cool Web Pages: A Sampler

Interactive Frog Dissection B
http://curry.edschool.virginia.edu/~insttech/frog/
See how a frog fits together by ... Guaranteed formaldehyde-free.

Le WebLouvre C
http://mistral.enst.fr/~pioch/louvre/
Tour the world-famous Louvre ...

Xmorphia D
http://www.ccsf.caltech.edu/ismap/image.html
Supercomputer-created images and MPEG movies.

NetBoy E
http://www.interaccess.com/netboy.html
Comics for the hardcore geek.

Shoemaker-Levy F
http://seds.lpl.arizona.edu/sl9/sl9.html
Everything you wanted to know about the comet collisions last July.

Gallery of Interactive On-Line Geometry G
http://www.geom.umn.edu/apps/gallery.html
Create Penrose tiles, generate Escher-like patterns, and use a prototype 3-D viewer through this page at the University of Minnesota.

Library of Congress H
http://lcweb.loc.gov/homepage/lchp.html
Offerings from the nation's library.

Kaleidospace I
http://kspace.com
Material from writers, artists, musicians, and videographers. The site charges a nominal fee to digitize and publicize unknowns.

Global Network Navigator J
http://nearnet.gnn.com/gnn/gnn.html
O'Reilly & Associates's online magazine and resource center.

The Web's Edge K
http://kzsu.stanford.edu/uwi.html
A well-designed launchpad to offbeat strands of the Web.
http://kzsu.stanford.edu/uwi/reviews-l.html
What's "Hot and Cool."

Sun Microsystems L
http://www.sun.com/
Product info, service and support areas, new-release announcements, and links to other sites using Sun hardware.

HotWired M
http://www.hotwired.com
What can we say? It's a cross between *Pee Wee's Playhouse* and *Wired* magazine.

WWW Robots, Wanderers, and Spiders N
http://web.nexor.co.uk/mak/doc/robots/robots.html
Programs that traverse the Web automatically.

Internet Underground Music Archive O
http://sunsite.unc.edu/ianc/index.html
Indie music on the Net.

Links from the Underground P
http://raptor.swarthmore.edu/jahall/
Links and commentary providing an alternative view of the Web.

quickly as they could, and there was no way to get through the backlog."

"We got calls from people saying, 'How can we get it?'" Andreessen recalls. "Then we got calls saying, 'What do we need to run it?' We even got a couple of calls saying, 'Do you need to have a computer?'"

As the Mosaic craze grew, commercial pressure on the young developers was also mounting. The NCSA's mission includes "technology transfer" – the licensing of its inventions to commercial companies. But the developers were not likely to see much of the profit. "Companies started to come to us," reports Andreessen. "They were saying: *'Let us have it, how much do we pay? We'll give you money!'"*

Neither Andreessen nor Wilson enjoyed being in an environment with many of the pressures of a commercial software company, including user sup- 150 ▸

Mosaic

◄121 port, and none of the rewards. "It wasn't clear where we stood," Wilson says. "All of a sudden we were working for money, but it wasn't admitted we were working *for money*. There was a lot of discontent *building up*." By early 1994, Wilson had left the NCSA and *joined* SPRY.

Andreessen also left the NCSA, departing in December 1993 with the intention of abandoning Mosaic development altogether. He moved to California and took a position with a small software company. But within a few months he had quit his new job and formed a partnership with SGI founder Jim Clark.

"At the NCSA," Andreessen explains, "the deputy director suggested that we should start a company, but we didn't know how. We had no clue. How do you start something like that? How do you raise the money? Well, I came out here and met Jim, and all of a sudden the answers starting falling into place."

In March, Andreessen and Clark flew back to Illinois, rented a suite at the University Inn, and invited about half a dozen of the NCSA's main Mosaic developers over for a chat. Clark spent some time with each of them alone. By May, virtually the entire ex-NCSA development group was working for Mosaic Communications.

Andreessen answers accusations that corporate Mosaic Communications "raided" nonprofit NCSA by pointing out that with the explosion of commercial interest in Mosaic, the developers were bound to be getting other offers to jump ship. "We originally were going to fly them out to California individually over a period of several weeks," Andreessen explains, "but Jim and I said, Wait a second, it does not make much sense to leave them available to be picked up by other companies. So we flew out to Illinois at the spur of the moment."

Since Mosaic Communications now has possession of the core team of Mosaic developers from NCSA, the company sees no reason to pay any licensing fees for NCSA Mosaic. Andreessen and his team intend to rewrite the code, alter the name, and produce a browser that looks similar and works better.

The Anti-Gates

Clark and Andreessen have different goals. For Jim Clark, whose old company led the revolution in high-end digital graphics, Mosaic Communications represents an opportunity to transform a large sector of the computer industry a second time. For Andreessen, Mosaic Communications offers a chance to keep him free from the grip of a company he sees as one of the forces of darkness – Microsoft.

"If the company does well, I do pretty well," says Andreessen. "If the company doesn't do well" – his voice takes on a note of mock despair – "I work at Microsoft."

The chair of Microsoft is anathema to many young software developers, but to Andreessen he is a particularly appropriate nemesis. Andreessen believes that Mosaic could become the standard front end to the Net, a universal gateway to the entire stream of digital information. The young developer hopes that the momentum toward a global data environment will create an insatiable demand for Mosaic Communications's proprietary browser. Mosaic, in this scenario, is the DOS/Windows of cyberspace, an achievement that would make its young creators the new millennium's first computer zillionaires.

Of course, there are a few barriers standing in the way, not least of which is the real-life Bill Gates, who is hardly prepared to cede the field. Microsoft has its own ideas about the front end of the Net. Gates is working with cable mogul John Malone to design a set-top box that will control digital televisions attached to the coaxial wires owned by the cable industry. In the short term, Microsoft is casually announcing that the new version of its Windows operating system will be "Internet-ready, right out of the box." Such promises may be mere braggadocio, but the young Mosaic developers know that in leaving the world of the Internet and going after the desktop market they are poaching on the estates of powerful industry notables. When they describe the future of Mosaic, Bill Gates is never far from their minds. "Microsoft, what are they going to do?" asks Andreessen. "The moment Microsoft jumps in, the rules change."

At SPRY, Chris Wilson expresses a hope that the momentum of the Web will keep Microsoft at bay. "It could be that Microsoft is going to announce the release of something that has a completely different form of networking," he says. "It is theoretically possible that they could crush us all. But I doubt that. Right now the World Wide Web and Mosaic have so much steam built up."

What the Mosaic vendors have going for them, aside from the sheer appeal of their browser, are the established technical and philosophical tendencies of the network world. The popularity of the World Wide Web rests upon the way it satisfies the desire of individuals and groups to make their information universally available, while not imposing any single standard of hardware or software. Tim Berners-Lee, who helped create the Web, is now directing an international effort to extend the Web's capabilities while maintaining an open and platform-neutral environment.

Based on this open environment, developers around the world are working on some stunning enhancements to the Web, including better page-layout techniques; artificial-intelligence search engines; smart, distributed data-storage methods; and even interactive, Web-

More Information on Mosaic:

To get the individual version of Mosaic right now, go to the NCSA Mosaic anonymous ftp distribution site: *ftp.ncsa.uiuc.edu*. Program files are in directory /Mosaic.

Among the many companies offering SLIP or PPP connections are Performance Systems Inc., +1 (703) 709 0300; CRL, +1 (415) 837 5300; and Netcom On-Line Communications Services Inc., +1 (408) 554 8649.

For more information about getting a Net connection that supports Mosaic and other graphical WWW browsers, send e-mail to *info-rama@wired.com*, containing the line "send getting wired" in the message body. The file will be returned to you automatically via e-mail. To stay abreast of goings-on at *HotWired*, *Wired*'s new WWW-based sister cyberstation, send e-mail to *info-rama@wired.com* containing the line "subscribe hotwired" in the message body. You will be added to an electronic mailing list that provides weekly updates on *HotWired*'s online activities.

based, virtual reality environments. David Raggett, who is on the technical staff of Hewlett-Packard's research labs in Bristol, England, and who is helping to develop the specifications for the next generation of Web documents, speaks of how the Web could accommodate the millions of new users expected to arrive in the coming months. He imagines the different computers on the Web sharing data in such a way that the most popular information is replicated onto many machines, while the least popular information lives on a single machine. Addresses, in the conventional sense, would disappear. No human being would know where any specific piece of information was stored. The Web would shift its data around automatically, while users could retrieve documents simply by knowing their names. The Web, in this scheme, becomes unlocatable and omnipresent.

At MIT, a researcher named John Mallery points out how primitive the Web's links are today. They are fun, he agreed, but they are not smart. You can find information on the Web only by drifting through the links other users have created or by knowing the specific address of the document. But if documents and parts of documents were catalogued in more complicated ways, the system itself could build links. Browsing a magazine on the Web might automatically generate links to other magazines. Looking at an archive of photographs of flowers might automatically create links to a botanical database. "With these kinds of systems," says Mallery, "the goal is referential integration. You've got all these people, and people are cultural – the individual has cultural software that he is running. As that culture is expressed electronically, you can integrate it into the Web. You can build a knowledge base that can draw on the experience of not just the individual or a limited group, but a whole country or planet." In Mallery's view, the Web is destined to become not only omnipresent, but also, in a sense, omniscient.

Perhaps the most intriguing experiment in Web development is the attempt to create a standard for interactive, virtual-reality environments. According to Hewlett-Packard's Raggett, some of the elements, such as giving a 3-D view and allowing movement and interactivity (for instance, clocks that tick louder as the user approaches), require surprisingly little bandwidth. And there is an ongoing effort to develop practical, present-day virtual reality tools. (For more information, visit *http://www.wired.com/vrml/*.) "The approach we are taking now," Raggett says, "is to keep it simple. Get some simple virtual-reality browsers out there. That will motivate people and begin to create opportunities."

"The overriding danger to an open standard is Microsoft," Andreessen says. "One way or another, Mosaic is going to be on every computer in the world."

The Commercial Conundrum

Interestingly, at the practical level of commercial Mosaic development, both Wilson and Andreessen expressed doubt about whether the World Wide Web can maintain its open yet unified environment. To keep the Web from fragmenting into

Mosaic

smaller communities with more rigid technical requirements, the authors of Web tools will have to share their ideas and coordinate the development of new standards. This is fine in the nonprofit research and academic worlds. But in the private sector, coordination could mean a sacrifice of competitive advantage. Mosaic Communications could hardly become the DOS of cyberspace if it developed its product in a way that encouraged competition from scores of other more or less interchangeable Mosaic browsers. Mosaic Communications has figured this out, which may be why Andreessen no longer shares much information with his colleagues outside the company.

HOTWIRED

Gary Wolf will appear in the Wired Auditorium on America Online to discuss Mosaic and the World Wide Web on October 12 from 9 p.m. to 10 p.m., Eastern Daylight Time (6 p.m. to 7 p.m., Pacific Daylight Time). From AOL type the keyword *wired* and click on the Wired Auditorium icon.

"At this point I see a lot of fragmentation," Wilson complains. "We are forging ahead in areas that need guidance – in security for instance. That is going to take a lot of standards work. I would like to see what happens with the other companies, and with Mosaic Communications especially. I haven't heard a lot from them."

The reason Wilson and other Mosaic developers have not heard much from Mosaic Communications lately, Andreessen admits, is that a unified standard is not of first importance to the company. "Our major concern is our products," he says. "On top of that, we would like to be in an open environment, where other browsers could read our documents. It makes companies and consumers more willing to buy in. But it can't be our primary concern.

"We are not going to let it slow us down," he continues. "If we are moving faster than everybody else, then we will simply publish what we have done. We will say, 'This is how it is done, this is how you write documents to it.' We will have our implementation out there, and we will be competing on the basis of quality."

As we talk, I sense that Andreessen anticipates that other Mosaic developers will be irritated by his approach. The reason is obvious: If Mosaic Communications releases a stunning version of Mosaic and everybody begins to use it, and if the new version or a later upgrade is not compatible with competing Web browsers, then the rest of the Mosaic companies are going to have to get in step with Mosaic Communications or go out of business. Mosaic Communications is going to be in the position of setting the standards. This top-down approach to standards development is well known: It's the Microsoft model. Andreessen admits that it does not always lead to the most logical standards or the best products. He pauses to tell a well-known Microsoft joke: "How many Microsoft engineers does it take to change a light bulb? None, they just declare darkness the standard."

Of course, a top-down effort to define the standards of the Web may simply fail. Most Web developers I spoke with seemed to think that Mosaic Com- 154 ●

The Rush to Commercialize: Companies Holding Licenses to NCSA's Mosiac

Amdahl Corporation
1250 East Arques
Sunnyvale, CA 94088-3470
Contact: Steve Telleen
Phone: +1 (408) 992 2693
E-mail: *slt50@oes.amdahl.com*
Product: Not announced.

Fujitsu Limited
17-25, Shinkamata 1 - Chome
Ota-ku
Tokyo 144, Japan
Contact: Yasuyo Kikuta
Phone: +81 (3) 3730 3174
Fax: +81 (3) 3735 4240
E-mail: *kikuta@aisys.se.fujitsu.co.jp*
Product: Infomosaic, a Japanese version of Mosaic.
Price: ¥5,000 (approx US$50).

InfoSeek Corporation
2620 Augustine Drive, Suite 250
Santa Clara, CA 95054
E-mail: *info@infoseek.com*
Product: No commercial Mosaic. May use Mosaic as part of a commercial database effort.

Quadralay Corporation
8920 Business Park Drive
Austin, TX 78759
Contact: Brian Combs
Phone: +1 (512) 346 9199
E-mail: *INFO@quadralay.com*
Product: Consumer version of Mosaic. Also using Mosaic in its online help and information product, GWHIS.
Price: US$249.

Quarterdeck Office Systems Inc.
150 Pico Boulevard
Santa Monica, CA 90405
Contact: Robert Kutnick
Phone: +1 (310) 314 4263
Fax: +1 (310) 314 4218
E-mail: *bob@qdeck.com*
Product: Not announced.

The Santa Cruz Operation Inc.
400 Encinal Street
Santa Cruz, CA 95060
Phone: 1(800) 726 8649
+1 (408) 425 7222
E-mail: *info@sco.com*
Product: Incorporating Mosaic into "SCO Global Access," a communications package for Unix machines that works with SCO's Open Server. Runs a graphical e-mail service and accesses newsgroups.
No separate price.

SPRY Inc.
316 Occidental Ave. South, Suite 200
Seattle, WA 98104
Phone: 1(800) 777 9638
+1 (206) 447 0300
E-mail: *info@spry.com*
Products: A communication suite: Air Mail, Air News, Air Mosaic, etc. Also producing Internet In a Box with O'Reilly & Associates.
Price: US$149-$399 for Air Series.

Spyglass Inc.
1800 Woodfield Drive
Savoy, IL 61874
Phone: +1 (217) 355 6000
Fax: +1 (217) 355 8925
E-mail: *info@spyglass.com*
Product: Relicensing to other vendors. Recently signed deal with Digital Equipment Corp., which will ship Mosaic with all its machines.

* * * * * * * * * * * * * *

NCSA Information:
Software Development Group
National Center for Supercomputing Applications, University of Illinois, Urbana-Champaign
Contact: Jae Allen
Phone: +1 (217) 244 3364
E-mail: *jallen@ncsa.uiuc.edu*

Mosaic Communications Corp.
650 Castro Street, Suite 500
Mountain View, CA 94041
Contact: Rosanne Siino
Phone: +1 (415) 254 1900
E-mail: *info@mcom.com*
WWW: http://mosaic.mcom.com

Appendix 5b: Article: "At Demo 95 it became clear the Web is our next-generation OS," by Bob Metcalfe, InfoWorld, 2/27/95

dia access, and so on. Will we
pany controlling the timing and
pr. vement? We would be foolish
lessons of history, and we may
g intervention until it inevitably
disruptions.
ope that the Judge Sporkin's re-
lead to a quick resolution that
present consumer benefits and
f future catastrophe.

John Wills

g arm' of Microsoft

: Talk about the clueless. It nev-
maze me that when the subject
and its unfair trade practices
iorant few start firing off letters
o lerful free enterprise is and
just leave Billy Boy alone. These
iusly haven't a clue as to how
d Microsoft got to where they

r bothered to read about Mi-
ng-arm tactics with Stac Elec-
r how they shut out other OS
their Volume Price Agreements.
believe Windows owns the mar-
s a great product that everybody
-yone who owns it bought it be-
he only product worth buying.

lost count of the people I have
r' through basic Windows op-
nave no idea why Windows is on
er. They just know that it must
're supposed to use because
does and because it comes
ed on their machine.
goobers have no idea why there
aws, and they can't understand
u. are sick to death of the Mi-
ng-machine. Clueless, just clue-

Dave Marshall
Houston

or Iway access?

T READY to cancel my sub-
I may have to reconsider after
ster's excellent response to Bob
ie use of post office kiosks to
road participation in the use of
g information infrastructure.
Editor, Jan. 30, page 48 and
er, Dec. 26, 1994/Jan. 2,1995,
especially gratified that he
at once the taxpayer, through
Department's Advanced Re-
Agency, had built the Internet,
tor is willing to take it over and
rom it.
ever, suggest a better candidate
blic access to the Iway than ei-
ost office or the local school,
al public library. The public li-
in the forefront of connecting
blic to computerized systems
1970s.
the banking community pro-
th four-digit access to banking
braries were introducing the
to the much more complex
line local catalogs, indexed
nd now magazine indexes.
ms have a wide experience in
ary people productively use
latabases.

Robert Newhard
70250.522@compuserve.com

At Demo 95 it became clear the Web is our next-generation OS

Demo 95 drew 700 personal computer industry insiders to Palm Springs this month. We celebrated premieres of eight hot new PC products there and sat down to demos of 70 others. All were carefully selected by InfoWorld Publishing Co.'s David Coursey for readers of his *P.C. Letter*, now in its 10th year. To read what Coursey says happened at Demo 95, ask for a free copy of his special issue at pcletter@mcimail.com or call (800) 432-2478. Now, here's what I say happened at Demo 95.

JOHN ILLE

First, because I'm now too often accused of Microsoft-bashing, I must say that Melinda French's demonstration of Microsoft's Bob was a great relief to the Demo crowd. We were reassured that Bob is a promising experiment in user interface design and that Bill Gates has indeed married well.

Second, Windows 95 (on Coursey's Vapor-List since December 1993) showed up at Demo for an on-stage product showdown against OS/2 Warp 3.0 (available since October 1994). OS/2 was demonstrated doing many more things at once than I'll ever want to do. Windows 95 survived one of the most adventurous demos I've ever seen, with the actual plugging in of various adapter cards during the playing of angry rock videos.

Iway-wise, Windows 95 reappeared in

Demo's on-line services session along with America Online, Apple's eWorld, CompuServe, Lotus' InterNotes, Microsoft Network (when it's ready, maybe in August), Netscape, and Prodigy. There was the usual agonizing over "content" vs. "community." But now the term "context" is clearing up how all this content and community fits into some grand scheme of things. And there was, of course, the mad scramble to get on-line services onto the Internet's World Wide Web.

Which brings us to Demo 95's parade of Web browsers. On stage for 5-minute demos were Frontier Technology with Super TCP Pro, Spry with Mosaic In A Box, Netscape Communications with Netscape Navigator and Netsite, California Software with InterAp, and Quarterdeck with Windows Personal Web Server. They were all showing everything new under the sun within their Web browsers, especially multithreading, which gives you something to do while large image files hyperlink down at less than ISDN speeds.

