FILE HISTORY

US 5,995,102

PATENT:	5,995,102
INVENTORS:	Rosen, James Samuel Schmitter, Thomas A. Hall, Mark S.
TITLE:	Server system and method for modifying a cursor image
APPLICATION NO:	US1997882580A
FILED:	25 JUN 1997
ISSUED:	30 NOV 1999
COMPILED:	19 MAY 2017

AMAZON.COM, INC.
EXHIBIT 1010
US 6,118,449

BEST COPY

05/25/97

Class	Subclass
345	332

ISSUE CLASSIFICATION

SCANNED

[redacted]

5995102

UTILITY SERIAL NUMBER	PATENT DATE	PATENT NUMBER
	NOV 30 1999	5995102

SERIAL NUMBER / FILING DATE	CLASS	SUBCLASS	GROUP ART UNIT	EXAMINER
	345	~~326~~ 339	2773	C. [illegible]

APPLICANTS

[illegible] None

[illegible] None

CPA

[illegible] ***** SMALL ENTITY *****

Foreign priority claimed 35 USC 119 conditions met / Verified and Acknowledged	AS FILED	STATE OR COUNTRY	SHEETS DRWGS.	TOTAL CLAIMS	INDEP. CLAIMS	FILING FEE RECEIVED	ATTORNEY'S DOCKET NO.
☐ yes ☒ no / ☐ yes ☐ no / Examiner's Initials	→	NY	9	75	5	$1,000.00	[illegible]

ADDRESS

[illegible] Wilmer, Cutler, Pickering, Hale and Dorr LLP
[illegible] 1 Commonwealth Square
Boston Massachusetts 02109

TITLE

[illegible]

U.S. DEPT. OF COMM./ PAT. & TM—PTO-436L (Rev.12-94)

PARTS OF APPLICATION FILED SEPARATELY

Applications Examiner

NOTICE OF ALLOWANCE MAILED		CLAIMS ALLOWED	
6-21-99	CHADWICK A. JACKSON / C.J. Assistant Examiner	Total Claims ~~77~~ 75	Print Claim 1

ISSUE FEE		DRAWING		
Amount Due	Date Paid	Sheets Drwg.	Figs. Drwg.	Print Fig.
$1,605.00	9-23-99	9	9	2

RAYMOND J. BAYERL
PRIMARY EXAMINER
ART UNIT 2773
Primary Examiner

ISSUE BATCH NUMBER 473

PREPARED FOR ISSUE 6-28-99

Label Area

WARNING: The information disclosed herein may be restricted. Unauthorized disclosure may be prohibited by the United States Code Title 35, Sections 122, 181 and 368. Possession outside the U.S. Patent & Trademark Office is restricted to authorized employees and contractors only.

Form PTO-436A (Rev 8/92)

CG

Formal Drawings (___ shts) set___

ISSUE FEE IN FILE

5,995,102

SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE

Transaction History

Date	Transaction Description
07-31-1997	Initial Exam Team nn
10-07-1997	IFW Scan & PACR Auto Security Review
11-12-1997	Notice Mailed--Application Incomplete--Filing Date Assigned
01-20-1998	Information Disclosure Statement (IDS) Filed
01-20-1998	Information Disclosure Statement (IDS) Filed
03-06-1998	Application Is Now Complete
03-09-1998	Application Dispatched from OIPE
04-30-1998	Case Docketed to Examiner in GAU
07-06-1998	Non-Final Rejection
07-13-1998	Mail Non-Final Rejection
07-27-1998	Change in Power of Attorney (May Include Associate POA)
12-17-1998	Change in Power of Attorney (May Include Associate POA)
12-17-1998	Response after Non-Final Action
12-17-1998	Request for Extension of Time - Granted
12-23-1998	Date Forwarded to Examiner
12-29-1998	Communication - Re: Power of Attorney (PTOL-308)
02-01-1999	Final Rejection
02-08-1999	Mail Final Rejection (PTOL - 326)
04-12-1999	Response after Final Action
04-15-1999	Date Forwarded to Examiner
04-22-1999	Advisory Action (PTOL-303)
04-23-1999	Mail Advisory Action (PTOL - 303)
05-11-1999	Examiner Interview Summary Record (PTOL - 413)
06-08-1999	Preliminary Amendment
06-08-1999	Continuing Prosecution Application - Continuation (ACPA)
06-08-1999	Mail Express Abandonment (During Examination)
06-08-1999	Express Abandonment (during Examination)
06-08-1999	Request for Extension of Time - Granted
06-17-1999	Miscellaneous Incoming Letter
06-18-1999	Date Forwarded to Examiner
06-21-1999	Mail Notice of Allowance
06-21-1999	Notice of Allowance Data Verification Completed
07-22-1999	Amendment after Notice of Allowance (Rule 312)
08-09-1999	Date Forwarded to Examiner
08-25-1999	Mail Response to 312 Amendment (PTO-271)
08-25-1999	Response to Amendment under Rule 312
09-20-1999	Workflow - Drawings Finished
09-20-1999	Workflow - Drawings Matched with File at Contractor
09-20-1999	Workflow - Drawings Received at Contractor
09-20-1999	Workflow - Drawings Sent to Contractor
09-23-1999	Workflow - File Sent to Contractor
09-23-1999	Issue Fee Payment Verified
11-08-1999	Workflow - Complete WF Records for Drawings
11-10-1999	Application Is Considered Ready for Issue
11-19-1999	Issue Notification Mailed
12-03-1999	Recordation of Patent Grant Mailed
01-23-2004	Information Disclosure Statement (IDS) Filed
01-23-2004	Information Disclosure Statement (IDS) Filed
12-08-2015	File Marked Found
01-14-2016	File Marked Found
01-29-2016	File Marked Found

PATENT APPLICATION

08882580

APPROVED FOR LICENSE ☐

INITIALS SEP 05 2001

CONTENTS

Date Entered or Counted			Date Received or Mailed
	1.	Application 9 papers.	
	2.	Ltr Re Dec.	11-12-97
	3.	Fee Surchg Dec & ~~Petty~~ Small	1-13-98
5-6-98	4.	IDS	1-30-98
7-6-98	5.	Rej. 3mos.	7-13-98
	6.	Associate P/A	7-27-98
	7.	Ext. (2)	12-17-98
	8.	Amdt A	12-17-98
	9.	Rev. P/Atty	Dec 17 1998
	10.	Notice of Acceptance	12/29/98
2-1-99	11.	Final Rej. 3mos.	2-8-99
4-15-99	12.	Amdt B (N.E.)	4-12-99
4-22-99	13.	Advisory Action	4-23-99 cw
5-12-99	14.	Interview Summary	5-11-99
6-16-99	15.	CPA Request + E.O.T 1 month	6-8-99
6-16-99	16.	Pre Amdt C	6-8-99
6-18-99	17.	Letter	6-17-99
6-21-99 N.Hilmond	18.	Notice of Allowance	6-21-99 cw
8-9-99	19.	Amdt D (1.312)	7-22-99
8-25-99 N.Hilmond	20.	Notice of Entry	8-25-99
11/8/99	21.	Formal Drawings 9 sht(s) acc 1	9/20/99
2-3-04	22.	IDS	1-23-04
	23.		
	24.		
	25.		
	26.		
	27.		
	28.		
	29.		
	30.		
	31.		
	32.		

(FRONT)

STAPLE AREA ☆U.S. GOVERNMENT PRINTING OFFICE: 1998-440-769

PATENT NUMBER

08/882,580

APPLICATION SERIAL NUMBER

Rosen et al

APPLICANT'S NAME (PLEASE PRINT)

IF REISSUE, ORIGINAL PATENT NUMBER

ORIGINAL CLASSIFICATION

CLASS	SUBCLASS
345	339

CROSS REFERENCE(S)

CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)			
345	334	145		
707	513			

INTERNATIONAL CLASSIFICATION

G	0	6	F		3/14
					/
					/
					/

GROUP ART UNIT: 2773

ASSISTANT EXAMINER (PLEASE STAMP OR PRINT FULL NAME): Chadwick A. Jackson

PRIMARY EXAMINER (PLEASE STAMP OR PRINT FULL NAME): Raymond J Bayer

PTO 270 (REV. 5-91)

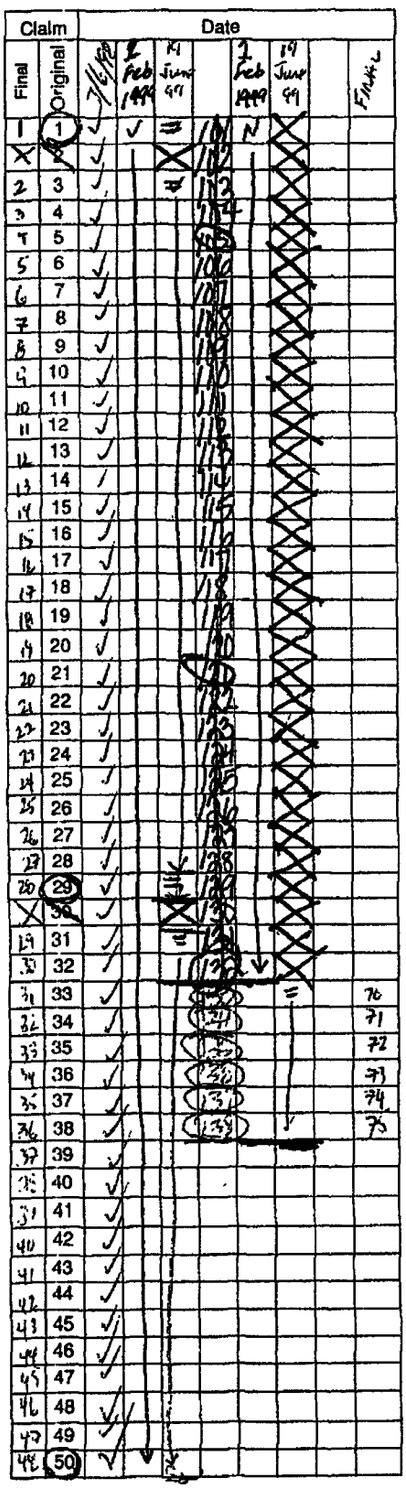
ISSUE CLASSIFICATION SLIP

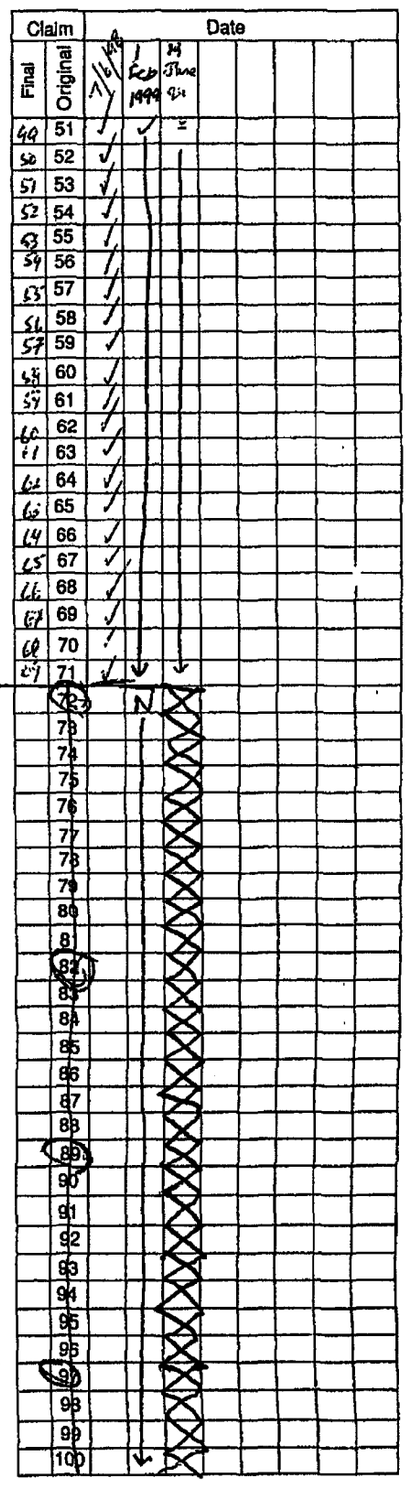
U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

Staple Issue Slip Here

POSITION		ID NO.	DATE
CLASSIFIER		5	7-5-77
EXAMINER		77480	11-7-97
TYPIST		11	11
VERIFIER		11	11
CORPS CORR.			
SPEC. HAND		E42	3-2-98
FILE MAINT.			
DRAFTING			

INDEX OF CLAIMS





SYMBOLS

✓ ..... Rejected
= ..... Allowed
- (Through numberal) Canceled
+ ..... Restricted
N ..... Non-elected
I ..... Interference
A ..... Appeal
O ..... Objected

(LEFT INSIDE)

[redacted]

SEARCHED

Class	Sub.	Date	Exmr.
345	302	6/1/98	C.S
	328		
	329		
	331		
	348		
395	200.33	7/3/98	C.S
	200.47		
	200.48		
	200.49		
same as above	update search	1/28/99	C.S
345	145	6/14/99	C.S
	302		
	328		
	329		
	331		
	339		
	334		
	335		
	345		
395	200.33		
	200.47		
	200.48		
	200.49		

INTERFERENCE SEARCHED

Class	Sub.	Date	Exmr.
345	145	6/19/99	C.S
	302		
	328		
	329		
	331		
	339		
	334		
	335		
	345		
395	200.33		
	200.47		
	200.48		
	200.49		

SEARCH NOTES

	Date	Exmr.
Consultation w/ Primary Raymond Bayer (PPE 2773)	7/1/98	C.S
GPIC TEXT SEARCH		
APS Text + Search	1/27/99	
APS TEXT SEARCH (CURSOR)/Ti, Ab	6/14/99	C.S
ISR Image Search	6/14/99	
Consultation w/ Primary Raymond Bayer		

US005995102A

United States Patent [19]

Rosen et al.

[11] Patent Number: 5,995,102

[45] Date of Patent: *Nov. 30, 1999

[54] **SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE**

[75] Inventors: **James Samuel Rosen**, New York, N.Y.; **Thomas A. Schmitter**, Cambridge, Mass.; **Mark S. Hall**, South Orange, N.J.

[73] Assignee: **Comet Systems, Inc.**, New York, N.Y.

[*] Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

[21] Appl. No.: **08/882,580**

[22] Filed: **Jun. 25, 1997**

[51] **Int. Cl.**[6] **G06F 3/14**

[52] **U.S. Cl.** **345/339**; 345/334; 345/145; 707/513

[58] **Field of Search** 345/302, 328, 345/329, 331, 348, 145, 339, 334, 335, 345; 395/200.33, 200.47, 200.48, 200.49

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,672,575	6/1987	Stephens	364/900
4,841,291	6/1989	Swix et al.	340/725
4,984,152	1/1991	Muller	364/200
5,157,768	10/1992	Hoeber et al.	395/157
5,179,656	1/1993	Lisle	395/159
5,347,628	9/1994	Brewer et al.	395/159
5,544,295	8/1996	Capps	395/152
5,559,943	9/1996	Cyr et al.	395/155
5,572,643	11/1996	Judson	395/793
5,596,694	1/1997	Capps	395/152
5,710,897	1/1998	Schneider	345/334
5,737,619	4/1998	Judson	395/761
5,740,549	4/1998	Reilly et al.	705/14
5,801,698	9/1998	Lection et al.	345/347

OTHER PUBLICATIONS

M. Brown , "WWW Plug–In Companions", 1996, pp. 14–18.

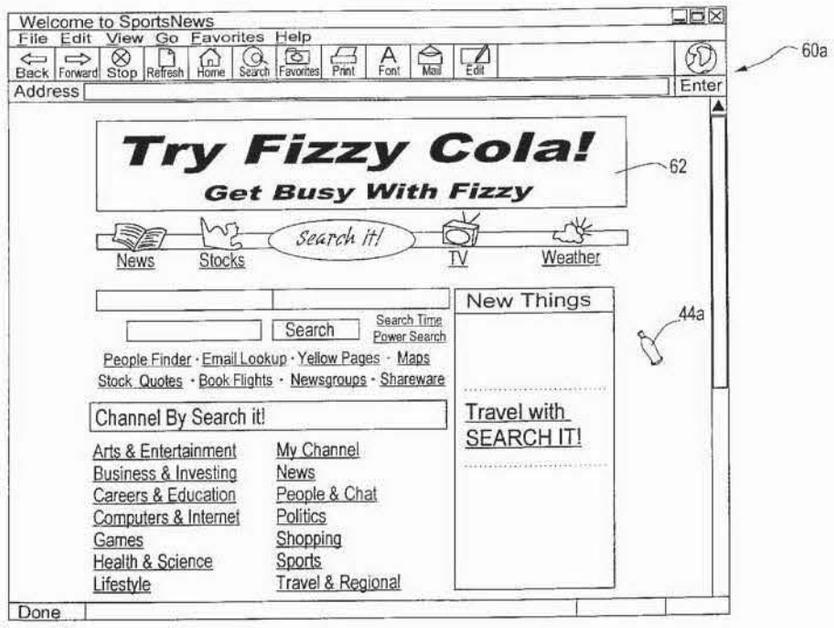
The Java Language Environment —A White Paper Goshing et al., May 1995, pp. 6–64.

Primary Examiner—Raymond J. Bayerl
Assistant Examiner—Chadwick A. Jackson
Attorney, Agent, or Firm—Weingarten, Schurgin, Gagnebin & Hayes LLP

[57] **ABSTRACT**

A system for modifying a cursor image, as displayed on a video monitor of a remote terminal, to a specific image having a desired shape and appearance. The system stores cursor image data corresponding to the specific image, and a cursor display code. The cursor display code contains information in response to which the cursor image is modified to the specific image. A server computer transmits specified information to the remote terminal. The information includes at least one cursor display instruction. The cursor display instruction is operable to modify, in conjunction with the cursor information and the cursor image data, a cursor image displayed by a display of the remote terminal in the shape and appearance of the specific image.

75 Claims, 9 Drawing Sheets



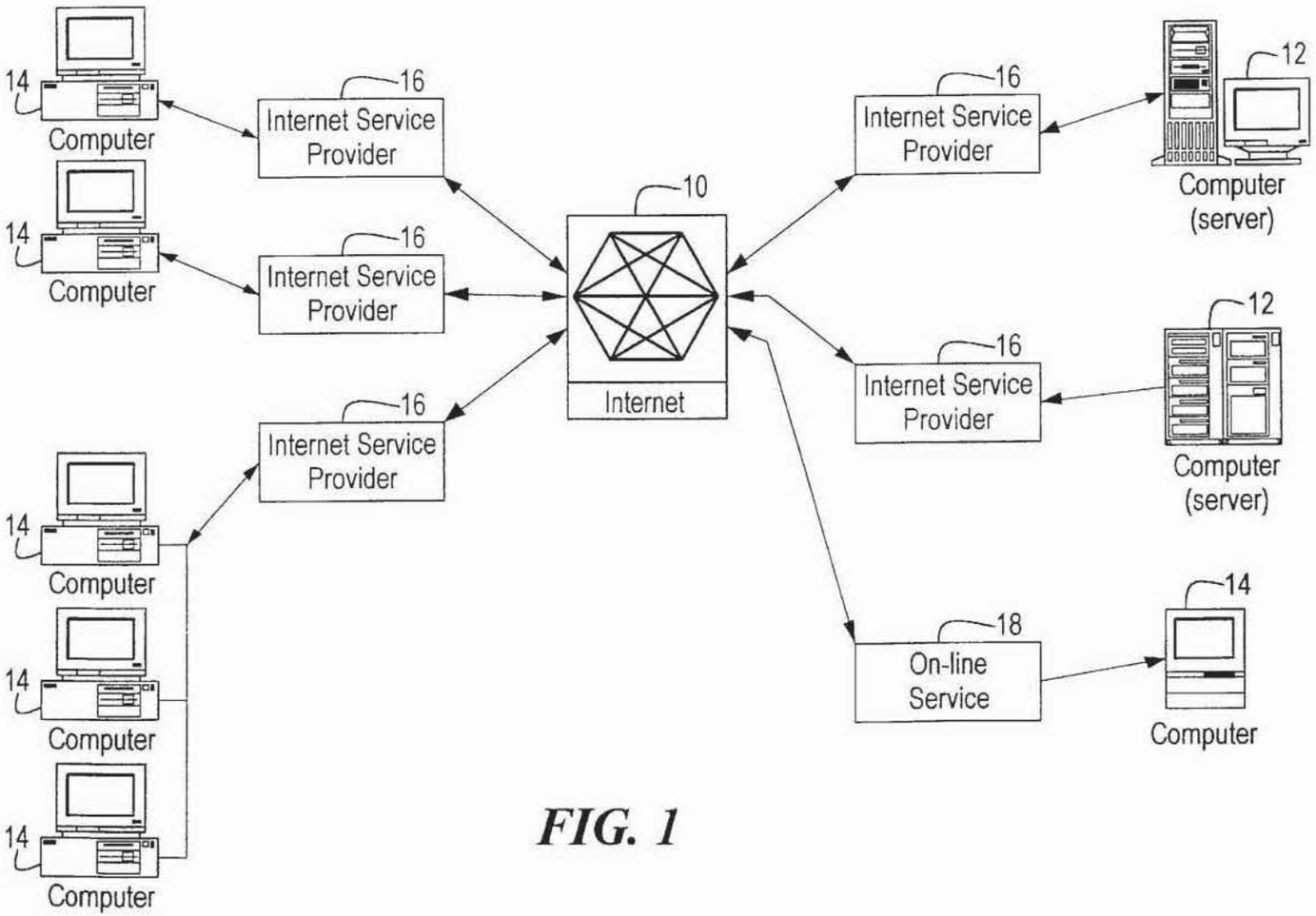

14
Computer
16
Internet Service Provider
10
Internet
12
Computer (server)
18
On-line Service

FIG. 1

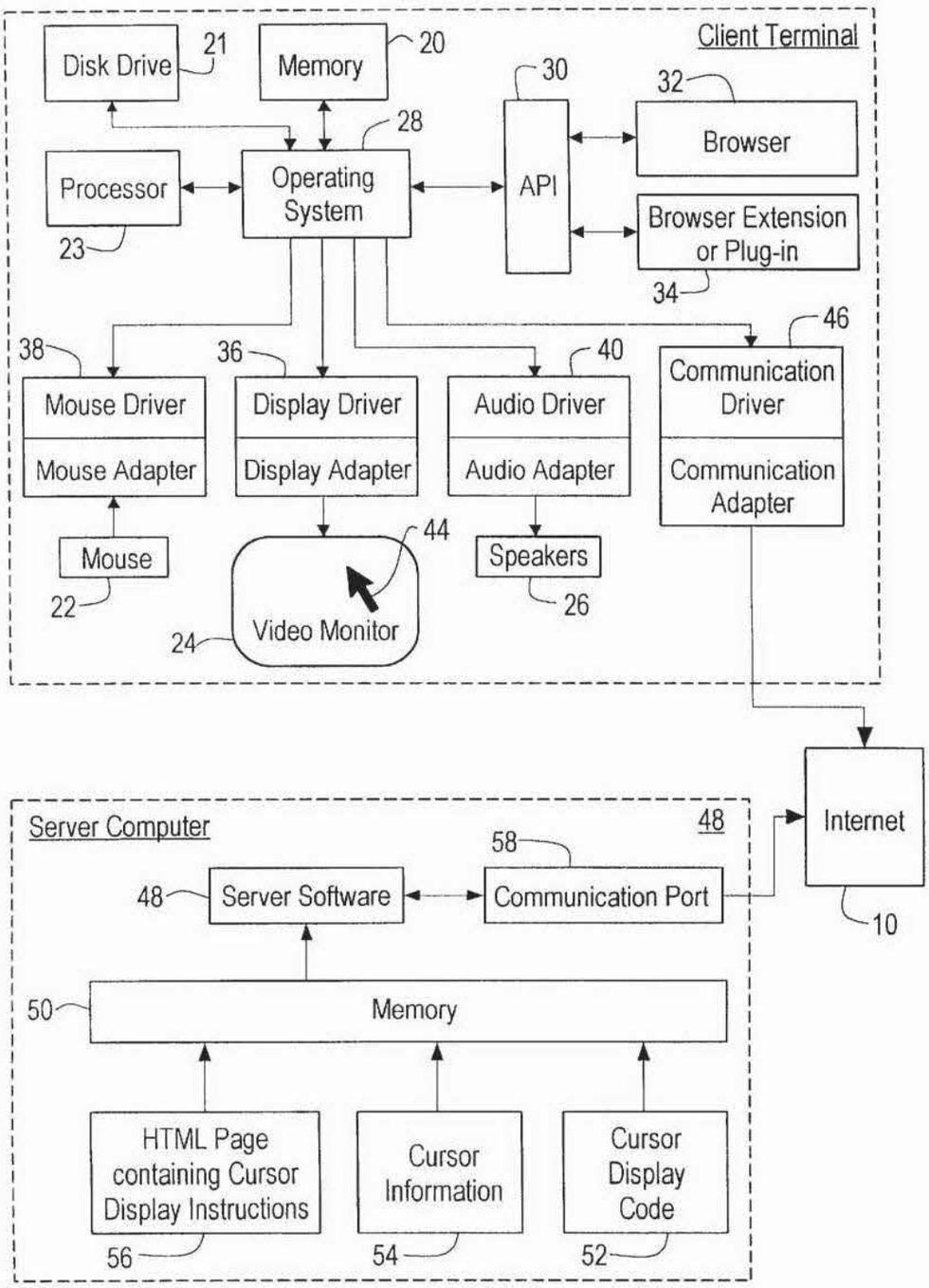

Client Terminal
Disk Drive
21
Memory
20
30
32
28
Processor
Operating System
API
Browser
Browser Extension or Plug-in
23
34
46
38
36
40
Mouse Driver
Mouse Adapter
Display Driver
Display Adapter
Audio Driver
Audio Adapter
Communication Driver
Communication Adapter
Mouse
22
44
Video Monitor
24
Speakers
26
Internet
10
Server Computer
48
58
48
Server Software
Communication Port
50
Memory
HTML Page containing Cursor Display Instructions
56
Cursor Information
54
Cursor Display Code
52

FIG. 2

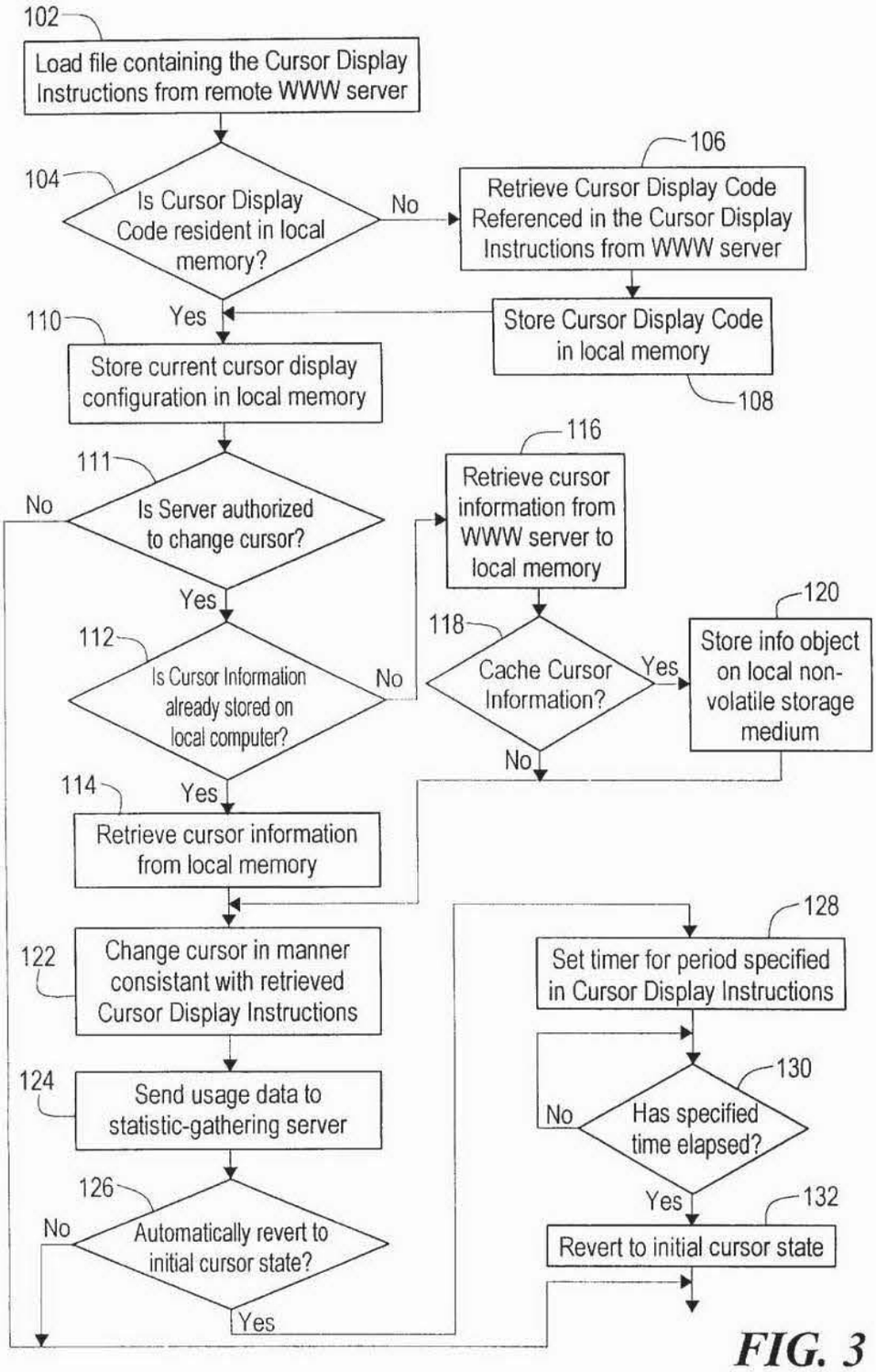

102
Load file containing the Cursor Display Instructions from remote WWW server
104
Is Cursor Display Code resident in local memory?
No
106
Retrieve Cursor Display Code Referenced in the Cursor Display Instructions from WWW server
Yes
Store Cursor Display Code in local memory
108
110
Store current cursor display configuration in local memory
111
Is Server authorized to change cursor?
No
Yes
112
Is Cursor Information already stored on local computer?
No
116
Retrieve cursor information from WWW server to local memory
118
Cache Cursor Information?
Yes
120
Store info object on local non-volatile storage medium
No
Yes
114
Retrieve cursor information from local memory
122
Change cursor in manner consistant with retrieved Cursor Display Instructions
124
Send usage data to statistic-gathering server
126
Automatically revert to initial cursor state?
No
Yes
128
Set timer for period specified in Cursor Display Instructions
130
Has specified time elapsed?
No
Yes
132
Revert to initial cursor state

FIG. 3

```
<OBJECT
        202.ID=cc1
        203.TYPE="application/x-oleobject"
        204.CLASSID="clsid:CB005660-D0C7-11cf-B7F6-00AA00A3F278"
        205.CODEBASE="http://cometsystems.com/controls/cc.cab#ver=4,70,
           0,1122"
        206.<PARAM NAME="CursorType" VALUE="1"
        207.<PARAM NAME="CursorImage"
           VALUE="http://cometsystems.com/library/images/acme.cur">
        208.<PARAM NAME="Counter" VALUE="http://
           cometsystems.com/accounting">
        209.<PARAM NAME="DisplayDuration" VALUE="5">
        210.<PARAM NAME="CacheCursor" VALUE="1">
        211.<PARAM NAME="ServerSignature" VALUE="54F5254A23BD988AB54">
        212.<PARAM NAME="DormantDelay" VALUE="600">
        213.<PARAM NAME="CursorTrajectoryMap" VALUE="http://
           cometsystems.com/maps/trajectory">
        214.<PARAM NAME="CursorPositionMap" VALUE="http://
           cometsystems.com/maps/position">
        215.<PARAM NAME="CursorVelocityMap"
           VALUE="http://cometsystems.com/maps/velocity">
        216.<PARAM NAME="CursorPositionMap" VALUE="http://
           cometsystems.com/maps/velocity">
        217.<PARAM NAME="CursorButtonMap" VALUE="http://
           cometsystems.com/maps/buttonstate">
        218.<PARAM NAME="ContentType" VALUE="5">
        219.<PARAM NAME="PriorityLevel" VALUE="1">
        220.<PARAM NAME="StreamBufferSize" VALUE="0">
        221.<PARAM NAME="SatelliteImage"
           VALUE="http://cometsystems.com/library/images/acmesat.bmp">
        222.<PARAM NAME="SatelliteXDisplacement" VALUE="-50">
        223.<PARAM NAME="SatelliteYDisplacement" VALUE="50">
        224.<PARAM NAME="ExtraDisplayParameters"
           VALUE="http://cometsystems.com/library/params/acme.prm">
</OBJECT>
```

FIG. 4

```
<script language="VBScript">
<!--\1`;
302.Sub window_onLoad()
303.    ccl.RememberCurrentCursor()
304.    ccl.SetNormalCursor("http://cometsystems.com/library/
           images/acme.cur")
305.end sub

306.Sub window_onUnload()
307.    ccl.Reset()
308.end sub
-->
</script>
```

FIG. 5

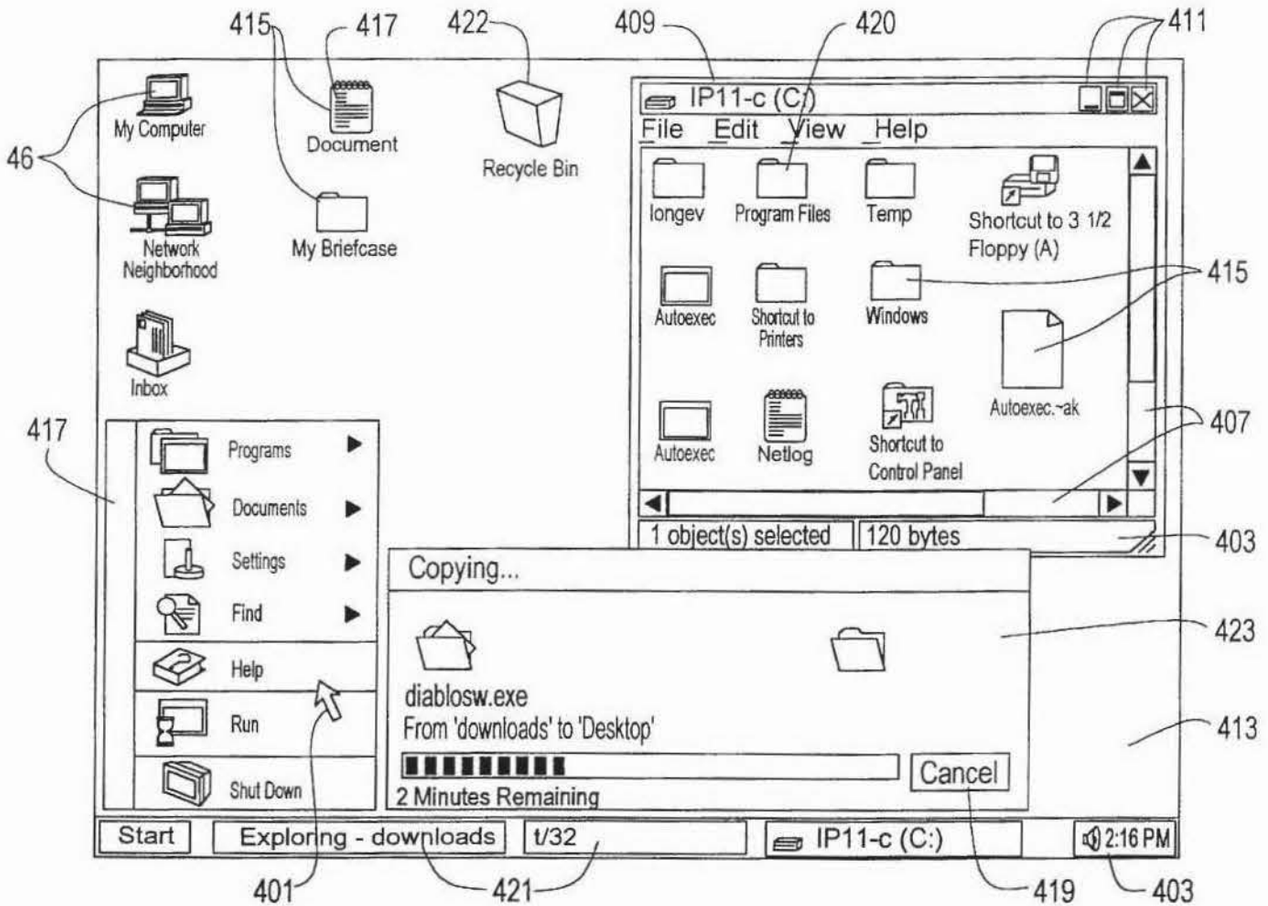

415
417
422
409
420
411
My Computer
Document
Recycle Bin
IP11-c (C:)
File Edit View Help
46
longev
Program Files
Temp
Shortcut to 3 1/2 Floppy (A)
Network Neighborhood
My Briefcase
Autoexec
Shortcut to Printers
Windows
415
Inbox
Autoexec
Netlog
Shortcut to Control Panel
Autoexec.~ak
407
417
Programs
Documents
Settings
Find
Help
Run
Shut Down
1 object(s) selected
120 bytes
403
Copying...
423
diablosw.exe
From 'downloads' to 'Desktop'
413
Cancel
2 Minutes Remaining
Start
Exploring - downloads
t/32
IP11-c (C:)
2:16 PM
401
421
419
403