America Online's InternetWorks comes as a DLL with its Windows client software. The DLL adds multithreading, OLE 2 support, persistent caching, progressive rendering, and APIs for custom development. Suddenly I saw Web browsers as, yes, client/server middleware for Internet screen scraping.

Multithreading and APIs are features of operating systems, and they bring us to this week's conceptual breakthrough: Web browsers are now viewed as a hot new application category. Soon, like Microsoft Windows before and Microsoft's Bob after, these former Web browsers will be euphemized as "operating environments" or some such, and every OS will have one. Right after that, as their APIs mature, they will graduate — like Windows may — to becoming full-fledged OSes all by themselves.

Consider files. The earliest OSes had only sequential access to unnamed files. Next came random access and the invention of file names and directories. Then structured data, with self-describing files, as in SQL databases.

The Web now advances files again, with hyperlinking — moving directory information *into files, with pointers not only* to other files but to pieces of other files, and not only to files on the local computer, but to files anywhere in the Web. It's suddenly clear that hyperlinking must be driven down into OS file systems. Why should hyperlinking be something you do only with files outside your computer?

Consider user interfaces. In the beginning there were batch job control languages. Then there were interactive command-line interfaces. And lately GUIs. Over the Internet, it was Telnet starting in the 1970s, then the X Window System, and now the Web browser — which is, yes, a client/server GUI hyperlinking OS.

Web browsers and servers are surely our next-generation OS. What I can't figure is whether the new Web OS standard will come from IBM, Apple, Microsoft, Oracle, Novell, or some new OS company we've heard little of.

Bob Metcalfe invented Ethernet in 1973 and founded 3Com Corp. in 1979. He receives E-mail at bob_metcalfe@infoworld.com via the Internet.

Appendix 5c: "Patent War Pending Over `Applets'", Interactive Week, 8/28/95

VOLUME 2 AUGUST 28, 1995 NUMBER 16

INTER@CTIVE WEEK

THE SINGLE SOURCE

####################5-DIGIT 60611
0000156281ZI E1R124 0206 002
RACHELLE TUNIK UP
EOLAS TECHNOLOGIES INC 0008
10 E ONTARIO #5106
CHICAGO IL 60611-2763

Webbed Fee

By Paul Noglows

The next big thing on the Internet may be "applets" — little programs that get delivered when needed to make browsers of the World Wide Web...

Netscape Unsafe In U.S.

By Brock N. Meeks

Just when you thought it was safe to go surfing again with your Netscape Navigator ... it isn't so.

After French student Damien Doligez hacked through the Netscape software early this month in Europe, Navigator users in the United States were told they were secure.

The U.S. version of the Navigator uses an encryption scheme that involves a 128-bit "key" for locking up data. Doligez cracked a much weaker 40-bit "export" version, Netscape said.

But the "U.S. version" in question is the *paid* version of the Navigator.

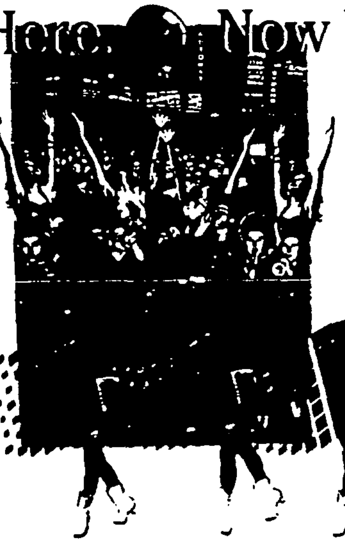
So what about the length of the key in the "evaluation" copies of the Navigator that millions of computer users have downloaded off the Net for free? 40 bits — the same key Doligez broke.

Translation: Users of "evaluation" copies are vulnerable to hacking.

The breaking of Netscape's secure transaction software also highlights a long-running battle between industry and the U.S. government over the export of encryption programs. Industry has lobbied the government for years to relax encryption export rules, and the White House may be starting to listen.

Browse At Your Own Risk 9
Crack In Crypto Stonewall 36

Win95: It's Here... Now What?



Well, it's finally over.

No longer does anyone have to worry about whether Microsoft Corp.'s Windows 95 operating system will arrive. Now, the worry will be how it plays out in the marketplace.

Bill Gates did his best last week to reassure customers, rivals, government investigators and just about anyone else who would listen that the new control program will only benefit personal computer users.

But ... Senior Writer ... describes ... Microsoft Network ... whether ... more or less ... about ... America Online Inc. and other competitors would have the Justice Department believe.

The answer may be more. Washington Bureau Chief Brock N. Meeks shows how the installation of Windows 95 has the effect of disabling computer users' existing Internet software, giving Microsoft's Internet Explorer an instant advantage over Netscape's Navigator and other browsers.

Meanwhile, Senior Writer Joe McGarvey shows how the whole Windows 95 transition may be a sideshow to a move by corporate America and individuals to the more powerful Windows NT operating system. And Silicon Valley Bureau Chief Tim Clark muses at all the shenanigans, watching the goings-on from his perch in the World Wide Web.

Launch 95 12
NT: The Real Thing 30
Socket To Them 31

...ce In Space

Brock N. Meeks

Everyone ... World Wide Web. But ... who seems to understand ... The Net ... 38

Hollywood's Hangup

It was strange casting: Three Bell companies trying to make it in Hollywood with the help of Creative Artists Agency power broker Michael Ovitz.

The twist came when Ovitz left his Bell buddies hanging and announced he was going over to The Walt Disney Co.

But the plot thickens. Tele-TV — the Bell Atlantic/Pacific Telesis/Nynex consortium formed amid much hoopla 10 months ago — has dumped CAA as chief content consultant. The casting call for new partners has attracted News Corp., NBC and Turner Broadcasting.

What happens next could depend on how swiftly the Bells move on the interactive TV networks they keep promising. And Tinseltown may lose interest completely before the final reel.

Heartbreak & Vine 8
Babes In Tinseltown 22

E 006305

UP FRONT

News, Noise and Analysis

AT&T *began assembling a financial war chest last week, selling $1 billion in securities and using its AT&T Universal Card loan assets for backing. Industry analysts believe that the securities sale is the first in a series — and they expect that AT&T will use the money to battle competition for local as well as long-distance telephone service.*

The Federal Communications Commission *cleared the way for CAI Wireless Systems Inc. and ACS Enterprises to merge by approving the transfer of wireless cable licenses between the two. This further advances the plans of Bell Atlantic and Nynex to use CAI's wireless cable network to distribute the content developed by Tele-TV, the unit they formed with Pacific Telesis.*

Creative Technologies *is heading into the game market. The sound card company is introducing 3D Blaster, a graphics board designed to enhance arcade-like games on personal computers. The three-dimensional accelerator is expected to be available in November for a street price of about $350.*

Banks are getting serious about going online, according to a recent report by ***Forrester Research.*** *The study indicates that 34 percent of banks now have a personal computer banking product available to customers, and 45 percent are developing one. Forrester says the specter of a merger between Microsoft Corp. and Intuit Inc. spurred the interest.*

Patent War Pending Over 'Applets'

By Paul Noglows

The price of winding through the Internet may be going up, if a small Chicago company succeeds in its attempt to extract licensing fees for inserting small computer programs into the software used to browse the World Wide Web.

Eolas Technologies Inc. announced last week that it has completed a licensing agreement with the University of California for the exclusive rights to a pending patent covering the use of embedded program objects, or "applets," within Web documents. Applets are poised to be the next big thing in Web browsers by making them truly interactive. Applets are tiny programs that will be downloaded automatically to a computer when a user wants to do something interactively with a browser, such as update a portfolio of stocks or hear a sound clip.

There's a perception that Java was there first, but that's simply not the case

If the patent is granted — an application from the University of California is under review by the U.S. Patent Office — Eolas stands to become a big company quickly by deriving a licensing fee from any outfit that supplies or uses applets.

Most affected will be browser companies, such as Netscape Communications Corp., Spyglass Inc. and Sun Microsystems Inc. Sun engineers, for instance, have been among the pioneers of the incorporation of applets into browsers, through Sun's Java programming language. Java is the basis for Hot Java, Sun's interactive browser, introduced earlier this year. Netscape also has said it will incorporate Java and its applet technology into the popular Navigator browser, which accounts for about three-fourths of all requests from Web servers.

Michael Doyle, chairman, chief executive and co-founder of Eolas and the former director of the UCSF Academic Computing Center in San Francisco, contends that his team of researchers invented the applets technology in 1993.

"Individuals involved at Netscape, Spyglass and Sun Microsystems saw our demonstrations in 1993," Doyle says. "Our technology has been widely discussed over the last year and we are not new players in this arena. There's a perception that Java was there first, but that's simply not the case."

Doyle says Eolas has been in discussions for months with user companies regarding both the licensing of the underlying technology (which his company has trademarked as Weblets) and associated products. While Eolas plans to provide royalty-free licenses to individual and academic users of applets, commercial users would be charged for each piece of software that uses the embedded applications. That charge could range from 50 cents per piece of software for heavy users (on the order of 1 million units) all the way up to $5 per unit for more limited usage.

Users of applets were reticent to discuss the University of California's patent application or Eolas' licensing plans. Spyglass spokesman Randy Pitzer says his company will wait to see if the patent is granted before commenting.

A Sun representative said the company is reviewing the patent application, and any comment now would be premature.

Netscape spokeswoman Kristina Lessing says her company would like to review the patent but has not been in negotiations with Eolas.

Despite these companies' current public caution, some experts expect them to vehemently oppose any development that takes money out of their pockets.

Whether a patent will be granted is anyone's guess. While Doyle says the University of California spent months researching the issue of whether the technology could be patented, the U.S. Patent Office has had a particularly difficult time in administering software patents. For instance, the Patent Office at one point issued Compton's New Media a patent for the concept of combining digital graphics, video, sound and text into "multimedia" presentations, only to rescind it later; Compton's is appealing that decision.

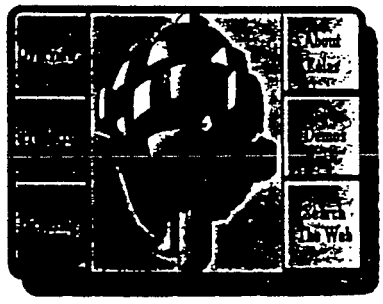
Eolas has not yet determined whether it will make its patent application public.

Doyle says applets could transform the Web into the preferred means for achieving interactive computing. That's because applets can run either on individuals' desktops or portable computers, or on more powerful computers in networks, known as servers. The user never knows whether the applet runs locally or remotely.

For that reason, Doyle says, the concept of an operating system can now be expanded beyond a program that runs on an individual machine to encompass large numbers of cooperative programs running on a web of computers all over the world.

"The World Wide Web becomes the operating system and the Internet becomes the computer," Doyle says.

Accordingly he adds, this Web operating system will eventually make irrelevant the issue of whether users are running Windows, Macintosh or Unix operating systems for their workstations.



Eolas on the Web at http://www.eolas.com

David Bennahum, author of the upcoming book *Coming Of Age In Cyberspace*, says existing operating system vendors could be hurt. "Who's left out in the cold in this new era? Folks who invested heavily in the personal computer paradigm. No one invested more than Microsoft," says Bennahum.

The acronym Eolas stands for Embedded Objects Linked Across Systems and is also the Gaelic word for Knowledge. ▲

Appendix 5d: "Eòlas Technologies Granted Exclusive Licensing Rights to Pending Web Applet Use", Communications Week, 9/28/95

firm the existen e of the new line, but sources briefed by NAC provided extensive details.

The sources said the vendor is set to introduce on Sept. 5 a line of

characterizes as a more typical configuration for $69,140, according to the briefed sources. That configuration features 128-megabyte main memory, 4-

looking to attract more departmental groups as NAC eyes Auspex's enterprise sites.

In October, Auspex, Santa Clara, Calif., will introduce low-

Auspex and NAC.

"Auspex has a great deal of flexibility, but it takes so much time to sit and manage the thing," he said. ■

Eolas Technologies Granted Exclusive Licensing Rights to Pending Web 'Applet' Use

By MARTIN MARSHALL

A tiny company in Chicago last week served notice to the World Wide Web community that it may someday be collecting royalties on applications launched from within a Web document.

Eolas Technologies Inc. announced it has been granted exclusive licensing rights to a pending patent covering the use of embedded program objects, or "applets," that are embedded within Web documents. If granted, the patent could affect commercial Web browsers such as those from Netscape Communications Corp. and SunSoft Inc., as well as in-house corporate applications involving in-line applications activated through the World Wide Web.

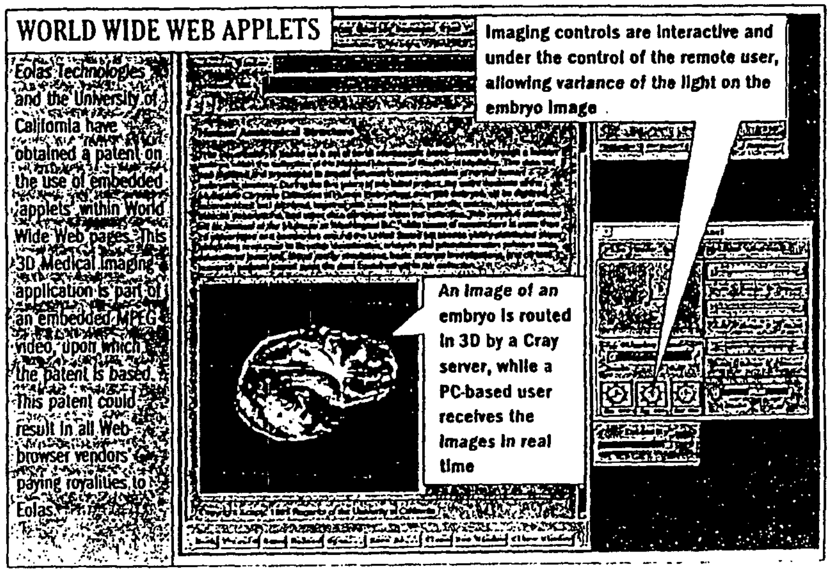
Targeting In-Line Apps

"We're not going after everything under the sun here. The licensing specifically applies to applications that are run in-line and interactively through Web documents," said Michael Doyle, CEO and founder of Eolas.

As an example, he said a Microsoft Corp. Excel spreadsheet transferred from a Web page and viewed from a local Excel application would not be covered by the patent. "But if it's embedded and interactive, that is, if you can manipulate it without leaving the Web browser, then it is covered," Doyle said.

The patent application was filed by the University of California, based upon work Doyle had done as director of UC San Francisco's Academic Computing Center. The University of California's Office of Technology Transfer granted Doyle's company the exclusive rights to licensing because, as Licensing Manager Candace Voelker explained, "We're not in a position to commercialize this technology like he can."

An official at one World Wide Web-browser company, who asked not to be named, said, "If the patent is granted, it will have very broad implications, including implications for the use of [Microsoft's] OLE [Object Linking and Embedding]."



Source Eolas Chart by Cheryl Gomandy

Netscape Communications officials were circumspect, noting that because patent-pending documents were secret, they could not examine the contents of the University of California's claim. "We were contacted by Dr. Doyle by fax in March, but no negotiations have taken place," said Roseanne Siino, director of Corporate Communications at Netscape Communications, based in Mountain View, Calif. ■

and Sun Microsystems Inc. s], but unlike other vendors that regard network management as a money-pit technology, Cabletron sees it as one of the company's crown jewels," said Greg Cline, director of network integration and management research at Business Research Group, Newton, Mass.

Network managers said the partnerships will give them added functionality. "[Spectrum] is more technically advanced than the others in that it is the only truly distributed platform and it will scale," said John Bullivant, a vice president in corporate technology at New York's J.P. Morgan & Co. Inc.

Deep Integration

Spectrum's object-oriented framework lets the new systems management applications be integrated much tighter than at the graphical user interface and several offer event and alarm integration. Some of the new applications go beyond that, integrating with Spectrum's Knowledge Base. That means the third-party product can populate Spectrum's database as a model or object.

Although several of the partner vendors are integrating their products with other management platforms, none of these other platforms have Knowledge Base integration. "Spectrum has seven different tool kits that enable us to integrate icons, screens, menus, alarms and events," said Bryon Drigian, president of Calypso Software Systems Inc., Manchester, N.H.

Calypso announced its Atrium Extendible Management System, a policy-based systems management application for managing distributed applications, workstations and servers.

Cabletron can be reached at 603-337-9400. ■

Appendix 5e: "Has Microsoft Reached its Peak?", Associated Press, 9/27/95

AP 27 Aug 95 4:06 EDT V0514

By EVAN RAMSTAD

Has Microsoft Reached Peak?

AP Business Writer

NEW YORK (AP) – It was a heady week for Microsoft Corp., lighting the Empire State Building, using a Rolling Stones song in a TV commercial and, oh yeah, putting Windows 95 on sale.

At rival Sun Microsystems Inc., chief technology officer Eric Schmidt was amused by the spectacle.

"When I saw Bill Gates on Larry King, that's when I thought, 'That's it. That's the high point,'" Schmidt said of the Microsoft chairman's appearance on the CNN celebrity interview show.

While Microsoft was showered with attention this year leading up to the Windows 95 launch, strong forces developed that could erode the dominant rol it has had in technology since the late 1980s.

Always watchful for the Next Big Thing, high-tech executives and analysts have been paying close attention to a change in the way personal computers are being used. Instead of simply running programs that reside on the machine itself, the PC has become the gateway to data anywhere.

More importantly, programs are no longer bound to the desktop or laptop device. Instead, they can reside in other computers and be downloaded when a person needs them.

The change threatens to diminish the existing structure of the PC software industry, in which application programs that perform specific functions are built around an operating system, like Microsoft's Windows, that runs the basics.

The rise of the Internet's World Wide Web and programs like the Netscape browser and Sun's Java programming language have shown people a simpler way to find and link data.

The danger for the existing PC software industry, and its leader Microsoft, comes from the corporate systems designers who have realized that such tools can be applied to an internal network and not just a public data network like the Web.

Technology magazines Upside and Forbes' ASAP this month carry articles suggesting Microsoft has fallen behind others in responding to the change.

And rivals are quick to tout their lead.

"The era of huge monolithic applications is coming to a close," said Michael Doyle, chief executive of Eolas Technologies Inc., a Chicago firm with patents for distributing small programs on-line.

While no one is suggesting that Microsoft will be insignificant in the future, the chance exists that it will lose the mantle of leadership just as IBM, Digital Equipment, and Wang did when technology shifted under them.

"The new paradigm is coming and the question is 'Will they be like Hewlett-Packard, which managed to make a technology change successfully, or like Digital, which failed to do so?'" Schmidt said.

Asked if people years from now would look back on the release of Windows 95 as Microsoft's shiniest moment, Nathan Myhrvold, the company's senior vice president for advanced technology, said, "I sure hope not! My God, no."

He continued, "This is clearly a great moment for the company and our marketers have really

outdone themselves. But the whole lesson for any technology company is we have to make this thing obsolete. If we don't, somebody else will."

And then, he identified the very condition that is both an opportunity and problem for Microsoft.

"The power of personal computers is not just the power of something being on a desk," Myhrvold said. "It's the power of connecting to others."

Microsoft's top products for a world in which the network is the computer are its Windows NT operating system, which is growing in sales, and The Microsoft Network on-line service, which became available with Windows 95 last week.

But the company was caught flat-footed by the burst of interest in the Web and the concept of jumping from one computer, represented by a Web site, to another.

After initially putting off development of a Web browser to go with Windows 95, Microsoft finally licensed one from Spyglass Inc. and modified it to go on sale last week. Enhancements that will bring it to the level of competitors like Netscape and Spry are months away, though.

Nonetheless, analysts are slow to count Microsoft out of anything.

"I don't believe this is the high point," said Aaron Goldberg, analyst at CI-Infocorp, a PC market research firm. "Microsoft still has little to no business outside the personal computer space and at some point that entire triple-digit billion-dollar opportunity has to heat up for them."

Analyst Robert Rosenberg said Microsoft has won some Windows NT customers among the the telecommunications firms he follows, which traditionally use Unix software for their sophisticated workstations.

"Microsoft is continuously surprising people because their products, while adequate, are certainly never innovative," said Rosenberg, president of Insight Research Corp. in Livingston, N.J. "But their marketing and ability to price and compete is unparalleled."

Sun's Schmidt is sure Microsoft will try to catch up.

"Microsoft is not stupid," he said. "They understand what I'm saying. Microsoft could do something like Netscape and Java too."

Unlike previous technology leaders, Microsoft's executives are quite open about the need to mow down their successful products. But actually shedding their allegiance to Windows, found on more than 100 million PCs despite its technical inelegance, is another matter.

"This whole phenomenon of computers being connected, whether by Internet or internally, is an incredibly powerful thing," Myhrvold said. "We hope to participate in this. For every way we participate, there's going to be 100 other ways that other entrepreneurial companies take the lead in."

Appendix 6: Web-Related Research Papers/Proposals by Eòlas Founders

Integrated Control of Distributed Volume Visualization Through the World-Wide-Web

Cheong S. Ang, M.S.
Programmer/Analyst
Center for Knowledge Management
University of California, San Francisco
email: cheong@ckm.ucsf.edu

David C. Martin, M.S.
Assistant Director
Center for Knowledge Management
University of California, San Francisco
email: dcmartin@almaden.ibm.com

Michael D. Doyle, Ph.D.
Chairman and CEO
Eolas Technologies Incorporated
7677 Oakport St., Ste. 646
Oakland CA, 94621
email: mddoyle@netcom.com

Abstract

The World-Wide-Web (WWW) has created a new paradigm for on-line information retrieval by providing immediate and ubiquitous access to digital information of any type from data respositories throughout the world. The web's development enables not only effective access for the generic users, but also more efficient and timely information exchanges among scientists and researchers. We have extended the capabilities of the web to include access to three-dimensional volume data sets with integrated control of a distributed client-server volume visualization system. This paper provides a brief background on the World-Wide-Web, an overview of the extensions necessary to support these new data types and a description of an implementation of this approach in a WWW-compliant distributed visualization system.

Available as both a PostScript document (6.33 M-bytes) and as a series of page images.

Polymap: A Versatile Client-Side Image Map for the Web

Cheong S. Ang, M.S.
*Michael D. Doyle, PhD.

University of California, San Francisco
Library and Center for Knowledge Management
San Francisco, California 94143-0840
and
**Eòlas Technologies Incorporated*
10 E. Ontario, Ste. 5106
Chicago, IL 60611

Abstract

Image mapping techniques have been used to improve the interactivity of graphics-based content on the World-Wide-Web (WWW). However, the current WWW image mapping technique requires a very cumbersome setup: a client-server architecture with the server acting as a point-in-polygon decoding engine to the inert client. While the advantages of having the server participate in the client/WWW browser users' activities are controversial, the need to move the decoding task to the browser is apparent, especially when better interactivity is desired. We propose a mechanism (Polymapping) that stores the hot-spot information in an otherwise unessential part of the image file -- the comment field of existing common image formats. Since adding data in the comment field does not violate most image format standards, this mechanism provides Polymap-enabled WWW browsers the information needed to achieve client-side hot-spot decoding while allowing non-enabled clients to remain dependent on the servers.

1. Introduction

Image maps facilitate the interactivity of the WWW by simulating the hot-spot mechanism in traditional multimedia applications. This feature allows WWW browser users to access URLs associated with objects within an inline image merely by clicking directly on the image. Much of the current popularity of the Web is related to the ease of use imparted by this functionality.

The currently popular image mapping implementation (ISMAP) passes the work of decoding the image map to the HTTP server, which is usually already overloaded with requests for documents, and for gateway access to non-HTTP (HyperText Transfer Protocol) [1] servers. This situation interferes with the efficiency of hot-spot decoding and doesn't allow realtime decoding of URLs, such as what users encounter when passing the mouse over text-based anchors. Seidman [8] and Doyle [4] both point out that the ISMAP standard also prevents the employment of image mapping on local file systems, which is necessary for one to distribute HTML-based applications on CDROM, where a WWW server is absent. The ISMAP implementation also imposes unnecessary network overhead upon image map decoding performance [8]. This degraded performance is the major reason for the lack of realtime interactive feedback for users, mentioned above.

Client-side image mapping is not a new concept. Ragget [7] introduced the FIG element in his HTML+

proposal. The FIG and SHAPE elements allow definition of shaped hypertext anchors in the associated image. Hot-spot decoding may be performed locally because the polygon information is downloaded as part of the hypertext document. However, complete support for the complicated FIG and SHAPE elements is claimed to need significant additional processing [8]. Furthermore, both elements may not degrade gracefully in the browsers which don't support them.

Although Seidman addressed the issues of processing complexity and graceful degrading in his HTML [3] extension for client-side image mapping, his solution requires an entity definition which may become complex quickly as the complexity of the polygons grows. The polygon map and the associated image in this case are kept as two separate files. Embedding a polygon map of many complex objects in an ASCII HTML document results in a very large file. Large document means increase in both transfer and parsing time.

Eòlas Technologies Inc. [4] developed and implemented a client-side image map protocol based upon the MetaMAP® technology, which encodes object identifiers into image pixel data using a patented (U.S. patent 4,847,604) approach. This resulted in an extremely efficient method for encapsulating URL information with image data, and for enabling realtime hotspot decoding. A disadvantage of this implementation, however, was that use of the resulting file format is constrained by the MetaMAP® patent. The Polymap work described in this paper was an attempt to allow advantages similar to those provided by the MetaMAP®-based file format, while allowing free use of the Polymap file format without concerns about patent issues relating to object encoding. Although the demonstrations of this technology we describe are based upon the GIF file format, and therefore commercial implementations may be subject to the Unisys LZW patent, the reader will note below that the Polymap approach is equally suitable for any of the several public domain file formats that allow comment fields, such as PNG or JPEG.

2. The Solution

Our solution to the above problems is to include the polygon map data in the image. Many existing standard image data formats (e.g GIF) have fields for application-specific data and/or comments, and adding data to those fields is legitimate. We utilize the comment field of the GIF 89 format to carry a compressed polygon map. With Polymap implementation, the HTML document does not differ in comparison to a simple inline image definition. Thus in addtion to client-side hot-spot decoding and better visual feedbacks, the polymap mechanism provides WWW browsers with the versatility to either perform polygon decoding themselves, or use the conventional image mapping, the ISMAP [6] technique.

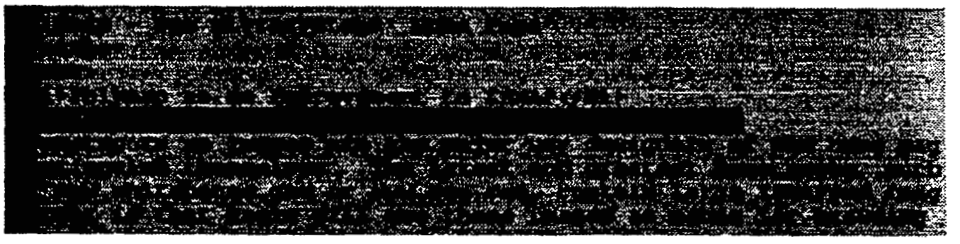


Fig 1: HTML source of the document in Fig 3.

Creating a Polymap image map is actually far simpler than setting up an image for ISMAP. We have implemented a rudimentary version of the Polymapping toolkit on the MS Windows platform (http://www.eolas.com/polymap/toolkit.html) which allows polygon vertex specifications by

point-and-click, or by free-hand contour drawing . The polygon data is compressed using the same compression engine that encodes the image data when the mapped image is saved. Since the Polymap technique does not require a separate map file, this is all a WWW server administrator has to do.

In addition, Polymapping does not interfere with the setup of an ISMAP, the server administrator is free to prepare a mapping file on the server side for Polymap ignorant browsers.

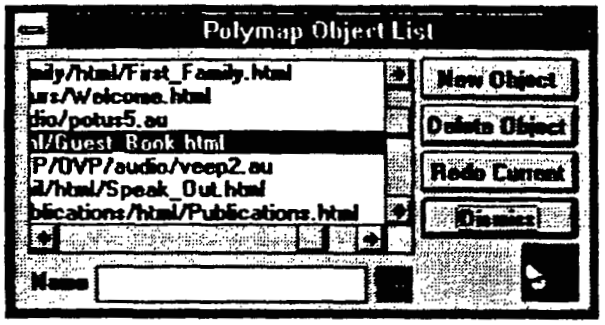


Fig 2a: Microsoft Windows version of the Polymapping toolkit (Object List).

Fig 2b: Microsoft Windows version of the Polymapping toolkit (Image Window), showing outlines around hot-spots, as each would be displayed to the user during imagemap decoding .

3. Implementation

The data in the GIF comment field is saved following a simple format:

```
GIF comment block indicator
PMAP1.0                 /* Polymap flag.  7 bytes of char */
N                       /* N bytes of data to follow.  N is a 4-byte unsigned intege
B                       /* The compression bit size. B is a 1-byte unsigned char */
N bytes of encoded data /* In GIF case, LZW encoded data */
GIF end of block indicator
```

The decoded data also has a straightforward format. The first unsigned short (2 bytes) designates the number of objects in the Polymap. Immediately following are null-terminated strings of the object names. In WWW Polymap applications, the object names are Universal Resource Locators (URLs) [3], but they may be any set of strings that is meaningful to a particular application. The name strings are followed by one or more blocks of the the following sequence:

0xFFFF
Object number
Polygon number
(x_1,y_1), (x_2,y_2),..., (x_N,y_N)

Enabling Polymap handling in WWW browsers merely requires extending the browsers' GIF reader comment handling routine to appropriately process the above format. There is no restriction on the implementation of the client-side decoding methods. Our browser, an enhanced version of NCSA Mosaic (downloadable via http://www.eolas.com/eMosaic/browser.html), deploys the windowing system's polygon and region functions to interactively highlight hot-spots as the cursor passes over them. The browser responds to mouse clicks with the same routine the text-based hot-spot respond function is using. Hence our browser reuses the hot-spot handling codes efficiently, and gives its users a consistent look-and-feel.

In the case of JPEG, another popular Internet image format, although the comment block is written as part of the file header, there are no restrictions on the format and processing of the comment block. However, the standard JPEG DCT (Discrete Cosine Transform) plus arithmetic-coded or Huffman encoding/decoding routines may not be reused because there will not be much sharable code between the lossless and lossy mode [5]. We may resort to several different solutions: (1) include the lossless spatial DPCM coding method, which will increase the size the Polymapping program and the size of the Polymap-enabled WWW browsers; (2) leave the Polygon data uncompressed as if it is a regular string of comment(s); or (3) a compromise between (1) and (2), use a smaller and simpler compressor.

A good way to realize (3) would be to use whichever image compression engine accessible internally by the WWW browser(s). Until another format becomes more prevalent, GIF's LZW encoding is a reasonable choice. The Polymap toolkit and the most common WWW browsers have GIF compression/decompression engines built-in, and the JPEG comment handling routine may invoke the engine when necessary. No extra code is needed.

Fig 2a and Fig 2b show the Polymapping toolkit. New objects are created simply by adding their names through the Polymap Object List dialog box. The user may redo or delete the selected object by a button press. Double clicking on an object in the object list will pop up a Name Change dialog, which allows the user to change the name of the object. The outline displayed for each object is also color-coded. The color icon beside the Name prompt reveals the color associated with the selected object. Clicking on the color icon will pop up a standard Microsoft Windows style Color Choice dialog that enables changing the color for the selected object.

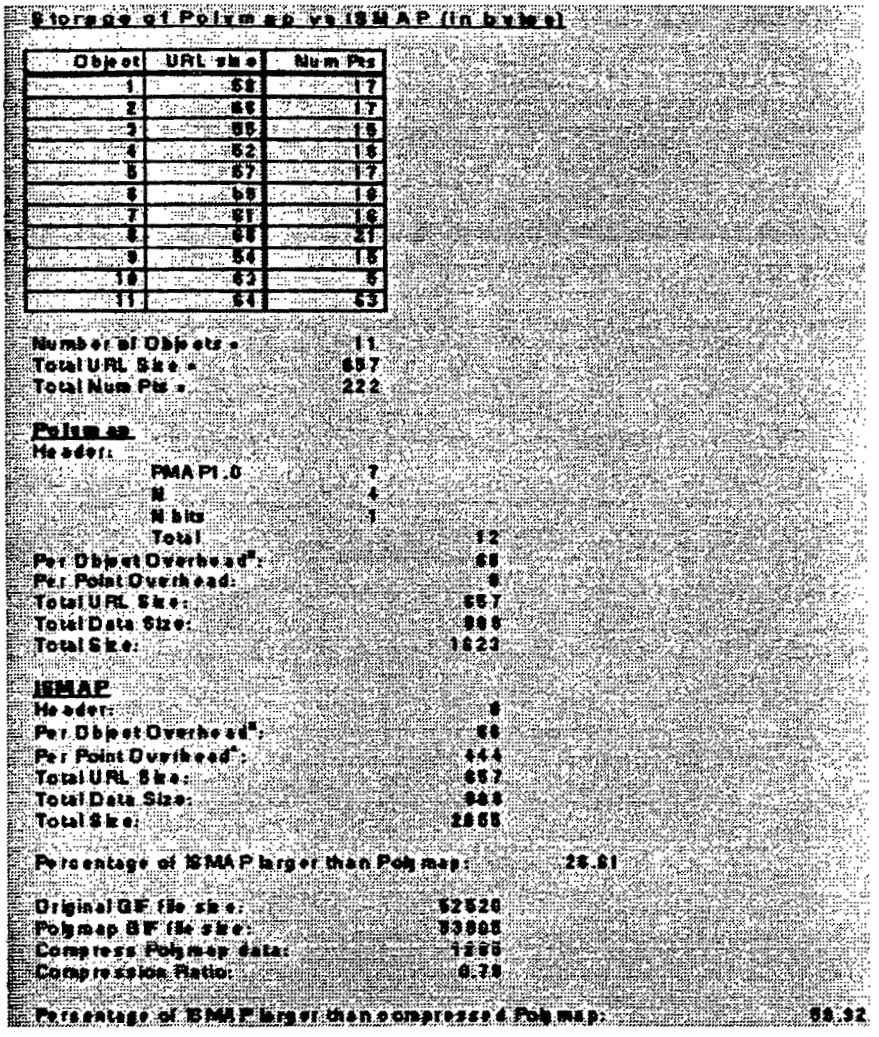
Fig 3 shows the Polymap-enabled NCSA Mosaic. Notice that the cursor is on a hot-spot, a hot cup of coffee, which is highlighted with the hot-spot outline. The URL indicator at the bottom of the image displays the location of the anchor.

Fig 3: Enhanced NCSA Mosaic with a Polymap HTML page.

4. Results

The results of the implementation are very encouraging. Our Polymapping implementation not only provides fast client-side decoding and realtime interactive feedback, but it also improves storage efficiency, setup speed and network communication efficacy.

The hot-spot polygon map of the image shown in Fig 2b would have consumed at least 2055 bytes of space if it had been stored in ASCII form (in an ISMAP file or HTML file as polygons[1]). Encapsulating the polygon data using the Polymap method yields an overhead of only 1285 bytes. The ISMAP file is 60% larger (Comparison in Table 1). Furthermore, Polymap storage effiency increases as the data size increases (sample comparison: Table 1 and Table 2).



Storage of Polymap vs ISMAP (in bytes)

Object	URL size	Num Pts
1	58	17
2	60	17
3	59	15
4	62	16
5	57	17
6	59	16
7	57	16
8	60	21
9	54	15
10	62	5
11	61	53

Number of Objects = 11
Total URL Size = 657
Total Num Pts = 222

Polymap

Header:
- PMAP1.0 7
- N 4
- N bits 1
- Total 12

Per Object Overhead*: 68
Per Point Overhead: 0
Total URL Size: 657
Total Data Size: 888
Total Size: 1623

ISMAP

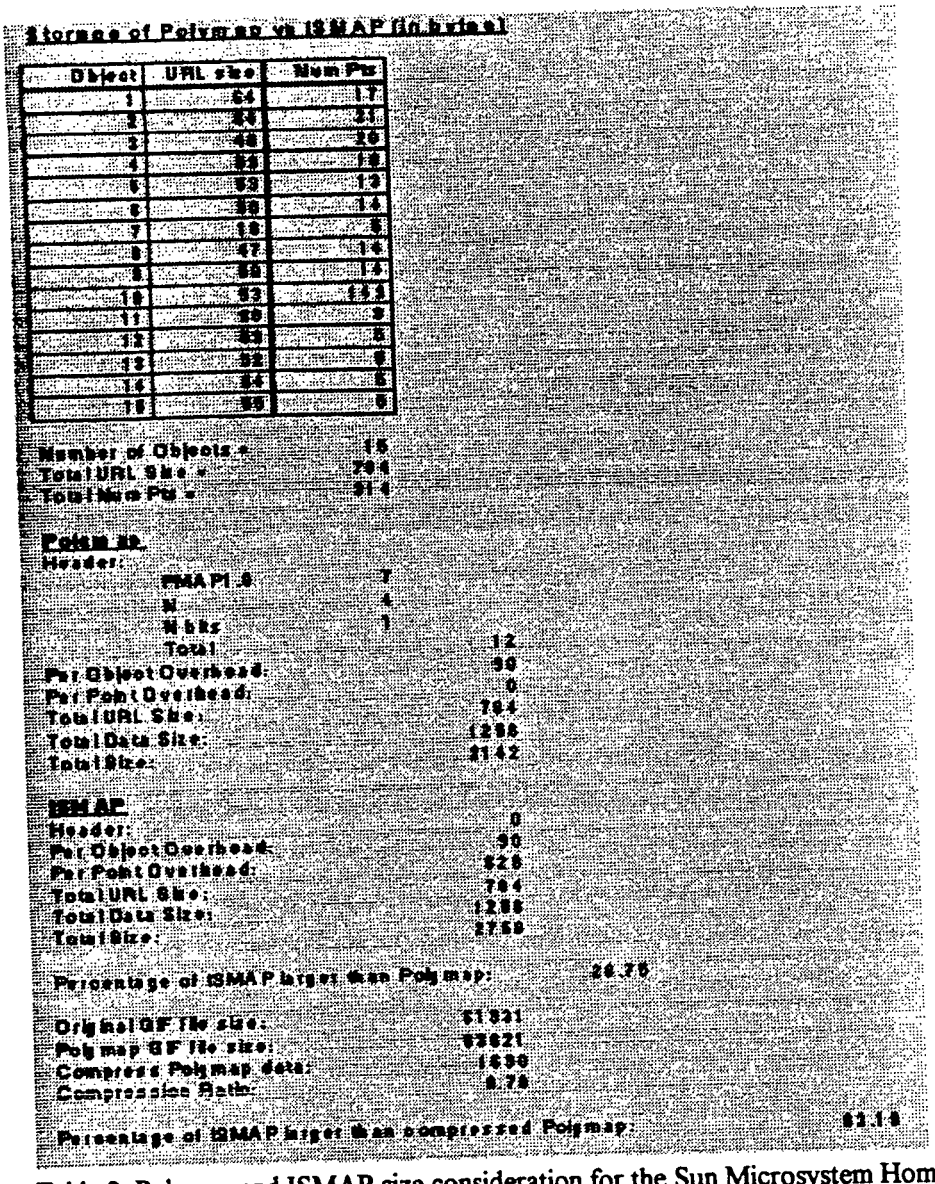
Header: 0
Per Object Overhead*: 68
Per Point Overhead*: 444
Total URL Size: 657
Total Data Size: 888
Total Size: 2865

Percentage of ISMAP larger than Polymap: 28.81

Original GIF file size: 52526
Polymap GIF file size: 53805
Compress Polymap data: 1285
Compression Ratio: 0.78

Percentage of ISMAP larger than compressed Polymap: 69.92

Table 1: Polymap and ISMAP size consideration for the White House image in Fig 2.