FIG. 6

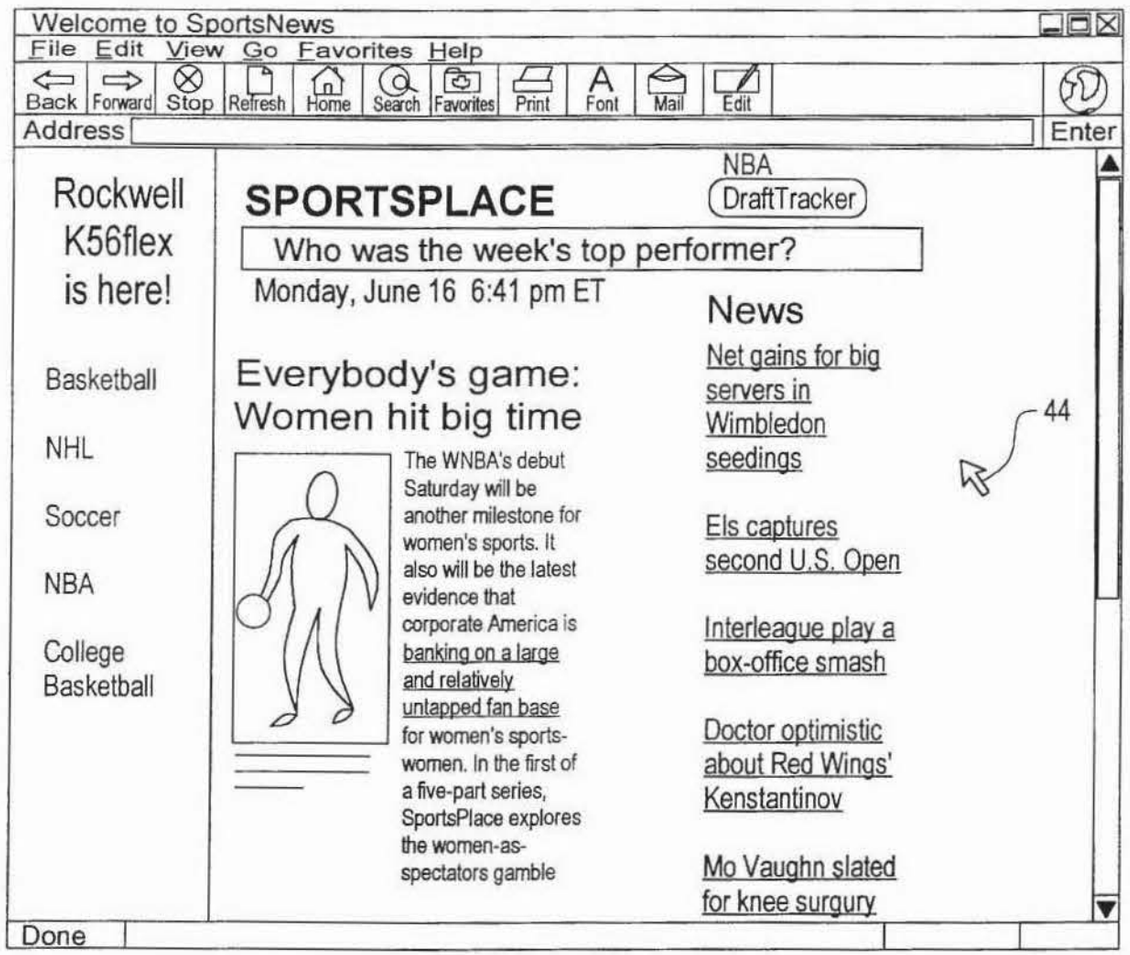

Welcome to SportsNews
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Font Mail Edit
Address
Enter
60
Rockwell K56flex is here!
Basketball
NHL
Soccer
NBA
College Basketball
SPORTSPLACE
NBA DraftTracker
Who was the week's top performer?
Monday, June 16 6:41 pm ET
Everybody's game: Women hit big time
The WNBA's debut Saturday will be another milestone for women's sports. It also will be the latest evidence that corporate America is banking on a large and relatively untapped fan base for women's sports-women. In the first of a five-part series, SportsPlace explores the women-as-spectators gamble
News
Net gains for big servers in Wimbledon seedings
Els captures second U.S. Open
Interleague play a box-office smash
Doctor optimistic about Red Wings' Kenstantinov
Mo Vaughn slated for knee surgury
44
Done

FIG. 7

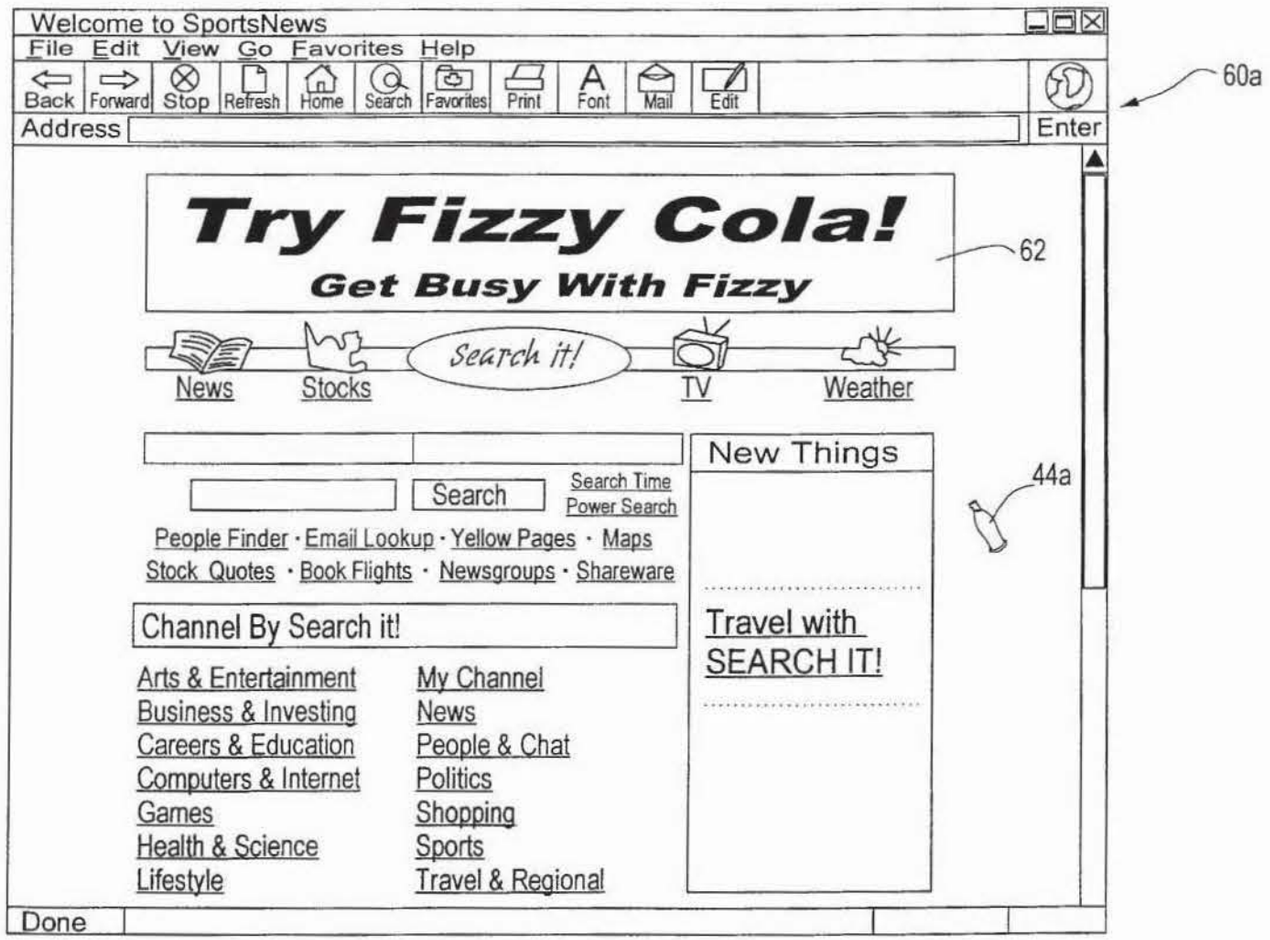

Welcome to SportsNews
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Font Mail Edit
Address
Enter
60a
Try Fizzy Cola!
Get Busy With Fizzy
62
News
Stocks
Search it!
TV
Weather
Search
Search Time
Power Search
People Finder · Email Lookup · Yellow Pages · Maps
Stock Quotes · Book Flights · Newsgroups · Shareware
Channel By Search it!
Arts & Entertainment
Business & Investing
Careers & Education
Computers & Internet
Games
Health & Science
Lifestyle
My Channel
News
People & Chat
Politics
Shopping
Sports
Travel & Regional
New Things
Travel with SEARCH IT!
44a
Done

FIG. 8

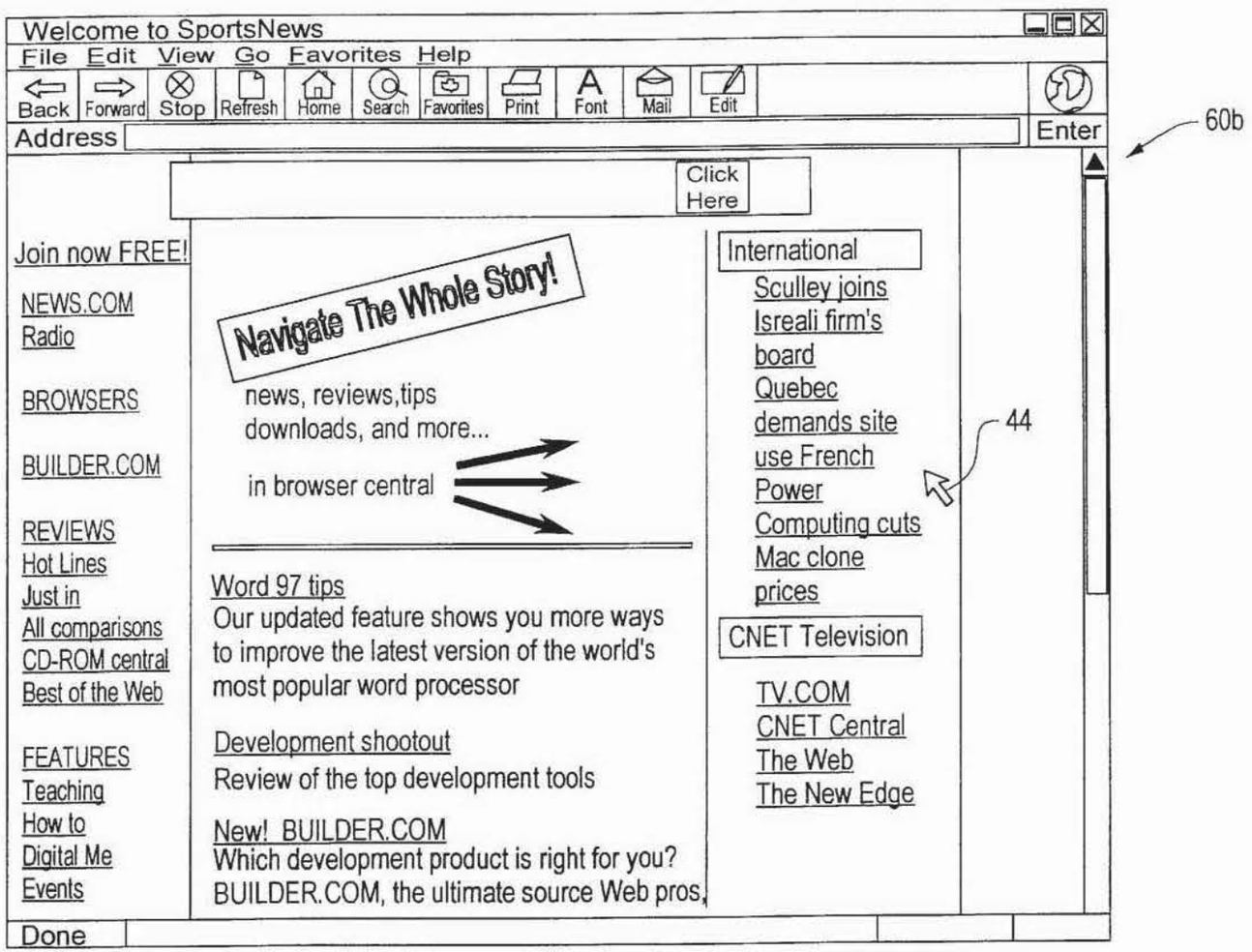

Welcome to SportsNews
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Font Mail Edit
Address
Enter
60b
Click Here
Join now FREE!
NEWS.COM
Radio
BROWSERS
BUILDER.COM
REVIEWS
Hot Lines
Just in
All comparisons
CD-ROM central
Best of the Web
FEATURES
Teaching
How to
Digital Me
Events
Navigate The Whole Story!
news, reviews,tips
downloads, and more...
in browser central
Word 97 tips
Our updated feature shows you more ways to improve the latest version of the world's most popular word processor
Development shootout
Review of the top development tools
New! BUILDER.COM
Which development product is right for you? BUILDER.COM, the ultimate source Web pros,
International
Sculley joins
Isreali firm's board
Quebec demands site use French
Power Computing cuts Mac clone prices
44
CNET Television
TV.COM
CNET Central
The Web
The New Edge
Done

FIG. 9

SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE

FIELD OF THE INVENTION

This invention relates to computer networks and software, and more particularly, to a server system capable of modifying a cursor image displayed on a remote client computer.

BACKGROUND OF THE INVENTION

The World Wide Web ("WWW" or "web") and online services such as America Online, in conjunction with faster and more powerful personal computers, have rendered the Internet and other interactive online computer networks accessible to millions of people all over the world. Concomitant with the emergence of this new communication medium, digital content providers have proliferated, providing online news, entertainment, games and all sorts of other content. As with other mass mediums, such as television, radio, and print publications, the entities that create such content seek to offset their expenses by selling advertising. With reference to the WWW, online advertising has become a multimillion dollar business, to the amount of approximately $300 million dollars in 1996.

The most common type of online advertisement exists in the form of "banner advertisements". Users of online services routinely encounter banner ads on the top, sides, and/or bottom of their video monitor screens when viewing a web page. Banner ads are generally square or rectangular boxes provided with some combination of graphics, color and text directed to the product or service being advertised. As such, the intention of these banner advertisements is to create impressions among online users and to convey some advertising message and/or logo. Banner ads are usually provided on a web page in the form of a "hyperlink", in which users who yield to the advertisements solicitation to "Click Here" are transported to the web site of the manufacturer of the product or service being advertised, or to some other screen which provides additional information about the product or service.

Unfortunately, banner ads occupy only a small portion of a web page. As the user scrolls down a page the banner ad disappears. Although online advertisers and content publishers have attempted to optimize the visibility of banner advertisements by placing them on a popular web page where they will have a greater chance of being seen, Internet users, nevertheless, can easily ignore or find ways to remove and eliminate from their view the banner ads which exist on the web pages they are viewing. As such, the banner ads are rendered ineffective in their aim to provide information about a product or service. Additionally, money spent to advertise a product may be wasted if users are able to ignore or remove the advertisements from the web pages they are viewing.

Another method of online advertising involves the use of "frames" on a web page. Frames are a feature supported by the recent versions of leading web navigating programs known as browsers, such as Netscape Navigator® and Microsoft's Internet Explorer®. Frames generally divide up a user's screen so that the user can, for example, independently scroll down each of numerous frames which appear on the web page being viewed on the user's screen. Like banner advertisements, frames can be aesthetically unappealing as well as confusing to the user. Additionally, placement of advertising frames on a web page generally results in cramping or decreasing the size of the main content frame which oftentimes renders the content in the main frame difficult to read. As a result, users have developed ways to reduce the size or even eliminate frames from the web page being viewed.

Another type of online advertising involves the self-appearing window which generally appears on its own as a user is using the Internet or browsing on the WWW. Such advertisements are relatively easy for a user to avoid as a user may simply re-size the window to make it smaller, drag another window or object in front of it to obscure it from view, close the advertising window, or simply ignore it and continue with the task being undertaken online. Recently, online advertisers have begun using self-appearing screens which are delivered via dialog boxes which dominate the main part of the screen. Although these dialog boxes can be removed when the user clicks on the appropriate place(s) on the dialog box, the self-appearing dialog boxes have a much higher rate of being seen by users. This follows because the dialog boxes take control of the user's screen for a preset amount of time and/or until the user clicks on the appropriate place(s) to make the dialog box disappear. The recent prevalence in the use of self-appearing dialog box advertising has resulted in a more intrusive method of advertising which has resulted in resentment among users who are accustomed to more passive online advertising methods such as the frames and banner advertisements which are more easily avoided and/or ignored.

Accordingly, there is a need for a simple means to deliver advertising elements, i.e. logos, animations, sound, impressions, text, etc., without the annoyance of totally interrupting and intrusive content delivery, and without the passiveness of ordinary banner and frame advertisements which can be easily ignored.

OBJECTS AND SUMMARY OF THE INVENTION

It is thus a general object of the present invention to provide a means for delivering online advertisements which are unintrusive and which are not easily ignored by a user.

A more specific object of the present invention is to provide a server system for modifying a cursor image to a specific image displayed on a video monitor of a remote user's terminal.

It is another object of the present invention to provide a server system for modifying a cursor image to a specific image displayed on a video monitor of a remote user's terminal for the purposes of providing on-screen advertising.

It is a further object of the present invention to provide a means for providing on-screen advertising transmitted online which does not interrupt the delivery of content and which is aesthetically appealing and which affords the advertiser a great degree of unintrusive exposure.

It is still a further object of the present invention to provide a system and a method for causing a remote user terminal to display a cursor image as specified by a server terminal.

It is also an object of the present invention to provide a system and method for causing a remote user terminal to display a cursor image as specified by a server terminal, wherein the cursor image corresponds to the content retrieved by the user terminal.

It is a further object of the present invention to provide a system and method for causing a remote user terminal to display a cursor image such as a corporate name or logo, a brand logo, an advertising or marketing icon or slogan, an animated advertising image, and a related audio clip, that

relate to an advertisement, such as a banner advertisement, that is included in the information content being retrieved by the user terminal.

It is an additional object of the present invention to provide a means for changing a cursor's appearance by sending data and control signals from a remote computer so that the cursor or pointer's appearance is associated with a portion of, or the entire content being displayed on the user's screen.

It is still an additional object of the present invention to provide a means for changing the appearance of a computer's cursor or pointer by sending data and control signals from a remote computer so that the cursor or pointer's appearance is associated with advertising messages.

These and other objects of the invention are realized in various embodiments of the present invention by providing a system for delivering advertising elements online without the annoyance resulting from the interruption of content delivery and without the passiveness of ordinary banner and frame advertisements which can be too easily ignored or bypassed or removed. An exemplary embodiment of the present invention is directed to a system that provides online advertising content using the on-screen cursor which is generally controlled by an input of positioning device known as a "mouse" or "mouse pointer". Nearly all online computer interfaces utilize a wired or remote control positioning device such as a mouse or roller or track ball which controls the cursor's movement on the screen. It is the cursor controlled by the mouse or positioning device which a user uses to "navigate" or move the cursor over objects, buttons, menus, scroll bars, etc., which appear on-screen and then clicking or in some cases double-clicking in order to activate a screen or task, or to commence an application or some function.

As a result of the prevalence of the use of the mouse, by many millions of users of online systems, a great deal of time is spent focused on the icons which represent the cursor or pointer as it may appear in some cases. Presently, pointer icons change from application to application and can also change within an application depending upon where on the screen the pointer is located, what state the computer exists in at a given moment, and what tools are being used, among other factors. Generally, pointers change shape to reflect an internal state of the computer or the present function within an application. While it is not new for pointers and cursors to change shape, pointers are not presently used to convey advertising. In conventional systems, the appearance of the cursor or pointer does not change to correspond with on-line content being displayed on the screen.

The present invention provides a means for enabling cursors and pointers to change color, shape, appearance, make sounds, display animation, etc., when the user's terminal or computer, known as the "client" or "user" terminal, which has a network connection, receives certain instructions from a remote or "server" computer attached to the network. In an exemplary embodiment of the present invention, the generic cursor or pointer icons used in many networking applications, such as black arrows, hands with a pointing finger, spinning wheels, hourglasses, wristwatches, and others, will change appearance, and in some cases may incorporate sound or animation, in a way that is linked and related to the content, such as a web page, which is being transmitted to and displayed on the client computer. The cursor or pointer may appear as a corporate or a brand logo which relates to advertising content within the web page being transmitted and displayed. The cursor or pointer image may also appear in a specified shape or color that is intended to convey a message that relates to the advertising content within the web page being transmitted and displayed.

An exemplary embodiment of the present invention comprises a combination of hardware and enabling software residing on the transmitting (server) computer or network server and/or on the receiving (client or user) computer or terminal which brings about the stated effect of enabling a computer's cursor or pointer to change appearance and in certain cases provide sound and animation which is linked and related to the content being transmitted to and displayed on the client computer or terminal. The transmitting computer and receiving computer or terminal advantageously include a processor, an operating system (OS) loaded thereon, a video monitor used to display a graphical user interface (GUI) and a Hypertext Transfer Protocol (HTTP) compliant web browser capable of loading and displaying hypertext documents transmitted over the Internet, although the invention is not limited in scope in that respect. For example, the receiving terminal may be any device that is able to communicate with a remote server, such as a user computer terminal, a user dumb terminal, or a television based system, such as a Web TV® terminal and other devices.

Preferably, coded information for bringing about the change in appearance of the cursor are embedded within the web page being loaded and viewed. In one embodiment of the present invention, the web page is written in Hypertext Markup Language (HTML) which is one of the most common standard page description languages used to develop web pages. Typically a web browser retrieves a web page to be loaded on a user's terminal. The retrieved web page in accordance with one embodiment of the invention contains a set of predetermined instructions referred to herein as cursor display instructions. The browser or browser extension interprets the information contained in cursor display instructions and instructs the operating system of the user's terminal via an application programming interface (API) to check its memory to determine if the user terminal is capable of loading the coded image, animation, and/or soundbite. If the image, etc. has been previously cached in the client computer memory, the cursor display instructions instruct one or more of the many devices controlled by the operating system in the user's terminal, such as the video monitor and audio speakers to display the desired images, animation and play desired sounds. If the image, etc. has not been previously cached in the client computer's memory, the browser or browser extension retrieves the information corresponding to the desired image from a remote server.

The present invention may serve to enhance banner advertisements which appear on a web page so as to remind users which company is sponsoring the particular page being viewed and to draw the user's attention to the banner advertisement. The present invention can also serve as a stand-alone branding vehicle as part of a "ubiquity campaign" to generate massive impressions among an audience of online users or can be simply used to make web sites more entertaining by providing animated, colorful cursors which may incorporate sound and/or animation, and which are configured so as to connote a relationship with the topic or subject of the web site.

The foregoing sets forth certain objects, features and advantages provided by exemplary embodiments of the present invention. Other objects and features of the present invention will become apparent from the following detailed description considered in conjunction with the accompanying drawings. It is to be understood, however, that the

drawings are designed solely for the purposes of illustration and not as a definition of the limits of the invention, for which reference should be made to the appended claims.

DETAILED DESCRIPTION OF THE DRAWINGS

In the drawings in which like reference characters denote similar elements throughout the several views:

FIG. **1** illustrates a diagrammatic representation of a computer network illustrating the interconnection of a plurality of computers in which the present invention is implemented;

FIG. **2** illustrates a client-server computer network supporting the hardware and software of the present invention;

FIG. **3** illustrates a flowchart diagram of an exemplary method of the present invention for obtaining information from a remote site for modifying a cursor image and implementing such information at numerous user sites;

FIG. **4** illustrates a portion of the Cursor Display Instructions which is referenced as a resource within an HTML document according to one embodiment of the present invention;

FIG. **5** illustrates a set of exemplary codes that cause the user terminal's cursor to be modified, then revert to its original shape in accordance with one embodiment of the present invention;

FIG. **6** illustrates a plurality of user interface attributes that may be remotely modified in accordance with one embodiment of the present invention; and

FIGS. **7–9** illustrate the appearance of a cursor prior to, during and after linking to a web page that contains cursor display instructions.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. **1** illustrates a computer network, such as Internet **10**, based on the client-server model. Internet **10** comprises a worldwide network of computers known as "servers" **12** which are accessible by "client computers" or "user terminals" **14**, which are typically used by individual users or comprise a collection of personal computers interconnected via a Local Area Network or LAN, which are capable of accessing the Internet via a private Internet service or access provider (ISP) **16**, such as the AT&T Worldnet Service® or the IBM Global Network®, or via an online service provider **18**, such as America Online®, Compuserve®, the Microsoft Network® or Prodigy® (to name the most popular online service providers). One of the most common applications of the Internet is to support the World Wide Web ("WWW" or "the web"), which is a collection of servers on the Internet that utilize the Hypertext Transfer Protocol (HTTP), a known application protocol that facilitates data exchange between client and server and provides users or clients **14** access to files which can include text, graphics, sound, video, etc., using a standard page description language referred to as Hypertext Markup Language (HTML).

Each client computer **14** as indicated in FIG. **1**, includes a "web browser" or browser loaded on the client computer's hard drive **21**. A browser is a common software tool which allows graphical user interface (GUI)-based access to Internet network servers **12** through Internet Service Providers, ISPs, **16** or online service providers **18**. A server **12** functions as a so-called "web site" which supports and maintains a plurality of files in the form of documents and pages. A Uniform Resource Locator or URL identifies a specific network path to a server **12** or some resource located on that server which has a known syntax for defining the network connection. The fundamental intrinsic capabilities of the browser are: (1) the ability to communicate with other computers using HTTP, and (2) the ability to process and present HTML documents to the user via a graphical user interface, GUI.

Recent versions of most browsers provide a plethora of other features beyond these two capabilities. For example, to increase its flexibility, the browser's intrinsic capabilities may be further extended through the use of software components, often called "controls" or "plug-ins". While the intrinsic capabilities of the browser are linked at compile-time ("statically"), the code which implements the capabilities of the control or plug-in component is linked with the browser's code at run-time ("dynamically"). By supporting these components through standard interface definitions, the browser's capabilities can be extended in ways never anticipated by its original manufacturer.

Another type of flexibility is offered when the browser implements some sort of command interpreter which is capable of interpreting and executing a code stream at run-time. In this case, the browser acts as a sort of "virtual machine" whose run-time behavior is completely governed by the code stream which it processes. The total scope of capabilities which can be realized with this approach is defined by the set of operations supported by the command interpreter.

Individually and collectively, these mechanisms provide a powerful and flexible platform which supports a wide range of Internet-based applications. Currently, some of the emerging standards govern the operation of these mechanisms, although the invention is not limited in scope in that respect. For example, Microsoft has created an interface definition for Windows "dynamic link libraries" and for ActiveX software components. Sun Microsystems has defined a software component model called JavaBeans. Sun has also created a virtual machine architecture and language called Java, which is supported via a variety of commercially available compilers. While a Java compiler translates source code into pseudo-code output called an "applet", which is in turn processed by the Java virtual machine, Microsoft, Sun, and others have also defined a set of HTML scripting languages whose source code is embedded directly in an HTML page. Microsoft's VBScript, JScript and Sun's JavaScript are examples of these embedded scripting languages.

The standard web page description language, HTML, provides basic document formatting and permits the web site developer to create and specify "links" or "hyperlinks" to other servers and files. Obtaining a web page or connecting to a web site requires the specification of a URL using an HTML-compliant client browser. After specifying the URL, client computer **14** initiates a request to server **12** identified in the link and connects to the web site and receives a web page. The request by client computer **14** to server **12** via the link is advantageously communicated via a TCP/IP (Transfer Control Protocol/Internet Protocol) communication, although the invention is not limited in this respect and other network connections or Internet protocols may be used.

Although an exemplary embodiment of the present invention is described based on the arrangement illustrated in FIG. **1**, it is noted that the invention is not limited in scope in that arrangement and other types of system connections may be employed. For example, a plurality of user terminals may be connected to an online provider via dedicated communica-

tion channels, such as telephone lines. In accordance with this embodiment, the server system provides certain information that causes the cursor image on the video monitor of the user terminal to display an image as specified by the server system. As a result, the server system remotely defines and manages the shape and appearance of the cursor image in accordance with a pre-specified condition. The shape and appearance of the cursor image may correspond to the actual content of the data being provided to the user. Furthermore, regardless of the actual content of the data being provided to the user, the shape and appearance of the cursor image may be specified by the server system such that a plurality of user terminals at a desired point in time receive appropriate instructions to display the specified cursor image.

FIG. **2** provides a block diagram of hardware and software which is representative of a client-server network system connected via the Internet according to one embodiment of the present invention. The user or client computer or user terminal **14** typically includes a number of hardware components and software subsystems which cooperate to deliver the wide range of capabilities demanded by a modern computer application or program. These include not only the basic computational processor **23** and memory **20**, but also a variety of input and output devices such as the keyboard (not shown), mouse **22**, video display monitor **24**, audio speakers **26**, non-volatile storage such as a hard drive **21** and network communications systems **46** such as a modem among other devices. User terminal **14** is controlled via an operating system ("OS") **28** which serves to organize all the disparate elements within the computer **14** and expose them in a consistent and organized way to a program which may need some or all of these capabilities. The interface between a program, which is generally loaded within the computer's memory **20**, and the systems under the control of the operating system **28** is commonly referred to as the Application Programming Interface ("API") **30**, which is essentially a library of functions which the program ("application") can invoke when it needs to interact with any of these hardware subsystems.

As illustrated, user terminal **14** contains a browser **32** loaded within the computer's memory **20**, and is adapted to communicate with a browser extension or browser plug-in **34**, both which are adapted to communicate with the operating system **28** via the application programming interface API **30**. As illustrated, operating system **28** is supplemented by a set of "drivers" which control and provide the operating system **28** with access to peripheral devices which are a part of user terminal **14**. The drivers include display driver **36** which controls and provides the operating system **28** with access to the cursor image or pointer **44** projected on video display monitor **24**, a mouse driver **38** which controls and provides the operating system **28** with access to mouse **22**, an audio driver **40** which controls and provides the operating system **28** with access to speakers **26**. Operating system **28** is configured to provide animated images to the video monitor. Furthermore, in accordance with another embodiment of the invention, the display driver may be configured to provide animated images to the video monitor. Operating system **28** also provides access to a communication port **46** such as a modem which serves as a communication interface to the Internet **10**.

With continued reference to FIG. **2**, user terminal **14** is connected to Internet **10** via a modem or some other communication interface such that information may be transmitted between user terminal **14** and Internet **10** via communication lines such as telephone cables or fiber optic networks, among other types of transmission systems. Internet **10** is also connected to numerous network servers, such as a simplified representation of a WWW server which is indicated as **48**. Server **48** is provided with memory **50** into which the contents of certain data files are loaded. Such data files, among others, include Cursor Display Code **52**, Cursor Information **54**, and an HTML page containing Cursor Display Instructions **56**, all of which are discussed in greater detail herein below. As illustrated in FIG. **2**, these data files **52, 54, 56** are shown residing on the same server computer. However, the interconnected nature of the WWW allows these data files **52, 54, 56** to exist anywhere on Internet **10**. For example, files **52** containing cursor display codes may be stored in various server systems, while files **54** containing cursor information may be stored in the same or other server systems, and files **56** containing HTML pages containing cursor display instructions may be stored in the same or yet other server systems.

In operation, WWW server **48** includes software which recognizes file requests received from WWW clients or users by communication port **58** and fulfills these requests by retrieving data stored in data files, i.e., Cursor Display Code **52**, Cursor Information **54**, and an HTML page containing Cursor Display Instructions **56**.

One of the characteristics of most recent software systems is the graphically oriented user interface (GUI) which is viewable on video monitor **24**. This graphical user interface helps to organize and filter the vast quantities of information which is accessible in a user terminal **14**. Fundamental to the graphical user interface is the pointing device, generally mouse **22** which allows the user to manipulate or input information into the user terminal **14**. Movement of mouse **22** is monitored by user terminal **14** which translates this movement into a corresponding movement of cursor **44** viewable on video monitor **24**. As such, operating system **28** may expose, as some subset of its API **30**, a set of functions which can be used to control aspects of the behavior and/or appearance of cursor **44**.

By combining the capabilities of browser extensions, such as indicated by **34** in FIG. **2**, with the capabilities to modify cursor **44**, it is possible for a WWW server, such as that indicated by **48** in FIG. **2**, to control the display characteristics of cursor **44** displayed on video monitor **24** of the user's computer **14**. By doing so, a cursor control arrangement is established which is capable of delivering information which supplements, enhances, or is completely independent of, other information transmitted from a server, such as indicated by **48**, through traditional means as via a communications port **58**. The basic conceptual components of such exemplary system for modifying cursor **44** comprises Cursor Display Code **52**, Cursor Information **54**, and Cursor Display Instructions **56**, discussed hereinabove with reference to FIG. **2**. Preferably, Cursor Display Code **52** comprises a set of instructions which are executed on the user terminal **14** and which interact directly with application programming interface **30** of the user terminal **14** and operating system **28** so as to accomplish the actual change of cursor **44**. Cursor Information **54** is, advantageously, a set of data which identifies the actual cursor image or images and corresponding audio content if desired. In one embodiment of the invention, Cursor Display Instruction **56** includes data that convey information that is used by Cursor Display Code **52** to control drivers, such as **36, 40, 46**, and to identify such things, which among others consist of: the physical location of Cursor Information **54**, the format of its representation, the intended manner and duration of its display, and information pertaining to how (and for how long) any cached Cursor Information **54** should be stored.

In general, the fundamental elements of the process of changing cursor **44** displayed on video monitor **24** of user terminal **14** are as follows: Cursor Display Instructions **56** are initially embedded inside an HTML document, e.g. a web page. When browser **32** of the user terminal **14** encounters Cursor Display Instructions **56**, Cursor Display Code **52** is retrieved then invoked. As part of the invocation, the browser passes to the Cursor Display Code coded information sufficient to specify the manner of the display. Cursor Display Code **52** then retrieves Cursor Information **54** either from within memory **20** of user terminal **14** or from storage at a remote site and then causes the Cursor Information to interact with the display system, such as display driver **36**, of user terminal **14** via the application programming interface **30** of operating system **28**. This interaction causes Cursor Information **54** to be accessed by the display driver **36** in order to accomplish the intended effect, e.g., the change or transformation of cursor **44** visible on video monitor **24**, and a corresponding sound information may be heard on speakers **26**.

FIG. **4** illustrates the Cursor Display Instruction as a resource within an HTML document which is retrieved from a remote server. The Cursor Display Instructions as shown in FIG. **4** are written for ActiveX® technology, although the invention is not limited in scope to that technology. Among the information included within this resource definition is an identifier of the Cursor Display Code (the ActiveX® control), and the ActiveX® control's physical location on the Internet. This information is listed in lines **202–205** which generally identifies the Cursor Display Code. Line **204** of the Cursor Display Instruction is an identifier which comprises a globally unique name, often called a "Class ID", and which allows a particular ActiveX® control to be distinguished from all other ActiveX® controls, such that the wrong ActiveX® control is prevented from being utilized or retrieved. The remainder of the Cursor Display Instruction listed in lines **206–224** include the ActiveX® parameters or argument list as discussed hereinafter with reference to FIG. **3**. The argument list includes parameters which provide information such as the type of cursor image (line **206**), where the image can be retrieved from if not already resident on the user computer (line **207**), where usage statistics are to be transmitted to (line **208**), how long a changed image should remain before reverting, if at all, to the initial image (line **209**), whether the cursor image is cached in the user terminal (line **210**), whether the transmitting server is authorized to send cursor display instructions (line **211**), the dormant delay duration (line **212**), the URL of a file which specifies cursor trajectory path (line **213**), the URL of a file which specifies how the cursor's shape should change based on its location on the screen (line **214**), the URL of a file which specifies how the cursor's shape should change based on its velocity (line **215**), the URL of a file which specifies how the cursor's shape should change based on modifications to the mouse button or keyboard state (line **217**), specification of the type of modification intended (line **218**), specification of the priority of intended modification (line **219**), specification that the modifications will occur as a result of the transfer of a series of data files (line **220**), the URL of a file which specifies the display of a satellite image that tracks the movement of the cursor image (line **221–223**), and location of additional display instructions (line **224**). It is noted that the invention is not limited in scope in this respect and other features may be included in the Cursor Display Instructions data.

One embodiment of this method in accordance with the present invention is set forth in greater detail in the flowchart illustrated in FIG. **3**. This embodiment is discussed with reference to the use of ActiveX® technology currently promoted by the Microsoft Corp. The ActiveX® technology provides a mechanism for defining the format of Cursor Display Instructions **56**, for defining, identifying, and in some instances dynamically retrieving Cursor Display Code **52**, and for implementing the interaction between Cursor Display Instructions **56** and the Cursor Display Code **52** as previously described. Although the flowchart in FIG. **3** is discussed with reference to ActiveX® technology, the invention is not limited in this respect, and other technologies for use with browser extensions or "plug-ins" may be utilized in accordance with various embodiments of the present invention as illustrated in FIG. **3**. Furthermore, additional embodiments in accordance with the principles of the present invention may be incorporated within other application software employed in the user terminal. For example, the operating system or the browser itself may be configured to incorporate the mechanism for receiving and recognizing the Cursor Display Instructions and in return provide additional instructions for changing the image or appearance of the cursor display.

With reference to FIG. **3**, in step **102**, browser **32** of user terminal **14** retrieves an HTML file containing Cursor Display Instructions **56**. The HTML file is retrieved when the user directs browser **32** to a remote WWW server site (such as, for example server **48** as indicated in FIG. **2**) by specifying the uniform resource locator, URL, of the site on the Internet where the HTML file is located. When the HTML file is retrieved, it is loaded from the remote WWW server site at which point browser **32** of user terminal **14** begins its routine parsing of the HTML document and eventually encounters a reference to an ActiveX® control or some other information coded in an appropriate programming language such as Sun Microsystem Inc.'s Java® or VBScript®, which is embedded in the Cursor Display Instructions **56** within the HTML document. The Cursor Display Code is capable of interacting with the application programming interface **30** of operating system **28** for the purpose of performing the change, transformation or "swap" of cursor **44** as it is presently displayed on video monitor **24**.

Upon encountering Cursor Display Instructions **56**, browser **32** recognizes Cursor Display Instructions **56** as a request to invoke the particular ActiveX® control with a particular argument list or set of parameters as illustrated in FIG. **4**. At step **104**, browser **32** examines Cursor Display Instructions **56** and uses a unique class identification within the Cursor Display Instructions **56** to determine whether Cursor Display Code **52** (ActiveX® control) is already resident within local memory **20** of user computer **14**.

If the Cursor Display Code **52** is not resident in local memory **14**, generally in the form of a browser extension or plug-in **34**, or if local memory contains an obsolete version of Cursor Display Code **52**, browser **32** attempts, at step **106**, to retrieve the ActiveX® control from a remote server on the Internet and store the Cursor Display Code in local memory **20** of user terminal **14** at step **108**. With reference to FIG. **4**, these steps correspond to lines **202–205**.

Cursor Display Code **52** retrieved in step **106** may be client-platform specific and may also be browser specific such that browser **32** may transmit specific details to the remote server so that the remote server can deliver the appropriate Cursor Display Code **52**.

In accordance with another embodiment of the invention, browser extension or plug-in **34** may be configured such that it can recognize Cursor Display Instructions based on any

one of the available technologies, such as Active X, JavaBeans, JavaScript or VBScript.

Furthermore, it is understood that data compression techniques may be used in order to reduce the amount of network traffic involved in the transmission of data over the Internet.

After Cursor Display Code **52** has been recognized by user terminal **14** as at step **104** or retrieved and loaded therein at steps **106** and **108**, operating system **28** is queried to determine the current cursor display configuration and this information is temporarily cached in local memory **20** of user terminal **14** at step **110** so that the cursor configuration may eventually be restored to its original state. Before any changes are made to cursor **44**, the system at step **111** determines whether server **48** is authorized to change cursor **44**. If authorization is not confirmed, no changes to cursor **44** transpire.

Step **112** is the first step which is executed from within the code of the ActiveX® control. At step **112**, the ActiveX® control determines whether the image specified (Cursor Information **54**) in the ActiveX® argument list which is to become the new cursor image exists in local memory **20** of user terminal **14**. If the specified image in the ActiveX® argument list exists in local memory **20**, it is retrieved therefrom at step **114**. An additional argument in the ActiveX® argument list (line **207**) identifies the location of this data on a remote server. If the specified image does not exist in local memory **20**, this data is utilized by the ActiveX® control to retrieve Cursor Information **54** at step **116** from the specified location.

At step **118**, an additional argument added within the ActiveX® control can be used to determine whether and for how long Cursor Information **54** should be cached in local memory **20**. At step **120** Cursor Information **54** is cached in local memory **20**. At step **122**, the cursor is caused to change in the manner consistent with the retrieved Cursor Display Instructions **56**. In an alternative embodiment, an additional step may be included which provides the user with the option of saving and storing the retrieved Cursor Information **54** in the computer's permanent memory on hard drive **21** even after the retrieved cursor is displayed. Storing the retrieved Cursor Information **54** in the computer's permanent memory saves time on the next occasion when the user loads a web page which requires the same cursor since the cursor is already stored within the computer's memory and need not be retrieved from a remote server.

Cursor Display Instructions **56** cause the invocation of an operating system function which causes the cursor to be displayed on video monitor **24**. More specifically, the ActiveX® control invokes the application programming interface **30** of operating system **28** which causes the cursor image displayed on video monitor **24** to change to the form intended as recited in the argument list. The changed cursor is not limited to image, and may also include animation as well as sound. It should also be appreciated that most computers utilize a multitude of cursor images depending upon the application and task which is being run on the computer. The invention is not limited to changing only a single cursor image and any and all cursor images controlled by the computer's display driver **36** may be caused to change.

At step **124** the ActiveX® control may send usage information to a particular remote server as coded in Cursor Display Instruction **56** or Cursor Display Code **52**. This information can be used to calculate the usage statistics of particular cursor images or cursor information and the context in which they are retrieved and viewed by users. In this particular embodiment, this information is conveyed as a data file transmitted to the remote server via HTTP. The invention is not, however, limited in the type of information and/or statistics which may be transmitted to the server, nor is the invention limited to being conveyed via HTTP as those skilled in the art will understand that such information may be conveyed via other transfer protocols. With reference to FIG. **4**, this step corresponds to line **208**. Additionally, the information may contain an identifying code for the server which issued the web page which contained the Cursor Display Instructions. This information could be used, for example, to verify that the issuing server has been granted the appropriate license to use the technology, by comparing a list of authorized servers or through digital signature validation.

In accordance with one embodiment of the present invention, the licensing arrangement is described in more detail, hereinafter. It is noted that licensing enforcement of the cursor display technology could be accomplished in several ways, and the invention is not limited in scope in that respect. As discussed previously, the server that transmits a web page may include the identity of the server in the form of a server ID within the Cursor Display Instructions. The user terminal then transmits the server ID to another server that among other things functions as a licensing body ("Licensing Body") so as to authenticate the server that transmits the web page as a valid licensee. Should this authentication fail, the execution of Cursor Display Instructions may not occur. In an alternative implementation, the execution of Cursor Display Instructions may be allowed to execute even if the issuer fails authentication. Such an infraction could be logged by the Licensing Body for use in enforcement through traditional channels. For performance reasons it may be desirable to collect the usage information for a plurality of Cursor Display Instructions as the user accesses multiple servers, and transmit the collection of information in batch form to the Licensing Body.

An alternative embodiment would involve the inclusion of an encrypted authentication code within the Cursor Display Instructions, as illustrated in line **211** of FIG. **4**, or via a separate exchange of data between the client and server. In order to ensure that this code could not be re-used by other, non-authorized sites, it could for example be derived from the server's IP address, the date and time at which it is generated, the argument list, or some other information that is accessible to the client. Another possibility would involve the transmission of a unique or pseudo-unique code, from the client to the server. Upon receipt of this authentication code, the client would perform a decryption and verify its authenticity. Under such circumstances, the server software could be augmented with an Authentication Code Module supplied by the Licensing Body which generates and encrypts this code. The mechanism by which this augmentation could occur is similar to that discussed previously in the context of extending the client browser. For example, the server software could be modified and statically linked to the Authentication Code. Alternatively, it could be dynamically linked at run-time. Another alternative would be to implement the Authentication Code as its own process on the server and facilitate an inter-process communication protocol such as the Common Gateway Interface ("CGI").

At step **126**, an ActiveX® control argument is used to determine whether the changed cursor should revert to its initial configuration. If it is intended to revert the changed cursor to its initial configuration, the reversion is paused at step **128** for a specified time period. After it is determined at step **130** that the specified time period has lapsed, the changed cursor reverts to its original configuration at step **132**.

Whether the cursor is caused to revert to its initial configuration is of concern to many users so as to ensure that the user's computer configuration is not permanently altered as a result of the process of changing the cursor. As such, additional alternative measures may be added into Cursor Display Instructions **54** such that the changed cursor could be restored to its original configuration when the ActiveX® control is loaded or unloaded, when the computer starts up, is rebooted or is shut down, when the browser is activated or shut down, when an animated cursor completes its animation sequence, when instructed by a remote server, or as a result of some user input such as setting an option in the browser or accessing another web page or site. An alternative to adding parameters to the Cursor Display Instructions would be to control the process of changing the cursor to its initial state by a control program downloaded by and executed on the client computer. An example written in VBScript and interacting with an ActiveX control is included in FIG. **5**.

Additionally, one of the significant attributes of this embodiment is the manner in which Cursor Display Code **52** is retrieved from a remote server if it is not located in the computer's local memory. Since Cursor Display Code **52** may be operating system or browser specific, it may be necessary that the server with which the user computer **14** is communicating be informed by user terminal **14** of the specific type of Cursor Display Code **52** which is desired. In another embodiment of the invention, browser extension or plug-in **34** may be configured such that it can recognize Cursor Display Instructions based on any available technology such as Active X and JavaScript.

The operation of steps **102**–**132** as set forth in FIG. **3**, may be illustrated pictorially in FIGS. **7**–**9**. FIG. **7** illustrates an example of a typical web page **60** as it would appear on a user's video monitor **24** having the standard arrow cursor **44**. In FIG. **8**, there is illustrated a different web page **60***a* having a banner advertisement **62** for Fizzy Cola which contains Cursor Display Instructions. When web page **60***a* loads, the Cursor Display Instructions cause arrow cursor **44** to change into a Fizzy cola bottle shaped cursor **44***a* in conjunction with the Fizzy Cola banner advertisement. As illustrated in FIG. **9**, if the user then loads a new web page **60***b* which is not provided with Cursor Display Instructions, the cola bottle shaped cursor of FIG. **8**, reverts to the standard arrow cursor **44**.

It is also understood that ActiveX® is but one of numerous technologies utilized over the Internet with which a user's computer may interact in bringing about the change or transformation of the cursor displayed on video monitor **24**. Other implementations may utilize different technologies such as Windows dynamic link libraries, VBScript and JScript from Microsoft, as well as Java, JavaScript and JavaBeans from Sun Microsystems Inc.. While these examples represent the dominant standards-based definitions, proprietary implementations could also be developed. Accordingly, while ActiveX® represents one embodiment of distributing and invoking Cursor Display Information **54** on a user's computer **14**, it is to be appreciated that there are a variety of alternative implementations, and this particular implementation should not be considered a limitation of the invention. For example, alternative versions of browser **32** may encapsulate the appropriate operating system application programming interface call within their own code modules such that a browser extension **34** is not required.

In yet another embodiment of the invention the tasks described in steps **102** through **132** may be employed cooperatively between browser and browser extension or plug-in **34**. Furthermore, browser **32** may employ a computational or processing engine such as an interpreter (as is the case with the Java® programming language, for example) which can extend the capabilities of browser **32** to a virtually unlimited degree.

It is also to be understood that in the course of carrying out the process of changing the cursor as discussed hereinabove, user terminal **14** may communicate with a multitude of remote servers as opposed to just a single server. For example, Cursor Display Codes may be retrieved from one remote server, Cursor Instructions may be retrieved from a second remote server, and the user terminal **14** may also be in communication with a third server to which it is transmitting the usage statistics.

Features identified in reference with FIG. **4** are described in more detail hereinafter. It is noted that in accordance with one embodiment of the invention, it may be desirable to modify the Cursor Display Code to improve its performance or enhance its capabilities. The server may transmit version information in the Cursor Display Instructions as illustrated in line **205** of FIG. **4**. The Cursor Display Code could compare this information with its own version information in order to determine whether it has been rendered obsolete by a more recent version. If so, the Cursor Display Code could retrieve the current version from a remote server and invoke execution on the new version.

In an alternative embodiment of the present invention the position, as well as the image, of the user terminal's cursor may be controlled by a remote server. This embodiment would be implemented within the Cursor Display Code **52** such that additional information could be passed to Cursor Display Code **52** via Cursor Display Instructions **56**. The additional information passed to Cursor Display Code **52** would contain code which indicates: (1) that the cursor position control is intended, (2) the conditions under which the cursor should be moved, and (3) the source of the data which specifies the particular movement that is intended. The latter could be stored in memory on a remote server and retrieved in a manner similar to retrieving Cursor Display Instructions **56** or the Cursor Display Code **52**. For example, if no user input is received for a specified interval, the cursor image could change and the position of the cursor could be set such that it follows a specified trajectory for several seconds, then reverts to its original state as illustrated by line **213** of FIG. **4**.

In accordance with another embodiment of the invention it is possible to vary the modification to the cursor as a function of cursor position. For example, the cursor pointer could be controlled such that it "points" to a specific location on the screen regardless of the cursor's location on the screen as illustrated in line **214** of FIG. **4**.

In accordance with another embodiment of the invention it is possible to vary the modification to the cursor as a function of cursor velocity. For example, the cursor image could change from a stationary bird to a bird with flapping wings only when the cursor is moved quickly across the screen as illustrated in line **215** of FIG. **4**. Furthermore, it is possible to vary the modification to the system-level user interface attributes as a function of mouse button state or keyboard state. For example, the image of a cube could be replaced with that of a jack-in-the-box when the mouse button is depressed.

In accordance with another embodiment of the invention, it is possible to modify other "system-level" attributes of the client computer's user interface, hereafter called "system-

level user interface attributes". These attributes, as illustrated in FIG. **6** are typically under the control of the operating system and, as such, they exist independently of the user "applications" (programs) and data which are stored on the computer and interact with that operating system. User applications interact with the operating system to deliver the computer's functionality to the user. Examples of user applications include word-processors, spreadsheets, web browsers, games, etc. The operating system may contribute certain user interface elements to the user interface of the applications running on it.

Because many of these attributes are inherited from the operating system by all applications running on that operating system, applications tend to exhibit a degree of commonality in their user interfaces. Examples of these attributes include: the shape and color of the cursor **401**, the shape and color of a status bar which displays current state information to the user **403**, the shape and color of the scroll bar which indicates the relative position and scope of the displayed sub-image to that of the underlying larger image to the user **407**, the shape and color of the title bar which displays current state information **409**, the shape and color of icons representing standard window operations such as close, minimize display size, restore display size, etc. **411**. Thus, these system level attributes may also be modified in response to Cursor Display Instructions data.

In addition, the operating system itself may have a user interface. Examples include: the images and sounds displayed when the computer starts or shuts down, the background image ("wallpaper") against which other graphical elements are displayed **413**, file catalogs and file selection mechanisms **415**, system icons **416**, file invocation mechanisms **417**, buttons **419**, process selection mechanisms **421**, etc. Further examples include the icons representing various system elements or information such as files **418**, groups of files **420**, files marked for deletion **422**, as well as standard, information bearing "dialog boxes", such as cancel, warning, illegal operation, stop, accept, continue, etc. **423**. The system may also support a set of audibly distinct waveforms which may be used to convey similar information to the user. These operating system user interfaces may also be modified in response to a Cursor Display Instruction data.

In yet another exemplary embodiment of the present invention a plurality of modifications to the system-level user interface attributes may occur simultaneously. For example, the cursor could animate while an audio waveform is playing, as the minimize display icon changes to a specific image.

A further feature of the invention is to accumulate information regarding the user's exposure time to various system-level user interface attribute modifications, and to vary the exposure to those modifications accordingly. For example, the client could transmit exposure data to the server and the server would select a version of the image based on that data. Furthermore, the exposure data could be transmitted as part of the usage statistics discussed previously.

Another feature of the invention is to monitor the load being placed on the client system by the user and schedule data exchange with the servers so that it occurs when it is least disruptive to the user's activities.

It is also possible to allow the user to control the level of interface modification he or she wishes to entertain. For example, the user could specify that only those modifications of specific types, as illustrated on line **218** of FIG. **4** or of specific priority should be delivered, or even that none be delivered, as illustrated at lines **218** and **219** of FIG. **4**. This specification could be implemented directly by the user on the client system, or could be implemented through communication with a remote server.

In accordance with another embodiment of the invention it is possible to transmit the image and/or audio data which specifies the modification as a series of data files which are delivered in a continuous stream to the client, as illustrated at line **220** of FIG. **4**. These files are exposed to the user before the complete set of data has been delivered, thereby providing the capability for the initiation of long animations or audio files before the entire quantity of data has been received by the client.

A further feature of the invention is to support the display of a "satellite" image which tracks the cursor's position on the screen. For example, the cursor image could be replaced with that of a mouse, and the image of a cat could be displayed near that mouse. When the cursor is moved, the satellite image moves accordingly at a specific offset, as illustrated at lines **221–223** of FIG. **4**.

A further feature of the invention is to provide a mechanism for the user to quickly establish a connection with a specific server based on the specific user interface attribute modification which is in effect when the mechanism is invoked. For example, the user could press a specific key sequence on the keyboard and immediately jump to the web site related to the cursor image which is currently displayed.

In accordance with another embodiment of the invention, it is possible to convey additional detailed Cursor Display Instructions as a separate file which is explicitly retrieved from a server by the Cursor Display Code, as illustrated at line **224** of FIG. **4**.

For each modification to the system-level user interface attributes, an appropriate set of display instructions must be transmitted to the client. These could take the form of additional parameters in the Cursor Display Instructions as discussed previously, or they could be represented within a code module which is received by and executed on the client. As discussed previously, Java, and its related technologies could be used for such a purpose, but use of these technologies should not considered a limitation of the invention.

It is noted that there are numerous ways in which a system-level user interface attribute modification is accomplished in accordance with the principles of the present invention. It is further noted that system level user interface attributes may be modified independently or in conjunction with cursor modification. Furthermore, the system-level user interface attribute modification may be related to specific information displayed on the rest of the user's screen (hereafter referred to as "specific information") in many different ways. Thus, the present invention is not limited in scope to how content providers may relate the system-level user interface attribute with the specific information. Rather, at least one of the goals of the present invention is to enable the content providers to modify the system-level user interface attribute whenever and wherever they see fit. For example, content providers may modify system-level user interface attributes at a remote user's terminal for advertising, entertainment, information delivery, celebrating an event, or other reasons, and therefore, the invention is not limited in scope in that respect. Furthermore, when a content provider elects to display a specified system-level user interface attribute in conjunction with and corresponding to specific information conveyed via the user's terminal, the cursor image and the background display data are deemed related.

Additional examples intended to illustrate some applications of the present invention are explained below, although the invention is not limited in scope to any one of these examples.

Thus, in accordance with one embodiment of the invention, a modified cursor might take the appearance of a "Fizzy Cola" bottle when a "Fizzy Cola" banner advertisement appears among the display data of a popular search engine's site. Similarly, the cursor can be modified for advertising purposes to represent Fizzy Cola's logo, its corporate mascot, images of its products or services, slogans, icons, brand images, advertising messages (the word "Thirsty?", for example), abstract suggestions (such as a straw or glass), etc.

Alternatively, Fizzy Cola, on its own site, or homepage, might have a picture of a bottle of Fizzy in the middle of the page (in the display data). A dynamic cursor image could then be used to show a person holding a straw in such a way that the straw always points from the user toward the top of the Fizzy bottle, no matter where the cursor moves on the screen. The straw, in this case, might be "attached" to the cursor image (part of the same image) or could be separate, "satellite" image, a "sprite," whose movement on the screen (in this case) is related to the movement of the cursor. Sprites, which can appear and disappear as desired, can enhance the invention by enabling the use of graphical elements which are associated with the cursor but which reside outside the limited cursor "space" (which in some systems may be, at maximum, 32 by 32 pixels). For the purposes of the invention, however, there should be no limitation to the size of the cursor.

Additional examples of modification to the cursor include rendering the cursor as a baseball bat (on a site with sports information), a pink but otherwise standard-shaped pointer (on a site about the Pink Panther), a witch-on-a-stick to celebrate Halloween, the Statue of Liberty to celebrate the Fourth of July, etc. All of the foregoing cursor images could be enhanced with related animations, such as the bat hitting the ball.

Similarly, the present invention can be used to replace not just the standard arrow but other standard cursors as well, such as the generic hand with pointing index finger (the icon commonly used in browsers to indicate that the pointer is positioned above a hot link). A site for children might, for example, replace this generic pointing-hand cursor with the pointing "paw" of a furry animal. A site dealing with horror movies might choose to replace this pointing hand with a bony skeleton-like hand.

Additional examples involve cursors with text or numbers. For example, the cursor might contain the text "Right-Click Now!" prompting users to click the right button of their mouse (where right-clicking on the mouse could, for example, trigger the delivery of a new page of display data). It may also be desirable in certain cases to put alphanumeric data in the cursor "space" to convey information to users, such as stock prices, baseball game scores, the temperature in Florida, etc. The data can be static, semi-static (i.e. updated periodically), or dynamic (updated frequently—possibly incorporating available streaming-data and data-compression technologies).

Use of associated sound, sprites, animations, and modified system display elements are provided as enhancements to the basic invention. For example, a Fizzy cola mascot could appear in the cursor space in conjunction with the speakers, attached to the user's machine, playing the sound of the mascot saying, "drink Fizzy!" Any time a content provider elects to incorporate said enhancements in conjunction with a new modified cursor image, the cursor image and said enhancements have been deemed related.

The present invention allows users to change cursor images; it also allows them to change them back. It may be desirable to revert the pointer to a previous or generic pointer image. Given the Fizzy Cola example above, if the page containing display data changes and there is no longer an advertisement for Fizzy, but rather an advertisement for its rival, Jazzy Cola, it may be desirable to ensure the removal of the Fizzy cursor image(s) and accompanying enhancements.

The foregoing examples are not intended to suggest limited uses for this invention; to the contrary, the examples are intended to illustrate the wide range of uses for this invention. The collective creativity of the online advertising, art, design, commerce, content publishing, and related industries will develop many novel and unforeseen ways to use the present invention. The versatility of the present invention should not be regarded as a limitation on its scope.

Thus, while there have been shown and described and pointed out fundamental novel features of the invention as applied to preferred embodiments thereof, it will be understood that various omissions and substitutions and changes in the form and details of the disclosed invention may be made by those skilled in the art without departing from the spirit of the invention. It is the intention, therefore, to be limited only as indicated by the scope of the claims appended hereto.

It is to be understood that the drawings are not necessarily drawn to scale, but that they are merely conceptual in nature.

We claim:

1. A server system for modifying a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said system comprising:

cursor image data corresponding to said specific image;

cursor display code, said cursor display code operable to modify said cursor image; and

a first server computer for transmitting specified content information to said remote user terminal, said specified content information including at least one cursor display instruction indicating a location of said cursor image data, said cursor display instruction and said cursor display code operable to cause said user terminal to display a modified cursor image on said user's display in the shape and appearance of said specific image, wherein said specified content information is transmitted to said remote user terminal by said first server computer responsive to a request from said user terminal for said specified content information, and wherein said specified content information further comprises information to be displayed on said display of said user's terminal, said specific image including content corresponding to at least a portion of said information to be displayed on said display of said user's terminal, and wherein said cursor display code is operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image responsive to displaying of said at least a portion of said information to be displayed on said display of said user's terminal.

2. The server system in accordance with claim **1**, wherein said specific image relates to at least a portion of said information to be displayed on said display of said user's terminal.

3. The server system in accordance with claim 2, wherein said specific image comprises advertising material related to at least a portion of said information to be displayed on said display of said user's terminal.

4. The server system in accordance with claim 3, wherein said advertising material further comprises a brand logo.

5. The server system in accordance with claim 3, wherein said advertising material further comprises a corporate mascot.

6. The server system in accordance with claim 3, wherein said advertising material further comprises images of a good or a service corresponding to said information to be displayed on said display of said user's terminal.

7. The server system in accordance with claim 3, wherein said advertising material further comprises messages relating to said information to be displayed on said display of said user's terminal.

8. The server system in accordance with claim 2, wherein said specific image has a shape and appearance corresponding to said information to be displayed on said display of said user's terminal.

9. The server system in accordance with claim 1, wherein at least a portion of said cursor image data and said cursor display code are disposed locally to said first server computer.

10. The server system in accordance with claim 1, wherein at least a portion of said cursor image data and said cursor display code are disposed within a second server computer located remotely to said first server computer.

11. The server system in accordance with claim 1, wherein at least a portion of said cursor image data and said cursor display code are disposed locally to said user's terminal.

12. The server system in accordance with claim 10 wherein said first server computer in response to a request from said user terminal transmits information stored in said second server computer to said user terminal.

13. The server system in accordance with claim 1, wherein said specified content information is transmitted in the form of HTML files that define a web page.

14. The server system in accordance with claim 13, wherein said cursor image data includes at least in part an advertisement for goods or services contained in said web page.

15. The server system in accordance with claim 1, wherein said user terminal includes a browser application responsive to said cursor display instruction, said browser application executing said cursor display code using parameters defined in said cursor display instruction.

16. The server system in accordance with claim 1, said cursor display instruction further comprising an image identifier indicating said cursor image data corresponding to said specific image.

17. The server system in accordance with claim 1 wherein said first server computer transmits said cursor image data in response to a request received from said remote user terminal indicating that a copy of said cursor image data is not stored in said remote user terminal.

18. The server system in accordance with claim 1 wherein said cursor display instruction further comprises an image identifier that corresponds to a graphic animation sequence.

19. The server system in accordance with claim 18, wherein said cursor display instruction further comprises instructions operable to modify said specific image to display said graphic animation sequence.

20. The server system in accordance with claim 1, wherein said cursor display instruction further comprises an audio identifier that corresponds to an audio information sequence.

21. The server system in accordance with claim 20, wherein said cursor display instruction further comprises instructions operable to play an audio clip corresponding to said audio information sequence.

22. The server system in accordance with claim 1, wherein said cursor display instruction further comprises information that controls a duration of time said specific image is displayed on said display of said remote user's terminal.

23. The server system in accordance with claim 1, wherein said specified content information is transmitted in the form of one or more hypertext objects.

24. The server system in accordance with claim 1, wherein said specified content information includes instructions executable by a virtual machine on said user terminal.

25. The server system in accordance with claim 1, wherein said specified content information includes HTML tags recognized by said cursor display code.

26. The server system in accordance with claim 1, wherein said cursor display code generates usage data for calculating usage statistics of said specific image.

27. The server system in accordance with claim 1, wherein said cursor display instruction transmitted to said remote user terminal initiates communication with a plurality of server systems for obtaining additional cursor image data.

28. A method for modifying an initial cursor image displayed on a display of a user terminal connected to at least one server, comprising:

receiving a request at said at least one server to provide specified content information to said user terminal;

providing said specified content information to said user terminal in response to said request, said specified content information including at least one cursor display instruction and at least one indication of cursor image data corresponding to a specific image; and

transforming said initial cursor image displayed on said display of said user terminal into the shape and appearance of said specific image in response to said cursor display instruction, wherein said specified content information includes information that is to be displayed on said display of said user's terminal, wherein said specific image includes content corresponding to at least a portion of said information to be displayed on said display of said user's terminal, and wherein said cursor display instruction indicates a cursor display code operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image responsive to displaying of said at least a portion of said information that is to be displayed on said display of said user's terminal.

29. The method in accordance with claim 28, wherein said transforming further comprises executing said cursor display code so as to display said specific cursor image while at least a portion of said information to be displayed is displayed on said display of said user's terminal.

30. The method in accordance with claim 28, wherein said displaying of said specific image further comprises displaying advertising material related to at least a portion of said information to be displayed.

31. The method in accordance with claim 30, wherein said advertising material further comprises a brand logo.

32. The method in accordance with claim 30, wherein said advertising material further comprises a corporate mascot.

33. The method in accordance with claim 30, wherein said advertising material further comprises images of a good or a service corresponding to said information to be displayed.

34. The method in accordance with claim 30, wherein said advertising material further comprises messages relating to said information to be displayed.

35. The method in accordance with claim 28, wherein said specific image has a shape and appearance relating to said information to be displayed.

36. The method in accordance with claim 28, wherein said specified content information further comprises HTML files that define a web page.

37. The method in accordance with claim 36, wherein said cursor image data corresponds to an advertisement for goods or services contained in said web page.

38. The method in accordance with claim 28, wherein said transforming further comprises:

employing a browser application including said cursor display code responsive to said cursor display instruction; and

executing said cursor display code by employing parameters defined in said cursor display instruction.

39. The method in accordance with claim 28 wherein said specified content information comprises an image identifier that corresponds to the location of data representing said specific image.

40. The method in accordance with claim 28, further comprising transmitting said cursor image data in response to a request received from said remote user terminal indicating that a copy of said cursor image data is not stored in said remote user terminal.

41. The method in accordance with claim 28, wherein said cursor display instructions further comprises an image identifier that corresponds to a graphic animation sequence.

42. The method in accordance with claim 41, further comprising modifying said remote user terminal's cursor to display said graphic animation sequence.

43. The server system in accordance with claim 28, wherein said cursor display instruction further comprises an audio identifier that corresponds to an audio information sequence.

44. The server system in accordance with claim 43, further comprising playing an audio clip corresponding to said audio information sequence responsive to displaying said specific image.

45. The method in accordance with claim 28 further comprising controlling a duration of time said specific image is displayed on said remote user's display.

46. The method in accordance with claim 28, further comprising providing usage data by said cursor display code for calculating usage statistics of said specific image, responsive to said cursor display instruction.

47. A computer storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform the method steps in claim 28.

48. An internet browser computer program stored on a computer readable medium, said internet browser configured to modify a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said internet browser comprising:

program code operable to receive specified content information from a remote server, said specified content information comprising information to be displayed on said remote user's terminal and at least one cursor display instruction, wherein said specific image includes content corresponding to at least a portion of said information to be displayed on said remote user's terminal, wherein said cursor display instruction indicates cursor image data corresponding to said specific image;

program code operable to recognize said cursor display instruction in connection with processing said information to be displayed on said display; and

program code operable to execute a cursor display code, responsive to said cursor display instruction and displaying of said at least a portion of said information to be displayed on said display, said cursor display code being operable to modify said cursor image to said specific image.

49. The internet browser in accordance with claim 48 further comprising program code operable to retrieve said cursor image data from a prespecified server, when said cursor image data is not stored in said user terminal.

50. The internet browser in accordance with claim 48 further comprising program code operable to retrieve said cursor display code from a prespecified server, when said cursor display code is not stored in said user terminal.

51. The internet browser in accordance with claim 48 further comprising program code operable to determine whether cursor display instructions received by said user terminal were transmitted by an authorized server.

52. The internet browser in accordance with claim 48 further comprising program code operable to transmit statistical information to a prespecified server so as to provide information relating to the usage of said specific image.

53. The internet browser in accordance with claim 48, wherein said specific image reverts back to its original shape and appearance after a prespecified duration.

54. The internet browser in accordance with claim 53 further comprising program code operable to store said cursor display code in a memory located locally to said user terminal.

55. The internet browser in accordance with claim 48 further comprising program code operable to provide audio clips corresponding to said display of said specific image.

56. The internet browser in accordance with claim 55 wherein information relating to said audio clips is contained within said cursor display instruction.

57. The internet browser in accordance with claim 48, further comprising program code operable to provide animated images corresponding to said specific image.

58. The internet browser in accordance with claim 57 wherein information relating to said animated images are contained within said cursor display instruction.

59. The internet browser in accordance with claim 48, wherein said specific image corresponds to at least a portion of said information to be displayed on said display of said user's terminal.

60. The internet browser in accordance with claim 48, wherein said specific image comprises advertising material related to at least a portion of said information to be displayed on said display of said user's terminal.

61. The internet browser in accordance with claim 60, wherein said advertising material further comprises a brand logo.

62. The internet browser in accordance with claim 60, wherein said advertising material further comprises a corporate mascot.

63. The internet browser in accordance with claim 60, wherein said advertising material further comprises images of a good or a service corresponding to said information to be displayed on said display of said user's terminal.

64. The internet browser in accordance with claim 60, wherein said advertising material further comprises messages relating to said information to be displayed on said display of said user's terminal.

65. The internet browser in accordance with claim 48, wherein said specific image has a shape and appearance

related to said information to be displayed on said display of said display of said user's terminal.

66. The internet browser in accordance with claim **48**, wherein said specified content information is transmitted in the form of at least one HTML file that defines a web page.

67. The internet browser in accordance with claim **48**, wherein said specified content information is transmitted in the form of one or more hypertext objects.

68. The internet browser in accordance with claim **48**, wherein said specified information content is executable at least in part by a virtual machine on said user terminal.

69. The internet browser in accordance with claim **48**, wherein said specified information content includes at least one instruction in an interpreted programming language.

70. A server system for modifying a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said system comprising:

cursor image data corresponding to said specific image;

cursor display code, said cursor display code operably to modify said cursor image; and

a first server computer for transmitting specified content information to said remote user terminal, said specified content information including at least one cursor display instruction indicating a location of said cursor image data, said cursor display instruction and said cursor display code operable to cause said user terminal to display a modified cursor image on said user's display in the shape and appearance of said specific image, wherein said specified content information is transmitted to said remote user terminal by said first server computer responsive to a request from said user terminal for said specified content information, and wherein said specified content information further comprises information to be displayed on said display of said user's terminal, said specific image including content corresponding to at least a portion of said information to be displayed on said display of said user's terminal, and wherein said cursor display code is operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image responsive to movement of said cursor image over a display of said at least a portion of said information to be displayed on said display of said user's terminal.

71. A server system for modifying a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said system comprising:

cursor image data corresponding to said specific image;

cursor display code, said cursor display code operable to modify said cursor image; and

a first server computer for transmitting specified content information to said remote user terminal, said specified content information including at least one cursor display instruction indicating a location of said cursor image data, said cursor display instruction and said cursor display code operable to cause said user terminal to display a modified cursor image on said user's display in the shape and appearance of said specific image, wherein said specified content information is transmitted to said remote user terminal by said first server computer responsive to a request from said user terminal for said specified content information, and wherein said specified content information further comprises information to be displayed on said display of said user's terminal, said specific image including content corresponding to at least a portion of said information to be displayed on said display of said user's terminal, and wherein said cursor display code is operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image responsive to movement of said cursor image over a specified location on said display of said user's terminal.

72. A method for modifying an initial cursor image displayed on a display of a user terminal connected to at least one server, comprising:

receiving a request at said at least one server to provide specified content information to said user terminal;

providing said specified content information to said user terminal in response to said request, said specified content information including at least one cursor display instruction and at least one indication of cursor image data corresponding to a specific image; and

transforming said initial cursor image displayed on said display of said user terminal into the shape and appearance of said specific image in response to said cursor display instruction, wherein said specified content information includes information that is to be displayed on said display of said user's terminal, wherein said specific image includes content corresponding to at least a portion of said information that is to be displayed on said display of said user's terminal, and wherein said cursor display instruction indicates a cursor display code operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image responsive to movement of said cursor image over a display of said at least a portion of said information to be displayed on said display of said user's terminal.

73. A method for modifying an initial cursor image displayed on a display of a user terminal connected to at least one server, comprising:

receiving a request at said at least one server to provide specified content information to said user terminal;

providing said specified content information to said user terminal in response to said request, said specified content information including at least one cursor display instruction and at least one indication of cursor image data corresponding to a specific image; and

transforming said initial cursor image displayed on said display of said user terminal into the shape and appearance of said specific image in response to said cursor display instruction, wherein said specified content information includes information that is to be displayed on said display of said user's terminal, wherein said specific image includes content corresponding to at least a portion of said information that is to be displayed on said display of said user's terminal, and wherein said cursor display instruction indicates a cursor display code operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image responsive to movement of said cursor image over a specified location on said display of said user's terminal.

74. An internet browser computer program stored on a computer readable medium, said internet browser configured to modify a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said internet browser comprising:

program code operable to receive specified content information from a remote server, said specified content information comprising information to be displayed on said remote user's terminal and at least one cursor display instruction, wherein said specific image includes content corresponding to at least a portion of said information to be displayed on said remote user's terminal, and wherein said cursor display instruction indicates cursor image data corresponding to said specific image;

program code operable to recognize said cursor display instruction in connection with processing said information to be displayed on said display; and

program code operable to execute a cursor display code, responsive to said cursor display instruction and movement of said cursor image over a display of said at least a portion of said information to on said remote user's terminal, said cursor display code being operable to modify said cursor image to said specific image.

75. An internet browser computer program stored on a computer readable medium, said internet browser configures to modify a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said internet browser comprising:

program code operable to receive specified content information from a remote server, said specified content information comprising information to be displayed on said remote user's terminal and at least one cursor display instruction, wherein said specific image includes content corresponding to at least a portion of said information to be displayed on said remote user's terminal, and wherein said cursor display instruction indicates cursor image data corresponding to said specific image;

program code operable to recognize said cursor display instruction in connection with processing said information to be displayed on said display; and

program code operable to execute a cursor display code, responsive to said cursor display instruction and movement of said cursor image over a specified location on said display, said cursor display code being operable to modify said cursor image to said specific image.

* * * * *

59391 U.S. PTO
08/882580
06/25/97

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

PTO-1556
(5/87)

71531 U.S. PTO
06/25/97

59391 U.S. PTO
08/882580
06/25/97

A/NO FEE

SOFER & HAROUN, L.L.P.
PATENT FILING TRANSMITTAL

ATTORNEY DOCKET NO: 658-002

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Box Patent Application
Commissioner of Patents and Trademarks
Washington, D.C. 20231

PATENT FILING TRANSMITTAL

Transmitted herewith for filing is the Patent Application of: Mark S. Hall, James S. Rosen, Thomas A. Schmitter

For: SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE

TYPE OF FILING

This new patent application is for a(n):

[X] Utility
[ ] Design
[ ] Plant
[ ] Divisional
[ ] Continuation
[ ] Continuation-in-part

Benefit of a prior filed application

☐ This is a continuation-in-part of copending application Serial No. _____ filed on _____ claims the benefit of an earlier filed U.S Patent Application under 35 USC 120.

☐ Please accord Applicant the benefit of the priority date of_____ to this case pursuant to 35 USC 119. Applicant's claim for priority is based on application _____ filed in _____ on that date.

Filing under 37 CFR 1.53 (Utility) or 37 CFR 1.153 (Design)

☐ This is an application filed pursuant to 37 CFR 1.53 or 37 CFR 1.153, permitting receipt of a filing date upon filing of a specification, at least one claim and necessary drawings.

☒ In the event any parts of this application are incomplete, please treat this as a filing under 37 CFR 1.53 or 37 CFR 1.153.

ENCLOSURES

Minimum requirements under 37 CFR 1.53(b) (Utility) or 37 CFR 1.153 (Design)

☒ 36 pages of specification;
☒ 15 pages of claims;
☒ 1 pages of abstract;

CERTIFICATE OF MAILING (37 CFR 1.10(a))

CERTIFICATE OF MAILING BY "EXPRESS MAIL" - Rule 10: I hereby certify that this correspondence is being deposited with the U. S. Postal Service "Express Mail Post Office to Addressee" Service under 37 CFR 1.10 as Express Mail No. _____addressed to the Commissioner of Patents & Trademarks, Washington, D.C. 20231 on _____ by_____.

Date:_____ Signed:_____

"Express Mail" Mailing Label No. EM 055 880 585 US

☒ _9_ sheets of informal drawings;
☒ A Declaration, Power of Attorney & Petition or listing of inventors;
and
☒ A postcard for return to us as proof of receipt of the above documents.

plus
☒ An Assignment of the invention to Comet Systems, Inc. and an Assignment cover sheet;
☐ Verified Statement Claiming Small Entity Status (37 CFR 1 9(f) and 1.27(b))
☐ Form PTO-1449 (IDS) and copies of the references listed thereon;
☐ A certified copy of _____(country) patent application number (priority document).
☐ A preliminary amendment;
☐ Declaration of Biological Deposit;
☐ Submission of sequence listing, computer readable copy and/or amendment relating thereto for biotechnology invention containing nucleotide and/or amino acid sequence;
☐ An associate power of attorney;
☐ Other.

DECLARATION OR OATH

The enclosed Declaration or Oath has been executed by:
☐ Inventor(s);
☐ Legal representative of the inventors (37 CFR 1.42 or 1 43);
☐ Joint inventor or person showing proprietary interest on behalf of an inventor who refused to sign or who cannot be reached and this is a petition required by 37 CFR 1.47 and the statement required by 37 CFR 1.47 is attached;
☒ Has not been executed and is enclosed for the purposes of identifying the inventors

INVENTORSHIP STATEMENT

The inventorship for all the claims in this application is:
☒ the same;
☐ not the same and, as an explanation, a statement is/ will be submitted

LANGUAGE

The application submitted herewith is:
☒ in English;
☐ in not in English and in terms of 37 CFR 1.52(d) a verified translation is
  ☐ attached
  ☐ not attached.

SOFER & HAROUN, L.L.P.
PATENT FILING TRANSMITTAL

ATTORNEY DOCKET NO: 658-002

FEE CALCULATION

The filing fee has been calculated as shown below:

	SMALL ENTITY	OR	OTHER THAN A SMALL ENTITY
BASIC FEE Design Patent	$160		$320
BASIC FEE Utility Patent	$385		$770
EXTRA FEES	RATE FEE		RATE FEE
TOTAL CLAIMS 71 MINUS 20 =	51	x 11 = $	x 22 = $
INDEP.CLAIMS 3 MINUS 3 =	0	x 40 = $	x 80 = $
☐ MULTIPLE DEP.CLAIM		+130 = $	+260 = $
☒ ASSIGNMENT		+ 40= $	+ 40 = $
☐ RULE 53 SURCHARGE		+ 65= $	+130 = $
TOTAL		$ 986.00	$ 0

FEE PAYMENT

☐ Attached is Check No. ___ in the sum of $____ to cover the filing and, if applicable, the assignment fee.

☐ Please charge Account No. 19-2825 the sum of $_____.

FEE DEFICIENCY

☐ The Commissioner is authorized to charge (or credit any overpayment) to deposit account No. 19-2825:

- ☐ Any additional filing fees required under 37 CFR 1.16, except Rule 53 filings, which will be paid within the time permitted by PTOL 1533.
- ☐ Assignment Recordal fees.