Storage of Polymap vs ISMAP (in bytes)

Object	URL size	Num Pts
1	64	17
2	54	31
3	48	26
4	33	18
5	53	13
6	50	14
7	18	3
8	47	14
9	53	13
10	53	153
11	20	3
12	53	5
13	32	0
14	54	5
15	35	0

Number of Objects = 15
Total URL Size = 704
Total Num Pts = 314

Polymap:
Header:
PMAP1.0 7
N 4
N bits 1
Total 12
Per Object Overhead: 90
Per Point Overhead: 0
Total URL Size: 704
Total Data Size: 1256
Total Size: 2142

ISMAP:
Header: 0
Per Object Overhead: 90
Per Point Overhead: 628
Total URL Size: 704
Total Data Size: 1256
Total Size: 2758

Percentage of ISMAP larger than Polymap: 28.75

Original GIF file size: 51831
Polymap GIF file size: 52621
Compress Polymap data: 1630
Compression Ratio: 0.78

Percentage of ISMAP larger than compressed Polymap: 81.14

Table 2: Polymap and ISMAP size consideration for the Sun Microsystem Home Page image in Fig 3.

[1] Polymap, like ISMAP, may also contain predefined geometric shapes such as circles. To facilitate comparison, we used only polygons.

[2] Each Polymap object has six bytes of the sequence (0xFFFF, object number, polygon number).

[3] Each ISMAP polygon object specification is preceeded by POLY url, thus six bytes, disregarding the size of variable, url, which has already been accounted for.

[4] Each ISMAP polygon vertex needs a comma between its x-coordinate and y-coordinate, and a space to separate itself from the next point or an end-of-line character if it is the last point.

5. Ongoing/Future work

We are currently adopting the geometric primitive specification of ISMAP into Polymap. More efforts will be put into refining the Polymapping toolkit and improving the polygon decoding speed on slower machines. Although our decoding engine only performs a sequential search, the performance is acceptable on machines at as low as 486 PC level. Enhancing the decoding engine with object elimination based on bounding boxes, and possibly use of a hash table will likely speed up the process.

Since there is a striking resemblance between the recently proposed Portable Network Graphics [2] (PNG) format and GIF in terms of overall image structure and layout, the Internet and other online (e.g. CompuServe) communities are shifting their attention from GIF to PNG. We are also planning on supporting PNG in future implementations of the Polymap toolkit.

6. Conclusions

The Polymap protocol provides a solution to the major problems confronting the ISMAP method of hot-spot interaction over the Web: redundant network access and dependence on the WWW server to decode local pointer events. Furthermore, because of the simplicity and efficiency of the implementation of this technique, we expect that Polymap will find wide acceptance on the Web. Polymap technology clearly demonstrates the possibility of realizing a complete hypergraphics multimedia system on the WWW based on the current electronic publishing infrastructure.

7. References

[1] Bernets-Lee, Tim, "Hypertext Transfer Protocol", Internet Draft.

[2] Boutell, Thomas, "PNG (Portable Network Graphics) Specification, Tenth Draft", URL: http://sunsite.unc.edu/~boutell/png.html.

[3] Connolly, Daniel W., "HTML 2.0 Specification Review Materials", URL: http://www.hal.com/users/connolly/html-spec.

[4] Doyle, Michael D., "MetaMAP®", URL: http://www.eolas.com/metamap/metamap.html

[5] Independent JPEG Group, "JPEG System Architecture", included in the publicly released JPEG 'C' Library, February 18, 1993.

[6] NCSA HTTPd Development Team, "NCSA Imagemap Tutorial", URL: http://hoohoo.ncsa.uiuc.edu/docs/setup/admin/Imagemap.html

[7] Ragget, Dave, "HTML+ Discussion Document", URL: http://info.cern.ch/hypertext/WWW/MarkUp/HTMLPlus/htmlplus_1.html.

[8] Siedman, James L., "An HTML Extension to Support Client-Side Image Maps", URL: http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/DDay/seidman/seidman.html.

University of Illinois
at Urbana-Champaign

Grants and Contracts Office
109 Coble Hall
801 South Wright Street
Champaign, IL 61820-6242

217 333-2186
217 333-2189 *fax*

January 27, 1994

U of I Ref. No. 94-NSF SBC-H-0227

MICHAEL D. DOYLE, DIRECTOR
UNIVERSITY OF CALIFORNIA
CENTER FOR KNOWLEDGE MANAGEMENT
530 PARNASSUS AVENUE, BOX 0840
SAN FRANSCISCO, CA 94143-0840

Title: DIGITAL LIBRARY PROPOSAL: EMBEDDED VISUALIZATION OBJECT FOR KNOWLEDGE ACCESS, CREATION...
Amount: $702,327.00
Period: OCT. 1, 1994 - SEPT. 30, 1998
Principal Investigator(s) JOSEPH B. HARDIN
Department NCSA
Type of Request: XX New Request, Supplement, Continuation, Renewal for Existing Award
Revision of Original Proposal Transmitted on _____;
Proposal Number _____;

Enclosed are copies of the above referenced proposal. This proposal has been approved for submission by the proper University administrative official(s).

Your consideration will be appreciated. Any contract or grant supporting the above described project must be issued in the University's corporate name, The Board of Trustees of the University of Illinois, Urbana, Illinois 61801.

Any questions of a non-technical nature regarding this proposal should be addressed to Willie Dozier, Carl Moore, Cindy Clennon or Steve Sanderson at the above telephone number.

Sincerely,

Wm. D. Morgan

Wm. D. Morgan, Associate Director
Grant and Contract Administration

WDM:ef

Enclosure 1 COPY

cc: BILL STRUTZ
UPS TRACKING #0563 9329 971

COVER SHEET FOR PROPOSALS TO THE NATIONAL SCIENCE FOUNDATION

For Consideration by NSF Organization Unit (Indicate the most specific unit known, i.e. program, division, etc.)	For NSF Use Only
University of California at San Francisco (prime contractor)	NSF Proposal Number
Program Announcement /Solicitation No./Closing Date: Research on Digital Libraries NSF 93-141 Feb 4, 94	

Date Received	Number of Copies	Division Assigned	Fund Code	File Location

Employer Identification Number (EIN) or Taxpayer Identification Number (TIN)	Show Previous Award No. If This is:	Is This Proposal Being Submitted to Another Federal Agency?
37-6000511	☐ A Renewal or ☐ An Accomplishment-Based Renewal	YES __ NO X ; If YES, List Acronym(s)

Name of Organization to Which Award Should be Made:	Address of Organization, Including Zip Code
The Board of Trustees of the University of Illinois	c/o Grants and Contracts 109 Coble Hall, MC-325 801 S. Wright St. Champaign, IL 61820-6242
Institutional Code (if known)	

Is Submitting Organization: ☐ For-Profit Organization; ☐ Small Business; ☐ Minority Business; ☐ Woman-Owned Business

Branch/Campus/Other Component (Where work is performed, if different)	Institutional Code (if known)
University of Illinois at Urbana-Champaign	

Digital Library Proposal: "Embedded Visualization Object for Knowledge Access, Creation and Management through the World Wide Web"

Requested Amount	Proposed Duration (1-60 months)	Requested Starting Date
$ 702,327	36 months	10/01/94

Check Appropriate Box(es) If This Proposal Includes Any of The Items Listed Below:

- ☐ Vertebrate Animals
- ☐ Human Subjects
- ☐ Historical Places
- ☐ Research Involving Genetically Engineered Organisms
- ☐ National Environmental Policy Act
- ☐ Proprietary and Privileged Information
- ☐ Disclosure of Lobbying Activities
- ☐ Facilitation for Scientists/Engineers with Disabilities
- ☐ Research Opportunity Award
- ☐ International Cooperative Activity: _____ Country/Countries

PI/PD Department	PI/PD Postal Address
NCSA	University of Illinois at Urbana-Champaign 1206 West Green Street Urbana, IL 61801
PI/PD Fax Number (217) 244-1987	

Names (Typed)	Social Security No.*	High Degree, Yr	Telephone Number	Electronic Mail Address
PI/PD Name				
Co-Pi/PD Hardin, Joseph B	348-42-2823	BA 1972	217-244-3473	hardin@ncsa.uiuc.edu
Co-Pi/PD				
Co-Pi/PD				
Co-Pi/PD				

NOTE: The FULLY SIGNED Certification Page must be submitted immediately following this Cover Sheet

*Submission of social security numbers is voluntary and will not affect the organization's eligibility for an award. However, they are an integral part of the NSF information system and assist in processing the proposal. SSN solicited under NSF Act of 1950, as amended.

NSF FORM 1207 (4/92)

University of Illinois
at Urbana-Champaign

National Center for
Supercomputing Applications

152 Computing Applications Building
605 East Springfield Avenue
Champaign, IL 61820

217 244-0072

Michael D. Doyle, Ph.D.
Director
Center for Knowledge Management
University of California, San Francisco
530 Parnassus Avenue, box 0840
San Francisco, CA 94143-0840

January 21, 1994

Michael:

This letter will serve as notification that SDG is pleased to collaborate with you on the Digital Library project "Embedded Visualization Object for Knowledge Access, Creation and Management through the World Wide Web." That scope of work for NCSA will focus on extensions of NCSA Mosaic and development of URN/URC methods.

We look forward to working with you on this proposed project.

Respectfully,

Joseph Hardin, Associate Director
Software Development Group

NSF/ARPA/NASA Digital Libraries RFP Response

Title: A Knowledge Managment Environment through the World Wide Web

Principal Investigator: Michael D. Doyle, Ph.D., UCSF Library and Center for Knowledge Managment

Specific Aims:

1) To develop a prototype knowledge management environment for the biomedical sciences which integrates access to online representations of the scientific literature, bibliographic databases, high-performance visualization technologies, large-scale scientific databases, and tools for authoring new-generation scientific publications.

1.a) To explore and evaluate the applicability of these tools in the areas of radiology and developmental & molecular biology.

2) To provide a means for relating digital forms of spatial, functional, and conceptual information as a basis for linking the biomedical scientific literature, through the Red Sage electronic journals project, to data resources provided through the Visible Human Project, The Human Brain Project, The Visible Embryo Project, The Human Genome Project, The Protein Database, and other large-scale biomolecular and biostructural databases.

2.a) To exploit these linking strategies in the creation of a set of integrated semi-automatic front ends to varied scientific databases accessible through the Internet.

2.b) To incorporate these linking methodologies into interactive authoring and editorial tools, allowing the creation of online publications that can embed visualizations and simulations which draw data from these Internet-accessible scientific databases.

3) To develop tools which provide access to interactive visualization and analysis of massive biomedical datasets through the Internet's World Wide Web distributed hypermedia network.

3.a) To refine and extend our existing algorithms enabling distributed visualization and analysis software "engines" which can be efficiently accessed by remote users through the Internet.

3.b) To refine and extend our existing algorithms to allow the display and real-time interactive control of three-and four-dimensional data visualization and analysis tools within hypermedia documents viewed using NCSA's Mosaic graphical browser to the World Wide Web.

3.c) To develop algorithms which use novel compression technologies for the optimized interactive remote control of computationally-intensive graphical applications through the Internet.

3.d) To integrate a,b & c into a system which allows real-time remote access to distributed parallel computational applications for visualization and analysis resources within a distributed hypermedia environment.

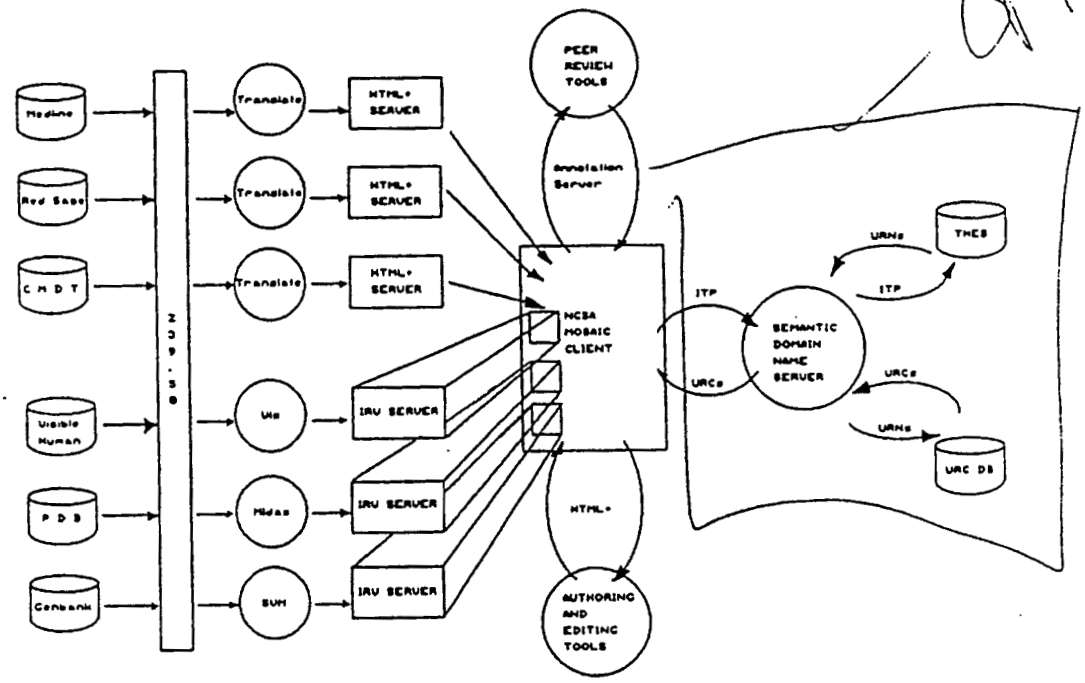
4) To explore extensions of the paradigm of scientific publishing which are made possible through use of current multimedia technologies in a networked environment, including:

4.a) publishing multidimensional datasets integrated with articles, eg: MRI and molecular data, perferred views, animations, interactive visualizations, interactive mathematical models, and

4.b) development of scientific authoring tools for publications which exist only in the networked environment.

4.b.1) This will include integration of HTML+ WYSIWYG authorial and editorial tools, multidimensional data visualization applications, molecular modelling and database management tools into an interactive scientific publishing environment.

System Diagram:



Definitions:		
	HTML+:	Hypertext Mark-up Language -- This is the language that World Wide Web databases are encoded in, and that Mosaic interprets.
	IRV Server:	UCSF CKM's Interactive Remote Visualization Server -- This allows interactive real-time visualization tools to be embedded into Mosaic documents.
	Vis:	UCSF CKM's distributed remote volume visualization tool
	Midas:	UCSF CGL's molecular visualization package
	SVM:	Sequence Visualization Module -- An as-yet unamed tool for graphical display of genetic sequence data.
	ITP:	Informal Text Phrase -- A user-entered search term, or a word or phrase of text that the user highlights from within a document.
	URN:	Universal Resource Name -- A persistent, location-independent identifier for an object.
	URL:	Universal Resource Location -- The address of an object. It contains enough information to identify a communications protocol and retrieve the object.
	URC:	Universal Resource Characteristics -- Any combination of one or more URNs or URLs with meta information (e.g. author, format, compression method).

Description:

The system will draw from of a number of fundamental databases including bibliographic data (Medline) in the form of MARC records, journal publication data (Red Sage) in the form of SGML header and Postscript files, encyclopedic reference text data (CMDT) stored in an object-oriented SGML database, volumetric anatomical data (Visible Human Project) stored as NCSA HDF datasets, protein structure data (Protein Data Bank) stored as PDB files, and genetic sequence data (Genbank) stored as compressed ASCII strings (?, I'm guessing about Genbank).

These databases will reside behind a Z39.50 interface layer which yeilds, to the requesting client, the respective dataset in its native form. This data then goes through a translation layer where the data is either translated directly into HTML+ (Medline, Red Sage, CMDT) or loaded into a native-data visualization tool (Visible Human, PDB, Genbank). The HTML+ code is then passed to a set of HTML+ servers, which can be browsed by the Mosaic client. The visualization data is handled differently. The graphical I/O of the relevant visualization tool is passed to an interactive remote visualization (IRV) server, which handles both mapping of the display output from the visualization tool onto embedded live-visualization windows within the Mosaic-browsable HTML+ documents, as well as capture of user-entered mouse and keyboard events within the visualization windows and transmission of those mouse and keyboard events back to the relevant visualization tools. The user, browsing the system with the project's enhanced version of the Mosaic client, is presented with data and visualizations derived from these various databases, yet embedded into coherent, multimedia Mosaic documents.

For multimedia documents that have been explicitly pre-composed, the linking of these various data resources can take the form of universal resource names (URNs) that are encoded as tags into the HTML+ documents. This is passed to the system's semantic domain name server, for resolution of the information object's location and retrieval means. The URNs are used as indices in order to look up the relevant universal resource characteristics (URCs) in a URC database, which yeilds the universal resource location (URL), or physical adress, of the information object in question.

Semi-automatic means will be provided for a user to search for arbitrary information objects on the system by either keying in a search word or phrase, or by highlighting a not-already-hyperlinked section of text that (s)he happens to be viewing within the Mosaic client at the time. This informal text phrase (ITP) is then passed to the semantic domain name server, which passes it on to a universal resource thesaurus (which will incorporate elements of the NLM's UMLS system). The thesaurus compares the ITP to its database of terms and phrases and returns a rank-ordered list of URNs that are likely to match the object in question. These URNs are then passed to the URC database for resolution of URLs that point to information objects on the Internet that are most likely to match the ITP that the user employed to initiate the search. The user is presented with a rank ordered set of textual descriptions of likely matches which are hyperlinked, via their URLs, to the data in question. Clicking upon a selection from this list loads the related data into the relevant visualization server (IRV) or HTML+ server, and a second Mosaic window pops up to allow viewing or interaction with that dataset.

A set of authoring and editing tools will be designed to allow the interactive WYSIWYG creation of HTML+ documents, as well as allowing the embedding of visualizations, etc., which can be created using the interactive remote visualization tools, and which can use data from the various scientific databases mentioned above. Alternatively, the author can use his/her own datasets, which would be uploaded to an Internet-accesible World Wide Web server. The journal editor can use the same set of tools to edit submitted articles and to communicate changes to the text with the author. This, of course, would occur in a private, access-controlled, area of the system, so that confidentiality of the material to be published can be controlled.

Other private, access-controlled HTML+ servers will be used to administer the peer review process. A modification of NCSA's Mosaic-based group annotation server will be developed to allow the jounal editor to exercise precise control and documentation of each reviewer's comments and suggestions.