☐ The filing fee and surcharge under 37 CFR 1.16, patent application processing fees under 37 CFR 1.17 and patent issue fees under 37 CFR 1.18 are intended to be paid by our firm as they arise. As no abandonment is intended by any inadvertent nonpayment of fees, the Commissioner is hereby authorized to charge payment of such fees as from time to time come due, if not paid prior to due date to our Deposit Account No. 19-2825 .

☐ A duplicate copy of this sheet is enclosed.

Respectfully submitted,

Dated: 6/25/97

Joseph Sofer
Reg. No., 34,438

SOFER & HAROUN, L.L.P.

Address:342 Madison Avenue, Suite 1921
New York, New York 10173
Telephone:(212)697-2800
Facsimile:(212)697-3004

PATENT Docket No. 658-002

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s) : Mark S. Hall, James S. Rosen, Thomas A. Schmitter
Serial No. : Not yet assigned Examiner: Not yet assigned
Filed : June 25, 1997 Group Art Unit: Not yet assigned
For : SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE

EXPRESS MAIL CERTIFICATE

Express Mail Label No. EM074816357US

Date of Deposit June 25, 1997

I hereby certify that the following attached paper(s) or fee

1) Patent Application
- 36 pgs. of specification; 15 pgs. of claims
2) Figs. 1-9
3) Patent Transmittal
4) Oath and Declaration - unexecuted
5) Assignment
6) Assignment Recordation
7) Check for $40
8) Express Mail Certificate
9) Return Postcard

is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. §1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Stacey Taitt
(Typed or printed name of person mailing paper(s) or fee)

(Signature of person mailing paper(s) or fee)

Mailing Address:
SOFER & HAROUN, LLP
342 Madison Avenue
Suite 1921
New York, New York 10173
Tel:(212)697-2800;fax (212)697-3004

EXPRESS MAIL NO. EM074816357US

APPLICATION FOR

UNITED STATES LETTERS PATENT

7552580.doc

SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE

Inventors:

James Rosen
Thomas Schmitter
Mark Hall

Field Of The Invention

This invention relates to computer networks and software, and more particularly, to a server system capable of modifying a cursor image displayed on a remote client computer.

Background Of The Invention

The World Wide Web ("WWW" or "web") and online services such as America Online, in conjunction with faster and more powerful personal computers, have rendered the Internet and other interactive online computer networks accessible to millions of people all over the world. Concomitant with the emergence of this new communication medium, digital content providers have proliferated, providing online news, entertainment, games and all sorts of other content. As with other mass mediums, such as television, radio, and print publications, the entities that create such content seek to offset their expenses by selling advertising. With reference to the WWW, online advertising has become a multimillion dollar business, to the amount of approximately $300 million dollars in 1996.

The most common type of online advertisement exists in the form of "banner advertisements". Users of online services routinely encounter banner ads on the top, sides, and/or bottom of their video monitor screens when viewing a web page. Banner ads are generally square or rectangular boxes provided with some combination of graphics, color and text directed to the product or service being advertised. As such, the intention of these banner advertisements is to create impressions among online users and to convey some advertising message and/or logo. Banner ads are usually provided on a web page in the form of a "hyperlink", in which users who

Express Mail No. EV339905025US

yield to the advertisement's solicitation to "Click Here" are transported to the web site of the manufacturer of the product or service being advertised, or to some other screen which provides additional information about the product or service.

Unfortunately, banner ads occupy only a small portion of a web page. As the user scrolls down a page the banner ad disappears. Although online advertisers and content publishers have attempted to optimize the visibility of banner advertisements by placing them on a popular web page where they will have a greater chance of being seen, Internet users, nevertheless, can easily ignore or find ways to remove and eliminate from their view the banner ads which exist on the web pages they are viewing. As such, the banner ads are rendered ineffective in their aim to provide information about a product or service. Additionally, money spent to advertise a product may be wasted if users are able to ignore or remove the advertisements from the web pages they are viewing.

Another method of online advertising involves the use of "frames" on a web page. Frames are a feature supported by the recent versions of leading web navigating programs known as browsers, such as Netscape Navigator® and Microsoft's Internet Explorer®. Frames generally divide up a user's screen so that the user can, for example, independently scroll down each of numerous frames which appear on the web page being viewed on the user's screen. Like banner advertisements, frames can be aesthetically unappealing as well as confusing to the user. Additionally, placement of advertising frames on a web page generally results in cramping or decreasing the size of the main content frame which oftentimes renders the content in the main frame difficult to read. As a result, users have developed ways to reduce the size or even

Doc. 0055 Filed 07/22/2008

eliminate frames from the web page being viewed.

Another type of online advertising involves the self-appearing window which generally appears on its own as a user is using the Internet or browsing on the WWW. Such advertisements are relatively easy for a user to avoid as a user may simply re-size the window to make it smaller, drag another window or object in front of it to obscure it from view, close the advertising window, or simply ignore it and continue with the task being undertaken online. Recently, online advertisers have begun using self-appearing screens which are delivered via dialog boxes which dominate the main part of the screen. Although these dialog boxes can be removed when the user clicks on the appropriate place(s) on the dialog box, the self-appearing dialog boxes have a much higher rate of being seen by users. This follows because the dialog boxes take control of the user's screen for a preset amount of time and/or until the user clicks on the appropriate place(s) to make the dialog box disappear. The recent prevalence in the use of self-appearing dialog box advertising has resulted in a more intrusive method of advertising which has resulted in resentment among users who are accustomed to more passive online advertising methods such as the frames and banner advertisements which are more easily avoided and/or ignored.

Accordingly, there is a need for a simple means to deliver advertising elements, i.e. logos, animations, sound, impressions, text, etc., without the annoyance of totally interrupting and intrusive content delivery, and without the passiveness of ordinary banner and frame advertisements which can be easily ignored.

Objects And Summary Of The Invention

It is thus a general object of the present invention to provide a means for delivering online advertisements which are unintrusive and which are not easily ignored by a user.

A more specific object of the present invention is to provide a server system for modifying a cursor image to a specific image displayed on a video monitor of a remote user's terminal.

It is another object of the present invention to provide a server system for modifying a cursor image to a specific image displayed on a video monitor of a remote user's terminal for the purposes of providing on-screen advertising.

It is a further object of the present invention to provide a means for providing on-screen advertising transmitted online which does not interrupt the delivery of content and which is aesthetically appealing and which affords the advertiser a great degree of unintrusive exposure.

It is still a further object of the present invention to provide a system and a method for causing a remote user terminal to display a cursor image as specified by a server terminal.

It is also an object of the present invention to provide a system and method for causing a remote user terminal to display a cursor image as specified by a server terminal, wherein the cursor image corresponds to the content retrieved by the user terminal.

It is a further object of the present invention to provide a system and method for causing a remote user terminal to display a cursor image such as a corporate name or logo, a brand logo, an advertising or marketing icon or slogan, an animated advertising image, and a related audio clip, that relate to an advertisement, such as a banner advertisement, that is included

in the information content being retrieved by the user terminal.

It is an additional object of the present invention to provide a means for changing a cursor's appearance by sending data and control signals from a remote computer so that the cursor or pointer's appearance is associated with a portion of, or the entire content being displayed on the user's screen.

It is still an additional object of the present invention to provide a means for changing the appearance of a computer's cursor or pointer by sending data and control signals from a remote computer so that the cursor or pointer's appearance is associated with advertising messages.

These and other objects of the invention are realized in various embodiments of the present invention by providing a system for delivering advertising elements online without the annoyance resulting from the interruption of content delivery and without the passiveness of ordinary banner and frame advertisements which can be too easily ignored or bypassed or removed. An exemplary embodiment of the present invention is directed to a system that provides online advertising content using the on-screen cursor which is generally controlled by an input of positioning device known as a "mouse" or "mouse pointer". Nearly all online computer interfaces utilize a wired or remote control positioning device such as a mouse or roller or track ball which controls the cursor's movement on the screen. It is the cursor controlled by the mouse or positioning device which a user uses to "navigate" or move the cursor over objects, buttons, menus, scroll bars, etc., which appear on-screen and then clicking or in some cases double-clicking in order to activate a screen or task, or to commence an application or some function.

As a result of the prevalence of the use of the mouse, by many millions of users of online systems, a great deal of time is spent focused on the icons which represent the cursor or pointer as it may appear in some cases. Presently, pointer icons change from application to application and can also change within an application depending upon where on the screen the pointer is located, what state the computer exists in at a given moment, and what tools are being used, among other factors. Generally, pointers change shape to reflect an internal state of the computer or the present function within an application. While it is not new for pointers and cursors to change shape, pointers are not presently used to convey advertising. In conventional systems, the appearance of the cursor or pointer does not change to correspond with on-line content being displayed on the screen.

The present invention provides a means for enabling cursors and pointers to change color, shape, appearance, make sounds, display animation, etc., when the user's terminal or computer, known as the "client" or "user" terminal, which has a network connection, receives certain instructions from a remote or "server" computer attached to the network. In an exemplary embodiment of the present invention, the generic cursor or pointer icons used in many networking applications, such as black arrows, hands with a pointing finger, spinning wheels, hourglasses, wristwatches, and others, will change appearance, and in some cases may incorporate sound or animation, in a way that is linked and related to the content, such as a web page, which is being transmitted to and displayed on the client computer. The cursor or pointer may appear as a corporate or a brand logo which relates to advertising content within the web page being transmitted and displayed. The cursor or pointer image may also appear in a specified shape or

color that is intended to convey a message that relates to the advertising content within the web page being transmitted and displayed.

An exemplary embodiment of the present invention comprises a combination of hardware and enabling software residing on the transmitting (server) computer or network server and/or on the receiving (client or user) computer or terminal which brings about the stated effect of enabling a computer's cursor or pointer to change appearance and in certain cases provide sound and animation which is linked and related to the content being transmitted to and displayed on the client computer or terminal. The transmitting computer and receiving computer or terminal advantageously include a processor, an operating system (OS) loaded thereon, a video monitor used to display a graphical user interface (GUI) and a Hypertext Transfer Protocol (HTTP) compliant web browser capable of loading and displaying hypertext documents transmitted over the Internet, although the invention is not limited in scope in that respect. For example, the receiving terminal may be any device that is able to communicate with a remote server, such as a user computer terminal, a user dumb terminal, or a television based system, such as a Web TV® terminal and other devices.

Preferably, coded information for bringing about the change in appearance of the cursor are embedded within the web page being loaded and viewed. In one embodiment of the present invention, the web page is written in Hypertext Markup Language (HTML) which is one of the most common standard page description languages used to develop web pages. Typically a web browser retrieves a web page to be loaded on a user's terminal. The retrieved web page in accordance with one embodiment of the invention contains a set of predetermined instructions

referred to herein as cursor display instructions. The browser or browser extension interprets the information contained in cursor display instructions and instructs the operating system of the user's terminal via an application programming interface (API) to check its memory to determine if the user terminal is capable of loading the coded image, animation, and/or soundbite. If the image, etc. has been previously cached in the client computer memory, the cursor display instructions instruct one or more of the many devices controlled by the operating system in the user's terminal, such as the video monitor and audio speakers to display the desired images, animation and play desired sounds. If the image, etc. has not been previously cached in the client computer's memory, the browser or browser extension retrieves the information corresponding to the desired image from a remote server.

The present invention may serve to enhance banner advertisements which appear on a web page so as to remind users which company is sponsoring the particular page being viewed and to draw the user's attention to the banner advertisement. The present invention can also serve as a stand-alone branding vehicle as part of a "ubiquity campaign" to generate massive impressions among an audience of online users or can be simply used to make web sites more entertaining by providing animated, colorful cursors which may incorporate sound and/or animation, and which are configured so as to connote a relationship with the topic or subject of the web site.

The foregoing sets forth certain objects, features and advantages provided by exemplary embodiments of the present invention. Other objects and features of the present invention will become apparent from the following detailed description considered in conjunction

with the accompanying drawings. It is to be understood, however, that the drawings are designed solely for the purposes of illustration and not as a definition of the limits of the invention, for which reference should be made to the appended claims.

Detailed Description Of The Drawings

In the drawings in which like reference characters denote similar elements throughout the several views:

Figure 1 illustrates a diagrammatic representation of a computer network illustrating the interconnection of a plurality of computers in which the present invention is implemented;

Figure 2 illustrates a client-server computer network supporting the hardware and software of the present invention;

Figure 3 illustrates a flowchart diagram of an exemplary method of the present invention for obtaining information from a remote site for modifying a cursor image and implementing such information at numerous user sites;

Figure 4 illustrates a portion of the Cursor Display Instructions which is referenced as a resource within an HTML document according to one embodiment of the present invention;

Figure 5 illustrates a set of exemplary codes that cause the user terminal's cursor to be modified, then revert to its original shape in accordance with one embodiment of the present invention;

Figure 6 illustrates a plurality of user interface attributes that may be remotely

modified in accordance with one embodiment of the present invention; and

Figures 7-9 illustrate the appearance of a cursor prior to, during and after linking to a web page that contains cursor display instructions.

Detailed Description Of The Presently Preferred Embodiment

Figure 1 illustrates a computer network, such as Internet 10, based on the client-server model. Internet 10 comprises a worldwide network of computers known as "servers" 12 which are accessible by "client computers" or "user terminals" 14, which are typically used by individual users or comprise a collection of personal computers interconnected via a Local Area Network or LAN, which are capable of accessing the Internet via a private Internet service or access provider (ISP) 16, such as the AT&T Worldnet Service® or the IBM Global Network®, or via an online service provider 18, such as America Online®, Compuserve®, the Microsoft Network® or Prodigy® (to name the most popular online service providers) One of the most common applications of the Internet is to support the World Wide Web ("WWW" or "the web"), which is a collection of servers on the Internet that utilize the Hypertext Transfer Protocol (HTTP), a known application protocol that facilitates data exchange between client and server and provides users or clients 14 access to files which can include text, graphics, sound, video, etc., using a standard page description language referred to as Hypertext Markup Language (HTML).

Each client computer 14 as indicated in Figure 1, includes a "web browser" or browser loaded on the client computer's hard drive 21. A browser is a common software tool

which allows graphical user interface (GUI)-based access to Internet network servers 12 through Internet Service Providers, ISPs, 16 or online service providers 18. A server 12 functions as a so-called "web site" which supports and maintains a plurality of files in the form of documents and pages. A Uniform Resource Locator or URL identifies a specific network path to a server 12 or some resource located on that server which has a known syntax for defining the network connection. The fundamental intrinsic capabilities of the browser are: (1) the ability to communicate with other computers using HTTP, and (2) the ability to process and present HTML documents to the user via a graphical user interface, GUI.

Recent versions of most browsers provide a plethora of other features beyond these two capabilities. For example, to increase its flexibility, the browser's intrinsic capabilities may be further extended through the use of software components, often called "controls" or "plug-ins". While the intrinsic capabilities of the browser are linked at compile-time ("statically"), the code which implements the capabilities of the control or plug-in component is linked with the browser's code at run-time ("dynamically"). By supporting these components through standard interface definitions, the browser's capabilities can be extended in ways never anticipated by its original manufacturer.

Another type of flexibility is offered when the browser implements some sort of command interpreter which is capable of interpreting and executing a code stream at run-time. In this case, the browser acts as a sort of "virtual machine" whose run-time behavior is completely governed by the code stream which it processes. The total scope of capabilities which can be realized with this approach is defined by the set of operations supported by the command

interpreter.

Individually and collectively, these mechanisms provide a powerful and flexible platform which supports a wide range of Internet-based applications. Currently, some of the emerging standards govern the operation of these mechanisms, although the invention is not limited in scope in that respect. For example, Microsoft has created an interface definition for Windows "dynamic link libraries" and for ActiveX software components. Sun Microsystems has defined a software component model called JavaBeans. Sun has also created a virtual machine architecture and language called Java, which is supported via a variety of commercially available compilers. While a Java compiler translates source code into pseudo-code output called an "applet", which is in turn processed by the Java virtual machine, Microsoft, Sun, and others have also defined a set of HTML scripting languages whose source code is embedded directly in an HTML page. Microsoft's VBScript, JScript and Sun's JavaScript are examples of these embedded scripting languages.

The standard web page description language, HTML, provides basic document formatting and permits the web site developer to create and specify "links" or "hyperlinks" to other servers and files. Obtaining a web page or connecting to a web site requires the specification of a URL using an HTML-compliant client browser. After specifying the URL, client computer 14 initiates a request to server 12 identified in the link and connects to the web site and receives a web page. The request by client computer 14 to server 12 via the link is advantageously communicated via a TCP/IP (Transfer Control Protocol/Internet Protocol) communication, although the invention is not limited in this respect and other network

connections or Internet protocols may be used.

Although an exemplary embodiment of the present invention is described based on the arrangement illustrated in Fig. 1, it is noted that the invention is not limited in scope in that arrangement and other types of system connections may be employed. For example, a plurality of user terminals may be connected to an online provider via dedicated communication channels, such as telephone lines. In accordance with this embodiment, the server system provides certain information that causes the cursor image on the video monitor of the user terminal to display an image as specified by the server system. As a result, the server system remotely defines and manages the shape and appearance of the cursor image in accordance with a pre-specified condition. The shape and appearance of the cursor image may correspond to the actual content of the data being provided to the user. Furthermore, regardless of the actual content of the data being provided to the user, the shape and appearance of the cursor image may be specified by the server system such that a plurality of user terminals at a desired point in time receive appropriate instructions to display the specified cursor image.

Figure 2 provides a block diagram of hardware and software which is representative of a client-server network system connected via the Internet according to one embodiment of the present invention. The user or client computer or user terminal 14 typically includes a number of hardware components and software subsystems which cooperate to deliver the wide range of capabilities demanded by a modern computer application or program. These include not only the basic computational processor 23 and memory 20, but also a variety of input and output devices such as the keyboard (not shown), mouse 22, video display monitor 24, audio

speakers 26, non-volatile storage such as a hard drive 21 and network communications systems 46 such as a modem among other devices. User terminal 14 is controlled via an operating system ("OS") 28 which serves to organize all the disparate elements within the computer 14 and expose them in a consistent and organized way to a program which may need some or all of these capabilities. The interface between a program, which is generally loaded within the computer's memory 20, and the systems under the control of the operating system 28 is commonly referred to as the Application Programming Interface ("API") 30, which is essentially a library of functions which the program ("application") can invoke when it needs to interact with any of these hardware subsystems.

As illustrated, user terminal 14 contains a browser 32 loaded within the computer's memory 20, and is adapted to communicate with a browser extension or browser plug-in 34, both which are adapted to communicate with the operating system 28 via the application programming interface API 30. As illustrated, operating system 28 is supplemented by a set of "drivers" which control and provide the operating system 28 with access to peripheral devices which are a part of user terminal 14. The drivers include display driver 36 which controls and provides the operating system 28 with access to the cursor image or pointer 44 projected on video display monitor 24, a mouse driver 38 which controls and provides the operating system 28 with access to mouse 22, an audio driver 40 which controls and provides the operating system 28 with access to speakers 26. Operating system 28 is configured to provide animated images to the video monitor. Furthermore, in accordance with another embodiment of the invention, the display driver may be configured to provide animated images to the video monitor. Operating system 28 also provides

access to a communication port 46 such as a modem which serves as a communication interface to the Internet 10.

With continued reference to Figure 2, user terminal 14 is connected to Internet 10 via a modem or some other communication interface such that information may be transmitted between user terminal 14 and Internet 10 via communication lines such as telephone cables or fiber optic networks, among other types of transmission systems. Internet 10 is also connected to numerous network servers, such as a simplified representation of a WWW server which is indicated as 48. Server 48 is provided with memory 50 into which the contents of certain data files are loaded. Such data files, among others, include Cursor Display Code 52, Cursor Information 54, and an HTML page containing Cursor Display Instructions 56, all of which are discussed in greater detail herein below. As illustrated in Figure 2, these data files 52, 54, 56 are shown residing on the same server computer. However, the interconnected nature of the WWW allows these data files 52, 54, 56 to exist anywhere on Internet 10. For example, files 52 containing cursor display codes may be stored in various server systems, while files 54 containing cursor information may be stored in the same or other server systems, and files 56 containing HTML pages containing cursor display instructions may be stored in the same or yet other server systems.

In operation, WWW server 48 includes software which recognizes file requests received from WWW clients or users by communication port 58 and fulfills these requests by retrieving data stored in data files, i.e., Cursor Display Code 52, Cursor Information 54, and an HTML page containing Cursor Display Instructions 56.

One of the characteristics of most recent software systems is the graphically oriented user interface (GUI) which is viewable on video monitor 24. This graphical user interface helps to organize and filter the vast quantities of information which is accessible in a user terminal 14. Fundamental to the graphical user interface is the pointing device, generally mouse 22 which allows the user to manipulate or input information into the user terminal 14. Movement of mouse 22 is monitored by user terminal 14 which translates this movement into a corresponding movement of cursor 44 viewable on video monitor 24. As such, operating system 28 may expose, as some subset of its API 30, a set of functions which can be used to control aspects of the behavior and/or appearance of cursor 44.

By combining the capabilities of browser extensions, such as indicated by 34 in Figure 2, with the capabilities to modify cursor 44, it is possible for a WWW server, such as that indicated by 48 in Figure 2, to control the display characteristics of cursor 44 displayed on video monitor 24 of the user's computer 14. By doing so, a cursor control arrangement is established which is capable of delivering information which supplements, enhances, or is completely independent of, other information transmitted from a server, such as indicated by 48, through traditional means as via a communications port 58. The basic conceptual components of such exemplary system for modifying cursor 44 comprises Cursor Display Code 52, Cursor Information 54, and Cursor Display Instructions 56, discussed hereinabove with reference to Figure 2. Preferably, Cursor Display Code 52 comprises a set of instructions which are executed on the user terminal 14 and which interact directly with application programming interface 30 of the user terminal 14 and operating system 28 so as to accomplish the actual change of cursor 44.

Cursor Information 54 is, advantageously, a set of data which identifies the actual cursor image or images and corresponding audio content if desired. In one embodiment of the invention, Cursor Display Instruction 56 includes data that convey information that is used by Cursor Display Code 52 to control drivers, such as 36, 40, 46, and to identify such things, which among others consist of: the physical location of Cursor Information 54, the format of its representation, the intended manner and duration of its display, and information pertaining to how (and for how long) any cached Cursor Information 54 should be stored.

In general, the fundamental elements of the process of changing cursor 44 displayed on video monitor 24 of user terminal 14 are as follows: Cursor Display Instructions 56 are initially embedded inside an HTML document, e.g. a web page. When browser 32 of the user terminal 14 encounters Cursor Display Instructions 56, Cursor Display Code 52 is retrieved then invoked. As part of the invocation, the browser passes to the Cursor Display Code coded information sufficient to specify the manner of the display. Cursor Display Code 52 then retrieves Cursor Information 54 either from within memory 20 of user terminal 14 or from storage at a remote site and then causes the Cursor Information to interact with the display system, such as display driver 36, of user terminal 14 via the application programming interface 30 of operating system 28. This interaction causes Cursor Information 54 to be accessed by the display driver 36 in order to accomplish the intended effect, e.g., the change or transformation of cursor 44 visible on video monitor 24, and a corresponding sound information may be heard on speakers 26.

Figure 4 illustrates the Cursor Display Instruction as a resource within an HTML document which is retrieved from a remote server. The Cursor Display Instructions as shown in

Figure 4 are written for ActiveX® technology, although the invention is not limited in scope to that technology. Among the information included within this resource definition is an identifier of the Cursor Display Code (the ActiveX® control), and the ActiveX® control's physical location on the Internet. This information is listed in lines 202-205 which generally identifies the Cursor Display Code. Line 204 of the Cursor Display Instruction is an identifier which comprises a globally unique name, often called a "Class ID", and which allows a particular ActiveX® control to be distinguished from all other ActiveX® controls, such that the wrong ActiveX® control is prevented from being utilized or retrieved. The remainder of the Cursor Display Instruction listed in lines 206-224 include the ActiveX® parameters or argument list as discussed hereinafter with reference to Figure 3. The argument list includes parameters which provide information such as the type of cursor image (line 206), where the image can be retrieved from if not already resident on the user computer (line 207), where usage statistics are to be transmitted to (line 208), how long a changed image should remain before reverting, if at all, to the initial image (line 209), whether the cursor image is cached in the user terminal (line 210), whether the transmitting server is authorized to send cursor display instructions (line 211), the dormant delay duration (line 212), the URL of a file which specifies cursor trajectory path (line 213), the URL of a file which specifies how the cursor's shape should change based on its location on the screen (line 214), the URL of a file which specifies how the cursor's shape should change based on its velocity (line 215), the URL of a file which specifies how the cursor's shape should change based on modifications to the mouse button or keyboard state (line 217), specification of the type of modification intended (line 218), specification of the priority of intended modification (line 219),

specification that the modifications will occur as a result of the transfer of a series of data files (line 220), the URL of a file which specifies the display of a satellite image that tracks the movement of the cursor image (line 221-223), and location of additional display instructions (line 224). It is noted that the invention is not limited in scope in this respect and other features may be included in the Cursor Display Instructions data.

One embodiment of this method in accordance with the present invention is set forth in greater detail in the flowchart illustrated in Figure 3. This embodiment is discussed with reference to the use of ActiveX® technology currently promoted by the Microsoft Corp. The ActiveX® technology provides a mechanism for defining the format of Cursor Display Instructions 56, for defining, identifying, and in some instances dynamically retrieving Cursor Display Code 52, and for implementing the interaction between Cursor Display Instructions 56 and the Cursor Display Code 52 as previously described. Although the flowchart in Figure 3 is discussed with reference to ActiveX® technology, the invention is not limited in this respect, and other technologies for use with browser extensions or "plug-ins" may be utilized in accordance with various embodiments of the present invention as illustrated in Figure 3. Furthermore, additional embodiments in accordance with the principles of the present invention may be incorporated within other application software employed in the user terminal. For example, the operating system or the browser itself may be configured to incorporate the mechanism for receiving and recognizing the Cursor Display Instructions and in return provide additional instructions for changing the image or appearance of the cursor display.

With reference to Figure 3, in step 102, browser 32 of user terminal 14 retrieves an

HTML file containing Cursor Display Instructions 56. The HTML file is retrieved when the user directs browser 32 to a remote WWW server site (such as, for example server 48 as indicated in Figure 2) by specifying the uniform resource locator, URL, of the site on the Internet where the HTML file is located. When the HTML file is retrieved, it is loaded from the remote WWW server site at which point browser 32 of user terminal 14 begins its routine parsing of the HTML document and eventually encounters a reference to an ActiveX® control or some other information coded in an appropriate programming language such as Sun Microsystem Inc.'s Java® or VBScript®, which is embedded in the Cursor Display Instructions 56 within the HTML document. The Cursor Display Code is capable of interacting with the application programming interface 30 of operating system 28 for the purpose of performing the change, transformation or "swap" of cursor 44 as it is presently displayed on video monitor 24.

Upon encountering Cursor Display Instructions 56, browser 32 recognizes Cursor Display Instructions 56 as a request to invoke the particular ActiveX® control with a particular argument list or set of parameters as illustrated in Figure 4. At step 104, browser 32 examines Cursor Display Instructions 56 and uses a unique class identification within the Cursor Display Instructions 56 to determine whether Cursor Display Code 52 (ActiveX® control) is already resident within local memory 20 of user computer 14.

If the Cursor Display Code 52 is not resident in local memory 14, generally in the form of a browser extension or plug-in 34, or if local memory contains an obsolete version of Cursor Display Code 52, browser 32 attempts, at step 106, to retrieve the ActiveX® control from a remote server on the Internet and store the Cursor Display Code in local memory 20 of user

terminal 14 at step 108. With reference to Figure 4, these steps correspond to lines 202-205.

Cursor Display Code 52 retrieved in step 106 may be client-platform specific and may also be browser specific such that browser 32 may transmit specific details to the remote server so that the remote server can deliver the appropriate Cursor Display Code 52.

In accordance with another embodiment of the invention, browser extension or plug-in 34 may be configured such that it can recognize Cursor Display Instructions based on any one of the available technologies, such as Active X, JavaBeans, JavaScript or VBScript.

Furthermore, it is understood that data compression techniques may be used in order to reduce the amount of network traffic involved in the transmission of data over the Internet.

After Cursor Display Code 52 has been recognized by user terminal 14 as at step 104 or retrieved and loaded therein at steps 106 and 108, operating system 28 is queried to determine the current cursor display configuration and this information is temporarily cached in local memory 20 of user terminal 14 at step 110 so that the cursor configuration may eventually be restored to its original state. Before any changes are made to cursor 44, the system at step 111 determines whether server 48 is authorized to change cursor 44. If authorization is not confirmed, no changes to cursor 44 transpire.

Step 112 is the first step which is executed from within the code of the ActiveX® control. At step 112, the ActiveX® control determines whether the image specified (Cursor Information 54) in the ActiveX® argument list which is to become the new cursor image exists in local memory 20 of user terminal 14. If the specified image in the ActiveX® argument list exists

in local memory 20, it is retrieved therefrom at step 114. An additional argument in the ActiveX® argument list (line 207) identifies the location of this data on a remote server. If the specified image does not exist in local memory 20, this data is utilized by the ActiveX® control to retrieve Cursor Information 54 at step 116 from the specified location.

At step 118, an additional argument added within the ActiveX® control can be used to determine whether and for how long Cursor Information 54 should be cached in local memory 20. At step 120 Cursor Information 54 is cached in local memory 20. At step 122, the cursor is caused to change in the manner consistent with the retrieved Cursor Display Instructions 56. In an alternative embodiment, an additional step may be included which provides the user with the option of saving and storing the retrieved Cursor Information 54 in the computer's permanent memory on hard drive 21 even after the retrieved cursor is displayed. Storing the retrieved Cursor Information 54 in the computer's permanent memory saves time on the next occasion when the user loads a web page which requires the same cursor since the cursor is already stored within the computer's memory and need not be retrieved from a remote server.

Cursor Display Instructions 56 cause the invocation of an operating system function which causes the cursor to be displayed on video monitor 24. More specifically, the ActiveX® control invokes the application programming interface 30 of operating system 28 which causes the cursor image displayed on video monitor 24 to change to the form intended as recited in the argument list. The changed cursor is not limited to image, and may also include animation as well as sound. It should also be appreciated that most computers utilize a multitude of cursor images depending upon the application and task which is being run on the computer.

The invention is not limited to changing only a single cursor image and any and all cursor images controlled by the computer's display driver 36 may be caused to change.

At step 124 the ActiveX® control may send usage information to a particular remote server as coded in Cursor Display Instruction 56 or Cursor Display Code 52. This information can be used to calculate the usage statistics of particular cursor images or cursor information and the context in which they are retrieved and viewed by users. In this particular embodiment, this information is conveyed as a data file transmitted to the remote server via HTTP. The invention is not, however, limited in the type of information and/or statistics which may be transmitted to the server, nor is the invention limited to being conveyed via HTTP as those skilled in the art will understand that such information may be conveyed via other transfer protocols. With reference to Figure 4, this step corresponds to line 208. Additionally, the information may contain an identifying code for the server which issued the web page which contained the Cursor Display Instructions. This information could be used, for example, to verify that the issuing server has been granted the appropriate license to use the technology, by comparing a list of authorized servers or through digital signature validation.

In accordance with one embodiment of the present invention, the licensing arrangement is described in more detail, hereinafter. It is noted that licensing enforcement of the cursor display technology could be accomplished in several ways, and the invention is not limited in scope in that respect. As discussed previously, the server that transmits a web page may include the identity of the server in the form of a server ID within the Cursor Display Instructions. The user terminal then transmits the server ID to another server that among other things functions as a

licensing body ("Licensing Body") so as to authenticate the server that transmits the web page as a valid licensee. Should this authentication fail, the execution of Cursor Display Instructions may not occur. In an alternative implementation, the execution of Cursor Display Instructions may be allowed to execute even if the issuer fails authentication. Such an infraction could be logged by the Licensing Body for use in enforcement through traditional channels. For performance reasons it may be desirable to collect the usage information for a plurality of Cursor Display Instructions as the user accesses multiple servers, and transmit the collection of information in batch form to the Licensing Body.

An alternative embodiment would involve the inclusion of an encrypted authentication code within the Cursor Display Instructions, as illustrated in line 211 of Fig. 4, or via a separate exchange of data between the client and server. In order to ensure that this code could not be re-used by other, non-authorized sites, it could for example be derived from the server's IP address, the date and time at which it is generated, the argument list, or some other information that is accessible to the client. Another possibility would involve the transmission of a unique or pseudo-unique code, from the client to the server. Upon receipt of this authentication code, the client would perform a decryption and verify its authenticity. Under such circumstances, the server software could be augmented with an Authentication Code Module supplied by the Licensing Body which generates and encrypts this code. The mechanism by which this augmentation could occur is similar to that discussed previously in the context of extending the client browser. For example, the server software could be modified and statically linked to the Authentication Code. Alternatively, it could be dynamically linked at run-time. Another alternative

would be to implement the Authentication Code as its own process on the server and facilitate an inter-process communication protocol such as the Common Gateway Interface ("CGI").

At step 126, an ActiveX® control argument is used to determine whether the changed cursor should revert to its initial configuration. If it is intended to revert the changed cursor to its initial configuration, the reversion is paused at step 128 for a specified time period. After it is determined at step 130 that the specified time period has lapsed, the changed cursor reverts to its original configuration at step 132.

Whether the cursor is caused to revert to its initial configuration is of concern to many users so as to ensure that the user's computer configuration is not permanently altered as a result of the process of changing the cursor. As such, additional alternative measures may be added into Cursor Display Instructions 54 such that the changed cursor could be restored to its original configuration when the ActiveX® control is loaded or unloaded, when the computer starts up, is rebooted or is shut down, when the browser is activated or shut down, when an animated cursor completes its animation sequence, when instructed by a remote server, or as a result of some user input such as setting an option in the browser or accessing another web page or site. An alternative to adding parameters to the Cursor Display Instructions would be to control the process of changing the cursor to its initial state by a control program downloaded by and executed on the client computer. An example written in VBScript and interacting with an ActiveX control is included in Figure 5.

Additionally, one of the significant attributes of this embodiment is the manner in which Cursor Display Code 52 is retrieved from a remote server if it is not located in the

computer's local memory. Since Cursor Display Code 52 may be operating system or browser specific, it may be necessary that the server with which the user computer 14 is communicating be informed by user terminal 14 of the specific type of Cursor Display Code 52 which is desired. In another embodiment of the invention, browser extension or plug-in 34 may be configured such that it can recognize Cursor Display Instructions based on any available technology such as Active X and JavaScript.

The operation of steps 102-132 as set forth in Figure 3, may be illustrated pictorially in Figures 7-9. Figure 7 illustrates an example of a typical web page 60 as it would appear on a user's video monitor 24 having the standard arrow cursor 44. In Figure 8, there is illustrated a different web page 60a having a banner advertisement 62 for Fizzy Cola which contains Cursor Display Instructions. When web page 60a loads, the Cursor Display Instructions cause arrow cursor 44 to change into a Fizzy cola bottle shaped cursor 44a in conjunction with the Fizzy Cola banner advertisement. As illustrated in Figure 9, if the user then loads a new web page 60b which is not provided with Cursor Display Instructions, the cola bottle shaped cursor of Figure 8, reverts to the standard arrow cursor 44.

It is also understood that ActiveX® is but one of numerous technologies utilized over the Internet with which a user's computer may interact in bringing about the change or transformation of the cursor displayed on video monitor 24. Other implementations may utilize different technologies such as Windows dynamic link libraries, VBScript and JScript from Microsoft, as well as Java, JavaScript and JavaBeans from Sun Microsystems Inc. While these examples represent the dominant standards-based definitions, proprietary implementations could

also be developed . Accordingly, while ActiveX® represents one embodiment of distributing and invoking Cursor Display Information 54 on a user's computer 14, it is to be appreciated that there are a variety of alternative implementations, and this particular implementation should not be considered a limitation of the invention. For example, alternative versions of browser 32 may encapsulate the appropriate operating system application programming interface call within their own code modules such that a browser extension 34 is not required.

In yet another embodiment of the invention the tasks described in steps 102 through 132 may be employed cooperatively between browser and browser extension or plug-in 34. Furthermore, browser 32 may employ a computational or processing engine such as an interpreter (as is the case with the Java® programming language, for example) which can extend the capabilities of browser 32 to a virtually unlimited degree.

It is also to be understood that in the course of carrying out the process of changing the cursor as discussed hereinabove, user terminal 14 may communicate with a multitude of remote servers as opposed to just a single server. For example, Cursor Display Codes may be retrieved from one remote server, Cursor Instructions may be retrieved from a second remote server, and the user terminal 14 may also be in communication with a third server to which it is transmitting the usage statistics.

Features identified in reference with Fig. 4 are described in more detail hereinafter. It is noted that in accordance with one embodiment of the invention, it may be desirable to modify the Cursor Display Code to improve its performance or enhance its capabilities. The server may transmit version information in the Cursor Display Instructions as illustrated in line 205 of Fig. 4.

The Cursor Display Code could compare this information with its own version information in order to determine whether it has been rendered obsolete by a more recent version. If so, the Cursor Display Code could retrieve the current version from a remote server and invoke execution on the new version..

In an alternative embodiment of the present invention the position, as well as the image, of the user terminal's cursor may be controlled by a remote server. This embodiment would be implemented within the Cursor Display Code 52 such that additional information could be passed to Cursor Display Code 52 via Cursor Display Instructions 56. The additional information passed to Cursor Display Code 52 would contain code which indicates: (1) that the cursor position control is intended, (2) the conditions under which the cursor should be moved, and (3) the source of the data which specifies the particular movement that is intended. The latter could be stored in memory on a remote server and retrieved in a manner similar to retrieving Cursor Display Instructions 56 or the Cursor Display Code 52. For example, if no user input is received for a specified interval, the cursor image could change and the position of the cursor could be set such that it follows a specified trajectory for several seconds, then reverts to its original state as illustrated by line 213 of Fig. 4.

In accordance with another embodiment of the invention it is possible to vary the modification to the cursor as a function of cursor position. For example, the cursor pointer could be controlled such that it "points" to a specific location on the screen regardless of the cursor's location on the screen as illustrated in line 214 of Fig. 4.

In accordance with another embodiment of the invention it is possible to vary the

modification to the cursor as a function of cursor velocity. For example, the cursor image could change from a stationary bird to a bird with flapping wings only when the cursor is moved quickly across the screen as illustrated in line 215 of Fig. 4. Furthermore, it is possible to vary the modification to the system-level user interface attributes as a function of mouse button state or keyboard state. For example, the image of a cube could be replaced with that of a jack-in-the-box when the mouse button is depressed.

In accordance with another embodiment of the invention, it is possible to modify other "system-level" attributes of the client computer's user interface, hereafter called "system-level user interface attributes". These attributes, as illustrated in Fig. 6 are typically under the control of the operating system and, as such, they exist independently of the user "applications" (programs) and data which are stored on the computer and interact with that operating system. User applications interact with the operating system to deliver the computer's functionality to the user. Examples of user applications include word-processors, spreadsheets, web browsers, games, etc. The operating system may contribute certain user interface elements to the user interface of the applications running on it.