Contributions:

UCSF CKM:

- Development of Z39.50-compliant experimental (subset) databases for storage of Visible Human data, PDB data, and Genbank sequence data.
- Cooperation with AT&T in the development of an object-oriented SGML-based database for the Handbook of Current Medical Diagnosis and Treatment (CMDT)
- Development of an experimental Z39.50 interface to Medline data (will be unnecessary if UC's DLA can provide such an interface to Melvyl Medline early enough into the project timeline)
- Development of translator servers to translate Medline MARC records, CMDT SGML data and Red Sage SGML/Postscript data into HTML+
- Development of a set of HTML+ documents that act as browsers to Medline, CMDT, and Red Sage
- Refinement and further development of Vis to allow better distribution of computation and better integration with Mosaic.
- Cooperation with CGL to adapt Midas for integration within Mosaic, and to identify and adapt a suitable program for graphical display of genetic sequence data.
- Refinement and further development of the interactive remote visualization server, and its incorporation (with NCSA's help) within the Mosaic environment.
- Development, in cooperation with NCSA, of an enhanced version of the Mosaic client to allow easier integration of external programs within Mosaic-readable documents.
- Development, in cooperation with Springer-Verlag and NCSA, of an interactive WYSIWYG editor for creation of HTML+ documents, and for embedding visualizations created using CKM's IRV tools, as well as development of a modified version of NCSA's group annotation server to support the peer review process.
- Development, in cooperation with AT&T, of an object-oriented SGML-based URC database
- Development, in cooperation with UCSF's CGL, UCSF's Radiology Dept., Washington Univ., and AT&T, of a Semantic domain name server and a URN Thesarus, based upon AT&T's object-oriented SGML database technology.
- Development, in cooperation with UCSF's CGL, UCSF's Radiology Dept., Washington Univ., and Springer-Verlag of a set of sample content for use in evaluating the effectiveness of the system, as well as for demonstration of the results of the project.

UCSF CGL:

- Cooperation with CKM to adapt Midas for integration within Mosaic, and to identify and adapt a suitable program for graphical display of genetic sequence data.
- Contributing to the refinement and further development of the interactive remote visualization server, and its incorporation (with NCSA's help) within the Mosaic environment.

- Development, in cooperation with UCSF's CKM, UCSF's Radiology Dept., Washington Univ., and AT&T, of a Semantic domain name server and a URN Thesarus, based upon AT&T's object-oriented SGML database technology.

- Development, in cooperation with UCSF's CKM, UCSF's Radiology Dept., Washington Univ., and Springer-Verlag of a set of sample content for use in evaluating the effectiveness of the system, as well as for demonstration of the results of the project.

Washington University:

- Development, in cooperation with UCSF's CKM, UCSF's CGL, and AT&T, of a Semantic domain name server and a URN Thesarus, based upon AT&T's object-oriented SGML database technology.

- Development, in cooperation with UCSF's CKM, and UCSF's CGL., and Springer Verlag of a set of sample content for use in evaluating the effectiveness of the system, as well as for demonstration of the results of the project.

AT&T Bell Laboratories:

- Development of Z39.50 interface to the RightPages server..

- Cooperation with CKM in the development of an object-oriented SGML-based database for the Handbook of Current Medical Diagnosis and Treatment (CMDT)

- Development, in cooperation with CKM, of an object-oriented SGML-based URC database

- Development, in cooperation with UCSF's CGL, UCSF's Radiology Dept., Washington Univ., and CKM, of a Semantic domain name server and a URN Thesarus, based upon AT&T's object-oriented SGML database technology.

Springer-Verlag:

- Development, in cooperation with UCSF's CKM and NCSA, of an interactive WYSIWYG editor for creation of HTML+ documents, and for embedding visualizations created using CKM's IRV tools, as well as development of a modified version of NCSA's group annotation server to support the peer review process.

- Development, in cooperation with UCSF's CKM, and UCSF's CGL., and Washington Univ. of a set of sample content for use in evaluating the effectiveness of the system, as well as for demonstration of the results of the project.

NCSA:

- Cooperation with CKM in developing an enhanced version of Mosaic to allow easier integration of a client modeule for CKM's interactive remote visualization server.

- Cooperation with CKM and Springer-Verlag in the modification of NCSA's group annotation server to facilitate the peer-review process.

Personnel:

Co-Investigators:

UCSF:

Library & CKM: Richard Lucier, David Martin, Zoe Stavri, Ph.D., Cheong Ang, Marc Salomon
Radiology: Tom Budinger, Ph.D.
Molecular & developmental Biology: Tom Ferrin, Ph.D., Charles Ordahl, PhD.

Washington University (molecular biology): Toni Kazic, PhD
Bell Laboratories: Ed Szurkowski, Guy Story
Springer Verlag: Bob Badger, PhD
NCSA: Joseph Hardin, PhD, & Mosaic development group
SFSU: Computer Science Dept. MS students

Timetable: 4 years

Budget: 1.2 $M/year

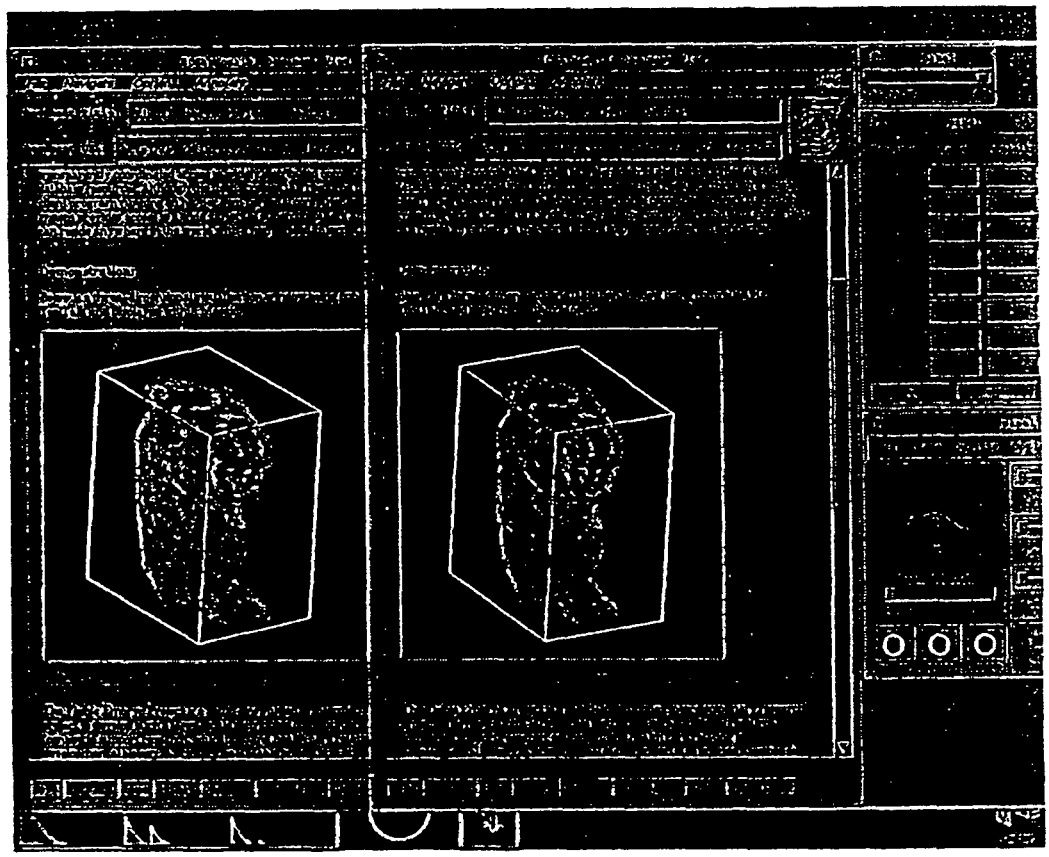


Figure 1: **A stereo-pair illustration of interactive real-time 3-dimensional human embryonic volume reconstructions embedded within an NCSA Mosaic document. This technology was developed by the Center for Knowledge Management at the University of California, San Francisco, and was demonstrated there in November, 1993.**

Appendix 7: Financial Projection Spreadsheets

7.a) 1995-1996

Eòlas				
Income Statement Projection (1,000's)				
1995-1996				
	1995	% of Rev	1996	% of Rev
Revenue	60		45,220	
Expenses				
Cost of Revenue	$3	5%	$4,930	11%
Engineering & Tech Support	$78	130%	$3,656	8%
Sales & Marketing	$85	142%	$11,747	26%
G&A	$490	817%	$2,514	6%
Total Expenses	$656	1094%	$22,846	51%
Operating Income	($596)	-994%	$22,374	49%
Net Interest Income (expense)	$13		$11	
Provision for taxes @40%	$0		$8,950	
Net Income	($609)	-1015%	$13,413	30%
Net Profit Margin	-1015%		30%	
Total Employees	18		50	
Revenue per Employee	$3		$904	

Eólas

World-Wide Operating Plan ($,000's)

1995-1996

	1995 Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	1995	1996 Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	1996
Revenue																										
License Revenue	0	0	0	0	0	0	0	0	0	0	0	0	0	66	66	66	492	492	492	1,196	1,196	1,196	8,216	8,216	8,216	33,816
Product Revenue	0	0	0	0	0	0	0	0	0	0	0	0	0	150	150	150	440	210	920	850	900	1,150	3,660	4,100	4,750	17,130
Service Revenue	0	0	0	0	0	0	0	0	80	0	0	0	0	80	22	22	22	85	147	141	186	210	236	977	1,032	1,087	4,181
Total	0	0	0	0	0	0	0	0	80	0	0	0	0	80	241	241	241	1,025	849	1,553	2,631	2,306	2,581	10,743	11,348	12,883	45,226
Expenses																										
Cost of Revenue																										
COGS	0	0	0	0	0	0	0	0	0	0	0	0	0	16	16	16	44	21	92	86	80	116	366	410	476	1,712
Royalties Paid @4%	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	20	20	20	48	48	48	340	340	340	957
Sales Comm.	0	0	0	0	0	0	0	3	0	0	0	0	3	12	12	12	81	42	78	122	116	139	837	847	805	2,361
Total	0	0	0	0	0	0	0	3	0	0	0	0	3	30	30	30	118	83	189	256	253	292	1,141	1,226	1,327	4,030
Administration																										
Salaries	0	10	14	14	14	14	17	20	20	20	20	20	202	43	64	66	62	62	62	62	62	62	62	62	62	766
T&E @ 30%	0	3	4	4	4	4	5	6	6	6	6	12	61	13	19	17	18	18	18	18	18	18	18	18	18	212
Other @6%	0	1	1	1	1	1	1	1	1	1	1	3	13	3	4	4	4	4	4	4	4	4	4	4	4	44
Total	0	14	19	19	19	19	23	28	28	28	28	36	276	58	74	79	84	84	84	84	84	84	84	84	84	967
Marketing																										
Salaries	0	0	0	0	0	0	0	0	0	0	0	16	16	17	17	17	38	38	38	43	43	43	43	43	43	431
T&E @ 30%	0	0	0	0	0	0	0	0	0	0	0	6	6	8	6	6	11	11	11	13	13	13	13	13	13	129
Ads	0	0	0	0	0	0	0	0	0	0	0	0	0	100	100	100	200	800	800	800	800	800	800	800	800	8,700
Direct Mktg	0	0	0	0	0	0	0	0	0	0	0	0	0	50	50	50	50	50	100	100	100	100	100	100	100	950
Collateral	0	0	0	0	0	0	0	0	0	0	0	0	0	60	60	60	60	60	60	75	75	75	75	75	75	750
PR	0	0	0	0	0	0	0	0	0	0	0	0	0	50	50	50	50	50	50	50	50	50	50	50	50	500
Trade Shows	0	0	0	0	0	0	0	0	0	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	240
Coop Ad	0	0	0	0	0	0	0	0	0	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	240
Travel	0	0	0	0	0	0	0	0	0	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	240
Other @5%	0	0	0	1	0	1	0	0	2	3	3	3	13	10	10	10	20	20	20	20	20	20	60	60	60	360
Total	0	0	0	1	0	1	0	0	2	3	3	3	13	10	10	10	20	20	20	20	20	20	60	60	60	360
Travel	0	0	0	0	0	0	0	0	0	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	240
Other @5%	0	0	0	0	0	0	0	0	0	0	0	0	0	20	20	20	20	20	20	20	20	20	20	20	20	240
Total	0	0	0	1	0	1	0	0	2	3	3	3	13	10	10	10	20	20	20	20	20	20	20	20	20	240
Sales																										
Salaries	0	0	0	0	0	0	0	0	0	0	0	0	0	11	11	11	17	28	28	28	28	28	28	28	28	276
T&E @ 30%	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	5	8	8	8	8	8	8	8	8	81
Travel	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	5	8	8	8	8	8	8	8	8	81
Other @10%	0	0	0	0	0	0	0	0	0	0	0	0	0	3	3	3	5	8	8	8	8	8	8	8	8	81
Total	0	0	0	0	0	0	0	0	0	0	0	0	0	20	20	20	32	52	52	52	52	52	52	52	52	519
Tech Support																										
Salaries	0	0	0	0	0	0	0	0	0	0	0	0	0	4	4	5	6	17	26	26	26	26	26	26	26	223
T&E @ 30%	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	5	8	8	8	8	8	8	8	67
Travel	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	5	8	8	8	8	8	8	8	67
Other @10%	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	5	8	8	8	8	8	8	8	67
Total	0	0	0	0	0	0	0	0	0	0	0	0	0	6	6	8	9	32	50	50	50	50	50	50	50	424
Engineering																										
Salaries	0	0	0	0	0	0	0	0	0	0	0	63	63	66	66	76	86	111	130	130	130	130	130	130	130	1,364
T&E @ 30%	0	0	0	0	0	0	0	0	0	0	0	18	18	20	20	24	29	33	36	36	36	36	36	36	36	381
Research	0	0	0	0	0	0	0	0	0	0	0	0	0	100	100	100	100	100	100	100	100	100	100	100	100	1,200
Cont. Labor	0	0	0	0	0	0	0	0	0	0	0	0	0	10	10	10	10	10	10	10	10	10	10	10	10	120
Travel	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2	3	3	3	3	3	3	30
Other @5%	0	0	0	0	0	0	0	0	0	0	0	3	3	10	10	11	12	13	13	13	14	14	14	14	14	153
Total	0	0	0	0	0	0	0	0	0	0	0	73	73	210	210	225	251	269	291	292	294	294	294	294	294	3,197
Overhead																										
Rent	0.0	0.4	0.7	0.7	0.7	0.7	0.7	1.1	1.1	1.1	1.1	6.5	14.6	7.4	7.7	8.8	13.3	14.7	16.8	16.8	17.8	17.8	17.8	17.8	17.8	173.3
Utilities	0.0	0.1	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	1.4	3.2	1.8	1.7	2.1	2.9	3.2	3.6	3.7	3.9	3.9	3.9	3.9	3.9	36.1
Insurance	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	10.0	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	10.0
Prof. Services	7	7	7	7	7	7	7	7	7	7	7	7	84	20	20	20	20	20	20	20	20	20	20	20	20	240
Office Suppl.	0.0	0.4	0.8	0.8	0.8	1.0	1.0	1.4	1.7	1.7	1.7	1.8	14.0	7.5	7.9	8.7	13.3	14.8	16.2	16.8	16.2	20.0	20.0	20.0	20.0	196.0
Depreciation	0.0	0.3	1.0	1.0	1.0	1.0	1.0	1.4	1.7	1.7	1.7	1.8	14.6	7.5	7.9	8.7	13.3	14.8	16.2	16.8	16.2	20.0	20.0	20.0	20.0	196.0
Rep. & Maint.	0.0	0.2	0.4	0.4	0.4	0.4	0.4	0.8	0.8	0.8	0.8	0.8	6.0	4.2	4.4	5.4	7.8	8.4	9.4	9.8	10.0	10.0	10.0	10.0	10.0	98.0
Phone	0.0	0.4	0.8	0.8	0.8	0.8	0.8	1.2	1.2	1.2	1.2	1.2	10.4	0.4	0.8	10.8	16.2	16.8	16.8	16.2	20.0	20.0	20.0	20.0	20.0	198.0
Postage	0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.2	1.1	2.5	1.3	1.3	1.8	2.3	2.5	2.8	2.8	3.0	3.0	3.0	3.0	3.0	29.7
Internet	0.0	0.5	1.0	1.0	1.0	1.0	1.0	1.5	1.5	1.5	1.5	9.0	24.5	10.5	11.0	13.5	19.0	21.0	23.5	24.0	25.0	25.0	25.0	25.0	25.0	247.5
Interest Expense	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.8	2.8	2.8	12.5	2.3	2.0	1.8	1.5	1.3	1.0	0.8	0.5	0.3	0.0	0.0	0.0	14.5
Other @10%	0.6	1.0	1.3	1.3	1.3	1.3	1.3	1.5	1.5	1.5	1.5	6.0	19.5	7.0	7.2	8.4	11.0	11.9	13.0	13.5	13.8	13.8	13.8	13.8	13.8	146.6
Total	8.3	11.9	14.8	14.8	14.8	14.8	14.5	17.9	17.8	19.6	19.6	54.0	224.8	79.4	81.9	94.4	122.1	132.0	144.4	146.7	151.9	151.6	151.4	151.4	151.4	1,558.3
Tot. Op. Exp.	8	26	34	37	36	37	40	53	52	50	50	229	660	778	796	844	1,102	1,488	1,686	1,646	1,903	2,255	3,555	3,221	3,321	22,857
Total Op. Inc	(8)	(26)	(34)	(37)	(36)	(37)	(40)	27	(52)	(50)	(50)	(229)	(660)	(538)	(555)	(603)	(137)	(639)	(136)	185	404	325	7,188	8,127	8,742	22,369
Head Count	0	1	2	2	2	2	2	3	3	3	3	18		21	22	27	38	42	47	48	50	50	50	50	50	
Notes:																										