Because many of these attributes are inherited from the operating system by all applications running on that operating system, applications tend to exhibit a degree of commonality in their user interfaces. Examples of these attributes include: the shape and color of the cursor 401, the shape and color of a status bar which displays current state information to the user 403, the shape and color of the scroll bar which indicates the relative position and scope of the displayed sub-image to that of the underlying larger image to the user 407, the shape and

color of the title bar which displays current state information 409, the shape and color of icons representing standard window operations such as close, minimize display size, restore display size, etc. 411. Thus, these system level attributes may also be modified in response to Cursor Display Instructions data.

In addition, the operating system itself may have a user interface. Examples include: the images and sounds displayed when the computer starts or shuts down, the background image ("wallpaper") against which other graphical elements are displayed 413, file catalogs and file selection mechanisms 415, system icons 416, file invocation mechanisms 417, buttons 419, process selection mechanisms 421, etc. Further examples include the icons representing various system elements or information such as files 418, groups of files 420, files marked for deletion 422, as well as standard, information bearing "dialog boxes", such as cancel, warning, illegal operation, stop, accept, continue, etc. 423. The system may also support a set of audibly distinct waveforms which may be used to convey similar information to the user. These operating system user interfaces may also be modified in response to a Cursor Display Instruction data.

In yet another exemplary embodiment of the present invention a plurality of modifications to the system-level user interface attributes may occur simultaneously For example, the cursor could animate while an audio waveform is playing, as the minimize display icon changes to a specific image.

A further feature of the invention is to accumulate information regarding the user's exposure time to various system-level user interface attribute modifications, and to vary the

exposure to those modifications accordingly. For example, the client could transmit exposure data to the server and the server would select a version of the image based on that data. Furthermore, the exposure data could be transmitted as part of the usage statistics discussed previously.

Another feature of the invention is to monitor the load being placed on the client system by the user and schedule data exchange with the servers so that it occurs when it is least disruptive to the user's activities.

It is also possible to allow the user to control the level of interface modification he or she wishes to entertain. For example, the user could specify that only those modifications of specific types, as illustrated on line 218 of Fig. 4 or of specific priority should be delivered, or even that none be delivered, as illustrated at lines 218 and 219 of Fig. 4. This specification could be implemented directly by the user on the client system, or could be implemented through communication with a remote server.

In accordance with another embodiment of the invention it is possible to transmit the image and/or audio data which specifies the modification as a series of data files which are delivered in a continuous stream to the client, as illustrated at line 220 of Fig. 4. These files are exposed to the user before the complete set of data has been delivered, thereby providing the capability for the initiation of long animations or audio files before the entire quantity of data has been received by the client.

A further feature of the invention is to support the display of a "satellite" image which tracks the cursor's position on the screen. For example, the cursor image could be replaced

with that of a mouse, and the image of a cat could be displayed near that mouse. When the cursor is moved, the satellite image moves accordingly at a specific offset, as illustrated at lines 221-223 of Fig. 4.

A further feature of the invention is to provide a mechanism for the user to quickly establish a connection with a specific server based on the specific user interface attribute modification which is in effect when the mechanism is invoked. For example, the user could press a specific key sequence on the keyboard and immediately jump to the web site related to the cursor image which is currently displayed.

In accordance with another embodiment of the invention, it is possible to convey additional detailed Cursor Display Instructions as a separate file which is explicitly retrieved from a server by the Cursor Display Code, as illustrated at line 224 of Fig. 4.

For each modification to the system-level user interface attributes, an appropriate set of display instructions must be transmitted to the client. These could take the form of additional parameters in the Cursor Display Instructions as discussed previously, or they could be represented within a code module which is received by and executed on the client. As discussed previously, Java, and its related technologies could be used for such a purpose, but use of these technologies should not considered a limitation of the invention.

It is noted that there are numerous ways in which a system-level user interface attribute modification is accomplished in accordance with the principles of the present invention. It is further noted that system level user interface attributes may be modified independently or in conjunction with cursor modification. Furthermore, the system-level user interface attribute

modification may be related to specific information displayed on the rest of the user's screen (hereafter referred to as "specific information") in many different ways. Thus, the present invention is not limited in scope to how content providers may relate the system-level user interface attribute with the specific information. Rather, at least one of the goals of the present invention is to enable the content providers to modify the system-level user interface attribute whenever and wherever they see fit. For example, content providers may modify system-level user interface attributes at a remote user's terminal for advertising, entertainment, information delivery, celebrating an event, or other reasons, and therefore, the invention is not limited in scope in that respect. Furthermore, when a content provider elects to display a specified system-level user interface attribute in conjunction with and corresponding to specific information conveyed via the user's terminal, the cursor image and the background display data are deemed related.

Additional examples intended to illustrate some applications of the present invention are explained below, although the invention is not limited in scope to any one of these examples.

Thus, in accordance with one embodiment of the invention, a modified cursor might take the appearance of a "Fizzy Cola" bottle when a "Fizzy Cola" banner advertisement appears among the display data of a popular search engine's site Similarly, the cursor can be modified for advertising purposes to represent Fizzy Cola's logo, its corporate mascot, images of its products or services, slogans, icons, brand images, advertising messages (the word "Thirsty?", for example), abstract suggestions (such as a straw or glass), etc.

Alternatively, Fizzy Cola, on its own site, or homepage, might have a picture of a

bottle of Fizzy in the middle of the page (in the display data). A dynamic cursor image could then be used to show a person holding a straw in such a way that the straw always points from the user toward the top of the Fizzy bottle, no matter where the cursor moves on the screen. The straw, in this case, might be "attached" to the cursor image (part of the same image) or could be separate, "satellite" image, a "sprite," whose movement on the screen (in this case) is related to the movement of the cursor. Sprites, which can appear and disappear as desired, can enhance the invention by enabling the use of graphical elements which are associated with the cursor but which reside outside the limited cursor "space" (which in some systems may be, at maximum, 32 by 32 pixels). For the purposes of the invention, however, there should be no limitation to the size of the cursor.

Additional examples of modification to the cursor include rendering the cursor as a baseball bat (on a site with sports information), a pink but otherwise standard-shaped pointer (on a site about the Pink Panther), a witch-on-a-stick to celebrate Halloween, the Statue of Liberty to celebrate the Fourth of July, etc. All of the foregoing cursor images could be enhanced with related animations, such as the bat hitting the ball.

Similarly, the present invention can be used to replace not just the standard arrow but other standard cursors as well, such as the generic hand with pointing index finger (the icon commonly used in browsers to indicate that the pointer is positioned above a hot link). A site for children might, for example, replace this generic pointing-hand cursor with the pointing "paw" of a furry animal. A site dealing with horror movies might choose to replace this pointing hand with a bony skeleton-like hand.

Additional examples involve cursors with text or numbers. For example, the cursor might contain the text "Right-Click Now!" prompting users to click the right button of their mouse (where right-clicking on the mouse could, for example, trigger the delivery of a new page of display data). It may also be desirable in certain cases to put alphanumeric data in the cursor "space" to convey information to users, such as stock prices, baseball game scores, the temperature in Florida, etc. The data can be static, semi-static (i.e. updated periodically), or dynamic (updated frequently - possibly incorporating available streaming-data and data-compression technologies).

Use of associated sound, sprites, animations, and modified system display elements are provided as enhancements to the basic invention. For example, a Fizzy cola mascot could appear in the cursor space in conjunction with the speakers, attached to the user's machine, playing the sound of the mascot saying, "drink Fizzy!" Any time a content provider elects to incorporate said enhancements in conjunction with a new modified cursor image, the cursor image and said enhancements have been deemed related.

The present invention allows users to change cursor images, it also allows them to change them back. It may be desirable to revert the pointer to a previous or generic pointer image. Given the Fizzy Cola example above, if the page containing display data changes and there is no longer an advertisement for Fizzy, but rather an advertisement for its rival, Jazzy Cola, it may be desirable to ensure the removal of the Fizzy cursor image(s) and accompanying enhancements.

The foregoing examples are not intended to suggest limited uses for this invention;

to the contrary, the examples are intended to illustrate the wide range of uses for this invention. The collective creativity of the online advertising, art, design, commerce, content publishing, and related industries will develop many novel and unforeseen ways to use the present invention. The versatility of the present invention should not be regarded as a limitation on its scope.

Thus, while there have been shown and described and pointed out fundamental novel features of the invention as applied to preferred embodiments thereof, it will be understood that various omissions and substitutions and changes in the form and details of the disclosed invention may be made by those skilled in the art without departing from the spirit of the invention. It is the intention, therefore, to be limited only as indicated by the scope of the claims appended hereto.

It is to be understood that the drawings are not necessarily drawn to scale, but that they are merely conceptual in nature.

Claims

We claim:

Sub a1

1. A server system for modifying a cursor image to a specific image having a desired shaped and appearance displayed on a video monitor of a remote user's terminal, said system comprising:

a first memory means for storing cursor information data corresponding to said specific image;

a second memory means for storing data corresponding to a cursor display code, said cursor display code containing information in response to which said cursor image is modified to said specific image;

a first server computer for transmitting a specified information to said remote user terminal, said information including a cursor display instruction data containing the location of said first and second memory means, such that said user terminal in accordance with said cursor display instruction data displays a cursor image on said user's video monitor in the shape and appearance of said specific image.

2. A server system in accordance with claim 1, wherein said specified information further comprises information that is intended to be displayed on said video monitor of said user's terminal.

3. A server system in accordance with claim 2, wherein said specific image

corresponds to at least a portion of said information intended to be displayed on said video monitor of said user's terminal.

4. The server system in accordance with claim 3, wherein said specific image comprises advertising material related to at least a portion of said information intended to be displayed.

5. The server system in accordance with claim 4, wherein said advertising material further comprises a brand logo.

6. The server system in accordance with claim 4, wherein said advertising material further comprises a corporate mascot.

7. The server system in accordance with claim 4, wherein said advertising material further comprises images of a good or a service corresponding to said information intended to be displayed.

8. The server system in accordance with claim 4, wherein said advertising material further comprises messages relating to said information intended to be displayed.

9. The server system in accordance with claim 3, wherein said specific image

has a shape and appearance representing the content of said information intended to be displayed.

10. The server system in accordance with claim 2, wherein said first and second memory means are disposed locally to said first server computer.

11. The server system in accordance with claim 2, wherein said first and second memory means are disposed within at least a second server computer located remotely to said first server computer.

12. The server system in accordance with claim 11, wherein the contents of said first and second memory means are also stored in a third memory means disposed locally to said user's terminal.

13. The server system in accordance with claim 11 wherein said server computer in response to a request from said user terminal transmits information stored in at least one of said first and second memory means to said user terminal.

14. A server system in accordance with claim 2, wherein said information intended to be displayed is transmitted in the form of HTML files that define a web page.

15. A server system in accordance with claim 14, wherein said cursor

information data corresponds to an advertisement for goods or services contained in said web page.

16. A server system in accordance with claim 15, wherein said user terminal includes a browser extension application responsive to said cursor display instruction data so that said extension application executes said cursor display code by employing parameters defined in said cursor display instructions.

17. A server system in accordance with claim 3 wherein said cursor information further comprises an image identifier that corresponds to data representing said specific image stored in said server system.

18. A server system in accordance with claim 4, wherein said server transmits said data representing said specific image in response to a request received from said remote user terminal indicating that a copy of said data representing said specific image is not stored in said remote user terminal.

19. A server system in accordance with claim 4, wherein said cursor display instruction data further comprises an image identifier that corresponds to a graphic animation sequence stored in said server system.

20. A server system in accordance with claim 19, wherein said cursor display instruction data further comprises instructions to modify said remote user terminal's cursor to display said graphic animation in response to activation of a link to a hypertext document selected by said user.

21. A server system in accordance with claim 20, wherein said cursor instruction data further comprises an audio identifier that corresponds to an audio information sequence stored in said server system.

22. A server system in accordance with claim 21, wherein said cursor display instruction data further comprises instructions to said remote user's terminal to play an audio clip corresponding to said audio information sequence in conjunction with said modification of said cursor display.

23. A server system in accordance with claim 4, wherein said cursor display instruction data further comprises information that controls the duration of time said cursor image is displayed on said remote user's video monitor.

24. A server system in accordance with claim 2, wherein said specified information transmitted in the form of hypertext objects are defined using the ActiveX® Internet programming technology.

Doc 2 - 5,960,409

25. A server system in accordance with claim 2, wherein said set of data files transmitted in the form of hypertext objects are defined using the Java® Internet programming language.

26. A server system in accordance with claim 2, wherein said set of data files transmitted in the form of hypertext objects are defined using the VBScript® Internet programming language.

27. A server system in accordance with claim 3, wherein said cursor display instruction data provides usage data to said server system for calculating usage statistics of said specific image.

28. A server system in accordance with claim 3, wherein said cursor display instruction data transmitted to said remote user terminal initiates communication with a plurality of server systems for obtaining additional cursor instruction data.

29. A method of modifying an initial cursor image displayed on a video monitor of a user terminal connected to one or more servers, which comprises the steps of:

receiving a request at said server to provide cursor display instructions to said user terminal;

providing a specified information to said user terminal in response to said request, said information including cursor display instruction data and cursor information corresponding to a specific image; and

transforming said initial cursor image displayed on said video monitor of said user terminal into the shape and appearance of said specific image in response to said cursor display instruction data.

30. A method in accordance with claim 29 wherein said step of providing specified information further comprises the step of providing information that is intended to be displayed on said video monitor of said user's terminal.

31. A method in accordance with claim 30, wherein said step of transforming further comprises the step of executing a cursor display code so as to display said specific cursor image such that it corresponds to at least a portion of said information intended to be displayed on said video monitor of said user's terminal.

32. A method in accordance with claim 31, wherein said step of displaying said specific cursor image further comprises the step of displaying advertising material related to at least a portion of said information intended to be displayed.

33. A method in accordance with claim 32, wherein said advertising material

further comprises a brand logo.

34. A method in accordance with claim 32, wherein said advertising material further comprises a corporate mascot.

35. A method in accordance with claim 32, wherein said advertising material further comprises images of a good or a service corresponding to said information intended to be displayed.

36. A method in accordance with claim 32, wherein said advertising material further comprises messages relating to said information intended to be displayed.

37. A method in accordance with claim 31, wherein said step of displaying said specific image further comprises the step of displaying a cursor image having a shape and appearance representing the content of said information intended to be displayed.

38. A method in accordance with claim 30, wherein said step of providing said specified information further comprises the step of providing information in the form of HTML files that define a web page.

39. A method in accordance with claim 38, wherein said cursor display

instruction data further comprises cursor information including specified images corresponding to an advertisement for goods or services contained in said web page.

40. A method in accordance with claim 31, wherein said step of transforming further comprises the steps of :

employing a browser extension application responsive to said cursor display instruction data; and

executing said cursor display code by employing parameters defined in said cursor display instruction data.

41. A method in accordance with claim 40 wherein said step of providing cursor information further comprises the step of providing an image identifier that corresponds to the location of data representing said specific image.

42. A method in accordance with claim 29, further comprising the step of transmitting said data representing said specific image in response to a request received from said remote user terminal indicating that a copy of said data representing said specific image is not stored in said remote user terminal.

43. A method in accordance with claim 42, wherein said step of providing specified information further comprises the step of providing an image identifier that corresponds

to a graphic animation sequence.

44. A method in accordance with claim 43, further comprising the step of modifying said remote user terminal's cursor to display said graphic animation in response to activation of a link to a hypertext document selected by said user.

45. A server system in accordance with claim 42, wherein said step of providing specified information further comprises the step of providing an audio identifier that corresponds to an audio information sequence.

46. A server system in accordance with claim 46, further comprising the step of playing an audio clip corresponding to said audio information sequence in conjunction with said modification of said cursor display.

47. A method in accordance with claim 29 further comprising the step controlling the duration of time said cursor image is displayed on said remote user's video monitor.

48. A method in accordance with claim 47, further comprising the step of providing usage data to said server system for calculating usage statistics of said specific image.

49. A computer storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform the method steps of claim 31.

50. An internet browser configured to modify a cursor image to a specific image having a desired shaped and appearance displayed on a video monitor of a remote user's terminal, said browser comprising:

means for retrieving information from a remote server intended to be displayed on said video monitor , said information comprising a cursor display instruction data, wherein said cursor display instruction data includes information corresponding to said specific image;

means for recognizing said cursor display instruction data; and

means for executing a cursor display code, said cursor display code including information in response to which said cursor image is modified to said specific image.

51. The internet browser in accordance with claim 50 further comprising means to retrieve said cursor information from a prespecified server, when said cursor information is not stored in said user terminal.

52. The internet browser in accordance with claim 50 further comprising means to retrieve said cursor display code from a prespecified server, when said cursor display code is not stored in said user terminal.

Sub a3

53. The internet browser in accordance with claim 50 further comprising means to communicate with a server system so as to determine whether cursor display instructions received by said user terminal were transmitted by an authorized server.

54. The internet browser in accordance with claim 50 further comprising means for transmitting statistical information to a prespecified server so as to provide information relating to the usage of said cursor display instruction data.

55. The internet browser in accordance with claim 50, wherein said specific cursor image reverts back to its original cursor shape and appearance after a prespecified duration.

56. The internet browser in accordance with claim 55 further comprising means to store said cursor display code in a memory means located within said user terminal.

57. The internet browser in accordance with claim 55 further comprising means to provide audio clips corresponding to said display of said specific cursor image.

58. The internet browser in accordance with claim 57 wherein information relating to said audio clips are contained within said cursor display instruction data.

59. The internet browser in accordance with claim 57 further comprising means to provide animated images corresponding to said specific cursor image.

60. The internet browser in accordance with claim 59 wherein information relating to said animated images are contained within said cursor display instruction data.

61. The internet browser in accordance with claim 50, wherein said specific image corresponds to at least a portion of said information intended to be displayed on said video monitor of said user's terminal.

62. The internet browser in accordance with claim 61, wherein said specific image comprises advertising material related to at least a portion of said information intended to be displayed.

63. The internet browser in accordance with claim 62, wherein said advertising material further comprises a brand logo.

64. The internet browser in accordance with claim 62, wherein said advertising material further comprises a corporate mascot.

65. The internet browser in accordance with claim 62, wherein said advertising

material further comprises images of a good or a service corresponding to said information intended to be displayed.

66. The internet browser in accordance with claim 62, wherein said advertising material further comprises messages relating to said information intended to be displayed.

67. The internet browser in accordance with claim 61, wherein said specific image has a shape and appearance representing the content of said information intended to be displayed.

68. The internet browser in accordance with claim 50, wherein said information intended to be displayed is transmitted in the form of HTML files that define a web page.

69. The internet browser in accordance with claim 50, wherein said specified information is defined using the ActiveX® Internet programming technology.

70. The internet browser in accordance with claim 50, wherein said specified information is defined using the Java® Internet programming language.

71. The internet browser in accordance with claim 50, wherein said specified information is defined using the VBScript® Internet programming language.

add ~~65~~ C''

Abstract

A server system for modifying a cursor image, is displayed on a video monitor of a remote user's terminal, to a specific image having a desired shape and appearance. The system comprises a first memory means for storing cursor information data corresponding to the specific image and a second memory means for storing data corresponding to a cursor display code. The cursor display code contains information in response to which the cursor image is modified to the specific image.

A first server computer is provided so as to transmit a specified information to the remote user terminal. The information includes a cursor display instruction data containing the location of the first and second memory means, such that the user terminal in accordance with the cursor display instruction data displays a cursor image on the user's video monitor in the shape and appearance of the specific image.

08550502.050297

PRINT OF DRAWINGS
AS ORIGINALLY FILED

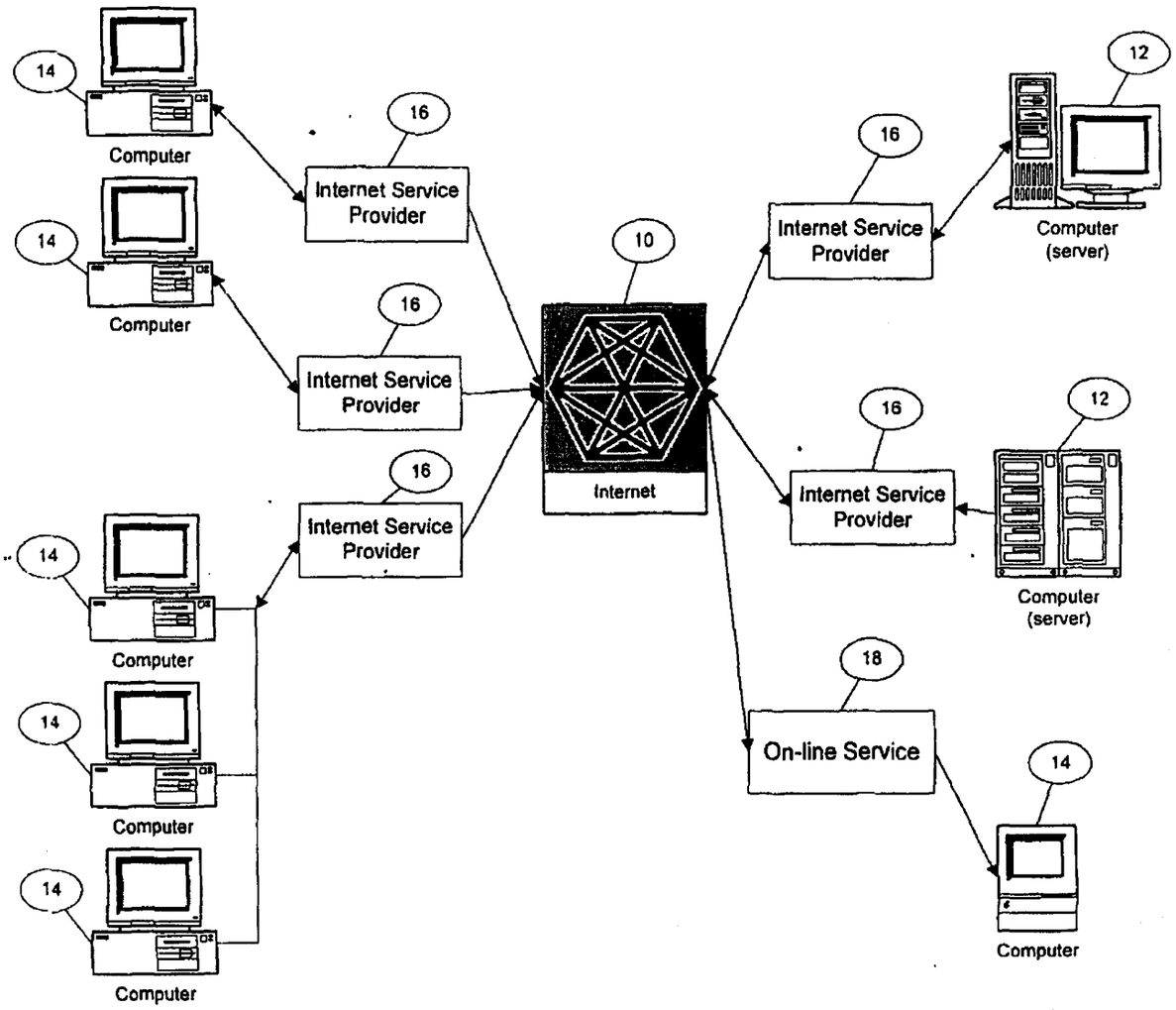


Figure 1

PRINT OF DRAWINGS
AS ORIGINALLY FILED

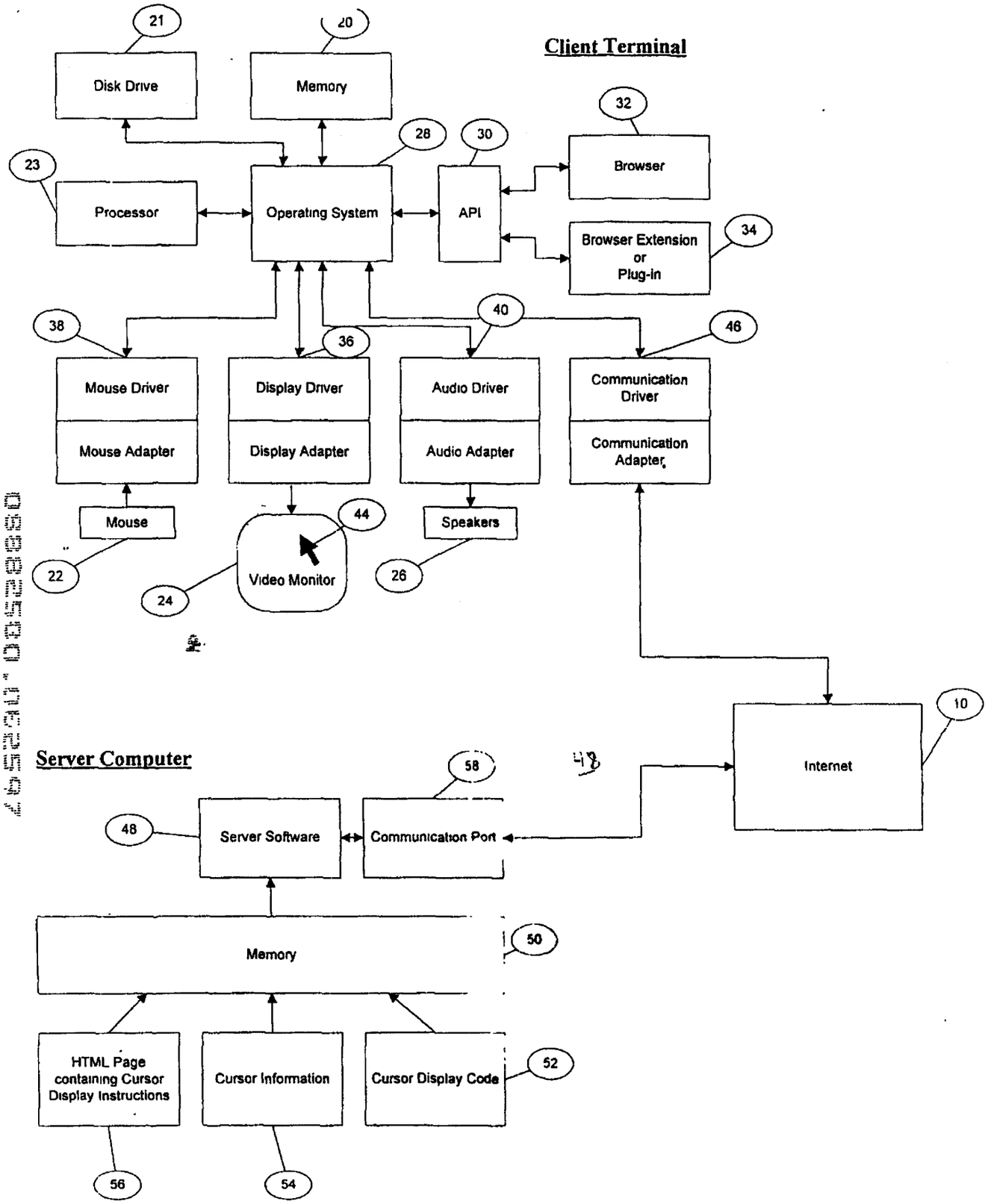


Figure 2

PRINT OF DRAWINGS
AS ORIGINALLY FILED

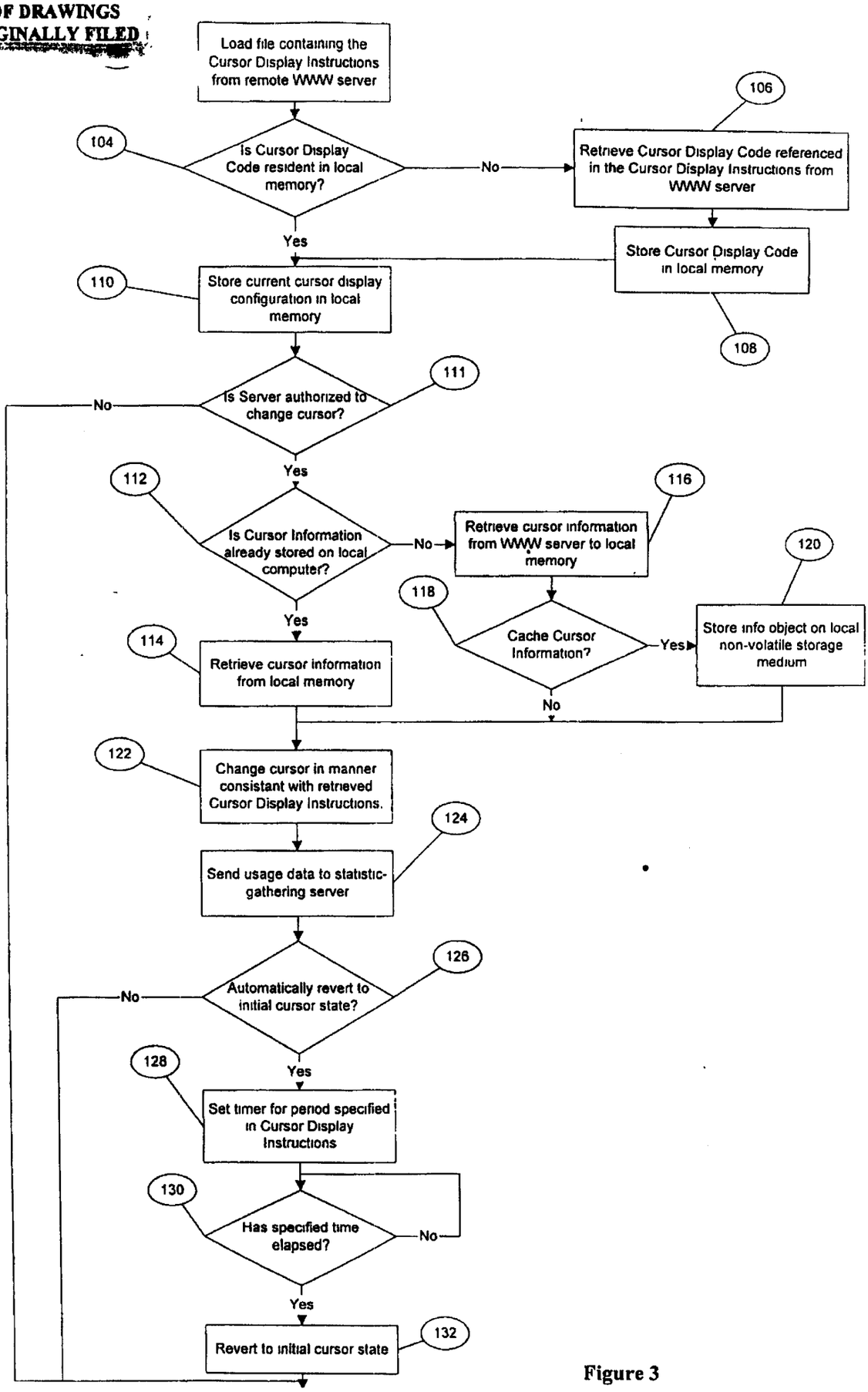


Figure 3

PRINT OF DRAWINGS
AS ORIGINALLY FILED

```
<OBJECT
        202.ID=cc1
        203.TYPE="application/x-oleobject"
        204.CLASSID="clsid:CB005660-D0C7-11cf-B7F6-00AA00A3F278"
        205.CODEBASE="http://cometsystems.com/controls/cc.cab#ver=4,70,0,
           1122"
        206.<PARAM NAME="CursorType" VALUE="1">
        207.<PARAM NAME="CursorImage"
           VALUE="http://cometsystems.com/library/images/acme.cur">
        208.<PARAM NAME="Counter" VALUE="http://
           cometsystems.com/accounting">
        209.<PARAM NAME="DisplayDuration" VALUE="5">
        210.<PARAM NAME="CacheCursor" VALUE="1">
        211.<PARAM NAME="ServerSignature" VALUE="54F5254A23BD988AB54">
        212.<PARAM NAME="DormantDelay" VALUE="600">
        213.<PARAM NAME="CursorTrajectoryMap" VALUE="http://
           cometsystems.com/maps/trajectory">
        214.<PARAM NAME="CursorPositionMap" VALUE="http://
           cometsystems.com/maps/position ">
        215.<PARAM NAME="CursorVelocityMap"
           VALUE="http://cometsystems.com/maps/velocity">
        216.<PARAM NAME="CursorPositionMap" VALUE="http://
           cometsystems.com/maps/velocity">
        217.<PARAM NAME="CursorButtonMap" VALUE="http://
           cometsystems.com/maps/buttonstate">
        218.<PARAM NAME="ContentType" VALUE="5">
        219.<PARAM NAME="PriorityLevel" VALUE="1">
        220.<PARAM NAME="StreamBufferSize" VALUE="0">
        221.<PARAM NAME="SatelliteImage"
           VALUE="http://cometsystems.com/library/images/acmesat.bmp">
        222.<PARAM NAME="SatelliteXDisplacement" VALUE="-50">
        223.<PARAM NAME="SatelliteYDisplacement" VALUE="50">
        224.<PARAM NAME="ExtraDisplayParameters"
           VALUE="http://cometsystems.com/library/params/acme.prm">
</OBJECT>
```

Figure 4

PRINT OF DRAWINGS
AS ORIGINALLY FILED

```
<script language="VBScript">
<!--\1`;
302.Sub window_onLoad()
303.    cc1.RememberCurrentCursor()
304.    cc1.SetNormalCursor("http://cometsystems.com/library/
           images/acme.cur")
305.end sub

306.Sub window_onUnload()
307.    cc1.Reset()
308.end sub
-->
</script>
```

Figure 5

7,656,870 - 09/552,998

BEST COPY

PRINT OF DRAWINGS
AS ORIGINALLY FILED

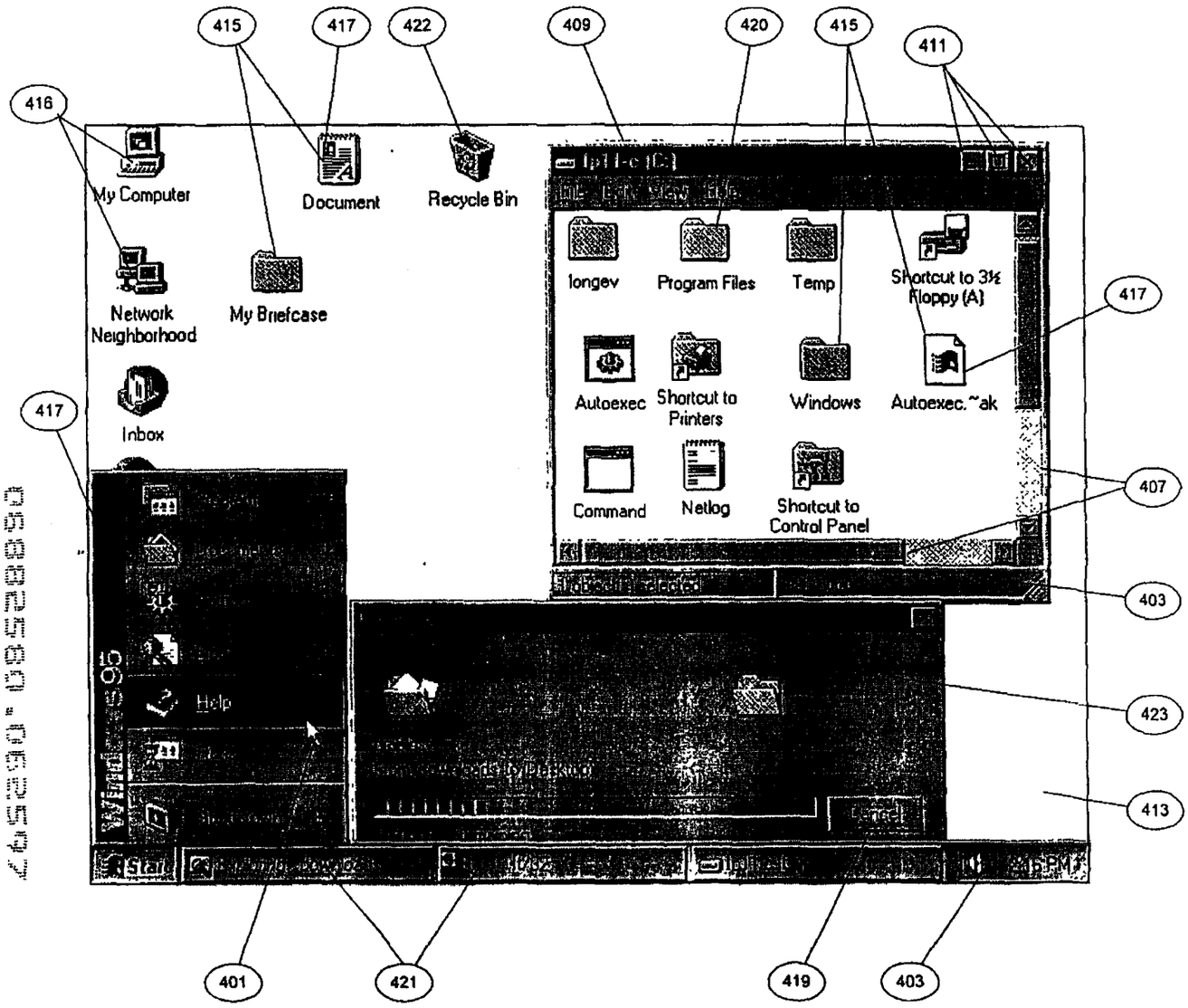


Figure 6

60

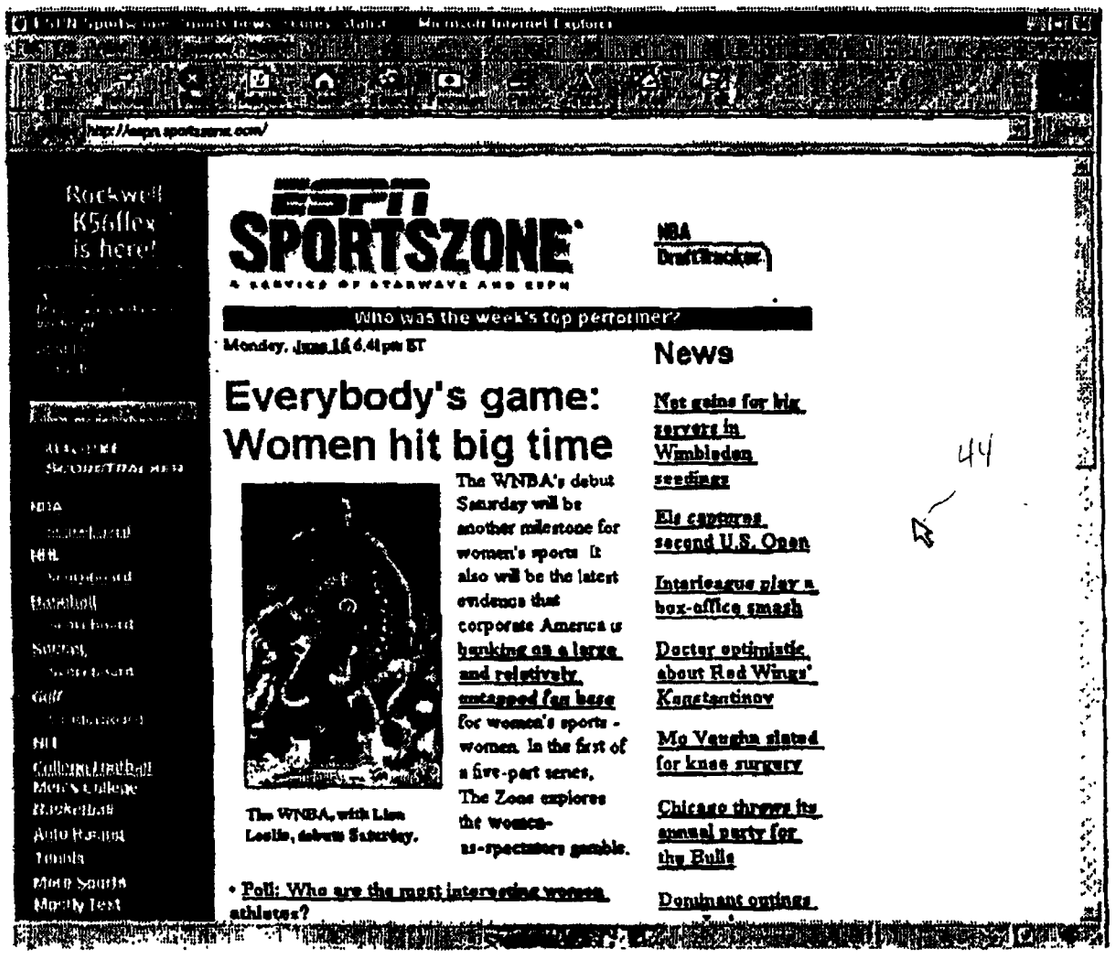

Rockwell
K56flex
is here!
ESPN
SPORTSZONE
A SERVICE OF STARWAVE AND ESPN
NBA
Who was the week's top performer?
Monday, June 16 6:41 pm ET
Everybody's game:
Women hit big time
The WNBA's debut Saturday will be another milestone for women's sports. It also will be the latest evidence that corporate America is banking on a large and relatively untapped fan base for women's sports - women. In the first of a five-part series, The Zone explores the women-as-spectators gamble.
The WNBA, with Lisa Leslie, debuts Saturday.
• Poll: Who are the most interesting women athletes?
News
Net gains for big servers in Wimbledon seedings
Elo captures second U.S. Open
Interleague play a box-office smash
Doctor optimistic about Red Wings' Konstantinov
Mo Vaughn slated for knee surgery
Chicago throws its annual party for the Bulls
Dominant outings
44