1995-1996

Eólas Technologies Incorporated Confidential

Page 2

Confidential - Subject to Protective Order
Eolas v. Microsoft CV99-C0626

E 006334

Eolas																											
Sales Projection																											
1995-1996																											
		1995													1996												
		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	1995	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	1996
Licenses																											
Weblet Tech Licenses																											
2M Units @	$1,000,000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	3
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$1,000	$1,000	$1,000	$3,000
500K Units @	$400,000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	6
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$800	$800	$800	$2,400
200K Units @	$200,000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	5	5	5	18
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$200	$200	$200	$1,000	$1,000	$1,000	$3,600
50K Units @	$100,000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	2	2	2	10	10	10	39
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$100	$100	$100	$200	$200	$200	$1,000	$1,000	$1,000	$3,900
10K Units @	$40,000	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	5	5	5	10	10	10	25	25	25	123
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$40	$40	$40	$200	$200	$200	$400	$400	$400	$1,000	$1,000	$1,000	$4,920
1K Units @	$5,000	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	8	8	8	20	20	20	40	40	40	210
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$10	$10	$10	$40	$40	$40	$100	$100	$100	$200	$200	$200	$1,050
100 Units @	$800	0	0	0	0	0	0	0	0	0	0	0	0	0	10	10	10	20	20	20	40	40	40	100	100	100	510
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$8	$8	$8	$16	$16	$16	$32	$32	$32	$80	$80	$80	$408
25 Units @	$250	0	0	0	0	0	0	0	0	0	0	0	0	0	20	20	20	30	30	30	50	50	50	150	150	150	750
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$5	$5	$5	$8	$8	$8	$13	$13	$13	$38	$38	$38	$188
5 Units @	$75	0	0	0	0	0	0	0	0	0	0	0	0	0	50	50	50	75	75	75	100	100	100	250	250	250	1,425
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$4	$4	$4	$6	$6	$6	$8	$8	$8	$19	$19	$19	$107
1 Unit @	$20	0	0	0	0	0	0	0	0	0	0	0	0	0	100	100	100	150	150	150	300	300	300	1,000	1,000	1,000	4,230
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$2	$2	$2	$3	$3	$3	$4	$4	$4	$20	$20	$20	$87
Subtotal Weblet Licenses		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$69	$69	$69	$472	$472	$472	$1,156	$1,156	$1,156	$6,156	$6,156	$6,156	$23,559
MetaMAP Licenses																											
Units @	$10,000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	4	4	4	6	6	6	36
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$20	$20	$20	$40	$40	$40	$60	$60	$60	$360
Products																											
WebRouter																											
100K Unit Sales @	$400,000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	3
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$400	$0	$0	$400	$0	$0	$400	$1,200
25K Unit Sales @	$250,000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	5
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$250	$0	$250	$0	$250	$0	$250	$0	$250	$1,250
10K Unit Sales @	$150,000	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	3	3	3	6	6	6	33
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$150	$150	$150	$150	$150	$150	$450	$450	$450	$900	$900	$900	$4,950
SDK																											
Unit Sales @	$200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100	200	500	500	500	1,000	1,000	5,000	5,000	13,800
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$20	$40	$100	$100	$100	$200	$200	$1,000	$1,000	$2,760
Weblets																											
Unit Sales @	$20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1,000	1,000	1,000	5,000	5,000	5,000	10,000	10,000	10,000	48,000
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$20	$20	$20	$100	$100	$100	$200	$200	$200	$960
WebWorks																											
Unit Sales @	$200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10,000	10,000	10,000	30,000
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$2,000	$2,000	$2,000	$6,000
Muse																											
Unit Sales @	$30	0	0	0	0	0	0	0	0	0	0	0	0	0	1,000	1,000	1,000	5,000	5,000	5,000	8,000	8,000	8,000	10,000	10,000	10,000	72,000
Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$30	$30	$30	$150	$150	$150	$240	$240	$240	$300	$300	$300	$2,160
Products Total		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$150	$150	$150	$440	$210	$920	$650	$900	$1,150	$3,550	$4,100	$4,760	$17,120
Services																											
Net Pub Consulting																											
Man Months @	$10,000	0	0	0	0	0	0	0	4	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
(10 MM per NetPub Lic.)		$0	$0	$0	$0	$0	$0	$0	$40	$0	$0	$0	$0	$40	$0	$0	$0	$0	$77	$0	$0	$0	$0	$0	$0	$0	$77
Support Contracts*																											
% of Lic & Prod Rev	10%	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$22	$22	$22	$93	$70	$141	$185	$210	$235	$977	$1,032	$1,097	$4,104
Summary																											
Product																											
License Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$69	$69	$69	$492	$492	$492	$1,196	$1,196	$1,196	$6,216	$6,216	$6,216	$23,919
Product Revenue		$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$150	$150	$150	$440	$210	$920	$650	$900	$1,150	$3,550	$4,100	$4,760	$17,120
Services Revenue		$0	$0	$0	$0	$0	$0	$0	$40	$0	$0	$0	$0	$40	$22	$22	$22	$93	$147	$141	$185	$210	$235	$977	$1,032	$1,097	$4,181
Total Revenue		$0	$0	$0	$0	$0	$0	$0	$40	$0	$0	$0	$0	$40	$241	$241	$241	$1,025	$849	$1,553	$2,031	$2,306	$2,581	$10,743	$11,348	$12,073	$45,220

*Support viewed on cash-in basis

Confidential - Subject to Protective Order
Eolas v. Microsoft CV99-C0626

E 006335

Eôlas
Payroll Plan
1995-1996

		1995													1996												
		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	1995		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	1996
# Employees	Ann.Sal.	0	1	2	2	2	2	2	3	3	3	3	18		Ann.Sal.	21	22	27	38	42	47	48	50	50	50	50	50	
Administration																												
President	125		10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	10.4	114.6	137.5	11.5	11.5	11.5	11.5	11.5	11.5	11.5	11.5	11.5	11.5	11.5	137.5
CTO	125													0.0	137.5		11.5	11.5	11.5	11.5	11.5	11.5	11.5	11.5	11.5	11.5	126.0
Executive VP	100												8.3	8.3	110.0	9.2	9.2	9.2	9.2	9.2	9.2	9.2	9.2	9.2	9.2	9.2	110.0
VP Admin & Legal	70							6.4	6.4	6.4	6.4	6.4	6.4	38.5	77.0	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	77.0
Admin	40			3.3	3.3	3.3	3.3		3.3	3.3	3.3	3.3	3.3	30.0	44.0	3.7	3.7	3.7	3.7	3.7	3.7	3.7	3.7	3.7	3.7	3.7	44.0
Admin	40												3.3	3.3	44.0	3.7	3.7	3.7	3.7	3.7	3.7	3.7	3.7	3.7	3.7	3.7	44.0
Admin	40													0.0	44.0			3.7	3.7	3.7	3.7	3.7	3.7	3.7	3.7	3.7	36.7
Admin	40													0.0	44.0				3.7	3.7	3.7	3.7	3.7	3.7	3.7	3.7	33.0
CFO	81												7.6	7.6	100.1	8.3	8.3	8.3	8.3	8.3	8.3	8.3	8.3	8.3	8.3	8.3	100.1
Total	671	0.0	10.4	13.8	13.8	13.8	13.8	16.8	20.2	20.2	20.2	20.2	38.4	202.3	738.1	42.7	54.2	57.0	61.6	61.6	61.6	61.6	61.6	61.6	61.6	61.6	706.3
Marketing																											
VP Marketing	100												8.3	8.3	110.0	9.2	9.2	9.2	9.2	9.2	9.2	9.2	9.2	9.2	9.2	9.2	110.0
Product Mgr	80												6.7	6.7	88.0	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	88.0
Product Mgr	80													0.0	88.0				7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	66.0
Marketing Mgr	60													0.0	66.0				5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	49.5
Marketing Mgr	60													0.0	66.0				5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	49.5
Marketing Mgr	60													0.0	66.0						5.5	5.5	5.5	5.5	5.5	5.5	
Admin	30													0.0	33.0				2.8	2.8	2.8	2.8	2.8	2.8	2.8	2.8	24.8
Total	470	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	15.0	15.0	517.0	16.5	16.5	16.5	37.8	37.8	37.8	43.1	43.1	43.1	43.1	43.1	387.8
Sales																											
Sales	80												5.0	5.0	66.0	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	66.0
Sales	60													0.0	66.0	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	66.0
Sales	60													0.0	66.0				5.5	5.5	5.5	5.5	5.5	5.5	5.5	5.5	49.5
Sales	60													0.0	66.0					5.5	5.5	5.5	5.5	5.5	5.5	5.5	44.0
Sales	60													0.0	66.0					5.5	5.5	5.5	5.5	5.5	5.5	5.5	44.0
Total	300	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.0	5.0	330.0	11.0	11.0	11.0	16.5	27.5	27.5	27.5	27.5	27.5	27.5	27.5	266.5
Tech Support																											
Tech Support Mgr	80													0.0	88.0			7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	7.3	73.3
Support Eng	50												4.2	4.2	55.0	4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	55.0
Support Eng	50													0.0	55.0			4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	45.8
Support Eng	50													0.0	55.0				4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	41.3
Support Eng	50													0.0	55.0				4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	41.3
Support Eng	50													0.0	55.0					4.6	4.6	4.6	4.6	4.6	4.6	4.6	32.1
Support Eng	50													0.0	55.0					4.6	4.6	4.6	4.6	4.6	4.6	4.6	32.1
Support Eng	50													0.0	55.0					4.6	4.6	4.6	4.6	4.6	4.6	4.6	32.1
Total	350	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.2	4.2	473.0	4.6	4.6	16.5	25.7	25.7	38.4	38.4	38.4	38.4	38.4	38.4	352.9
Engineering																											
Chief Engineer	100												8.3	8.3	110.0	9.2	9.2	9.2	9.2	9.2	9.2	9.2	9.2	9.2	9.2	9.2	110.0
Senior Scientist	100												8.3	8.3	110.0	9.2	9.2	9.2	9.2	9.2	9.2	9.2	9.2	9.2	9.2	9.2	110.0
Sr. Designer	70													0.0	77.0	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	77.0
Win Eng	70												5.8	5.8	77.0	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	77.0
Win Eng	70												5.8	5.8	77.0	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	77.0
Win Eng	70													0.0	77.0			6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	64.2
Win Eng	70													0.0	77.0				6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	57.8
Win Eng	70													0.0	77.0				6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	57.8
Win Eng	70													0.0	77.0				6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	57.8
Unix Eng	70												5.8	5.8	77.0	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	77.0
Unix Eng	70												5.8	5.8	77.0	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	77.0
Unix Eng	70													0.0	77.0					6.4	6.4	6.4	6.4	6.4	6.4	6.4	51.3
Unix Eng	70													0.0	77.0					6.4	6.4	6.4	6.4	6.4	6.4	6.4	51.3
Mac Eng	70												5.8	5.8	77.0	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	77.0
Mac Eng	70												5.8	5.8	77.0	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	77.0
Mac Eng	70													0.0	77.0				6.4	6.4	6.4	6.4	6.4	6.4	6.4	6.4	57.8
Sys Admin	50													0.0	55.0	4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	4.6	55.0
QA Engineer	48													0.0	52.8			4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	4.4	44.0
QA Engineer	48													0.0	52.8					4.4	4.4	4.4	4.4	4.4	4.4	4.4	30.8
QA Engineer	48													0.0	52.8					4.4	4.4	4.4	4.4	4.4	4.4	4.4	30.8
QA Engineer	48													0.0	52.8							4.4	4.4	4.4	4.4	4.4	22.0
QA Engineer	48													0.0	52.8							4.4	4.4	4.4	4.4	4.4	22.0
QA Engineer	48													0.0	52.8								4.4	4.4	4.4	4.4	22.0
Total	1300	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	51.7	51.7	1,540.0	87.8	87.8	78.7	87.8	110.7	118.5	118.5	128.3	128.3	128.3	128.3	1,303.7
Grand Total	2,791.0	0.0	10.4	13.8	13.8	13.8	13.8	16.8	20.2	20.2	20.2	20.2	110.3	273.7	3,388.1	131.6	143.1	169.5	222.7	235.5	258.0	263.5	272.3	272.3	272.3	272.3	2,752.7

Eolas
Cap. Expenditure Plan
1995-1996

	1995												
ITEM	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	1995
PCs	$0	$8,000	$8,000	$0	$0	$0	$0	$8,000	$0	$0	$0	$90,000	$108,000
PC Software	$0	$3,000	$3,000	$0	$0	$0	$0	$3,000	$0	$0	$0	$45,000	$54,000
Furniture	$0	$2,000	$2,000	$0	$0	$0	$0	$2,000	$0	$0	$0	$30,000	$36,000
Phones	$0	$500	$500	$0	$0	$0	$0	$500	$0	$0	$0	$7,500	$8,000
Network	$0	$0	$10,000	$0	$0	$0	$10,000	$0	$0	$0	$0	$0	$20,000
Printers	$0	$0	$1,000	$0	$0	$0	$1,000	$0	$0	$0	$0	$1,000	$3,000
Copiers	$0	$0	$3,000	$0	$0	$0	$0	$0	$0	$0	$3,000	$0	$6,000
Fax	$0	$0	$0	$0	$0	$0	$1,000	$0	$0	$0	$0	$0	$1,000
Total	$0	$11,500	$25,500	$0	$0	$0	$12,000	$11,500	$0	$0	$3,000	$173,500	$237,000
ITEM													
PCs	0	1	1	0	0	0	0	1	0	0	0	15	18
PC Software	0	1	1	0	0	0	0	1	0	0	0	15	18
Furniture	0	1	1	0	0	0	0	1	0	0	0	15	18
Phones	0	1	1	0	0	0	0	1	0	0	0	15	18
Network	0	0	1	0	0	0	1	0	0	0	0	0	2
Printers	0	0	1	0	0	0	1	0	0	0	0	1	3
Copiers	0	0	1	0	0	0	0	0	0	0	1	0	2
Fax	0	0	0	0	0	0	1	0	0	0	0	0	1
Headcount	0	1	2	2	2	2	2	3	3	3	3	18	

	1996												
ITEM	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	1996
PCs	$18,000	$6,000	$30,000	$66,000	$24,000	$30,000	$6,000	$12,000	$0	$0	$0	$0	$182,000
PC Software	$9,000	$3,000	$15,000	$33,000	$12,000	$15,000	$3,000	$6,000	$0	$0	$0	$0	$96,000
Furniture	$6,000	$2,000	$10,000	$22,000	$8,000	$10,000	$2,000	$4,000	$0	$0	$0	$0	$64,000
Phones	$1,500	$500	$2,500	$5,500	$2,000	$2,500	$500	$1,000	$0	$0	$0	$0	$16,000
Network	$0	$0	$10,000	$0	$0	$0	$0	$10,000	$0	$0	$0	$0	$20,000
Printers	$0	$0	$0	$1,000	$0	$0	$0	$0	$0	$0	$1,000	$0	$2,000
Copiers	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0
Fax	$0	$0	$0	$0	$0	$0	$0	$1,000	$0	$0	$0	$0	$1,800
Total	$34,500	$11,500	$67,500	$127,500	$46,000	$57,500	$11,500	$34,000	$0	$0	$1,000	$0	$391,000
ITEM													0
PCs	3	1	5	11	4	5	1	2	0	0	0	0	32
PC Software	3	1	5	11	4	5	1	2	0	0	0	0	32
Furniture	3	1	5	11	4	5	1	2	0	0	0	0	32
Phones	3	1	5	11	4	5	1	2	0	0	0	0	32
Network	0	0	1	0	0	0	0	1	0	0	1	0	2
Printers	0	0	0	1	0	0	0	0	0	0	0	0	2
Copiers	0	0	0	0	0	0	0	0	0	0	0	0	0
Fax	0	0	0	0	0	0	0	1	0	0	0	0	1
Headcount	21	22	27	38	42	47	48	50	50	50	50	50	

E 006337

Eólas

Cash Flow Projection

1995-1996

	1995												1996											
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Cash Sources																								
Beginning Balance	$30	$106	$66	$0	$462	$427	$390	$253	$161	$100	$440	$378	$4,678	$3,848	$2,996	$2,296	$1,348	$6,843	$6,679	$4,831	$4,342	$3,997	$3,629	$1,879
Equity Neutral				$499								$5,000	$0			$0	$7,000							
Sales	$0	$0	$0	$0	$0	$0	$0	$0	$0	$60	$0	$0	$0	$0	$341	$241	$341	$1,026	$844	$1,663	$2,031	$2,306	$2,501	$10,743
Debt Financing	$86	$0	$0	$0	$0	$0	$0	$0	$0	$330	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0
Total Cash In	$115	$106	$66	$499	$462	$427	$390	$253	$161	$490	$440	$5,378	$4,678	$3,848	$3,336	$2,626	$8,481	$7,868	$6,919	$6,364	$6,372	$6,303	$6,130	$12,618
Cash Uses																								
Cost of Revenue	$0	$0	$0	$0	$0	$0	$0	$5	$0	$0	$0	$0	$3	$30	$30	$30	$116	$85	$198	$214	$263	$282	$1,141	$1,229
Sales & Marketing	$0	$0	$0	$3	$2	$3	$2	$8	$7	$12	$12	$39	$407	$407	$407	$406	$829	$1,017	$1,198	$1,193	$1,512	$2,371	$1,881	$2,017
Engineering & Tech Supp.	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$78	$218	$218	$348	$298	$302	$332	$333	$345	$346	$346	$346	$345
Administration	$0	$14	$16	$16	$16	$19	$23	$29	$28	$28	$28	$64	$68	$74	$79	$84	$84	$84	$84	$84	$84	$84	$84	$84
Overhead	$0	$12	$15	$15	$15	$15	$14	$17	$17	$20	$20	$68	$79	$82	$84	$122	$132	$144	$147	$152	$152	$151	$151	$151
Repay Debt	$0	$0	$0	$0	$0	$0	$86	$0	$0	$0	$0	$0	$30	$30	$30	$30	$30	$30	$30	$30	$30	$30	$30	$0
Acquired Licenses	$0	$0	$0	$0	$0	$0	$0	$26	$0	$0	$0	$300	$0	$0	$0	$0	$0	$60	$126	$0	$0	$0	$0	$0
Capital Exp.	$0	$12	$28	$0	$0	$0	$12	$12	$0	$0	$3	$174	$36	$12	$88	$128	$48	$48	$12	$34	$0	$0	$1	$0
Total Cash Out	$0	$38	$59	$37	$34	$37	$137	$92	$52	$60	$62	$702	$828	$850	$954	$1,298	$1,539	$1,799	$2,099	$2,042	$2,376	$3,273	$3,734	$3,899
Ending Bal.	$106	$68	$8	$462	$427	$390	$253	$161	$109	$440	$378	$4,678	$3,848	$2,996	$2,296	$1,348	$6,843	$6,679	$4,831	$4,342	$3,997	$3,029	$1,879	$8,729
Debt Financing																								
Debt Outstanding	$86	$86	$86	$86	$86	$86	$0	$0	$0	$330	$330	$330	$300	$270	$240	$210	$180	$150	$120	$90	$60	$30	$0	$0

Notes: 1. Assumes 60 day collection cycle on sales
2. Assumes no terms on all payments

Financial Projection Spreadsheets

7.b) 1997-1998

PLEASE KEEP CONFIDENTIAL

Eòlas				
Income Statement Projection (1,000's)				
1997-1998				
	1997	% of Rev	1998	% of Rev
Revenue	471,730		1,935,692	
Expenses				
Cost of Revenue	$55,459	12%	$227,435	12%
Engineering & Tech Support	$16,166	3%	$42,646	2%
Sales & Marketing	$68,404	15%	$269,686	14%
G&A	$7,336	2%	$17,001	1%
Total Expenses	$147,365	31%	$556,767	29%
Operating Income	$324,365	69%	$1,378,925	71%
Net Interest Income (expense)	$0		$0	
Provision for taxes @40%	$129,746		$551,570	
Net Income	$194,619	41%	$827,355	43%
Net Profit Margin	41%		43%	
Total Employees	165		330	
Revenue per Employee	$2,859		$5,866	

Eolas

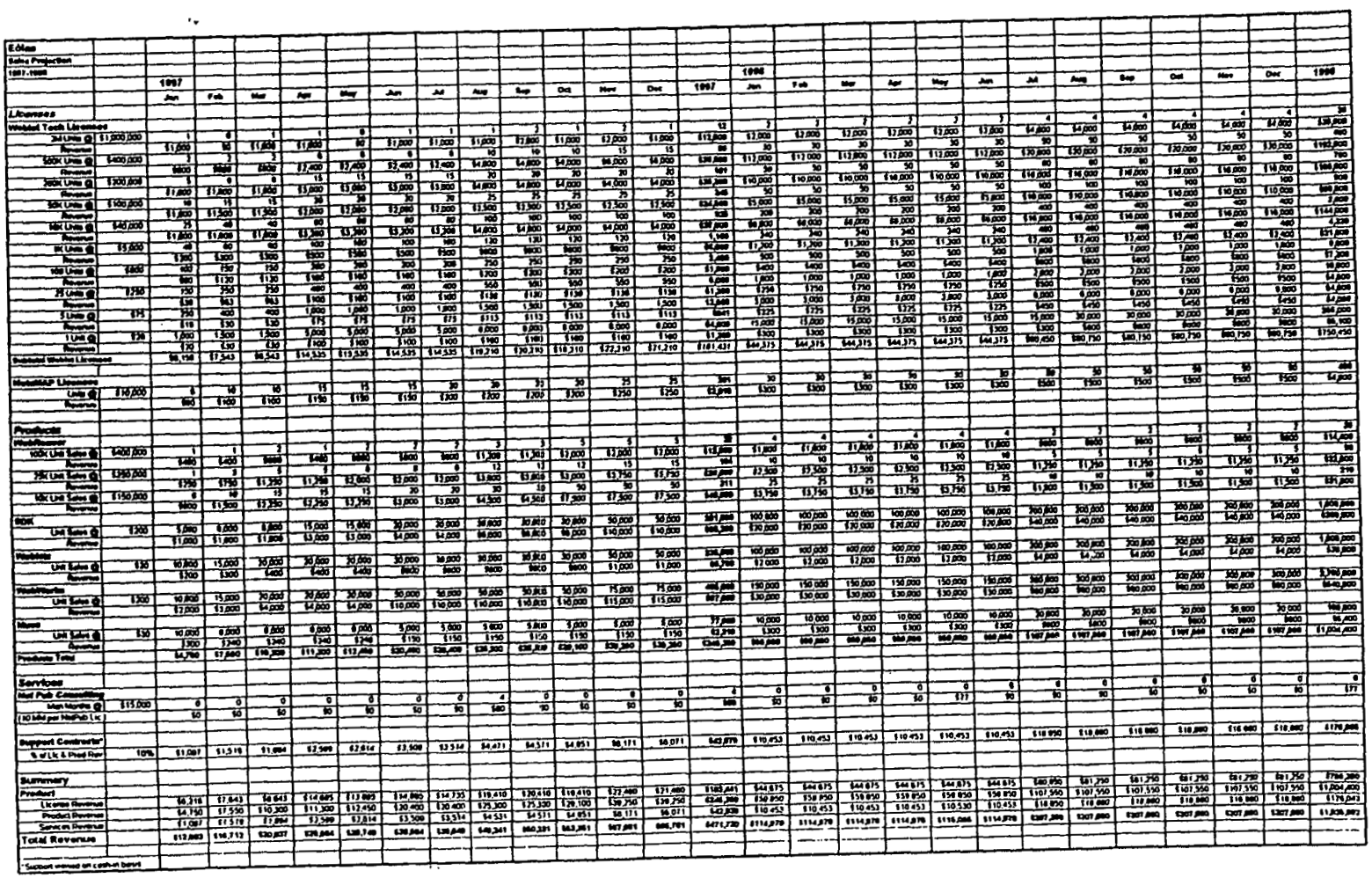
World-Wide Operating Plan (1,000's)

1997-1998

	1997 Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	1997
Revenue													
License Revenue	6,216	7,643	8,643	14,665	13,886	14,688	14,736	19,410	20,410	19,410	22,400	21,440	183,441
Product Revenue	4,790	7,560	10,300	11,360	12,400	20,400	20,400	25,300	25,300	25,100	39,250	39,250	245,350
Services Revenue	1,097	1,610	1,894	2,500	2,614	3,509	3,514	4,531	4,571	4,851	6,171	6,071	42,939
Total	12,063	16,713	20,837	28,504	28,740	38,504	38,640	49,241	50,301	53,311	67,801	66,761	471,730
Expenses													
Cost of Revenue													
COGS	475	766	1,030	1,130	1,346	2,040	2,040	2,530	2,530	2,510	3,925	3,926	14,535
Royalties Paid @4%	249	305	348	567	547	567	808	776	816	776	898	888	7,330
Sales Comm.	803	836	1,042	1,429	1,437	1,930	1,932	2,462	2,514	2,856	3,394	3,330	22,507
Total	1,527	1,906	2,410	3,147	3,330	4,567	4,562	5,790	5,960	6,154	8,217	8,132	55,450
Administration													
Salaries	136	136	136	136	136	136	203	203	203	203	203	203	2,030
TB @ 30%	41	41	41	41	41	41	61	61	61	61	61	61	608
Other @5%	9	9	9	9	9	9	13	13	13	13	13	13	132
Total	185	185	185	185	185	185	277	277	277	277	277	277	2,771
Marketing													
Salaries	97	97	97	97	97	97	146	146	146	146	146	146	1,452
TB @ 30%	29	29	29	29	29	29	44	44	44	44	44	44	436
Ads	2,000	2,000	2,000	2,000	2,000	2,000	5,000	5,000	5,000	5,500	5,000	5,000	42,000
Direct Mktg	600	600	500	500	600	600	600	750	750	750	750	750	7,250
Collateral	100	100	100	100	100	100	150	200	200	200	200	200	2,050
PR	75	75	75	75	75	75	75	100	100	100	100	100	1,025
Trade Shows	300	300	300	300	300	300	300	500	600	600	600	600	4,900
Coop Ad	50	50	50	50	50	50	50	75	75	75	75	75	725
Travel	100	100	100	100	100	100	100	150	150	150	150	150	1,450
Other @5%	166	166	166	166	166	166	310	348	348	348	348	348	2,940
Total	3,486	3,486	3,486	3,486	3,486	3,486	6,852	7,312	7,312	7,312	7,312	7,312	64,037
Sales													
Salaries	121	121	121	121	121	121	182	182	182	182	182	182	1,815
TB @ 30%	36	36	36	36	36	36	54	54	54	54	54	54	545
Travel	120	120	120	120	120	120	100	100	100	100	100	100	1,320
Other @10%	14	14	14	14	14	14	21	21	21	21	21	21	210
Total	291	291	291	291	291	291	437	437	437	437	437	437	4,367
Tech Support													
Salaries	87	87	87	87	87	87	130	130	130	130	130	130	1,301
TB @ 30%	26	26	26	26	26	26	39	39	39	39	39	39	390
Total	113	113	113	113	113	113	169	169	169	169	169	169	1,691
Engineering													
Salaries	282	282	282	282	282	282	424	424	424	424	424	424	4,235
TB @ 30%	85	85	85	85	85	85	127	127	127	127	127	127	1,271
Research	500	500	500	500	500	500	750	750	750	750	750	750	7,500
Contr. Labor	50	50	50	50	50	50	50	50	50	50	50	50	600
Travel	10	10	10	10	10	10	20	20	20	20	20	20	180
Other @5%	46	46	46	46	46	46	69	69	69	69	69	69	690
Total	973	973	973	973	973	973	1,430	1,430	1,430	1,430	1,430	1,430	14,475
Overhead													
Rent	38.5	38.5	38.5	38.5	38.5	38.5	57.8	57.8	57.8	57.8	57.8	57.8	577.6
Utilities	8.5	8.5	8.5	8.5	8.5	8.5	12.7	12.7	12.7	12.7	12.7	12.7	127.1
Insurance	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	10.0
Prof. Services	50	50	50	50	50	50	75	75	75	75	75	75	750
Office Suppl.	44.0	44.0	44.0	44.0	44.0	44.0	66.0	66.0	66.0	66.0	66.0	66.0	660.0
Depreciation	0.0	0.0	8.4	8.4	8.4	8.4	18.3	18.3	18.3	18.3	18.4	18.4	111.3
Rep. & Maint.	22.0	22.0	22.0	22.0	22.0	22.0	33.0	33.0	33.0	33.0	33.0	33.0	330.0
Phone	44.0	44.0	44.0	44.0	44.0	44.0	66.0	66.0	66.0	66.0	66.0	66.0	660.0
Postage	6.6	6.6	6.6	6.6	6.6	6.6	9.9	9.9	9.9	9.9	9.9	9.9	99.0
Internet	55.0	55.0	55.0	55.0	55.0	55.0	82.5	82.5	82.5	82.5	82.5	82.5	825.0
Interest Expense	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Other @10%	26.9	26.9	27.0	27.0	27.0	27.0	42.2	42.2	42.2	42.2	42.2	42.2	415.0
Total	296.3	296.3	296.8	296.8	296.8	296.8	464.2	464.2	464.2	464.2	464.3	464.3	4,565.1
Tot. Op. Exp.	6,651	7,229	7,742	8,471	8,554	9,867	14,830	15,867	15,959	16,453	19,316	19,321	147,365
Total Op. Inc	5,412	9,484	13,095	20,152	20,186	28,712	24,010	33,374	34,322	36,858	48,485	46,940	324,365
Head Count	110	110	110	110	110	110	165	165	165	165	165	165	
Notes:													

	1998 Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	1998
Revenue													
License Revenue	44,875	44,875	44,875	44,875	44,875	44,875	80,900	81,250	81,250	81,250	81,250	81,250	765,250
Product Revenue	59,850	59,850	59,850	59,850	59,850	59,850	107,560	107,560	107,560	107,560	107,560	107,560	1,004,400
Services Revenue	10,453	10,453	10,453	10,453	10,530	10,453	18,890	18,800	18,800	18,800	18,800	18,800	178,043
Total	114,878	114,878	114,878	114,878	115,055	114,878	207,350	207,600	207,600	207,600	207,600	207,600	1,935,097
Expenses													
Cost of Revenue													
COGS	5,986	5,986	5,986	5,986	5,986	5,986	10,756	10,756	10,756	10,756	10,756	10,756	100,440
Royalties Paid @4%	1,707	1,707	1,707	1,707	1,707	1,707	3,236	3,250	3,250	3,250	3,250	3,250	30,210
Sales Comm.	5,749	5,749	5,749	5,749	5,753	5,749	10,300	10,304	10,304	10,304	10,304	10,304	95,765
Total	13,521	13,521	13,521	13,521	13,525	13,521	24,301	24,300	24,300	24,300	24,300	24,300	227,435
Administration													
Salaries	298	298	298	298	298	298	447	447	447	447	447	447	4,468
TB @ 30%	89	89	89	89	89	89	134	134	134	134	134	134	1,340
Other @5%	19	19	19	19	19	19	29	29	29	29	29	29	290
Total	406	406	406	406	406	406	610	610	610	610	610	610	6,095
Marketing													
Salaries	213	213	213	213	213	213	319	319	319	319	319	319	3,194
TB @ 30%	64	64	64	64	64	64	96	96	96	96	96	96	958
Ads	10,000	10,000	10,000	10,000	10,000	10,000	20,000	20,000	20,000	20,000	20,000	20,000	180,000
Direct Mktg	1,500	1,500	1,500	1,500	1,500	1,500	3,000	3,000	3,000	3,000	3,000	3,000	27,000
Collateral	400	400	400	400	400	400	800	800	800	800	800	800	7,200
PR	200	200	200	200	200	200	400	400	400	400	400	400	3,600
Trade Shows	1,000	1,000	1,000	1,000	1,000	1,000	2,000	2,000	2,000	2,000	2,000	2,000	18,000
Coop Ad	150	150	150	150	150	150	300	300	300	300	300	300	2,700
Travel	300	300	300	300	300	300	600	600	600	600	600	600	5,400
Other @5%	691	691	691	691	691	691	1,376	1,376	1,376	1,376	1,376	1,376	12,403
Total	14,510	14,510	14,510	14,510	14,510	14,510	28,891	28,891	28,891	28,891	28,891	28,891	260,464
Sales													
Salaries	266	266	266	266	266	266	400	400	400	400	400	400	3,963
TB @ 30%	80	80	80	80	80	80	120	120	120	120	120	120	1,190
Travel	240	240	240	240	240	240	300	300	300	300	300	300	3,240
Other @10%	29	29	29	29	29	29	44	44	44	44	44	44	440
Total	615	615	615	615	615	615	923	923	923	923	923	923	9,230
Tech Support													
Salaries	191	191	191	191	191	191	286	286	286	286	286	286	2,862
TB @ 30%	57	57	57	57	57	57	86	86	86	86	86	86	858
Total	248	248	248	248	248	248	372	372	372	372	372	372	3,720
Engineering													
Salaries	621	621	621	621	621	621	932	932	932	932	932	932	9,317
TB @ 30%	186	186	186	186	186	186	280	280	280	280	280	280	2,795
Research	1,000	1,000	1,000	1,000	1,000	1,000	3,000	3,000	3,000	3,000	3,000	3,000	24,000
Contr. Labor	10	10	10	10	10	10	10	10	10	10	10	10	120
Travel	40	40	40	40	40	40	100	100	100	100	100	100	840
Other @5%	93	93	93	93	93	93	216	216	216	216	216	216	1,854
Total	1,950	1,950	1,950	1,950	1,950	1,950	4,537	4,537	4,537	4,537	4,537	4,537	38,926
Overhead													
Rent	77.0	77.0	77.0	77.0	77.0	77.0	115.5	115.5	115.5	115.5	115.5	115.5	1,155.0
Utilities	16.9	16.9	16.9	16.9	16.9	16.9	25.4	25.4	25.4	25.4	25.4	25.4	254.1
Insurance	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	0.8	10.0
Prof. Services	150	150	150	150	150	150	300	300	300	300	300	300	2,700
Office Suppl.	88.0	88.0	88.0	88.0	88.0	88.0	132.0	132.0	132.0	132.0	132.0	132.0	1,320.0
Depreciation	36.0	36.0	36.3	36.3	36.3	36.3	71.4	71.7	71.7	71.7	71.8	71.8	647.1
Rep. & Maint.	44.0	44.0	44.0	44.0	44.0	44.0	66.0	66.0	66.0	66.0	66.0	66.0	660.0
Phone	88.0	88.0	88.0	88.0	88.0	88.0	132.0	132.0	132.0	132.0	132.0	132.0	1,320.0
Postage	13.2	13.2	13.2	13.2	13.2	13.2	19.8	19.8	19.8	19.8	19.8	19.8	198.0
Internet	110.0	110.0	110.0	110.0	110.0	110.0	165.0	165.0	165.0	165.0	165.0	165.0	1,650.0
Interest Expense	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Other @10%	62.4	62.4	62.4	62.4	62.4	62.4	102.8	102.8	102.8	102.8	102.8	102.8	991.4
Total	686.3	686.3	686.6	686.7	686.7	686.7	1,130.0	1,131.1	1,131.1	1,131.1	1,131.1	1,131.1	10,905.8
Tot. Op. Exp.	31,845	31,845	31,846	31,846	31,850	31,846	60,824	60,853	60,853	60,853	60,853	60,853	556,767
Total Op. Inc	83,032	83,032	83,032	83,032	83,105	83,032	146,526	146,827	146,827	146,827	146,827	146,827	1,378,335
Head Count	220	220	220	220	220	220	330	330	330	330	330	330	
Notes:													