Figure 7

PRINT OF DRAWINGS
AS ORIGINALLY FILED

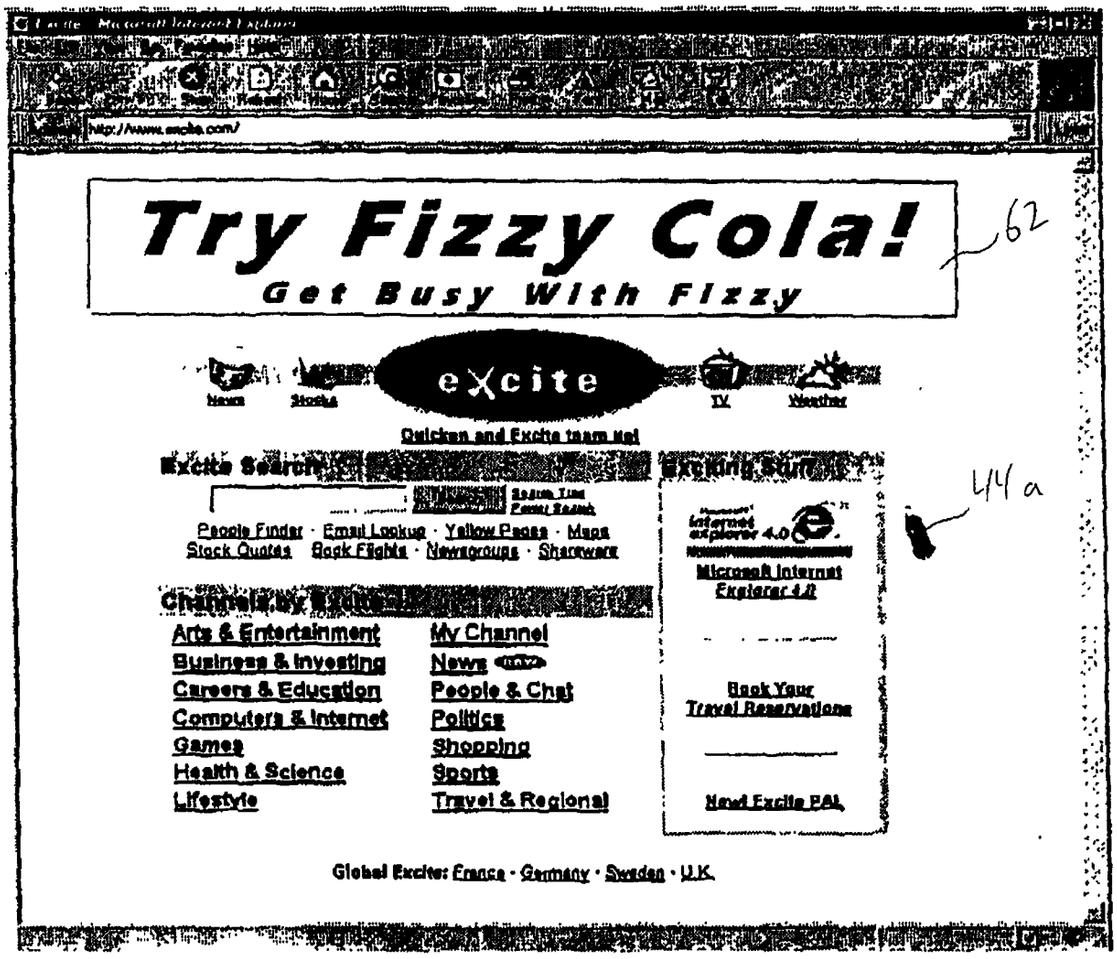


Figure 8

PRINT OF DRAWINGS
AS ORIGINALLY FILED

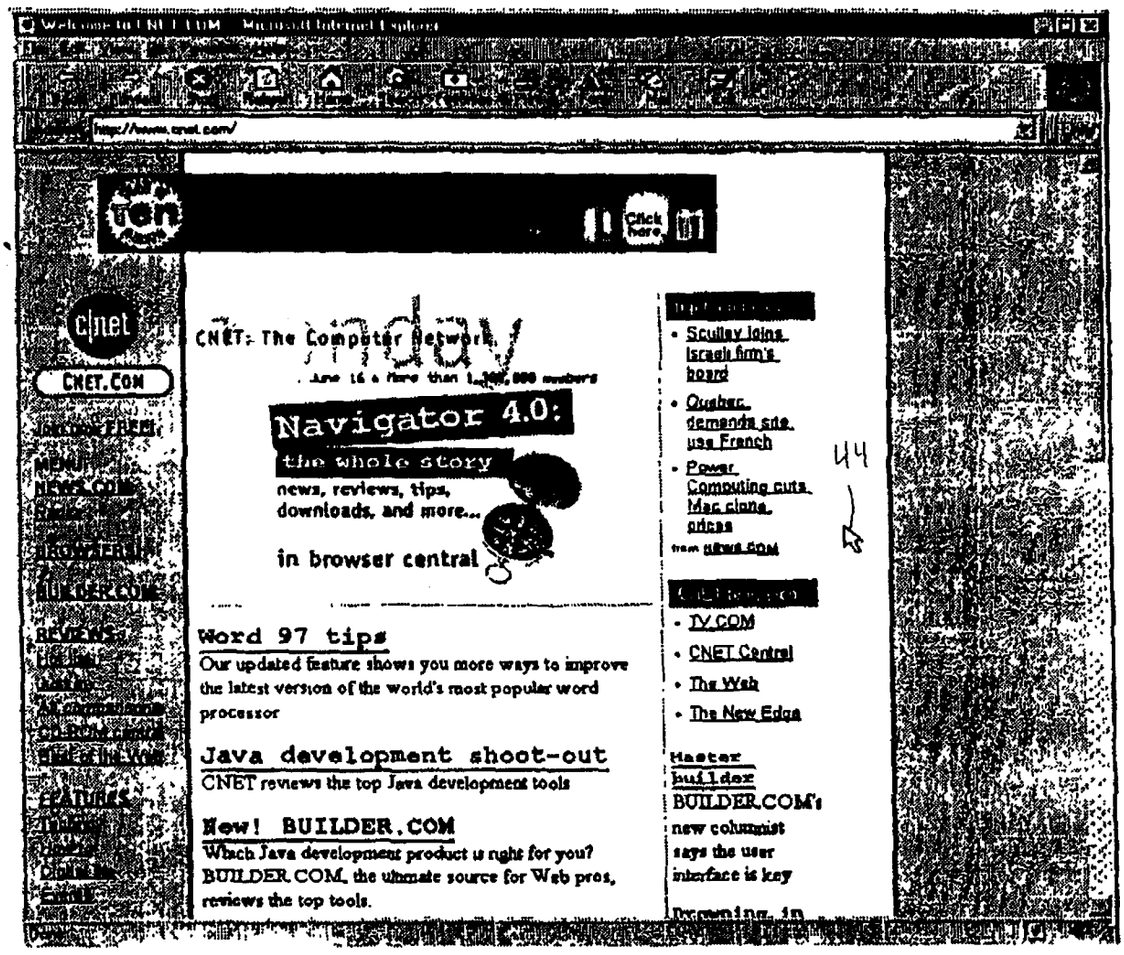


Figure 9

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

COMBINED DECLARATION, POWER OF ATTORNEY & PETITION

TYPE OF DECLARATION

[X] Utility
[ ] Design
[ ] Supplemental
[ ] Divisional
[ ] Continuation
[ ] Continuation-in-part
[ ] National Stage of the PCT

INVENTORSHIP AND SPECIFICATION IDENTIFICATION

My residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first and sole inventor (if only one is listed below) or a joint inventor (if plural inventors are named below) of the invention entitled:

SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE

as described and claimed in the specification which

[X] is attached hereto.
[ ] was filed on _____
  [ ] as U.S. Serial Number. 08/_____; or
  [ ] Express Mail No._____ (as serial number not yet known); and
  [ ] was amended on _____.
[ ] was described and claimed in PCT International Application No. PCT/ /_____ filed on _____; and
  ☐ as amended under PCT Article 19 and/or 34 on _____.

REVIEW OF PAPERS AND DUTY OF CANDOR

I have reviewed and understand the contents of the attached specification including the drawing and claims as amended by any amendment referred to below; and
I acknowledge my duty to disclose information of which I am aware which is material to the examination of this application, in accordance with 37 CFR 1.56(a); and
☐ in compliance with this duty there is attached an information disclosure statement.

PRIORITY CLAIMS

I hereby claim **foreign priority** benefits under Title 35, United States Code, § 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign applications for patent applications for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed
_____ (Number)	_____ (Country)	_____ (Day/Month/Year/Filed)	[ ] Yes [ ] No
_____ (Number)	_____ (Country)	_____ (Day/Month/Year/Filed)	[ ] Yes [ ] No
_____ (Number)	_____ (Country)	_____ (Day/Month/Year/Filed)	[ ] Yes [ ] No

I hereby claim the benefit under Title 35, United States Code, § 120 of any **United States Application(s)** listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States Application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, § 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application.

_____ (Application Serial No.)	_____ (Filing Date)	_____ (Status: Patented, Pending, Abandoned)
_____ (Application Serial No.)	_____ (Filing Date)	_____ (Status: Patented, Pending, Abandoned)

DECLARATION

I declare that all statements made above of my own knowledge are true and all statements made on information and belief are believed to be true; and these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

POWER OF ATTORNEY

I hereby appoint the following patent attorneys and/or patent agent(s) with full power of appointment, substitution and revocation to prosecute this application, to make alterations and amendments thereto, to receive the patent, and to transact all business in the Patent Office connected therewith.

Joseph Sofer (Reg. No. 34,438)
Robert M. Haroun (Reg. No. 34,345)

Please address all telephone calls and correspondence to:

Joseph Sofer
SOFER & HAROUN, L.L.P.
342 Madison Avenue
Suite 1921
New York, New York 10173
Telephone:(212) 697-2800
Facsimile:(212) 697-3004

2005-0525.658-002

PETITION

Wherefore, I pray that Letters Patent be granted to me for the invention or discovery described and claimed in the above-mentioned specification and claims, and I hereby subscribe my name to the foregoing Declaration, Power of Attorney & Petition with references to the above-identified specification and claims.

SIGNATURES

Name of joint or
first inventor: James Samuel Rosen

Home Address: 333 East 23rd Street, Apt. 8D, New York, NY 10010

Post Office Address: Same as above
Citizenship: US

Inventor's Signature: _____ Date:_____

Name of joint or
second inventor: Thomas A. Schmitter

Home Address: 197 Eight Street, Apt. #715, Charlestown, MA 02129
Post Office Address: P.O Box 397154, Cambridge, Massachusetts, 02139
Citizenship: US

Inventor's Signature: _____ Date:_____

Name of joint or
third inventor: Mark S. Hall

Home Address: 156 Irving Avenue, South Orange, NJ 07079
Post Office Address: _____
Citizenship: UK

Inventor's Signature: _____ Date:_____

Document 1-5 Filed 05/05/2008

Application Assignment Record

According to the application transmittal letter, an assignment recording ownership was filed with this application; however, a copy of this record was not located in the original file history record obtained from the United States Patent and Trademark Office. Upon your request, we will attempt to obtain the assignment documents from the Assignment Recordation Branch of of the United States Patent and Trademark Office or from a related application case (if applicable). Please note that additional charges will apply for this service.

This page is not part of the official USPTO record. It has been determined that content identified on this document is missing from the original file history record.

Transaction History Date 1997-11-12
Date information retrieved from USPTO Patent Application Information Retrieval (PAIR) system records at www.uspto.gov

BEST COPY

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NUMBER	FILING/RECEIPT DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NO./TITLE
08/883,580	06/25/97	ROSEN J	6748-002

0232/1112

JOSEPH SOFER
SOFER & HAROUN
342 MADISON AVE
SUITE 1921
NEW YORK NY 10173

NOT ASSIGNED

2312

DATE MAILED: 11/12/97

NOTICE TO FILE MISSING PARTS OF APPLICATION
Filing Date Granted

An Application Number and Filing Date have been assigned to this application. However, the items indicated below are missing. The required items and fees identified below must be timely submitted ALONG WITH THE PAYMENT OF A SURCHARGE for items 1 and 3-6 only of $ 130 for a ☑ large entity ☐ small entity in compliance with 37 CFR 1.27. The surcharge is set forth in 37 CFR 1.16(e). Applicant is given TWO MONTHS FROM THE DATE OF THIS NOTICE within which to file all required items and pay any fees required above to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

If all required items on this form are filed within the period set above, the total amount owed by applicant as a ☑ large entity ☐ small entity (verified statement filed), is $ 2042.

☑ 1. The statutory basic filing fee is:
- ☑ missing.
- ☐ insufficient.

Applicant must submit $ 790 to complete the basic filing fee and/or file a verified small entity statement claiming such status (37 CFR 1.27).

☑ 2. Additional claim fees of $ 1122, including any multiple dependent claim fees, are required.
Applicant must either submit the additional claim fees or cancel additional claims for which fees are due.

☐ 3. The oath or declaration:
- ☐ is missing.
- ☐ does not cover the newly submitted items.
- ☐ does not identify the application to which it applies.
- ☐ does not include the city and state or foreign country of applicant's residence.

An oath or declaration in compliance with 37 CFR 1. 63, including residence information and identifying the application by the above Application Number and Filing Date is required.

☑ 4. The signature(s) to the oath or declaration is/are:
- ☑ missing.
- ☐ by a person other than inventor or person qualified under 37 CFR 1.42, 1.43, or 1.47.

A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.

☐ 5. The signature of the following joint inventor(s) is missing from the oath or declaration:

_____

An oath or declaration listing the names of all inventors and signed by the omitted inventor(s), identifying this application by the above Application Number and Filing Date, is required.

☐ 6. A $_____ processing fee is required since your check was returned without payment (37 CFR 1.21(m)).

☐ 7. Your filing receipt was mailed in error because your check was returned without payment.

☐ 8. The application does not comply with the Sequence Rules.
See attached "Notice to Comply with Sequence Rules 37 CFR 1.821-1.825."

☐ 9. OTHER:

Direct the response and any questions about this notice to "Attention: Box Missing Parts."

A copy of this notice MUST be returned with the response.

Customer Service Center
Initial Patent Examination Division (703) 308-1202

FORM PTO-1533 (REV.7-96) PART 3 - OFFICE COPY *U.S. GPO: 1996-404-496/40515

jc575 U.S. PTO
01/13/98

Attorney Docket No.: 658-002 Patent

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

----------X

In re Application of
James S. Rosen, Thomas A. Schmitter, :
Mark S. Hall

Serial No.: 08/882,580 :

Filed: June 25, 1997 :

For: SERVER SYSTEM AND METHOD FOR :
MODIFYING A CURSOR IMAGE

----------X

RESPONSE TO NOTICE TO FILE MISSING PARTS OF APPLICATION FILING DATE GRANTED

Hon. Commissioner of Patents
and Trademarks
Washington, D.C. 20231

S I R:

Enclosed is a Declaration and check in the amount $1,021 in payment of the application fees and surcharge, along with a copy of PTO-1533, Notice to File Missing Parts of Application. Also enclosed is a Verified Statement Claiming Small Entity Status.

Any additional fees required in connection with this application may be charged to Deposit Account No. 19-2825.

Respectfully submitted,
SOFER & HAROUN, LLP

By: _____
Joseph Sofer
Reg. No. 34,438
342 Madison Avenue, Suite 1921
New York, New York 10173
(212) 697-2800

Dated: January 9, 1998

jc575 U.S. PTO
01/13/98

Sector 1

PATENT

Docket No. 658-002

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s)	: Rosen et al.	Group Art Unit: 2312
Serial No.	: 08/882,580	Examiner: Not yet assigned
Filed	: June 25, 1997	
For	: SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE	

CERTIFICATE OF MAILING (37 C.F.R. 1.8a)

#3

HON. COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

Sir:

I hereby certify that the attached Response to Notice to File Missing Parts, Form PTO-1533, Declaration, Check for $1021, Verified Statement Claiming Small Entity Status and Return Postcard along with any paper(s) referred to as being attached or enclosed and this Certificate of Mailing are being deposited with the United States Postal Service on the date shown below with sufficient postage as first-class mail in an envelope addressed to the: Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Respectfully submitted,

SOFER & HAROUN, L.L.P.

By: _____
Stacey Taitt

Date 1/9/98

Mailing Address.

SOFER & HAROUN, L.L.P.
342 Madison Avenue
Suite 1921
New York, New York 10173
Tel:(212) 697-2800
Fax:(212) 697-3004

01/13/1998 JWASHING 00000059 08882580
01 FC:201 395.00 DP
02 FC:203 561.00 DP
03 FC:205 65.00 DP

jc575 U.S. PTO
01/13/98

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

COMBINED DECLARATION, POWER OF ATTORNEY & PETITION

TYPE OF DECLARATION

[X] Utility
[ ] Design
[ ] Supplemental
[ ] Divisional
[ ] Continuation
[ ] Continuation-in-part
[ ] National Stage of the PCT

INVENTORSHIP AND SPECIFICATION IDENTIFICATION

My residence, post office address and citizenship are as stated below next to my name; I believe I am the original, first and sole inventor (if only one is listed below) or a joint inventor (if plural inventors are named below) of the invention entitled:

SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE

as described and claimed in the specification which

[ ] is attached hereto.
[X] was filed on June 25, 1997
  [X] as U.S. Serial Number. 08/ 882,580; or
  [ ] Express Mail No._____ (as serial number not yet known); and
  [ ] was amended on _____.
[ ] was described and claimed in PCT International Application No. PCT/ /_____ filed on _____; and
  ☐ as amended under PCT Article 19 and/or 34 on _____.

REVIEW OF PAPERS AND DUTY OF CANDOR

I have reviewed and understand the contents of the attached specification including the drawing and claims as amended by any amendment referred to below; and
I acknowledge my duty to disclose information of which I am aware which is material to the examination of this application, in accordance with 37 CFR 1.56(a); and
☐ in compliance with this duty there is attached an information disclosure statement.

PRIORITY CLAIMS

I hereby claim **foreign priority** benefits under Title 35, United States Code, § 119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign applications for patent applications for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)			Priority Claimed	
(Number)	(Country)	(Day/Month/Year/Filed)	[ ] Yes	[ ] No
(Number)	(Country)	(Day/Month/Year/Filed)	[ ] Yes	[ ] No
(Number)	(Country)	(Day/Month/Year/Filed)	[ ] Yes	[ ] No

I hereby claim the benefit under Title 35, United States Code, § 120 of any **United States Application(s)** listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States Application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, § 1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application.

(Application Serial No.)	(Filing Date)	(Status: Patented, Pending, Abandoned)
(Application Serial No.)	(Filing Date)	(Status: Patented, Pending, Abandoned)

DECLARATION

I declare that all statements made above of my own knowledge are true and all statements made on information and belief are believed to be true; and these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and such willful false statements may jeopardize the validity of the application or any patent issuing

thereon.

POWER OF ATTORNEY

I hereby appoint the following patent attorneys and/or patent agent(s) with full power of appointment, substitution and revocation to prosecute this application, to make alterations and amendments thereto, to receive the patent, and to transact all business in the Patent Office connected therewith.

Joseph Sofer (Reg. No. 34,438)
Robert M. Haroun (Reg. No. 34,345)

Please address all telephone calls and correspondence to:

Joseph Sofer
SOFER & HAROUN, L.L.P.
342 Madison Avenue
Suite 1921
New York, New York 10173
Telephone:(212) 697-2800
Facsimile:(212) 697-3004

PETITION

Wherefore, I pray that Letters Patent be granted to me for the invention or discovery described and claimed in the above-mentioned specification and claims, and I hereby subscribe my name to the foregoing Declaration, Power of Attorney & Petition with references to the above-identified specification and claims.

SIGNATURES

Name of joint or
first inventor: **James Samuel Rosen**

Home Address: **333 East 23rd Street, Apt. 8D, New York, NY 10010**
Post Office Address: **Same as above**
Citizenship: **US**

Inventor's Signature: _____ Date: 12-23-97

Name of joint or
second inventor: **Thomas A. Schmitter**

Home Address: **197 Eight Street, Apt. #715, Charlestown, MA 02129**
Post Office Address: **P.O Box 397154, Cambridge, Massachusetts, 02139**
Citizenship: **US**

Inventor's Signature: _____ Date: 12/23/97

Name of joint or
third inventor: **Mark S. Hall**

Home Address: **156 Irving Avenue, South Orange, NJ 07079**
Post Office Address: **Same as above**

Citizenship: **UK**

Inventor's Signature: _____ Date: 1/5/98

BEST COPY

1c575 U.S. PTO 01/13/98

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NUMBER	FILING/RECEIPT DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NO./TITLE
08/802,580	06/25/97	ROSEN	J 658-002

0232/1112

JOSEPH SOFER
SOFER & HARDIN
342 MADISON AVE
SUITE 1921
NEW YORK NY 10173

NOT ASSIGNED

3

2312

DATE MAILED: 11/12/97

NOTICE TO FILE MISSING PARTS OF APPLICATION
Filing Date Granted

An Application Number and Filing Date have been assigned to this application. However, the items indicated below are missing. The required items and fees identified below must be timely submitted ALONG WITH THE PAYMENT OF A SURCHARGE for items 1 and 3-6 only of $ 130 for a ☑ large entity ☐ small entity in compliance with 37 CFR 1.27. The surcharge is set forth in 37 CFR 1.16(e). Applicant is given TWO MONTHS FROM THE DATE OF THIS NOTICE within which to file all required items and pay any fees required above to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

If all required items on this form are filed within the period set above, the total amount owed by applicant as a ☑ large entity ☐ small entity (verified statement filed), is $ 2042.

☑ 1. The statutory basic filing fee is:
- ☑ missing.
- ☐ insufficient.

Applicant must submit $ 790 to complete the basic filing fee and/or file a verified small entity statement claiming such status (37 CFR 1.27).

☑ 2. Additional claim fees of $ 1122, including any multiple dependent claim fees, are required.
Applicant must either submit the additional claim fees or cancel additional claims for which fees are due.

☐ 3. The oath or declaration:
- ☐ is missing.
- ☐ does not cover the newly submitted items.
- ☐ does not identify the application to which it applies.
- ☐ does not include the city and state or foreign country of applicant's residence.

An oath or declaration in compliance with 37 CFR 1. 63, including residence information and identifying the application by the above Application Number and Filing Date is required.

☑ 4. The signature(s) to the oath or declaration is/are:
- ☑ missing.
- ☐ by a person other than inventor or person qualified under 37 CFR 1.42, 1.43, or 1.47.

A properly signed oath or declaration in compliance with 37 CFR 1.63, identifying the application by the above Application Number and Filing Date, is required.

☐ 5. The signature of the following joint inventor(s) is missing from the oath or declaration:

_____

An oath or declaration listing the names of all inventors and signed by the omitted inventor(s), identifying this application by the above Application Number and Filing Date, is required.

☐ 6. A $_____ processing fee is required since your check was returned without payment (37 CFR 1.21(m)).

☐ 7. Your filing receipt was mailed in error because your check was returned without payment.

☐ 8. The application does not comply with the Sequence Rules.
See attached "Notice to Comply with Sequence Rules 37 CFR 1.821-1.825."

☐ 9. OTHER:

Direct the response and any questions about this notice to "Attention: Box Missing Parts."

A copy of this notice MUST be returned with the response.

Customer Service Center
Initial Patent Examination Division (703) 308-1202

FORM PTO-1533 (REV.7-96) **PART 2 - COPY TO BE RETURNED WITH RESPONSE** *U.S. GPO: 1996-404-496/40515

jc575 U.S. PTO
01/13/98

VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY STATUS
(37 CFR 1.9(f) and 1.27(c)) - SMALL BUSINESS CONCERN

#3

Applicant or Patentee: J. Rosen, T. Schmitter, M. Hall
Attorney's Docket No.: 658-002
Serial or Patent No.: 08/882,580
Filed or Issued: June 25, 1997
Title: SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE

I hereby declare that I am

[ ] the owner of the small business concern identified below:
[X] an official of the small business concern empowered to act on behalf of the concern identified below:

NAME OF SMALL BUSINESS CONCERN Comet Systems, Inc.
ADDRESS OF CONCERN 66 West Broadway, Suite 306
New York, NY 10007

I hereby declare that the above identified small business concern qualifies as a small business concern as defined in 13 CFR 121.12, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees to the United States Patent and Trademark Office in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention described in:

[ ] the specification filed herewith with title listed above.
[X] the application identified above.
[ ] the patent identified above.

If the rights held by the above identified small business concern are not exclusive, each individual, concern or organization having rights to the invention must file separate verified statements averring to their status as small entities, and no rights to the invention are held by any person, other than the inventor, who would not qualify as an independent inventor under 37 CFR 1.9(c) if that person made the invention or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d), or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern or organization having any rights in the invention is listed below:

[ ] no such person, concern, or organization exists.
[ ] each such person, concern or organization listed below

FULL NAME: _____

ADDRESS: _____

[ ] INDIVIDUAL [ ] SMALL BUSINESS CONCERN [ ] NONPROFIT ORGANIZATION

jc575 U.S. PTO 01/13/98

Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING James Rosen

TITLE OF PERSON OTHER THAN OWNER _____

ADDRESS OF PERSON SIGNING 333 East 23rd Street, Apt. 8D

New York, NY 10010

SIGNATURE J-Ro DATE 12-23-97

GP 2312
2772

PATENT

Docket No. 658-002



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s)	: Rosen et al.	Group Art Unit: 2312
Serial No.	: 08/882,580	Examiner: Not yet assigned
Filed	: June 25, 1997	
For	: SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE	

CERTIFICATE OF MAILING (37 C.F.R. 1.8a)

HON. COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

Sir:

I hereby certify that the attached Information Disclosure Statement, Form PTO-1449, 10 References and Return Postcard along with any paper(s) referred to as being attached or enclosed and this Certificate of Mailing are being deposited with the United States Postal Service on the date shown below with sufficient postage as first-class mail in an envelope addressed to the: Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Respectfully submitted,

SOFER & HAROUN, L.L.P.

By: _____
Stacey Taitt

Date: 1/16/98

RECEIVED
MAR 24 98
GROUP 2600

Mailing Address

SOFER & HAROUN, L L.P.
342 Madison Avenue
Suite 1921
New York, New York 10173
Tel:(212) 697-2800
Fax:(212) 697-3004

OIPE
JAN 20 1998
PATENT & TRADEMARK OFFICE 2030

64
6-6-98
B.H. Hirsch

PATENT Docket No. 658-002

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s) : Rosen et al.

Serial No. : 08/882,580 Group Art Unit: ~~2312~~ 2773

Filed : June 25, 1997

For : SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE

INFORMATION DISCLOSURE STATEMENT

HON. COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

Sir:

This Information Disclosure Statement is filed in accordance with 37 C.F R. §§1.56, 1.97 and 1.98. The items listed on Form PTO-1449, a copy of which is enclosed, may be deemed to be pertinent to the above-identified application and are made of record to assist the Patent and Trademark Office in its examination of this application. The Examiner is respectfully requested to fully consider the items and to independently ascertain their teaching.

1. [ ] For each of the following items listed on the enclosed copy of Form PTO-1449 that is not in the English language, an English language translation of that item or a portion thereof or a concise explanation of the relevance of that item is enclosed:
   A:
   B:

2. [ ]For each of the following items listed on the enclosed copy of Form PTO-1449 that is not in the English language, a concise explanation of the relevance of that item is incorporated in the specification of the above-identified application.

3. [ ]Any copy of the items listed on the enclosed copy of Form PTO-1449 that is not enclosed with this Information Disclosure Statement was previously cited by or submitted to the Patent and Trademark Office in the prior [ ] Continuation, [ ] Divisional or [ ] Continuation-In-Part application under 37 C.F.R. §1.60, U.S. Serial No. _____ _____, filed _____

4. [X]No fee is due under 37 C F.R. §1.17(p) for this Information Disclosure Statement since it is being filed in compliance with:

[ ] 37 C F.R. §1 97(b)(1), within three months of the filing date of the above-identified application.

[ ] 37 C.F R. §1.97(b)(2), within three months of the date of entry into the national stage as set forth in §1.491 in an international application.

[X] 37 C F.R. §1 97(b)(3), before the mailing date of a first Office action on the merits.

5. [ ] No fee is due under 37 C.F.R §1.17(p) for this Information Disclosure Statement since it is being filed in compliance with 37 C.F.R. §1.97(c), after the period specified in paragraph 4 above but before the mailing date of a final action or a Notice of Allowance (where there has been no prior final action), and is accompanied by one of the certifications pursuant to 37 C F.R. §1 97(e) set forth in paragraph 9 below.

6. [ ] A fee is due under 37 C.F.R. §1.17(p) for this Information Disclosure Statement since it is being filed in compliance with 37 C.F.R. §1.97(c), after the period specified in paragraph 4 above but before the mailing date of a final action or a notice of allowance (where there has been no prior final action):

[ ] A check in the amount of $240.00 is enclosed in payment of the fee.

[ ] Charge the fee to Deposit Account No. 19-2825. Order No. _____. A DUPLICATE COPY OF THIS SHEET IS ATTACHED.

7. [ ] A fee is due under 37 C.F.R §1.17(I)(1) for this Information Disclosure Statement since it is being filed in compliance with 37 C.F.R. §1.97(d), after the mailing date of a final action or a notice of allowance, whichever comes first, but before payment of the issue fee, and is accompanied by:

a one of the certifications pursuant to 37 C.F.R. §1 97(e) set forth in paragraph 9 below; and

b. the attached petition requesting consideration of this Information Disclosure Statement; and

c. the fee due under 37 C.F.R. §1.17(I)(1) which is paid as set forth in paragraph 10 below.

8 [ ] A fee is due under 37 C.F.R §1.17(I)(1) for this Information Disclosure Statement since it is being filed in compliance with:

a. [ ] 37 C.F R §1 313(b)(3), after the issue fee has been paid and information cited in this Information Disclosure Statement may render at least one claim unpatentable and is accompanied by the attached Petition To Withdraw Application From Issue,

b. [ ] 37 C.F.R. §1.313(b)(5), after the issue fee has been paid and information cited in this Information Disclosure Statement is to be considered in a Continuation application upon abandonment of the instant application and is accompanied by the attached Petition To Withdraw Application From Issue.

c. The fee due under 37 C.F.R. §1.17(I)(1) is paid as set forth in paragraph 10 below.

9. [ ] I hereby certify that each item of information contained in this Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of this Information Disclosure Statement.

[ ] I hereby certify that no item of information in the Information Disclosure Statement filed herewith was cited in a communication from a foreign patent office in a counterpart foreign application or, to my knowledge after making reasonable inquiry, was known to any individual designated in §1.56(c) more than three months prior to the filing of this Information Disclosure Statement.

10. [ ] A check in the amount of $130.00 is enclosed in payment of the fee due under 37 C.F.R. §1.17(I)(1).

[ ] Charge the fee due under 37 C.F.R. §1.17(I)(1) to Deposit Account No. . Order No. _____. A DUPLICATE COPY OF THIS SHEET IS ATTACHED.

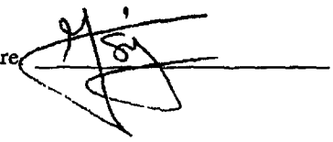
[X] The Commissioner is hereby authorized to charge any additional fees which may be required for this Information Disclosure Statement, or credit any overpayment to Deposit Account No. 19-2825, Order No. 658-002 .
A DUPLICATE COPY OF THIS SHEET IS ATTACHED.

Respectfully submitted,

SOFER & HAROUN, L.L.P.

By: Joseph Sofer
Registration No. 34,438

Dated: 1/16/98

Signature: 

Mailing Address:

Joseph Sofer, Esq.
SOFER & HAROUN, LLP
342 Madison Avenue, Suite 1921
New York, New York 10173

OIPE JAN 20 1998 JC36 PATENT & TRADEMARK OFFICE

Sheet 1 of 1

FORM PTO-1449 DEPARTMENT OF COMMERCE (Rev.2-3) PATENT AND TRADEMARK OFFICE	ATTY. DOCKET NO. 658-002	SERIAL NO. 08/882,580
INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Use several sheets if necessary)	APPLICANT Rosen et al.	
	FILING DATE June 25, 1997	GROUP ~~2312~~ 2773

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
C.G	4 6 7 2 5 7 5	6/9/87	Stephens	364	900	5/31/83
C.G	4 8 4 1 2 9 1	6/20/89	Swix et al.	340	725	9/21/87
C.G	4 9 8 4 1 5 2	1/8/91	Muller	364	200	10/6/87
C.G	5 1 5 7 7 6 8	10/20/92	Hoeber et al.	395	157	5/17/91
C.G	5 1 7 9 6 5 6	1/12/93	Lisle	395	159	1/19/89
C.G	5 3 4 7 6 2 8	9/13/94	Brewer et al.	395	159	1/18/90
C.G	5 5 4 4 2 9 5	8/6/96	Capps	395	152	5/27/92
C.G	5 5 5 9 9 4 3	9/24/96	Cyr et al.	395	155	6/27/94
C.G	5 5 7 2 6 4 3	11/5/96	Judson	395	793	10/19/95
C.G	5 5 9 6 6 9 4	1/21/97	Capps	395	152	4/8/96

FOREIGN PATENT DOCUMENTS

	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
						YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

RECEIVED MAR 24 98 GROUP 2600

EXAMINER	DATE CONSIDERED
	6/25/98

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

BEST COPY

Transaction History Date 1998-07-13
Date information retrieved from USPTO Patent Application Information Retrieval (PAIR) system records at www.uspto.gov

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NO
08/882,540	06/25/97	ROSEN J	658-002

LM02/0713

JOSEPH SOFER
SOFER & HAROUN
342 MADISON AVE
SUITE 1921
NEW YORK NY 10173

EXAMINER	
JACKSON, C	
ART UNIT	**PAPER NUMBER**
2773	#5

DATE MAILED: 07/13/98

This is a communication from the examiner in charge of your application.
COMMISSIONER OF PATENTS AND TRADEMARKS

OFFICE ACTION SUMMARY

☒ Responsive to communication(s) filed on 6/25/97.

☐ This action is **FINAL**.

☐ Since this application is in condition for allowance except for formal matters, **prosecution as to the merits is closed** in accordance with the practice under *Ex parte Quayle*, 1935 D.C. 11; 453 O.G. 213.

A shortened statutory period for response to this action is set to expire 3 month(s), or thirty days, whichever is longer, from the mailing date of this communication. Failure to respond within the period for response will cause the application to become abandoned. (35 U.S.C. § 133). Extensions of time may be obtained under the provisions of 37 CFR 1.136(a).

Disposition of Claims

☒ Claim(s) 1-71 is/are pending in the application.

Of the above, claim(s) _____ is/are withdrawn from consideration.

☐ Claim(s) _____ is/are allowed.

☒ Claim(s) 1-71 is/are rejected.

☒ Claim(s) 1 is/are objected to.

☐ Claims _____ are subject to restriction or election requirement.

Application Papers

☒ See the attached Notice of Draftsperson's Patent Drawing Review, PTO-948.

☐ The drawing(s) filed on _____ is/are objected to by the Examiner.

☐ The proposed drawing correction, filed on _____ is ☐ approved ☐ disapproved.

☒ The specification is objected to by the Examiner.

☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119

☐ Acknowledgement is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d).

☐ All ☐ Some* ☐ None of the CERTIFIED copies of the priority documents have been

☐ received.

☐ received in Application No. (Series Code/Serial Number) _____.

☐ received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

*Certified copies not received: _____.

☐ Acknowledgement is made of a claim for domestic priority under 35 U.S.C. § 119(e).

Attachment(s)

☒ Notice of Reference Cited, PTO-892

☒ Information Disclosure Statement(s), PTO-1449, Paper No(s). #4

☐ Interview Summary, PTO-413

☒ Notice of Draftsperson's Patent Drawing Review, PTO-948

☐ Notice of Informal Patent Application, PTO-152

– SEE OFFICE ACTION ON THE FOLLOWING PAGES –

★ US GPO 1996-409 290/40029

PTOL-326 (Rev. 10/95)

DETAILED ACTION

Claim Objections

1. Although applicant's claim 1 meets the requirements of 35 USC § 112, the grammar and use of terminology could be improved. The language of claim 1 reciting "said cursor display code containing information" is awkward. Suggested language is "said cursor display code providing information". As a result, the examiner request that the applicant, during the normal review and/or rewriting of the claims, take into consideration these editorial comments and make changes as necessary.