E 006341



Confidential - Subject to Protective Order
Eolas v. Microsoft CV99-C0626

E 006342

Eolas																												
Payroll Plan																												
1997-1998																												
		1997														1998												
		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	1997		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	1998
# Employees (*4	Ann.Sal.	110	110	110	110	110	110	165	165	165	165	165	165		Ann.Sal.	220	220	220	220	220	220	330	330	330	330	330	330	
Administration																												
President	151.25	25.2	25.2	25.2	25.2	25.2	25.2	37.8	37.8	37.8	37.8	37.8	37.8	378.1	166.4	55.5	55.5	55.5	55.5	55.5	55.5	83.2	83.2	83.2	83.2	83.2	83.2	831.0
CTO	151.25	25.2	25.2	25.2	25.2	25.2	25.2	37.8	37.8	37.8	37.8	37.8	37.8	378.1	166.4	55.5	55.5	55.5	55.5	55.5	55.5	83.2	83.2	83.2	83.2	83.2	83.2	831.0
Executive VP	121	20.2	20.2	20.2	20.2	20.2	20.2	30.3	30.3	30.3	30.3	30.3	30.3	302.5	133.1	44.4	44.4	44.4	44.4	44.4	44.4	66.6	66.6	66.6	66.6	66.6	66.6	665.5
VP Admin & Legal	84.7	14.1	14.1	14.1	14.1	14.1	14.1	21.2	21.2	21.2	21.2	21.2	21.2	211.8	93.2	31.1	31.1	31.1	31.1	31.1	31.1	46.6	46.6	46.6	46.6	46.6	46.6	465.9
Admin	48.4	8.1	8.1	8.1	8.1	8.1	8.1	12.1	12.1	12.1	12.1	12.1	12.1	121.0	53.2	17.7	17.7	17.7	17.7	17.7	17.7	26.6	26.6	26.6	26.6	26.6	26.6	266.2
Admin	48.4	8.1	8.1	8.1	8.1	8.1	8.1	12.1	12.1	12.1	12.1	12.1	12.1	121.0	53.2	17.7	17.7	17.7	17.7	17.7	17.7	26.6	26.6	26.6	26.6	26.6	26.6	266.2
Admin	48.4	8.1	8.1	8.1	8.1	8.1	8.1	12.1	12.1	12.1	12.1	12.1	12.1	121.0	53.2	17.7	17.7	17.7	17.7	17.7	17.7	26.6	26.6	26.6	26.6	26.6	26.6	266.2
Admin	48.4	8.1	8.1	8.1	8.1	8.1	8.1	12.1	12.1	12.1	12.1	12.1	12.1	121.0	53.2	17.7	17.7	17.7	17.7	17.7	17.7	26.6	26.6	26.6	26.6	26.6	26.6	266.2
CFO	110.11	18.4	18.4	18.4	18.4	18.4	18.4	27.5	27.5	27.5	27.5	27.5	27.5	275.3	121.1	40.4	40.4	40.4	40.4	40.4	40.4	60.6	60.6	60.6	60.6	60.6	60.6	605.6
Total	811.91	135.3	135.3	135.3	135.3	135.3	135.3	203.0	203.0	203.0	203.0	203.0	203.0	2,028.8	893.1	297.7	297.7	297.7	297.7	297.7	297.7	446.6	446.6	446.6	446.6	446.6	446.6	4,465.5
Marketing																												
VP Marketing	121	20.2	20.2	20.2	20.2	20.2	20.2	30.3	30.3	30.3	30.3	30.3	30.3	302.5	133.1	44.4	44.4	44.4	44.4	44.4	44.4	66.6	66.6	66.6	66.6	66.6	66.6	665.5
Product Mgr	96.8	16.1	16.1	16.1	16.1	16.1	16.1	24.2	24.2	24.2	24.2	24.2	24.2	242.0	106.5	35.5	35.5	35.5	35.5	35.5	35.5	53.2	53.2	53.2	53.2	53.2	53.2	532.4
Product Mgr	96.8	16.1	16.1	16.1	16.1	16.1	16.1	24.2	24.2	24.2	24.2	24.2	24.2	242.0	106.5	35.5	35.5	35.5	35.5	35.5	35.5	53.2	53.2	53.2	53.2	53.2	53.2	532.4
Marketing Mgr	72.6	12.1	12.1	12.1	12.1	12.1	12.1	18.2	18.2	18.2	18.2	18.2	18.2	181.5	79.9	26.6	26.6	26.6	26.6	26.6	26.6	39.9	39.9	39.9	39.9	39.9	39.9	399.3
Marketing Mgr	72.6	12.1	12.1	12.1	12.1	12.1	12.1	18.2	18.2	18.2	18.2	18.2	18.2	181.5	79.9	26.6	26.6	26.6	26.6	26.6	26.6	39.9	39.9	39.9	39.9	39.9	39.9	399.3
Marketing Mgr	72.6	12.1	12.1	12.1	12.1	12.1	12.1	18.2	18.2	18.2	18.2	18.2	18.2	181.5	79.9	26.6	26.6	26.6	26.6	26.6	26.6	39.9	39.9	39.9	39.9	39.9	39.9	
Admin	48.4	8.1	8.1	8.1	8.1	8.1	8.1	12.1	12.1	12.1	12.1	12.1	12.1	121.0	53.2	17.7	17.7	17.7	17.7	17.7	17.7	26.6	26.6	26.6	26.6	26.6	26.6	266.2
Total	580.8	96.8	96.8	96.8	96.8	96.8	96.8	145.2	145.2	145.2	145.2	145.2	145.2	1,452.0	638.9	213.0	213.0	213.0	213.0	213.0	213.0	319.4	319.4	319.4	319.4	319.4	319.4	2,795.1
Sales																												
Sales	72.6	12.1	12.1	12.1	12.1	12.1	12.1	18.2	18.2	18.2	18.2	18.2	18.2	181.5	79.9	26.6	26.6	26.6	26.6	26.6	26.6	39.9	39.9	39.9	39.9	39.9	39.9	399.3
Sales	72.6	12.1	12.1	12.1	12.1	12.1	12.1	18.2	18.2	18.2	18.2	18.2	18.2	181.5	79.9	26.6	26.6	26.6	26.6	26.6	26.6	39.9	39.9	39.9	39.9	39.9	39.9	399.3
Sales	72.6	12.1	12.1	12.1	12.1	12.1	12.1	18.2	18.2	18.2	18.2	18.2	18.2	181.5	79.9	26.6	26.6	26.6	26.6	26.6	26.6	39.9	39.9	39.9	39.9	39.9	39.9	399.3
Sales	72.6	12.1	12.1	12.1	12.1	12.1	12.1	18.2	18.2	18.2	18.2	18.2	18.2	181.5	79.9	26.6	26.6	26.6	26.6	26.6	26.6	39.9	39.9	39.9	39.9	39.9	39.9	399.3
Sales	72.6	12.1	12.1	12.1	12.1	12.1	12.1	18.2	18.2	18.2	18.2	18.2	18.2	181.5	79.9	26.6	26.6	26.6	26.6	26.6	26.6	39.9	39.9	39.9	39.9	39.9	39.9	399.3
Sales	72.6	12.1	12.1	12.1	12.1	12.1	12.1	18.2	18.2	18.2	18.2	18.2	18.2	181.5	79.9	26.6	26.6	26.6	26.6	26.6	26.6	39.9	39.9	39.9	39.9	39.9	39.9	399.3
Sales	72.6	12.1	12.1	12.1	12.1	12.1	12.1	18.2	18.2	18.2	18.2	18.2	18.2	181.5	79.9	26.6	26.6	26.6	26.6	26.6	26.6	39.9	39.9	39.9	39.9	39.9	39.9	399.3
Sales	72.6	12.1	12.1	12.1	12.1	12.1	12.1	18.2	18.2	18.2	18.2	18.2	18.2	181.5	79.9	26.6	26.6	26.6	26.6	26.6	26.6	39.9	39.9	39.9	39.9	39.9	39.9	399.3
Sales	72.6	12.1	12.1	12.1	12.1	12.1	12.1	18.2	18.2	18.2	18.2	18.2	18.2	181.5	79.9	26.6	26.6	26.6	26.6	26.6	26.6	39.9	39.9	39.9	39.9	39.9	39.9	399.3
Sales	72.6	12.1	12.1	12.1	12.1	12.1	12.1	18.2	18.2	18.2	18.2	18.2	18.2	181.5	79.9	26.6	26.6	26.6	26.6	26.6	26.6	39.9	39.9	39.9	39.9	39.9	39.9	399.3
Total	726	121.0	121.0	121.0	121.0	121.0	121.0	181.5	181.5	181.5	181.5	181.5	181.5	1,815.0	798.6	266.2	266.2	266.2	266.2	266.2	266.2	399.3	399.3	399.3	399.3	399.3	399.3	3,993.0
Tech Support																												
Tech Support Mgr	96.8	16.1	16.1	16.1	16.1	16.1	16.1	24.2	24.2	24.2	24.2	24.2	24.2	242.0	106.5	35.5	35.5	35.5	35.5	35.5	35.5	53.2	53.2	53.2	53.2	53.2	53.2	532.4
Support Eng	60.5	10.1	10.1	10.1	10.1	10.1	10.1	15.1	15.1	15.1	15.1	15.1	15.1	151.3	66.6	22.2	22.2	22.2	22.2	22.2	22.2	33.3	33.3	33.3	33.3	33.3	33.3	332.8
Support Eng	60.5	10.1	10.1	10.1	10.1	10.1	10.1	15.1	15.1	15.1	15.1	15.1	15.1	151.3	66.6	22.2	22.2	22.2	22.2	22.2	22.2	33.3	33.3	33.3	33.3	33.3	33.3	332.8
Support Eng	60.5	10.1	10.1	10.1	10.1	10.1	10.1	15.1	15.1	15.1	15.1	15.1	15.1	151.3	66.6	22.2	22.2	22.2	22.2	22.2	22.2	33.3	33.3	33.3	33.3	33.3	33.3	332.8
Support Eng	60.5	10.1	10.1	10.1	10.1	10.1	10.1	15.1	15.1	15.1	15.1	15.1	15.1	151.3	66.6	22.2	22.2	22.2	22.2	22.2	22.2	33.3	33.3	33.3	33.3	33.3	33.3	332.8
Support Eng	60.5	10.1	10.1	10.1	10.1	10.1	10.1	15.1	15.1	15.1	15.1	15.1	15.1	151.3	66.6	22.2	22.2	22.2	22.2	22.2	22.2	33.3	33.3	33.3	33.3	33.3	33.3	332.8
Support Eng	60.5	10.1	10.1	10.1	10.1	10.1	10.1	15.1	15.1	15.1	15.1	15.1	15.1	151.3	66.6	22.2	22.2	22.2	22.2	22.2	22.2	33.3	33.3	33.3	33.3	33.3	33.3	332.8
Support Eng	60.5	10.1	10.1	10.1	10.1	10.1	10.1	15.1	15.1	15.1	15.1	15.1	15.1	151.3	66.6	22.2	22.2	22.2	22.2	22.2	22.2	33.3	33.3	33.3	33.3	33.3	33.3	332.8
Total	423.5	86.7	86.7	86.7	86.7	86.7	86.7	130.1	130.1	130.1	130.1	130.1	130.1	1,300.8	572.7	190.8	190.8	190.8	190.8	190.8	190.8	286.2	286.2	286.2	286.2	286.2	286.2	2,861.7
Engineering																												
Chief Engineer	121	20.2	20.2	20.2	20.2	20.2	20.2	30.3	30.3	30.3	30.3	30.3	30.3	302.5	133.1	44.4	44.4	44.4	44.4	44.4	44.4	66.6	66.6	66.6	66.6	66.6	66.6	665.5
Senior Scientist	121	20.2	20.2	20.2	20.2	20.2	20.2	30.3	30.3	30.3	30.3	30.3	30.3	302.5	133.1	44.4	44.4	44.4	44.4	44.4	44.4	66.6	66.6	66.6	66.6	66.6	66.6	665.5
Sr. Designer	84.7	14.1	14.1	14.1	14.1	14.1	14.1	21.2	21.2	21.2	21.2	21.2	21.2	211.8	93.2	31.1	31.1	31.1	31.1	31.1	31.1	46.6	46.6	46.6	46.6	46.6	46.6	465.9
Win Eng	84.7	14.1	14.1	14.1	14.1	14.1	14.1	21.2	21.2	21.2	21.2	21.2	21.2	211.8	93.2	31.1	31.1	31.1	31.1	31.1	31.1	46.6	46.6	46.6	46.6	46.6	46.6	465.9
Win Eng	84.7	14.1	14.1	14.1	14.1	14.1	14.1	21.2	21.2	21.2	21.2	21.2	21.2	211.8	93.2	31.1	31.1	31.1	31.1	31.1	31.1	46.6	46.6	46.6	46.6	46.6	46.6	465.9
Win Eng	84.7	14.1	14.1	14.1	14.1	14.1	14.1	21.2	21.2	21.2	21.2	21.2	21.2	211.8	93.2	31.1	31.1	31.1	31.1	31.1	31.1	46.6	46.6	46.6	46.6	46.6	46.6	465.9
Win Eng	84.7	14.1	14.1	14.1	14.1	14.1	14.1	21.2	21.2	21.2	21.2	21.2	21.2	211.8	93.2	31.1	31.1	31.1	31.1	31.1	31.1	46.6	46.6	46.6	46.6	46.6	46.6	465.9
Win Eng	84.7	14.1	14.1	14.1	14.1	14.1	14.1	21.2	21.2	21.2	21.2	21.2	21.2	211.8	93.2	31.1	31.1	31.1	31.1	31.1	31.1	46.6	46.6	46.6	46.6	46.6	46.6	465.9
Unix Eng	84.7	14.1	14.1	14.1	14.1	14.1	14.1	21.2	21.2	21.2	21.2	21.2	21.2	211.8	93.2	31.1	31.1	31.1	31.1	31.1	31.1	46.6	46.6	46.6	46.6	46.6	46.6	465.9
Unix Eng	84.7	14.1	14.1	14.1	14.1	14.1	14.1	21.2	21.2	21.2	21.2	21.2	21.2	211.8	93.2	31.1	31.1	31.1	31.1	31.1	31.1	46.6	46.6	46.6	46.6	46.6	46.6	465.9
Unix Eng	84.7	14.1	14.1	14.1	14.1	14.1	14.1	21.2	21.2	21.2	21.2	21.2	21.2	211.8	93.2	31.1	31.1	31.1	31.1	31.1	31.1	46.6	46.6	46.6	46.6	46.6	46.6	465.9
Unix Eng	84.7	14.1	14.1	14.1	14.1	14.1	14.1	21.2	21.2	21.2	21.2	21.2	21.2	211.8	93.2	31.1	31.1	31.1	31.1	31.1	31.1	46.6	46.6	46.6	46.6	46.6	46.6	465.9
Mac Eng	84.7	14.1	14.1	14.1	14.1	14.1	14.1	21.2	21.2	21.2	21.2	21.2	21.2	211.8	93.2	31.1	31.1	31.1	31.1	31.1	31.1	46.6	46.6	46.6	46.6	46.6	46.6	465.9
Mac Eng	84.7	14.1	14.1	14.1	14.1	14.1	14.1	21.2	21.2	21.2	21.2	21.2	21.2	211.8	93.2	31.1	31.1	31.1	31.1	31.1	31.1	46.6	46.6	46.6	46.6	46.6	46.6	465.9
Mac Eng	84.7	14.1	14.1	14.1	14.1	14.1	14.1	21.2	21.2	21.2	21.2	21.2	21.2	211.8	93.2	31.1	31.1	31.1	31.1	31.1	31.1	46.6	46.6	46.6	46.6	46.6	46.6	465.9
Sys Admin	60.5	10.1	10.1	10.1	10.1	10.1	10.1	15.1	15.1	15.1	15.1	15.1	15.1	151.3	66.6	22.2	22.2	22.2	22.2	22.2	22.2	33.3	33.3	33.3	33.3	33.3	33.3	332.8
QA Engineer	58.00	9.7	9.7	9.7	9.7	9.7	9.7	14.5	14.5	14.5	14.5	14.5	14.5	145.2	63.9	21.3	21.3	21.3	21.3	21.3	21.3	31.9	31.9	31.9	31.9	31.9	31.9	319.4
QA Engineer	58.00	9.7	9.7	9.7	9.7	9.7	9.7	14.5	14.5	14.5	14.5	14.5	14.5	145.2	63.9	21.3	21.3	21.3	21.3	21.3	21.3	31.9	31.9	31.9	31.9	31.9	31.9	319.4
QA Engineer	58.00	9.7	9.7	9.7	9.7	9.7	9.7	14.5	14.5	14.5	14.5	14.5	14.5	145.2	63.9	21.3	21.3	21.3	21.3	21.3	21.3	31.9	31.9	31.9	31.9	31.9	31.9	319.4
QA Engineer	58.00	9.7	9.7	9.7	9.7	9.7	9.7	14.5	14.5	14.5	14.5	14.5	14.5	145.2	63.9	21.3	21.3	21.3	21.3	21.3	21.3	31.9	31.9	31.9	31.9	31.9	31.9	319.4
QA Engineer	58.00	9.7	9.7	9.7	9.7	9.7	9.7	14.5	14.5	14.5	14.5	14.5	14.5	145.2	63.9	21.3	21.3	21.3	21.3	21.3	21.3	31.9	31.9	31.9	31.9	31.9	31.9	319.4
Total	1573	282.3	282.3	282.3	282.3	282.3	282.3	423.5	423.5	423.5	423.5	423.5	423.5	4,235.0	1,863.4	621.1	621.1	621.1	621.1	621.1	621.1	931.7	931.7	931.7	931.7	931.7	931.7	9,317.0
Grand Total	3,388.2	601.2	601.2	601.2	601.2	601.2	601.2	901.8	901.8	901.8	901.8	901.8	901.8	9,017.5	3,947.7	1,322.6	1,322.6	1,322.6	1,322.6	1,322.6	1,322.6	1,983.9	1,983.9	1,983.9	1,983.9	1,983.9	1,983.9	19,430.3

Eólas

Cap. Expenditure Plan

1997-1998

ITEM	1997 Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	1997	1998 Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	1998
PCs	$0	$0	$0	$0	$0	$0	$330,000	$0	$0	$0	$0	$0	$330,000	$330,000	$0	$0	$0	$0	$0	$660,000	$0	$0	$0	$0	$0	$990,000
PC Software	$0	$0	$0	$0	$0	$0	$165,000	$0	$0	$0	$0	$0	$165,000	$165,000	$0	$0	$0	$0	$0	$330,000	$0	$0	$0	$0	$0	$495,000
Furniture	$0	$0	$0	$0	$0	$0	$110,000	$0	$0	$0	$0	$0	$110,000	$110,000	$0	$0	$0	$0	$0	$220,000	$0	$0	$0	$0	$0	$330,000
Phones	$0	$0	$0	$0	$0	$0	$27,500	$0	$0	$0	$0	$0	$27,500	$27,500	$0	$0	$0	$0	$0	$55,000	$0	$0	$0	$0	$0	$82,500
Network	$0	$0	$10,000	$0	$0	$0	$10,000	$0	$0	$0	$0	$0	$20,000	$0	$0	$10,000	$0	$0	$0	$0	$10,000	$0	$0	$0	$0	$20,000
Printers	$0	$0	$1,000	$0	$0	$0	$1,000	$0	$0	$0	$0	$1,000	$3,000	$0	$0	$0	$1,000	$0	$0	$0	$0	$0	$0	$1,000	$0	$2,000
Copiers	$0	$0	$3,000	$0	$0	$0	$0	$0	$0	$0	$0	$3,000	$6,000	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0
Fax	$0	$0	$0	$0	$0	$0	$1,000	$0	$0	$0	$0	$0	$1,000	$1,000	$0	$0	$0	$0	$0	$0	$1,000	$0	$0	$0	$0	$1,000
Total	$0	$0	$14,000	$0	$0	$0	$644,500	$0	$0	$0	$3,000	$1,000	$662,500	$632,500	$0	$10,000	$1,000	$0	$0	$1,285,000	$11,000	$0	$0	$1,000	$0	$1,820,500
																										0
ITEM																										
PCs	0	0	0	0	0	0	55	0	0	0	0	0	55	55	0	0	0	0	0	110	0	0	0	0	0	165
PC Software	0	0	0	0	0	0	55	0	0	0	0	0	55	55	0	0	0	0	0	110	0	0	0	0	0	165
Furniture	0	0	0	0	0	0	55	0	0	0	0	0	55	55	0	0	0	0	0	110	0	0	0	0	0	165
Phones	0	0	0	0	0	0	55	0	0	0	0	0	55	55	0	0	0	0	0	110	0	0	0	0	0	165
Network	0	0	1	0	0	0	1	0	0	0	0	0	2	0	0	1	0	0	0	0	1	0	0	0	0	2
Printers	0	0	1	0	0	0	1	0	0	0	0	1	3	0	0	0	1	0	0	0	0	0	0	1	0	2
Copiers	0	0	1	0	0	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
Fax	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1
Headcount	110	110	110	110	110	110	185	165	165	165	165	165		220	220	220	220	220	220	330	330	330	330	330	330	

Confidential - Subject to
Protective Order
Eolas v. Microsoft CV99-C0626

E 006344

Eolas																								
Cash Flow Projection																								
1997-1998																								
	1997												1998											
	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Cash Sources																								
Beginning Balance	$30	$4,830	$12,404	$16,730	$23,571	$34,415	$51,106	$63,321	$83,564	$102,779	$133,000	$162,440	$194,000	$101,825	$216,711	$287,004	$305,200	$442,546	$516,034	$573,201	$617,010	$753,132	$900,375	$1,026,017
Equity Raised																								
Sales	$12,863	$16,712	$12,863	$16,713	[illegible]	[illegible]	$28,749	[illegible]	[illegible]	$49,341	$80,201	$63,361	[illegible]	[illegible]	$114,870	$114,878	$114,978	$114,870	[illegible]	$114,870	[illegible]	[illegible]	[illegible]	[illegible]
Debt Financing	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0
Total Cash In	$12,863	[illegible]	[illegible]	$33,471	[illegible]	[illegible]	$79,836	$101,823	$122,343	$153,811	$194,171	$215,820	[illegible]	[illegible]	[illegible]	$442,362	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	$1,087,256	$1,233,817
Cash Uses																								
Cost of Revenue	$1,327	[illegible]	$2,416	$3,147	$3,320	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	$8,317	$8,122	[illegible]	$13,821	$13,821	$13,821	$13,821	$13,826	$13,821	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Sales & Marketing	[illegible]	[illegible]	$4,796	[illegible]	[illegible]	[illegible]	[illegible]	$10,211	$10,263	$10,417	$11,143	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	$40,162	$40,190	$40,190	$40,190	$40,190	$40,190
Engineering & Tech Supp.	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	$1,808	[illegible]	[illegible]	$1,808	$1,808	$2,190	$2,190	$2,190	$2,190	$2,190	$2,190	[illegible]	[illegible]	[illegible]	$4,309	[illegible]	$4,300
Administration	$106	$106	$106	$106	$106	$106	$277	$277	$277	$277	$277	$277	$406	$406	$406	$406	$406	$406	$610	$610	$610	$610	$610	$610
Overhead	$296	$296	$297	$297	$297	$297	$464	$464	$464	$464	$464	$464	[illegible]	[illegible]	$667	$667	$667	$667	$1,131	$1,131	$1,131	$1,131	$1,131	$1,131
Repay Debt	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0
Acquired Licenses	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$200	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$1,000
Capital Exp.	$0	$0	$14	$0	$0	$0	$645	$0	$0	$0	$3	$1	$633	$0	$10	$1	$0	$0	$1,206	$11	$0	$0	$1	$0
Total Cash Out	$7,264	[illegible]	[illegible]	[illegible]	[illegible]	$11,011	$16,807	$18,229	$19,473	$19,131	$21,713	$21,811	[illegible]	$37,694	$37,796	$37,698	$37,699	$37,699	$61,617	$71,216	$71,237	$71,237	$71,230	$72,837
Ending Bal.	$4,830	$12,404	$16,730	$23,571	$34,415	$51,106	$63,329	$83,594	$102,779	$133,000	$162,440	$194,000	$101,825	$216,711	$287,004	$305,200	$442,546	$516,034	$573,201	$617,010	$753,132	$900,375	$1,026,017	$1,190,000
Debt Financing																								
Debt Outstanding	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0	$0
Notes:	1. Assumes 60 day collection cycle on sales																							
	2. Assumes no loans on all payments																							

This sheet contains the expense assumptions that are used in the Operating Plan.

	Cost per Person/Month		
Rent	$350		
Office Suppl.	$400		
Depreciation		months	
Rep. & Maint.	$200		
Phone	$400		
Postage	$60		
Internet	$500		
Insurance	$10,000	per year	
COGS	10%	of Revenue	
Commission	5%	of Revenue	
ITEM PRICE			
PCs	$6,000		
PC Software	$3,000		
Furniture	$2,000		
Phones	$500		
Network	$10,000		
Printers	$1,000		
Copiers	$3,000		
Fax	$1,000		