Specification

2. The abstract of the disclosure is objected to because it is not provided in a single paragraph. Correction is required. See MPEP § 608.01(b).

Claim Rejections - 35 USC § 112

3. Claims 24-26, 29-49 and 69-71 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. The applicant's use of trademark in each of the aforementioned claims makes them indefinite where there characteristics may change from time to time and yet it may continue to be sold under the same trademark. See MPEP 608.01(v). RB

4. Claim 29 recites the limitation "said server" in line 20, page 20. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103© and potential 35 U.S.C. 102(f) or (g) prior art under 35 U.S.C. 103(a).

7. Claims 1-3, 9-14, 17, ~~28~~ 27-31, 37-38, ~~41~~ 48, 47, 49-52, ~~55~~ 54-61 and 67-68 are rejected under 35 U.S.C. 103(a) as being unpatentable over Schneider (US Patent # 5,710,897). RB

Schneider ('897) teaches the applicant's invention as claimed in **claims 1-3, 9-14, 17, 28-31, 37-38, 41, 47, 49-52, 55-61 and 67-68** as follows:

as per claim 1, a first memory means for storing cursor information data corresponding to said specific image [local or remote memory (e.g., floppy drive, hard drive) for storing a pointer graphic file corresponding to a pointer graphic to be used by the operating system, see Schneider ('897), col. 5, lines 65- col. 6, lines 1-10; col. 7, lines 28-31; col. 8, lines 10-12, and 55-60] ;

a second memory means for storing data corresponding to a cursor display code [local or remote storage media, such as ROM, CD-ROM, floppy disk, or hard drive, for storing computer program product corresponding to pointer graphics manager program, see Schneider ('897), col. 9, lines 6-8; abstract], said cursor display code containing information in response to which said cursor image is modified to said specific image [the pointer graphics manager program containing instructions in response to which the pointer graphic can be customized, see Schneider ('897), col. 8, lines 52-col. 9, lines 1-67; abstract];

While Schneider ('897) suggest that the pointer graphic manager can be used on a computer system designed to give independent computing power to a group of user, wherein files, such as the pointer graphic manager program containing functions for modifying systems cursors, located on the remote system can be delivered, it does not specifically disclose that the information includes data containing the location of said first and second memory means. See Schneider ('897), col. 3, lines 58-63; col. 9, lines 7-18. However, it is well known in the art that the distribution of data over a network (e.g., pointer files or pointer graphics manager program file located remotely) transmits the location of the file accessed. As a result, it would have been obvious to one skilled in the art at the time the invention was made that the location (e.g.,

directory) of an accessed file would be transmitted when executing the apparatus of Schneider ('897) in a distributed environment because it enables communication between the remote and local systems.

as per claims 2-3, 9, 30-31, 37 and 67 [the files includes instructions for the display of windows associated with the functions of the pointer graphics manager, and the pointer graphic file corresponding to the pointer graphic to be displayed on the monitor of the user's terminal, see discussion independent claim 1, supra.].

as per claims 10-13 and 56, [the pointer graphic manager program file and the pointer graphic files can be on the same or different computers where the first and second memory corresponds to memory known in the art such as hard or floppy drive or CD-ROM of a computer as described above with respect to independent claim 1; the third memory is the RAM of the end-user station where the program or pointer graphic files retrieved from a remote location would be stored].

as per claims 14, 38 and 68, [as described in claim 1, in addition to the functions of the pointer graphics program other information can be distributed remotely by a server, such as an HTML page].

as per claims 17 and 41, [each pointer graphic file must have the PTR postscript and file name that corresponds to the pointer graphic to be displayed, see Schneider ('897), col. 6, lines 1-10].

as per claims 27, 48 and 54 [It would have been obvious to a developer creating the system of Schneider ('897) to maintain a log of the pointer of most interest to user in-order to provide and make those pointers available as well as to identify the pointer that are of least interest so that they can be eliminated. This would provide an efficient use of systems resources where memory space would not be consumed by pointer graphics files that are rarely used].

as per claim 28, [the a function that effects the display of the cursor such as the find object function of Schneider ('897) transmitted to the user terminal enables communication with a plurality of remote sites where the pointer graphics file can be located, see Schneider ('897),col. 7, lines 24-30].

as per claims 29 and 49, receiving a request at said server to provide cursor display instructions to said user terminal [receiving a request at a computer system designed to give independent computing power to a group of user to execute the pointer graphic manager program file on an end-user terminal to provide it functions, see Schneider ('897), col. 3, lines 58-63; col. 9, lines 7-18; col. 7, lines 24-30];

providing a specified information to said user terminal in response to said request, said information including cursor display instructions data and cursor information corresponding to a specified image [providing the pointer graphics manager program, including program functions, and pointer graphic file corresponding to pointer graphic];

transforming said initial cursor image displayed on said video monitor of said user terminal into the shape and appearance of said specific image in response to said cursor display instruction data [changing a pointer displayed into the shape and appearance of the pointer graphic modified by the functions of the pointer graphic manager program file, see Schneider ('897), col. 8, lines 52-col. 9, lines 1-67; abstract].

as per claims 47 and 55, [the pointer event or function determines how long the pointer will be displayed, wherein as the system changes states the pointer also changes states accordingly, Schneider ('897), col. 5, lines 65-col. 6, lines 1-65].

as per claim 50, an Internet browser configured to modify a cursor image [software (e.g., the operating system, programs and various device drivers), wherein the pointer graphic manager establishes pointer choices available to the user through modification of the operating systems pointers and is applicable to any graphical user interface that typically uses pointer graphics, see Schneider, col. 5, lines 47-55; col. 8, lines 55-60.]

means for retrieving information from a remote server intended to be displayed on said video monitor, said information comprising a cursor display instruction data wherein said cursor display instruction data includes information corresponding to said specific image [means for retrieving pointer graphic manager program file data from a remote resource intended to be displayed on a monitor, the file comprising providing for the find object function and wherein the find object function has the function of locating a pointer graphic file corresponding to pointer graphic, see Schneider ('897), col. 7, lines 25-32; col. 8, lines 55-60];

means for recognizing said cursor display instruction data [means for retrieving the pointer graphic];

means for executing a cursor display code, said cursor display code including information in response to which said cursor image is modified to said specific image [means for executing the pointer graphic manager program, the program including the find object function in response to which the cursor image is replaced by the pointer graphic selected, see Schneider ('897), col. 7, lines 25-32; col. 8, lines 55-60].

as per claims 51-52, [the pointer graphics manager program file can be located on a predefined server not stored on the users terminal].

as per claims 57-60, [it is old and well know in the art to provide for animation and audio of pointers].

as per claim 61, [pointer graphic associated with pointer graphic file transmitted corresponds to a portion of the data displayed on the monitor of the users].

8. Claims 4-8, 15, 18-23, 32-36, 42-46 and 62-66 are rejected under 35 U.S.C. 103(a) as being unpatentable over Schneider (US Patent # 5,710,897) in view of Judson (US Patent # 5,572,643). RB

The combination of Schneider ('897) and Judson ('643) teaches the applicant's invention as claimed in **claims 4-8, 15, 18-23, 32-36, 42-46, and 62-66** as follows:

as per claims 4, 15, 32, and 62, Schneider ('897) teaches that a pointer graphic is provided for various states, events or commands (e.g., functions) of the system and changes depending on the mode of operation. See Schneider ('897), abstract; col. 5, lines 65-col. 6, lines 1-10. For example, Schneider ('897) provides a wait pointer that is invoked when the system is busy. See Schneider ('897), col. 6, lines 13-17. Schneider ('897) also discloses that the pointers may be edited or modified as desired by the user and stored for later use. See Schneider ('897), id. However, Schneider is silent on providing a pointer graphic for when the system is in the mode of displaying advertising material. Judson ('643), provides for the display of advertising material during the downtime (i.e., when the system is busy) of downloading a hypertext document on a web page. See Judson ('643), abstract. One having ordinary skill in the art would recognize that the downtime and downloading of advertising material that occurs in the apparatus of Judson ('643) is an event and thus the pointer could be provided that changed to represent this event. One having ordinary skill in the art would also recognize that each pointer graphic depicted in the system of Schneider ('897) relates to the current event or mode of operation of the system. As a result, it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide a customized pointer set using the system of Schneider ('897) for the event of displaying advertising material in the apparatus of Judson ('643), wherein the pointer graphic relates to the advertising material, because it would provide the user with an option of selecting a pointer that best suites there individual taste and sense of context interest

(i.e., whether they are more interested in knowing that the system is busy or that the system is displaying an advertisement).

as per claims 5-8, 33-36 and 63-66, [the advertising material can include advertisements messages, logos, etc., see Judson ('643), abstract.].

as per claims 18 and 42, [the system of Schneider will not change the pointer image if it is not found and provides the user with the ability to locate a pointer graphic using find object function, see Schneider ('897), col. 7, lines 24-30].

as per claims 19-22 and 43-46, [it is old and well know in the art to provide for animation and audio of pointers].

as per claim 23, [see discussion of claim 47, supra].

9. Claims 16, 40 and 53 are rejected under 35 U.S.C. 103(a) as being unpatentable over Schneider (US Patent # 5,710,897) in view of M. Brown, "WWW Plug-Ins Companion", 1996 (hereinafter "Plug-Ins"). RB

Schneider ('897) discloses an application that allows a user to modify a cursor at the operating system level. However, it does not disclose that it enables a browser's intrinsic capabilities to be increased. Plug-Ins discloses that Plug-Ins extend and enhance the capabilities of a browser. See Plug-Ins, pages 14-18. One having ordinary skill in the art would recognize that modifying a pointer on an operating systems level affects all the applications running on the system, including a browser application. One having ordinary skill in the art would also recognize that providing customized pointer sets to an operating system would provide a browser

application running on an end-user terminal with the ability to utilize resources not previously available to the browser application. As a result, it would have been obvious to one skill in the art at the time the invention was made to provide the apparatus of Schneider ('897) as a plug-in where it would allow a user to provide a browser running on an end-user's terminal to display and utilize pointers not previously available to the browser application.

Conclusion

10. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Capps (US Patents # 5,596,694, 5,544,295) teaches the use of animation.

Judson (US Patent # 5,737,619) teaches the downloading of information while waiting for a hypertext document to be downloaded.

Reilly et al. (US Patent # 5,740,549) discloses a system for displaying advertisements on an use terminal that has met a predefined idleness criteria.

11. Response to this action should be mailed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231. If applicant desires to fax a response, (703) 308-9051 may be used for formal communications or (703) 305-9724 for informal or draft communications. Please label "PROPOSED" OR "DRAFT" for informal facsimile communications. For after final responses, please label "AFTER FINAL" or "EXPEDITED PROCEDURE' on the document. Hand

delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington VA., Sixth Floor (Receptionist).

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chadwick A. Jackson, whose telephone number is (703) 308-9572. The examiner can normally be reached Mon-Thu from 7:30 a.m. - 6:00 p.m. ET. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Kim, can be reached at (703) 305-3821

13. Communication via Internet e-mail regarding this application, other than those under 35 U.S.C. 132 or which otherwise require a signature, may be used by the applicant and should be addressed to [**matt.kim@uspto.gov**]. All Internet e-mail communications will be made of record in the application file. PTO employees do not engage in Internet communications where there exists a possibility that sensitive information could be identified or exchanged unless the record includes a properly signed express waiver of the confidentiality requirements of 35 U.S.C. 122. This is clearly set forth in the Interim Internet Usage Policy published in the Official Gazette of the Patent and Trademark Office on February 25, 1997 at 1195 OG 89.

14. Any inquiry of a general nature or relating to the status of this application or proceedings should be directed to the group receptionist whose telephone number is (703) 305-3900.

Chadwick A. Jackson

July 6, 1998

RAYMOND J. BAYERL
PRIMARY EXAMINER
ART UNIT 2773

Notice of References Cited

Application No.	Applicant(s)		
08/882,580	Rosen et al.		
Examiner	Group Art Unit	Page 1 of 1	
C. Jackson	2773		

U.S. PATENT DOCUMENTS

*		DOCUMENT NO	DATE	NAME	CLASS	SUBCLASS
	A	5,710,897	Jan 20, 1998	Schneider	345	334
	B	5,737,619	Apr. 7, 1998	Judson	395	761
	C	5,740,549	Apr. 14, 1998	Reilly et al	705	14
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

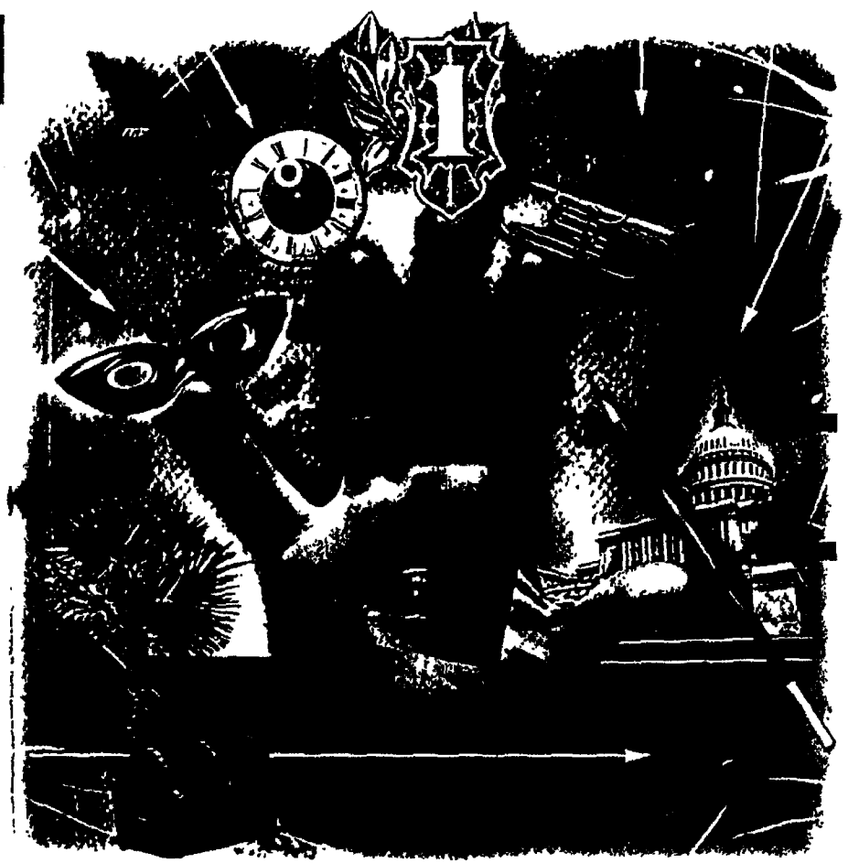
*		DOCUMENT NO.	DATE	COUNTRY	NAME	CLASS	SUBCLASS
	N						
	O						
	P						
	Q						
	R						
	S						
	T						

NON-PATENT DOCUMENTS

*		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	U	M. Brown, "www Plug-In Companions", 1996, pg 14-18	1996
	V		
	W		
	X		

A Copy of this reference is not being furnished with this Office action.
(See M.P.E.P. 707.05(a).)

#5



WWW PLUG-INS COMPANION

Que

Create powerful Web pages with Netscape™ plug-i

Works with Netscape and Internet Explorer

Incorporate multimedia, applications, and other li content into Web pages

Learn how to use popular multimedia plug-ins inclu Shockwave™ and Real Aud

CD-ROM includes over 100 VRML, sound, graphics, multimedia, prod and navigational plug-ins along wi development tools and server soft

$39.99

CD-ROM included

Works with Netscape™ and Internet Explorer™!

Brown

Que

World Wide Web

WWW PLUG-INS COMPANION

Mark B



WWW PLUG-INS COMPANION

Written by Mark R. Brown with

Simeon M. Greene • Galen Grimes • John Jung
Bernie Roehl • David Wall • Joe Weber

QUE®

WWW Plug-Ins Companion

Copyright© 1996 by Que® Corporation.

All rights reserved. Printed in the United States of America. No part of this book may be used or reproduced in any form or by any means, or stored in a database or retrieval system, without prior written permission of the publisher except in the case of brief quotations embodied in critical articles and reviews. Making copies of any part of this book for any purpose other than your own personal use is a violation of United States copyright laws. For information, address Que Corporation, 201 W. 103rd Street, Indianapolis, IN, 46290. You may reach Que's direct sales line by calling 1-800-428-5331.

Library of Congress Catalog No.: 96-69599

ISBN: 0-7897-0845-0

This book is sold *as is*, without warranty of any kind, either express or implied, respecting the contents of this book, including but not limited to implied warranties for the book's quality, performance, merchantability, or fitness for any particular purpose. Neither Que Corporation nor its dealers or distributors shall be liable to the purchaser or any other person or entity with respect to any liability, loss, or damage caused or alleged to have been caused directly or indirectly by this book.

99 98 97 96 4 3 2 1

Interpretation of the printing code: the rightmost double-digit number is the year of the book's printing; the rightmost single-digit number, the number of the book's printing. For example, a printing code of 96-1 shows that the first printing of the book occurred in 1996.

CHAPTER 1

How Plug-Ins Work

Although Netscape is a pretty versatile Web browser, you'll still encounter many files on the Web that Netscape can't display—video files, audio files, [illegible] files, strange document formats, and even compressed files. [illegible] play or display these files online, you'll need to install helper plug-ins or helper applications.

Plug-ins [illegible] supplement Netscape's native capabilities, expanding and enhancing the type of content that can be displayed over the World Wide Web and corporate [illegible]. Using plug-ins, your Web browser can display animation, multimedia, audio, interactive applications, and video inline, right on the page, without launching external helper application programs.

Unlike a browser's built-in display capabilities, which are limited to generic file formats like GIF and JPEG images, plug-ins offer [illegible] extension that can include just about any content [illegible]

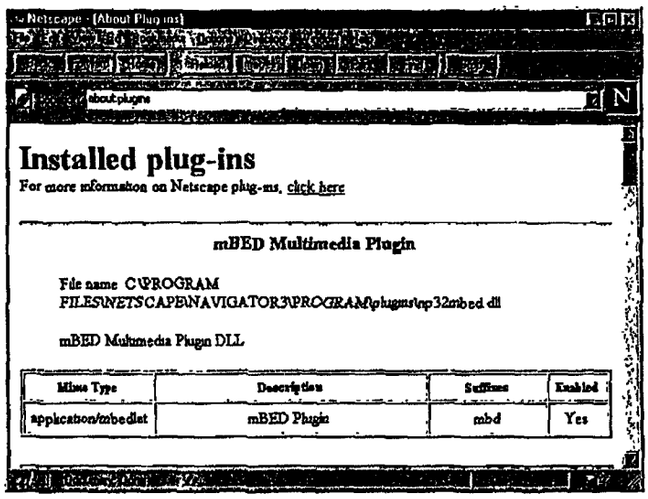


FIG. 1.4
The internally generated About Plug-Ins Netscape page lists the MIME types for which you have plug-ins installed.

NOTE You can find out more about MIME types by obtaining the Internet Working Group's Request for Comments (RFC) document on the topic. You can download this document by pointing Netscape to the following address:

ftp://ds.internic.net

Look for the directory /RFC and the file name RFC1521.TXT.

You can also enter into discussions about MIME on UseNet. Just point Netscape's newsreader to the group **comp.mail.mime.**

How Plug-Ins "Plug In"

Plug-ins are simply feature add-ons that can understand and interpret files that Netscape can't handle itself. They extend the capabilities of Netscape in much the way that plug-in software modules are used to extend the capabilities of other products such as Adobe PhotoShop. They are essentially transparent, appearing as enhancements and supplements to the Netscape browser itself. The Netscape user interface remains relatively constant no matter what plug-ins are installed—if you're displaying an inline QuickTime movie, for example, the parts of the display that handle page navigation, scrolling, and so on aren't affected by the plug-in's presence.

In more technical terms, plug-ins are dynamic code modules that are a part of Netscape's application programming interface (API) for integrating third-party software into Netscape. It's a part of Netscape Corporation's "open systems" philosophy regarding Netscape Navigator; this approach allows third-party developers to use Netscape to integrate their products into the Web seamlessly.

NOTE Although this book often refers specifically to Netscape, Microsoft's Internet Explorer can also use plug-ins. Over time, you can expect to see many more plug-ins released specifically for Internet Explorer. Almost everything that this book says about Netscape plug-ins also applies directly to Internet Explorer plug-ins.

In fact, the latest version of Internet Explorer can even use Netscape plug-ins directly, so that you can add your favorite Netscape plug-ins to either browser. ■

Plug-ins enable you to customize Netscape to interact with third-party products and industry media standards. They are meant to supplement and complement, not supplant, other interapplication architectures such as Windows OLE (Object Linking and Embedding) and Java. Plug-ins can accomplish the following tasks:

- Create a window in Netscape for displaying information, as in a Video for Windows movie player.
- Execute an application such as a Musical Instrument Digital Interface (MIDI) player.
- Generate data for Netscape or other plug-ins. For example, a plug-in might create an index on the fly.
- Provide interapplication communication. For example, a plug-in might transfer data to a stand-alone spreadsheet program.
- Override a native Netscape capability and supply its own implementation. For example, a plug-in might provide an improved .GIF viewer
- Link to and receive data from Uniform Resource Locators (URLs). For example, some plug-ins can download stock quotes.

Because plug-ins are platform-specific, you must have a different version of each plug-in for every operating system that you use, such as Windows, Windows 95, UNIX, or the Macintosh operating system. Regardless of your platform, however, Netscape plug-ins should be functionally equivalent across all platforms.

TIP Many plug-ins ship with the copy of Netscape that you purchased or downloaded. These plug-ins are already designed for your platform. However, if you find other plug-ins that you want either to purchase or download from the Internet, make sure that they are for your specific platform.

For most users, the use of plug-ins is totally transparent. When Netscape starts up, it checks to see which plug-ins have been installed; if it encounters a data MIME type for which a plug-in is registered, Netscape launches the plug-in to handle that data. When you leave the page that contains that data, the plug-in unloads, freeing up system resources.

You activate a plug-in only by opening a Web page that initiates it; usually, you don't even see the plug-in at work. For example, after you install the Shockwave for Director plug-in,

you notice no difference in the way that Netscape functions until you come across a Web page that features Shockwave content (see fig. 1.5).

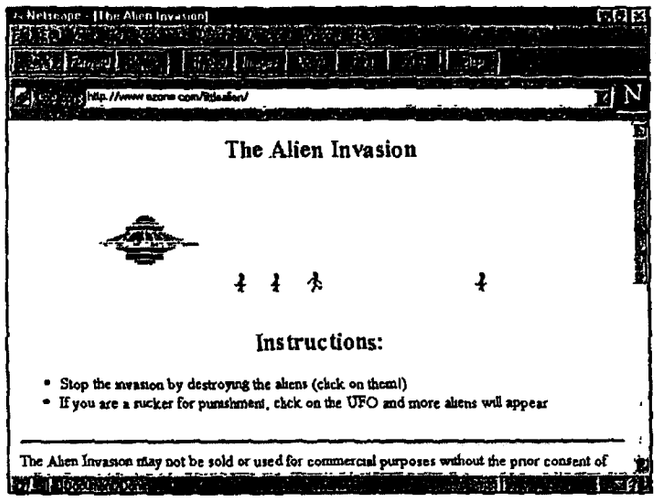


FIG. 1.5
The Shockwave for Director plug-in in action, letting you blast Space Aliens inline on a Web page.

When the Netscape client launches, it notes any available plug-ins, but does not load any into random access memory (RAM). This way, a plug-in resides in memory only when needed; however, you still need to be aware of memory allocation, because many plug-ins can be in use at any one time. Plug-ins simply reside on disk until you need them. As soon as you move to another HTML page that doesn't require a plug-in, it is deleted from RAM.

At its most fundamental level, a plug-in can access a URL and retrieve MIME data just like a standard Netscape client. This data is streamed to the plug-in as it arrives from the network, making it possible to implement viewers and other interfaces that can display information progressively as it arrives from the server.

If a plug-in requires more data than a single data stream can supply, the plug-in can request multiple, simultaneous data streams, so long as the user's system supports such data streams.

Plug-ins can also handle data in the "old-fashioned" way: caching data for display only when it has all been downloaded. For instance, a plug-in can draw a simple frame and introductory graphic while the bulk of the data is streaming off the network into the Netscape cache.

If Netscape or another plug-in needs data while a plug-in is active, the plug-in can generate the needed data. Therefore, plug-ins not only process data, but also generate it. For example, a plug-in can be a data translator or filter.

The integration of plug-ins with Netscape is quite elegant and flexible, making the most of asynchronous processes and multithreaded data. All plug-ins are associated with a MIME type not native to the Netscape client, and can be associated with multiple MIME types. Netscape can concurrently run multiple instances of the same plug-in if the page contains several plug-in-compatible data files of the same type.

Netscape's plug-in API also attempts to address the concerns of programmers, providing a high degree of flexibility and cross-platform support to plug-in developers.

Plug-ins are a godsend for applications developers, who can extend the utility of existing products into the burgeoning Internet market by developing a quick and easy Netscape plug-in that reads existing data files, instead of developing a whole new product. Not only does this save developers time and effort, it lets them ride into a huge market on the coattails of Netscape, the Internet's most popular browser. Applications developers are happy, because their market expands quickly and almost for free; Web content developers are happy, because they have new formats that they can provide; Web users are happy, because they have new ways in which to use the Web; and even Netscape is happy, because its Web browser becomes more powerful and useful without any additional effort on its part. Everybody wins!

CAUTION

You should be keenly aware of the potential system security problems that plug-ins present. *Plug-ins have full access to all the data on your computer system.* They are written by third parties whom you may or may not know. Plug-ins are delivered by servers over whom you have no control. Each of these factors poses a potential security risk. Make sure that you trust the plug-in developer and the plug-in server before you install a plug-in on a system in which a security breach could cause serious problems.

For example, a plug-in could easily be developed that scans through the Windows 95 registry looking for passwords, then passes them back to the plug-in developer through the Internet. Although you're unlikely to encounter such an insidious plug-in at a major developer's site, you might want to exercise more caution when downloading a plug-in from an individual's Web site.

Almost worse is the case of a poorly written plug-in that means no harm, but through sloppy programming manages to reformat your hard drive or trash your system registry. *Be careful.*

The Three Kinds of Plug-Ins

After you install a plug-in on your machine and a Web page initiates the plug-in, it manifests itself in three potential ways:

- Embedded
- Full-screen
- Hidden

An *embedded* plug-in appears as a visible, rectangular window integrated into a Web page. This window might not look any different than a window created by a graphic, such as an embedded .GIF or JPEG picture. The main difference between the previous windows supported by Netscape and those created by plug-ins is that plug-in windows can support a much wider range of interactivity and movement, and thereby remain dynamic rather than static.

Embedded plug-ins can read and note mouse clicks, mouse location, mouse movement, keyboard input, and input from virtually any other input device. In this way, a plug-in can support the full range of user events required to produce sophisticated applications.

Examples of embedded plug-ins include a Moving Picture Expert Group (MPEG) movie player, the QuickTime movie player, the Shockwave for Macromedia Director player, or Video for Windows player like VDOLive (see fig. 1.6).

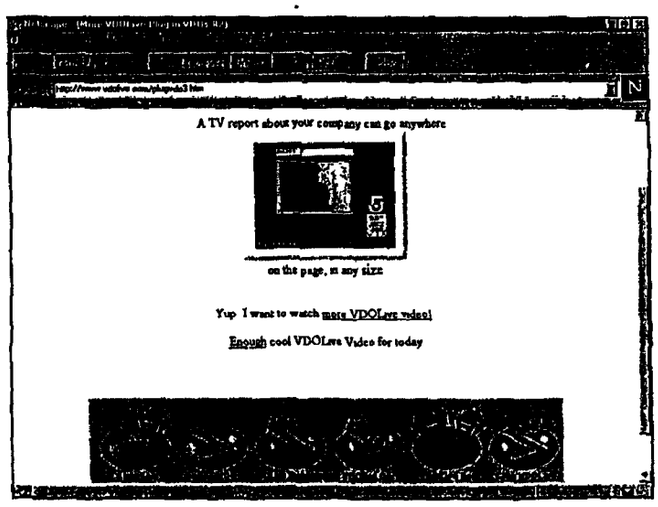


FIG. 1.6
VDOLive is an example of an embedded plug-in that can seamlessly integrate a window within an HTML document.

A *full-screen* plug-in takes over the entire current Netscape window to display its own content. This is necessary when a Web page is designed to display data that HTML does not support. An example of this type of plug-in is the Adobe Acrobat viewer (see fig. 1.7).

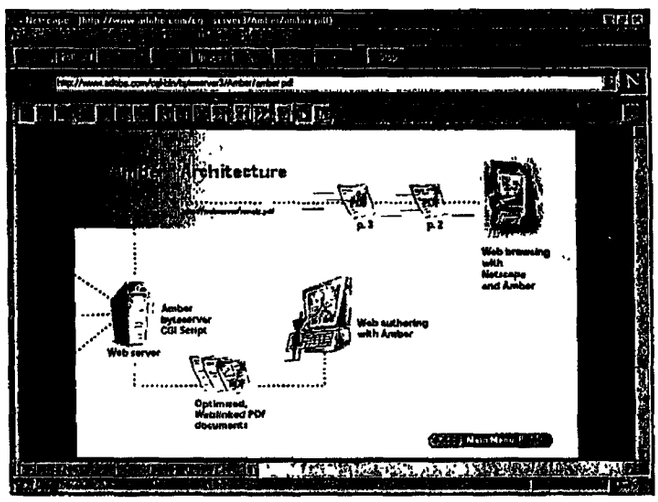


FIG. 1.7
The Adobe Acrobat viewer is a full-screen plug-in that even incorporates its own control bar.

If you view an Acrobat page using the Netscape plug-in, the page displays just like any other Web page, but retains the look and functionality of an Acrobat document viewed in Adobe's stand-alone viewer. For instance, if a Web site uses Acrobat to display an online manual for a product, the site might enable you to scroll, print, and interact with the page just as if the stand-alone Acrobat Reader program were displaying it.

You can invoke some plug-ins in either embedded or full-screen mode. For example, you can launch Netscape's Video for Windows sample plug-in as either an inline window or a full-screen player, depending on the display mode specified in the page that you access.

A *hidden* plug-in doesn't have any visible elements, but works strictly behind the scenes to add some feature to Netscape that is otherwise not available. Examples of possible hidden plug-ins include MIDI music players or file decompression engines. A MIDI player plug-in could read MIDI data from a Web page whenever it's encountered, and automatically play the data through your local hardware without so much as even displaying a control panel. Similarly, a decompression engine might function much the way that it does on commercial online services—decompressing data in real time in the background—or delaying decompression until the user logs off the Internet.

Regardless of the plug-ins that you use, and whether they are embedded, full-screen, or hidden, the rest of Netscape's user interface remains relatively constant and available. Therefore, even if Netscape's main window is displaying an Acrobat page, you still can access Netscape's menus and navigational controls.

What Plug-Ins Can (and Can't) Do

The current version of the Netscape plug-in API supports four broad areas of functionality. Plug-ins can do the following:

- Draw into and receive events from a native window element that is a part of the Netscape window hierarchy
- Obtain MIME data from the network through URLs
- Generate data for consumption by Netscape or other plug-ins
- Override and implement protocol handlers

Netscape plug-ins are ideally suited to take advantage of platform-independent protocols, architectures, languages, and media types such as Java, Virtual Reality Modeling Language (VRML), and MPEG. Although plug-ins should be functionally equivalent across platforms, they should also complement platform-specific protocols and architectures such as OLE 2.

This book loosely groups plug-ins into three major categories: multimedia, VRML, and business applications. Multimedia plug-ins are the heart and soul of plug-ins development. This category includes inline players for audio, video, animations, and multimedia presentations. VRML plug-ins display three-dimensional, online "worlds" created using the Virtual Reality Modeling Language. These "worlds" are filled with 3-D objects that you can move around and through and with which you can (occasionally) interact. Quite a few VRML plug-ins are available, each having its own strengths, weaknesses, and additions to the VRML standard. Business plug-ins are a hot area of development, with plug-ins now available for the Adobe Acrobat portable document format, inline Excel-compatible spreadsheets, Word documents, and more. Some plug-ins even enable you to develop your own applications for use with Netscape; early examples include indexing programs, stock quote grabbers, and even a graphic world clock.

You can even write your own plug-ins. Netscape offers software developer's kits (SDKs) for Windows, Macintosh, and UNIX system plug-ins development. These kits are available for free downloading from Netscape's Web site. Chapter 21, "Creating Your Own Plug-Ins," provides details on how to download, install, and use these SDKs.

File History Content Report

The following content is missing from the original file history record obtained from the United States Patent and Trademark Office. No additional information is available.

Document Date - 1998-07-13

Document Title - Notice of Formal Drawings Required

This page is not part of the official USPTO record. It has been determined that content identified on this document is missing from the original file history record.

GP2312



PATENT

Docket No. 658-002

RECEIVED
98 JUL 30 AM 10:12
GROUP 2700

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s)	:	Rosen et al.	Group Art Unit: 2312
Serial No.	:	08/882,580	Examiner: Not yet assigned
Filed	:	June 25, 1997	
For	:	SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE	

CERTIFICATE OF MAILING (37 C.F.R. 1.8a)

HON. COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

Sir:

I hereby certify that the attached Associate Power of Attorney and Return Postcard along with any paper(s) referred to as being attached or enclosed and this Certificate of Mailing are being deposited with the United States Postal Service on the date shown below with sufficient postage as first-class mail in an envelope addressed to the: Commissioner of Patents and Trademarks, Washington, D.C. 20231.

Respectfully submitted,

SOFER & HAROUN, L.L.P.

By: _____
Stacey Taitt

Date: 7/24/98

Mailing Address:

SOFER & HAROUN, L.L.P.
342 Madison Avenue, Suite 1921
New York, New York 10173
Tel:(212) 697-2800
Fax:(212) 697-3004

Serial No.: 08/882,580

OIPE JC57
JUL 27 1998
PATENT & TRADEMARK OFFICE

Docket No.: 658-002P

#6

ASSOCIATE POWER OF ATTORNEY

Please recognize: ADAM C. SOLOMON, Reg. No. 43,142;

Comet Systems, Inc.
180 Maiden Lane, 20th Floor
New York, New York 10007

RECEIVED
98 JUL 30 AM 10:12
GROUP 2700

as an associate attorney in the above-mentioned application, with full power to prosecute said application, to make alterations and amendments therein, and to transact all business in the Patent and Trademark Office connected therewith.

Telephone calls should be made to Joseph Sofer by dialing Area Code (212) 697-2800.

In addition, all written communications are to be addressed to Sofer & Haroun, LLP, 342 Madison Avenue, Suite 1921, New York, NY 10173.

Joseph Sofer
Attorney for Applicant(s)
Reg. No. 34,438

Transaction History Date 1998-12-17
Date information retrieved from USPTO Patent Application Information Retrieval (PAIR) system records at www.uspto.gov

GAU 2773

09/97 FORM 9

Application No.: 08/882,580
Filed: June 25, 1997
Group Art Unit: 2773

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
Ten Post Office Square
Boston, Massachusetts 02109
Telephone: (617) 542-2290
Telecopier: (617) 451-0313

RECEIVED
DEC 22 1998
Group 2700

DEC 17 1998

Date: December 14, 1998

ASSISTANT COMMISSIONER FOR PATENTS
Washington, D.C. 20231

Attorney
Docket No.: COMET-001XX

#7

Sir:

In re application of: James S. Rosen, et al.

Entitled: SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE

Transmitted herewith is an **amendment** in the above-identified application. The following checked items are applicable:

[X] A Petition for Extension of Time for 2 month(s) is hereby made, under §1.136(a); a check in the amount of $200.00 is enclosed per §1.17.

[X] In the event a Petition for Extension of Time is required by this paper and not otherwise provided, such Petition is hereby made and authorization is provided herewith to charge Deposit Account No. 23-0804 for the cost of such extension.

[X] David A. Dagg is hereby appointed Associate Attorney by:
Registration No.: 37,809

Attorney of Record: Victor B. Lebovici
Registration No.: 30,864

[X] Other: Revocation of Power and Appointment of New Power of Attorney; substitute Abstract.

CLAIMS AFTER AMENDMENT:	MINUS PRIOR PAID CLAIMS:	EQUALS PRESENT EXTRA CLAIMS:	RATE:	ADDITIONAL FEE:
Independent	10 - 3	= 7	x $78.00 =	546.00
Total	132 - 71	= 61	x $18.00 =	1,098.00
[ ] Multiple Dependent Claims (1st presentation)			+ $260.00 =	0.00
			SUBTOTAL ADDITIONAL FEE	1,644.00
Small Entity filing, divide by 2. Previously submitted or verified statement must be attached (per §1.9, §1.27, §1.28)				822.00
			TOTAL ADDITIONAL FEE	822.00

[ ] No additional fee. [X] The fee has been calculated above; a check in the amount of $ 822.00 is enclosed.

12/21/1998 [illegible] 00000170 [illegible]
01 FC:216 190.00 OP

[X] **The Commissioner is hereby authorized to charge payment of any additional filing fees under §1.16 associated with this communication or credit any overpayment to Deposit Account No. 23-0804.**

I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on December 14, 1998.

Attorney of Record: David A. Dagg
Registration No.: 37,809

SUBMIT IN TRIPLICATE

DAD/rec
Enc.
132804

Repln. Ref: [illegible]
DAA:23804
FC: 704

OIPE JC54 DEC 17 1998 PATENT & TRADEMARK OFFICE

PATENT RECEIVED
DEC 22 1998
Group 2700

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application	:	James S. Rosen et al.
Application No.	:	08/882,580
Filed	:	June 25, 1997
For	:	SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE
Examiner	:	C. Jackson
Attorney's Docket	:	COMET-001XX

#8/a Tich 12-23-98

Group Art Unit: 2773

* *

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on December 17, 1998.

By: David A. Dagg
David A. Dagg
Registration No. 37,809
Attorney for Applicant(s)

* *

AMENDMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

In response to the Office Action dated July 13, 1998, please amend the above identified patent application as follows:

12/21/1998 AHONDAMNE 00000190 08882580

02 FC:202 273.00 OP
03 FC:203 549.00 OP

In the Specification:

Please amend the specification as follows:

On page 3, at line 5, please insert --,-- following "smaller".

On page 7, at line 15, please insert --a-- after "as".

On page 7, at line 20, please insert --a-- after "on".

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

In the Claims:

Please amend the Claims as follows:

Sub C' a'

1. (Amended) A server system for modifying a cursor image to a specific image having a desired shape [shaped] and appearance displayed on a display [video monitor] of a remote user's terminal, said system comprising:

[a first memory means for storing] cursor image [information] data corresponding to said specific image;

[a second memory means for storing data corresponding to a] cursor display code, said cursor display code operable to modify [containing information in response to which] said cursor image [is modified to said specific image];

a first server computer for transmitting [a] specified content information to said remote user terminal, said specified content information including at least one [a] cursor display instruction indicating [data containing the] a location of said [first and said second memory means] cursor image data, [such that said user terminal in accordance with] said cursor display instruction [data] and said cursor display code operable to cause said user terminal to display [displays] a modified cursor image on said user's display [video monitor] in the shape and appearance of said specific image, wherein said specified content information is

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

transmitted to said remote user terminal by said first server computer responsive to a request from said user terminal for said specified content information.

2. (Amended) The [A] server system in accordance with claim 1, wherein said specified content information further comprises information [that is intended] to be displayed on said display [video monitor] of said user's terminal, and wherein said cursor display code is operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image responsive to displaying of said information to be displayed on said display of said user's terminal.

3. (Amended) The [A] server system in accordance with claim 2, wherein said specific image relates [corresponds] to at least a portion of said information [intended] to be displayed on said display [video monitor] of said user's terminal.

4. (Amended) The server system in accordance with claim 3, wherein said specific image comprises advertising material related to at least a portion of said information [intended] to be displayed on said display of said user's terminal.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

6 7. (Amended) The server system in accordance with claim ~~4~~ 3, wherein said advertising material further comprises images of a good or a service corresponding to said information [intended] to be displayed on said display of said user's terminal.

7 8. (Amended) The server system in accordance with claim ~~4~~ 3, wherein said advertising material further comprises messages relating to said information [intended] to be displayed on said display of said user's terminal.

8 9. (Amended) The server system in accordance with claim ~~3~~ 2, wherein said specific image has a shape and appearance corresponding to [representing the content of] said information [intended] to be displayed on said display of said user's terminal.

9 ~~10~~. (Amended) The server system in accordance with claim 1 [2], wherein at least a portion of said cursor image data and said cursor display code [first and second memory means] are disposed locally to said first server computer.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

10
~~11~~. (Amended) The server system in accordance with claim 1 [2], wherein at least a portion of said cursor image data and said cursor display code [first and second memory means] are disposed within [at least] a second server computer located remotely to said first server computer.

11
~~12~~. (Amended) The server system in accordance with claim 1 [11], wherein at least a portion of said cursor image data and said cursor display code [the contents of said first and second memory means] are [also stored in a third memory means] disposed locally to said user's terminal.

Claim 20 cont.

12 10
~~13~~. (Amended) The server system in accordance with claim ~~11~~ wherein said first server computer in response to a request from said user terminal transmits information stored in said second server computer [at least one of said first and second memory means] to said user terminal.

13
~~14~~. (Amended) The [A] server system in accordance with claim 1 [2], wherein said specified content information [intended to be displayed] is transmitted in the form of HTML files that define a web page.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

14 ~~18~~. (Amended) The [A] server system in accordance with claim 13 ~~14~~, wherein said cursor image [information] data includes at least in part [corresponds to] an advertisement for goods or services contained in said web page.

15 ~~16~~. (Amended) The [A] server system in accordance with claim 1 [15], wherein said user terminal includes a browser [extension] application responsive to said cursor display instruction, [data so that] said browser [extension] application executing [executes] said cursor display code using [by employing] parameters defined in said cursor display instruction [instructions].

16 ~~17~~. (Amended) The [A] server system in accordance with claim 1, [3 wherein] said cursor [information] display instruction further comprising [comprises] an image identifier indicating said cursor image [that corresponds to] data corresponding to [representing] said specific image [stored in said server system].

17 ~~18~~. (Amended) The [A] server system in accordance with claim 1 [4,] wherein said first server computer transmits said cursor image data [representing said specific image] in response to a request received from said remote user terminal indicating that a copy of

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

said cursor image data [representing said specific image] is not stored in said remote user terminal.

18
~~19~~. (Amended) The [A] server system in accordance with claim 1 [4,] wherein said cursor display instruction [data] further comprises an image identifier that corresponds to a graphic animation sequence [stored in said server system].

Cancel.

19 18
~~20~~. (Amended) The [A] server system in accordance with claim ~~19~~, wherein said cursor display instruction [data] further comprises instructions operable to modify said [remote user terminal's cursor] specific image to display said graphic animation sequence [in response to activation of a link to a hypertext document selected by said user].

20
~~21~~. (Amended) The [A] server system in accordance with claim 1 [20], wherein said cursor display instruction [data] further comprises an audio identifier that corresponds to an audio information sequence [stored in said server system].

21 20
~~22~~. (Amended) The [A] server system in accordance with claim ~~21~~, wherein said cursor display instruction [data] further comprises instructions operable [to said remote user's terminal] to play an

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

audio clip corresponding to said audio information sequence [in conjunction with said modification of said cursor display].

22 ~~23~~. (Amended) The [A] server system in accordance with claim 1 [4], wherein said cursor display instruction [data] further comprises information that controls a [the] duration of time said specific [cursor] image is displayed on said display of said remote user's terminal [remote user's video monitor].

23 ~~24~~. (Amended) The [A] server system in accordance with claim 1 [2], wherein said specified content information is transmitted in the form of one or more hypertext objects [are defined using the ActiveX® Internet programming technology].

24 ~~25~~. (Amended) The [A] server system in accordance with claim 1 [2], wherein said specified content information [set of data files transmitted in the form of hypertext objects include] includes instructions executable by a virtual machine on said user terminal [are defined using the Java® Internet programming language].

25 ~~26~~. (Amended) The [A] server system in accordance with claim 1 [2], wherein said specified content information [set of data files transmitted in the form of hypertext objects] includes HTML tags

Cont.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

recognized by said cursor display code [are defined using the VBSScript® Internet programming language].

26
~~27~~. (Amended) The [A] server system in accordance with claim 1 [3], wherein said cursor display code generates [instruction data provides] usage data [to said server system] for calculating usage statistics of said specific image.

29
~~28~~. (Amended) The [A] server system in accordance with claim 1 [3], wherein said cursor display instruction [data] transmitted to said remote user terminal initiates communication with a plurality of server systems for obtaining additional cursor image [instruction] data.

Sub C3

~~29~~. (Amended) A method for [of] modifying an initial cursor image displayed on a display [video monitor] of a user terminal connected to at least one server [one or more servers], comprising [which comprises the steps of]:

receiving a request at said at least one server to provide specified content information [cursor display instructions] to said user terminal;

providing [a] said specified content information to said user terminal in response to said request, said specified content

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

information including at least one cursor display instruction [data] and at least one indication of cursor image data [information] corresponding to a specific image; and

transforming said initial cursor image displayed on said display [video monitor] of said user terminal into the shape and appearance of said specific image in response to said cursor display instruction [data].

30. (Amended) The [A] method in accordance with claim 29 wherein said [step of providing] specified content information includes [further comprises the step of providing] information that is [intended] to be displayed on said display [video monitor] of said user's terminal, and wherein said cursor display instruction indicates a cursor display code operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image responsive to displaying of said specified content information on said display of said user's terminal.

31. (Amended) The [A] method in accordance with claim 30, wherein said [step of] transforming further comprises [the step of] executing said [a] cursor display code so as to display said specific cursor image while [such that it corresponds to] at least

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

a portion of said information [needed] to be displayed on said display [video monitor] of said user's terminal.

32. (Amended) The [A] method in accordance with claim 30 [31], wherein said [step of] displaying of said specific [cursor] image further comprises [the step of] displaying advertising material related to at least a portion of said information [intended] to be displayed.

31 ~~33~~. (Amended) The [A] method in accordance with claim 30 ~~22~~, wherein said advertising material further comprises a brand logo.

32 ~~34~~. (Amended) The [A] method in accordance with claim 30 ~~22~~, wherein said advertising material further comprises a corporate mascot.

33 ~~35~~. (Amended) The [A] method in accordance with claim 30 ~~22~~, wherein said advertising material further comprises images of a good or a service corresponding to said information [intended] to be displayed.

34 ~~36~~. (Amended) The [A] method in accordance with claim 30 ~~22~~, wherein said advertising material further comprises messages relating to said information [intended] to be displayed.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Sub C5

37. (Amended) The [A] method in accordance with claim 30 [31], wherein ~~said~~ [step of displaying said] specific image [further comprises the step of displaying a cursor image having] has a shape and appearance relating to [representing the content of] said information [intended] to be displayed.

A Cont.

36 28

~~38~~. (Amended) The [A] method in accordance with claim ~~29~~ [30], wherein said [step of providing said] specified content information further comprises [the step of providing information in the form of] HTML files that define a web page.

37 36

~~39~~. (Amended) The [A] method in accordance with claim ~~38~~, wherein said [cursor display instruction data further comprises cursor information including specified images] cursor image data corresponds [corresponding] to an advertisement for goods or services contained in said web page.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

38 ~~40~~. (Amended) The [A] method in accordance with claim 28 ~~29~~ [31], wherein said [step of] transforming further comprises [the steps of]:

employing a browser [extension] application including said cursor display code responsive to said cursor display instruction [data]; and

executing said cursor display code by employing parameters defined in said cursor display instruction [data].

Accept

39 ~~41~~. (Amended) The [A] method in accordance with claim 28 ~~29~~ [40] wherein said [step of providing] specified content [cursor] information comprises [the step of providing] an image identifier that corresponds to the location of data representing said specific image.

40 ~~42~~. (Amended) The [A] method in accordance with claim 28 ~~29~~, further comprising [the step of] transmitting said cursor image data [representing said specific image] in response to a request received from said remote user terminal indicating that a copy of said cursor image data [representing said specific image] is not stored in said remote user terminal.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

41 28
~~46~~. (Amended) The [A] method in accordance with claim ~~29~~ [42], wherein said [step of providing] cursor display instructions [specified information] further comprises [the step of providing] an image identifier that corresponds to a graphic animation sequence.

42 41
~~44~~. (Amended) The [A] method in accordance with claim ~~43~~, further comprising [the step of] modifying said remote user terminal's cursor to display said graphic animation sequence [in response to activation of a link to a hypertext document selected by said user].

43 28
~~45~~. (Amended) The [A] server system in accordance with claim ~~29~~ [42], wherein said [step of providing] cursor display instruction [specified information] further comprises [the step of providing] an audio identifier that corresponds to an audio information sequence.

44 43
~~46~~. (Amended) The [A] server system in accordance with claim ~~45~~ [46], further comprising [the step of] playing an audio clip corresponding to said audio information sequence [in conjunction with said modification of said cursor display] responsive to displaying said specific image.

Amdt.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

45 28
~~47~~. (Amended) The [A] method in accordance with claim ~~29~~ further comprising [the step] controlling a [the] duration of time said specific [cursor] image is displayed on said remote user's display [video monitor].

46 28
~~48~~. (Amended) The [A] method in accordance with claim ~~29~~ [47], further comprising [the step of] providing usage data by said cursor display code [to said server system] for calculating usage statistics of said specific image, responsive to said cursor display instruction.

47 28
~~49~~. (Amended) A computer storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform the method steps in claim ~~29~~ [31].

Sub Cd
~~50~~. (Amended) An internet browser configured to modify a cursor image to a specific image having a desired shape [shaped] and appearance displayed on a display [video monitor] of a remote user's terminal, said browser comprising:

means for receiving [retrieving] specified content information from a remote server [intended to be displayed on said video monitor], said specified content information comprising information to be displayed on said remote user's terminal and at least one [a]

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

cursor display instruction [data], wherein said cursor display instruction [data includes information] indicates cursor image data corresponding to said specific image;

means for recognizing said cursor display instruction [data] in connection with processing said information to be displayed on said display; and

means for executing a cursor display code, responsive to said cursor display instruction, said cursor display code [including information in response to which said cursor image is modified to said specific image] being operable to modify said cursor image to said specific image.

51. (Amended) The internet browser in accordance with claim 50 further comprising means to retrieve said cursor image data [information] from a prespecified server, when said cursor image data [information] is not stored in said user terminal.

53. (Amended) The internet browser in accordance with claim 50 further comprising means [to communicate with a server system so as] to determine whether cursor display instructions received by said user terminal were transmitted by an authorized server.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

54. (Amended) The internet browser in accordance with claim 50 further comprising means for transmitting statistical information to a prespecified server so as to provide information relating to the usage of said specific image [cursor display instruction data].

55. (Amended) The internet browser in accordance with claim 50, wherein said specific [cursor] image reverts back to its original [cursor] shape and appearance after a prespecified duration.

56. (Amended) The internet browser in accordance with claim 55 further comprising means to store said cursor display code in a memory located locally to [within] said user terminal.

57. (Amended) The internet browser in accordance with claim 50 [55] further comprising means to provide audio clips corresponding to said display of said specific [cursor] image.

58. (Amended) The internet browser in accordance with claim 57 wherein information relating to said audio clips is [are] contained within said cursor display instruction [data].

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Sub C10

59. (Amended) The internet browser in accordance with claim 50 [57] further ~~comprising~~ means to provide animated images corresponding to said ~~specific~~ [cursor] image.

58 57

~~60.~~ (Amended) The internet browser in accordance with claim ~~59~~ wherein information relating to said animated images are contained within said cursor display instruction [data].

59 48

~~61.~~ (Amended) The internet browser in accordance with claim ~~50,~~ wherein said specific image corresponds to at least a portion of said information [intended] to be displayed on said display [video monitor] of said user's terminal.

A3 cont.

60 48

~~62.~~ (Amended) The internet browser in accordance with claim ~~50~~ [61], wherein said specific image comprises advertising material related to at least a portion of said [information intended to be displayed] information to be displayed on said display of said user's terminal.

A4

63 60

~~65.~~ (Amended) The internet browser in accordance with claim ~~62,~~ wherein said advertising material further comprises images of a good or a service corresponding to said [information intended to be

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

displayed] information to be displayed on said display of said user's terminal.

64 60 ~~60~~. (Amended) The internet browser in accordance with claim ~~62~~, wherein said advertising material further comprises messages relating to said [information intended to be displayed] information to be displayed on said display of said user's terminal.

65 48 ~~67~~. (Amended) The internet browser in accordance with claim ~~50~~ [61], wherein said specific image has a shape and appearance related to said [representing the content of said information intended to be displayed] information to be displayed on said display of said display of said user's terminal.

66 48 ~~68~~. (Amended) The internet browser in accordance with claim ~~50~~, wherein said specified content information [intended to be displayed] is transmitted in the form of at least one HTML file [files] that defines [define] a web page.

67 48 ~~69~~. (Amended) The internet browser in accordance with claim ~~50~~, wherein said specified content [specified] information [is defined using the ActiveX® Internet programming technology] is transmitted in the form of one or more hypertext objects.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

68 48

70. (Amended) The internet browser in accordance with claim 50, wherein said specified information content is executable at least in part by a virtual machine on said user terminal [specified information is defined using the Java® Internet programming language].

Ed Final.

69 48

71. (Amended) The internet browser in accordance with claim 50, wherein said specified information content includes at least one instruction in an interpreted programming language [specified information is defined using the VBScript® Internet programming language].

Please add the following new claims:

---

72. A system for replacing an existing cursor image with at least one new cursor image responsive to displaying contents of a web page, comprising:

15 a cursor display code module operable to process at least one cursor display instruction associated with said web page, said processing of said at least one cursor display instruction providing said new cursor image responsive to said web page being displayed, said at least one cursor display instruction including indication of said cursor image.

EINGARTEN, SCHURGIN,
AGNEBIN & HAYES LLP
EL (617) 542-2290
AX (617) 451-0313

73. The system as in claim 72, wherein said at least one cursor display instruction is stored within said web page.

74. The system as in claim 72, wherein said at least one cursor display instruction includes identification of said cursor display code module.

75. The system as in claim 72, wherein said cursor display code module is dynamically linkable with an internet browser.

76. The system as in claim 72, wherein said at least one cursor display instruction is stored in a server system, and wherein said web page includes indication of a location of said at least one cursor display instruction on said server system.

77. The system as in claim 72, wherein said at least one cursor display instruction includes indication of a location of said cursor image.

78. The system as in claim 77, wherein said indication of said cursor image includes a location of a local copy of said cursor image on a local computer system.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

79. The system as in claim 77, wherein said indication of said cursor image is a location of said cursor image on a server computer system.

80. The system as in claim 79, wherein said indication of said cursor image is a uniform resource locator.

81. The system as in claim 76, wherein said indication of said at least one cursor display instruction on said server system is a uniform resource locator.

82. A method for authenticating modification of a cursor image, comprising:

requesting specified content information from a first server system;

receiving said specified content information, said specified content information including at least one cursor display instruction, said cursor display instruction including authentication data;

transmitting, responsive to said specified content information, an authentication request to a second server system, said authentication request including said authentication data; and

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

receiving, from said second server system, an authentication message indicating whether said first server system is authorized to modify said cursor image.

83. The method of claim 82, further comprising executing said cursor display instructions responsive to said authentication message indicating that said first server system is authorized to modify said cursor image on said client system.

84. The method of claim 82, further comprising preventing execution of said cursor display instructions responsive to said authentication message indicating that said first server system is not authorized to modify said cursor image.

85. The method of claim 82, further comprising executing said cursor display instructions responsive to said authentication message indicating that said first server system is not authorized to modify said cursor image, and logging an unauthorized request event at said second server system.

86. The method of claim 82, wherein said authentication data includes identification of said first server system.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

87. The method of claim 82, wherein said authentication data includes an encrypted authentication code.

88. The method of claim 82, further comprising said client system requesting said authentication data responsive to receipt of said specified content information.

89. A method for authenticating an operation which modifies a cursor image displayed by a client system, comprising:

requesting specified content information from a server system;

receiving said specified content information, said specified content information including at least one cursor display instruction, said cursor display instruction operable to modify an existing cursor image displayed by said client system, said cursor display instruction including an encrypted authentication code;

decrypting said encrypted authentication code;

determining whether to execute said cursor display instruction in response to said decrypted authentication code.

90. The method of claim 89 wherein said authentication code is based at least in part on an identity of said server system.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

91. The method of claim 90 wherein said identity of said server system is an IP address of said server system.

92. The method of claim 89 wherein said authentication code is based at least in part on a time at which it is generated.

93. The method of claim 89 wherein said authentication code is based at least in part on a date on which it is created.

94. The method of claim 89 wherein said authentication code is based at least in part on a value transmitted from said client system to said server system.

95. The method of claim 94 wherein said value is a unique code.

96. The method of claim 94 wherein said value is a pseudo-unique code.

97. A method for authenticating an operation which modifies a cursor image displayed by a client system, comprising:

requesting, by said client system, specified content information from a server system;

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

receiving, by said client system, said specified content information, said specified content information including at least one cursor display instruction, said cursor display instruction operable to modify an existing cursor image displayed by said client system;

requesting, by said client system responsive to recognizing said cursor display instruction, an encrypted authentication code;

decrypting, by said client system, said encrypted authentication code; and

determining whether to execute said cursor display instruction in response to said decrypted authentication code.

98. The method of claim 97 wherein said authentication code is based at least in part on an identity of said server system.

99. The method of claim 98 wherein said identity of said server system is an IP address of said server system.

100. The method of claim 97 wherein said authentication code is based at least in part on a time at which it is generated.

101. The method of claim 97 wherein said authentication code is based at least in part on a date on which it is created.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

102. The method of claim 97 wherein said authentication code is based at least in part on a value transmitted from said client system to said server system.

103. The method of claim 102 wherein said value is a unique code.

104. The method of claim 102 wherein said value is a pseudo-unique code.

105. A system for modifying a system level attribute in a client system responsive to displaying the contents of a web page, comprising:

a code module operable to process at least one attribute modifying instruction associated with said web page, said code module processing said at least one attribute modifying instruction modifying said system level attribute in said client system responsive to processing said web page to display the contents of the web page by said client system.

106. The system of claim 105, wherein said system level attribute is a shape of a status bar displayed by said client system.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

107. The system of claim 105, wherein said system level attribute is a color of a status bar displayed by said client system.

108. The system of claim 105, wherein said system level attribute is a shape of a scroll bar displayed by said client system.

109. The system of claim 105, wherein said system level attribute is a color of a scroll bar displayed by said client system.

110. The system of claim 105, wherein said system level attribute is a shape of a status bar displayed by said client system.

111. The system of claim 105, wherein said system level attribute is a shape of a title bar displayed by said client system.

112. The system of claim 105, wherein said system level attribute is a color of a title bar displayed by said client system.

113. The system of claim 105, wherein said system level attribute is a shape of an icon representing a standard window operation.

114. The system of claim 105, wherein said system level attribute is a color of an icon representing a standard window operation.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

115. The system of claim 105, wherein said system level attribute is an appearance of a background image against which other graphical images are displayed.

116. The system of claim 105, wherein said system level attribute is a file selection mechanism.

117. The system of claim 105, wherein said system level attribute is a file invocation mechanism.

118. The system of claim 105, wherein said system level attribute is a process selection mechanism.

119. The system of claim 105, wherein said system level attribute is an audible waveform used to convey information to a user.

120. The system of claim 105, said code module further responsive to a user specification of permitted modifications.

121. A system for replacing an existing cursor image with content information responsive to displaying a contents of a web page, comprising:

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

a cursor display code module operable to process at least one cursor display instruction associated with said web page, said processing of said at least one cursor display instruction replacing said existing cursor with said content information responsive to said web page being displayed, said at least one cursor display instruction including indication of said content information.

122. The system of claim 121, wherein said content information includes alphanumeric data.

123. The system of claim 121, wherein said content information includes at least one stock price.

124. The system of claim 121, wherein said content information includes at least one baseball game score.

125. The system of claim 121, wherein said content information includes a temperature.

126. The system of claim 121 wherein said content information is static.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

127. The system of claim 121 wherein said content information is updated periodically.

128. The system of claim 121 wherein said content information is updated as new content information becomes available.

129. The system of claim 121 wherein said content information includes financial information.

130. The system of claim 72 wherein said cursor display code is further operable to display a second web page indicated by said at least one cursor display instruction responsive to a specific user input received while said web page is being displayed, wherein said specific user input is indicated by at least one cursor display instruction.

131. The system of claim 105 wherein said specific user input is a right click on a mouse.

132. A computer program product including a computer readable medium, said computer readable medium having a computer program stored thereon, said computer program for replacing an existing

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

cursor image with at least one new cursor image responsive to displaying a contents of a web page, comprising:

cursor display program code operable to process at least one cursor display instruction associated with said web page, said processing of said at least one cursor display instruction providing said new cursor image responsive to said web page being displayed, said at least one cursor display instruction including indication of said cursor image.

In the Abstract:

Please find enclosed herewith a proposed substitute Abstract. Amendment of the Abstract in accordance with the substitute Abstract is respectfully requested.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Application No.: 08/882,580
Filed: June 25, 1997
Group Art Unit: 2773

REMARKS

This amendment is filed responsive to the Examiner's Action dated July 13, 1998. All rejections and objections of the Examiner are respectfully traversed. Reconsideration is respectfully requested.

Claims 72-132 have been added.

Claims 1, 29, 50, 72, 82, 89, 98, 105, 121 and 132 are independent.

Claims 1-132 are currently pending.

At paragraph 1 of the Office Action, the Examiner requested that the Applicant take into consideration certain editorial comments when revising the claims. Applicant respectfully notes the Examiner's request, and has made changes responsive to the Examiner's comments.

At paragraph 2 of the Office Action, the Examiner objected to the Abstract. Enclosed herewith is a substitute Abstract that is believed to meet all requirements of the Examiner.

At paragraph 3 of the Office Action, the Examiner rejected various claims under 35 U.S.C. §112, second paragraph. Amendments to the claims are believed to satisfy all requirements of the Examiner in this regard.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

At paragraph 4 of the Office Action, the Examiner rejected claim 29 due to insufficient antecedent basis. Amendments to the claim are believe to satisfy all requirements of the Examiner in this regard.

At paragraphs 5-7 of the Office Action, the Examiner rejected the independent claims 1, 29 and 50 under 35 U.S.C. §103(a), citing U.S. Patent No. 5,710,897 of Schneider (Schneider hereafter). Applicant respectfully traverses this rejection.

Nowhere in Schneider is there disclosed or suggested any system or method for modifying a cursor image by transmitting content information to a remote user's terminal in response to a request for that content information, where the content information includes at least one cursor display instruction to modify the cursor image, as set forth in the present independent claims 1 and 29. In distinct contrast, Schneider discloses a software system enabling *a user* to select a pointer graphics folder in order to customize *system pointers*. In clear contradistinction to the the presently claimed invention, the modification of the system pointer set by the user, as well as any subsequent modifications of the pointers in Schneider, are performed *completely independent from any specific content information*.

Specifically, the Schneider system enables an end user to edit the appearance of a number of default system pointer images through

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

a graphical user interface. See Abstract, and column 8, lines 58-59. The user selected pointer images of Schneider are specific only to the device on which they are defined, and their display is not related to any specific content being displayed on the device. See column 2, lines 23-24. Accordingly, the modified pointer sets in Schneider reside in a directory on the system boot drive. See column 8, lines 10-12. Pointer sets in Schneider may only be moved from device to device by explicit user directives. See column 7, lines 14-23. Accordingly, Schneider completely lacks even a hint of the desirability of associating a cursor image with specific content information, regardless of the underlying system on which the content is being displayed, as is provided by the present invention as set forth in claims 1 and 29.

Moreover, Schneider's system for mouse pointer modification subsequent to user selection of a pointer set teaches away from the presently claimed invention by associating each pointer image in the pointer set with a content independent system operation. See column 2, lines 18-21, and lines 30-42. For example, cursor images in Schneider are changed when the mouse pointer moves over any window ***regardless of what is being displayed within that window***. Similarly, a different mouse pointer is also shown by the Schneider system whenever the mouse pointer moves over a sizing border of a window, without any consideration of what specific content is being

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

displayed within the window being sized. See column 5, lines 55-65.

With regard to independent claim 50, nowhere in Schneider is there disclosed or suggested any system for processing cursor display instructions in connection with processing content information received from a remote server.

In view of the foregoing, Applicants respectfully submit that the independent claims 1, 29 and 50 are patentably distinct over the cited reference.

Further in paragraphs 5-7, as well as in paragraphs 8-9 of the Office Action, the Examiner rejected the claims depending from the above discussed independent claims, also citing U.S. Patent No. 5,737,619 of Judson (Judson hereafter), and "WWW Plug-Ins Companion" of M. Brown (Brown hereafter). Applicant respectfully traverses this rejection.

Judson discloses a system for reducing the waiting time normally associated with downloading hypertext. Judson teaches displaying information to the user between linking and downloading of a hypertext document identified by a link. In the Figures 4-5 and 8 showing the operation of Judson, as well as the accompanying text, no cursors of any kind are shown or described. Nothing in Judson suggests even the possibility of associating the modification of a cursor image with content information as in the

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

present independent claims 1, 29 and 50, and accordingly as in those claims depending either directly or indirectly therefrom.

The Brown reference teaches plug-in technology to complement a browser in order to expand and enhance the type of content that can be delivered over the World Wide Web (WWW). Brown also includes no teaching or suggestion of modifying any cursor image. Brown teaches that the "user interface remains relatively constant no matter what plug-ins are installed." In clear contradistinction to the system of claims 1, 29 and 50, Brown further states: "the parts of the page that handle page navigation, scrolling and so on aren't effected by the plug-ins presence". See page 15. Accordingly, Brown teaches a system that is specifically designed to operate while ***not modifying such parts of the page as the cursor***.

Brown further teaches three specific categories of plug-ins having specific information display characteristics. These are indicated as: 1) embedded, 2) full screen, and 3) hidden. These categories teach the display of information in a regular window integrated into a web page, the display of the information in a full screen, and storing of hidden information respectively. See Brown, beginning at page 9. Thus Brown several specific information display strategies, but fails to include any teaching or suggestion of even the need to modify a cursor image.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

With regard to independent claims 72 and 132, nowhere in the cited references, taken either independently or in combination, is there disclosed or suggested any system or method for replacing a cursor image using at least one cursor display instruction associated with a web page, where the cursor display instruction provides the new cursor image responsive to the web page being displayed.

With regard to newly added claims 82, 89 and 98, nowhere in the disclosed references, taken either independently or in combination, is there disclosed or suggested any system or method for authenticating the modification of a cursor image.

With regard to newly added independent claim 105, nowhere in the cited references, taken either independently or in combination, is there disclosed or suggested any system or method for modifying a system level attribute in a client system responsive to processing a web page to display the contents of the web page on the client system.

With regard to claim 121, nowhere in the cited references, taken either independently or in combination, is there disclosed or suggested any system or method for replacing a cursor with content information responsive to a web page being displayed.

For the reasons cited above Applicant respectfully urges that the above discussed independent claims are patentably distinct over

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

the cited references. As to the remaining claims, they depend either directly or indirectly from the above discussed independent claims, and are believed to patentably distinct over the cited references for at least the reasons stated above. Accordingly, reconsideration of all currently pending claims is respectfully requested.

As all pending claims are believed to be allowable, the application is believed to be in condition for allowance. Favorable action is respectfully requested.

The Examiner is encouraged to telephone the undersigned attorney to discuss any matter which would expedite allowance of the present application.

Respectfully submitted,

JAMES S. ROSEN ET AL.

By: David A. Dagg

David A. Dagg
Registration No. 37,809
Attorney for Applicant(s)

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
Ten Post Office Square
Boston, MA 02109
Telephone: (617) 542-2290
Telecopier: (617) 451-0313

Dated: December 14, 1998

DAD/gjn
Enclosure

131288

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

08/882,580

DEC 17 1998

ABSTRACT

A system for modifying a cursor image, as displayed on a video monitor of a remote terminal, to a specific image having a desired shape and appearance. The system stores cursor image data corresponding to the specific image, and a cursor display code. The cursor display code contains information in response to which the cursor image is modified to the specific image. A server computer transmits specified information to the remote terminal. The information includes at least one cursor display instruction. The cursor display instruction is operable to modify, in conjunction with the cursor information and the cursor image data, a cursor image displayed by a display of the remote terminal in the shape and appearance of the specific image.



12/24/98

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application	:	~~Joseph Sofer~~ James Rosen et al.
Application No.	:	08/882,580
Filed	:	June 25, 1997
For	:	SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE
Examiner	:	
Attorney's Docket	:	COMET-001XX

11/23/98 #9

Group Art Unit: 2312

* *

RECEIVED
DEC 22 1998
Group 2700

REVOCATION OF POWER AND
APPOINTMENT OF NEW POWER OF ATTORNEY

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

As an officer of the Assignee having the entire right, title, and interest in the above-identified application for United States Patent, evidenced by the Assignment as recorded on Reel 8631, Frame 0209, I hereby revoke all previous powers and respectfully request the appointment of:

Stanley M. Schurgin, Registration No. 20,979
Charles L. Gagnebin III, Registration No. 25,467
Paul J. Hayes, Registration No. 28,307
Victor B. Lebovici, Registration No. 30,864
Eugene A. Feher, Registration No. 33,171; and
Beverly E. Hjorth, Registration No. 32,033
Holliday C. Heine, Registration No. 34,346
Gordon R. Moriarty, Registration No. 38,973

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
Ten Post Office Square
Boston, Massachusetts 02109

and

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313



Application No.: 08/882,580
Filed: June 25, 1997
Group Art Unit: 2312

Adam C. Solomon, Registration No. 43,142
Comet Systems, Inc.
180 Maiden Lane, 20th Floor
New York, New York 10007

RECEIVED
DEC 22 1998
Group 2700

as attorneys with full powers.

Please direct all further correspondence to the following address:

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
Ten Post Office Square
Boston, Massachusetts 02109

Respectfully submitted,

By: _____
Dean Margolis
Chief Executive Officer
Comet Systems, Inc.

Date: 11/23/98

131259

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

BEST COPY

Transaction History Date 1998-12-29
Date information retrieved from USPTO Patent Application Information Retrieval (PAIR) system records at www.uspto.gov

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

SERIAL NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NO.
[illegible]	05/25/97	ROSEN	058-002

EXAMINER	
ART UNIT	PAPER NUMBER
[illegible]	10

JOSEPH SCHER
SCHER & HAROUN LLP
345 MADISON AVE
[illegible]
NEW YORK NY 10173

DATE MAILED: 12/29/98

This is in response to the Power of Attorney filed 12/17/98.

☐ 1. The Power of Attorney to you in this application **has been revoked** by the applicant. Future correspondence will be mailed to the new address of record. 37 CFR 1.33.

☐ 2. The Power of Attorney to you in this application **has been revoked** by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record. (37 CFR 1.33).

☐ 3. The withdrawal as attorney in this application **has been accepted.** Future correspondence will be mailed to the new address of record. 37 CFR 1.33.

This is a communication from the Patent and Trademark Office

☐ 4. The Power of Attorney in this application **is accepted.** Correspondence in this application will be mailed to the below-noted address as provided by 37 CFR 1.33.

☐ 5. The Power of Attorney in this application **is not accepted** for the reason(s) checked below:

☐ a. The Power of Attorney is from an assignee and the Certificate required by 37 CFR 3.73 (b) has not been received.

☐ b. The person signing for the assignee has omitted their empowerment to sign on behalf of the assignee.

☐ c. The inventor(s) is without authority to appoint attorneys since the assignee has intervened as provided by 37 CFR 3.71.

☐ d. The signature of _____, a co-inventor in this application, has been omitted. The Power of Attorney will be entered upon receipt of confirmation signed by said co-inventor.

☐ e. The person(s) appointed in the Power of Attorney is not registered to practice before the U. S. Patent & Trademark Office.

☐ f. The revocation is not signed by the applicant, the assignee of the entire interest, or **one** particular principal attorney having the authority to revoke.

[illegible] SCHURGIN GAGNEBIN & HAYES LLP
[illegible] TEN POST OFFICE SQUARE
[illegible] MA 02109

VERLENE D. GREEN
SUPERVISORY LEGAL INSTRUMENTS EXAMINER
GROUP 2700

This is a communication from the Patent and Trademark Office

PTOL-305 (REV. 7/89) RETAIN THIS COPY IN THE APPLICATION FILE COPY A

Transaction History Date 1999-02-08
Date information retrieved from USPTO Patent Application Information Retrieval (PAIR) system records at www.uspto.gov

BEST COPY

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.
09/360,500	06/25/99	ROSEN	850-P001

LM61/0208
WEINGARTEN SCHURGIN GAGNEBIN & HAYES LLP
TEN POST OFFICE SQUARE
BOSTON MA 02109

EXAMINER	
JACKSON, C	
ART UNIT	**PAPER NUMBER**
2772	11

DATE MAILED: 02/03/99

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner of Patents and Trademarks

PTO-90C (Rev. 2/95)
*U.S GPO 1998-437-638/80022

1- File Copy

Office Action Summary	Application No. 08/882,580	Applicant(s) Rosen et al.
	Examiner C. Jackson	Group Art Unit 2773

—The MAILING DATE of this communication appears on the cover sheet beneath the correspondence address—

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, such period shall, by default, expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).

Status

☒ Responsive to communication(s) filed on 12/17/98.

☒ This action is **FINAL.**

☐ Since this application is in condition for allowance except for formal matters, **prosecution as to the merits is closed** in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 1 1; 453 O.G. 213.

Disposition of Claims

☒ Claim(s) 1-132 is/are pending in the application.

Of the above claim(s) 72-132 is/are withdrawn from consideration.

☐ Claim(s) _____ is/are allowed.

☒ Claim(s) 1-71 is/are rejected.

☐ Claim(s) _____ is/are objected to.

☐ Claim(s) _____ are subject to restriction or election requirement.

Application Papers

☐ See the attached Notice of Draftsperson's Patent Drawing Review, PTO-948.

☐ The proposed drawing correction, filed on _____ is ☐ approved ☐ disapproved.

☐ The drawing(s) filed on _____ is/are objected to by the Examiner.

☐ The specification is objected to by the Examiner.

☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. § 119 (a)-(d)

☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 11 9(a)-(d).

☐ All ☐ Some* ☐ None of the CERTIFIED copies of the priority documents have been

☐ received.

☐ received in Application No. (Series Code/Serial Number) _____.

☐ received in this national stage application from the International Bureau (PCT Rule 1 7.2(a)).

*Certified copies not received: _____.

Attachment(s)

☐ Information Disclosure Statement(s), PTO-1449, Paper No(s). _____

☐ Interview Summary, PTO-413

☐ Notice of Reference(s) Cited, PTO-892

☐ Notice of Informal Patent Application, PTO-152

☐ Notice of Draftsperson's Patent Drawing Review, PTO-948

☐ Other _____

Office Action Summary

DETAILED ACTION

1. This action is responsive to communications: Amendment, filed on 12/17/98. This action is final.

2. The present title of the invention is "SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE" having claims 1-132 as Amended. In the Amendment, filed on 12/17/98, claims 1-4, 7-62 and 65-71 were amended, and claims 72-132 were added.

Election/Restriction

3. Newly submitted claims 72-132 directed to inventions that are independent or distinct from the invention originally claimed for the following reasons:

-- claims 82-104 are directed toward authentication; class 340, subclass 825.34 or class 395, subclass 200.59.

-- claims 72-81 and 105-132 are directed toward hypermedia document processing; class 707, subclass 501.

Since applicant has received an action on the merits for the originally presented invention, this invention has been constructively elected by original presentation for prosecution on the merits. Accordingly, claim 72-132 are withdrawn from consideration as being directed to a non-elected invention. See 37 CFR 1.142(b) and MPEP § 821.03.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103© and potential 35 U.S.C. 102(f) or (g) prior art under 35 U.S.C. 103(a).

6. Claims 1-3, 9-14, 18-24, 27-31, 37-38, 43-52, 54-61 and 67-69 are rejected under 35 U.S.C. 103(a) as being unpatentable over Schneider (US Patent # 5,710,897).

Schneider ('897) teaches the applicant's invention as claimed in claims 1-3, 9-14, 18-25, 27-31, 37-38, 43-52, 54-61 and 67-69 as follows:

as per claim 1, cursor image data corresponding to said specific image [a pointer graphic corresponding to a pointer graphic to be used by the operating system, see Schneider ('897), col. 5, lines 65- col. 6, lines 1-10; col. 7, lines 28-31; col. 8, lines 10-12, and 55-60] ;

cursor display code, said cursor display code operable to modify said cursor image [pointer graphic file set, the pointer graphic file set is operable to modify a displayed pointer graphic based on the occurrence of an event, see Schneider ('897), col. 5, line 55-col. 6, line 11];

a first server computer for transmitting specified content information to a remote terminal [a personal computer designed to give users independent computing power, where information, such as applications, documents, files, ect., are transmitted to a remote user. See Schneider ('897), col. 3, lines 58-68].

Schneider ('897) does not explicitly discuss transmitting specified content information that includes instructions indicating the location of a cursor image file. However, as discussed above, the system of Schneider ('897) is operable on a networked environment. In a networked environment specified content information is transmitted to a remote users in response to a request from the user for the information. Specified content information that can be transmitted includes application, application windows and files. An application that is executed on a network includes instructions to be used by the operating system regarding the handling of events. For example, the remote launching of a word processing application would typically generate an application window having a client area, menus and toolbars. Instructions would be provided to the operating system defining what operations can be performed on the window and what events should occur in response to other events. Accordingly, an application that provides for resizing of a window would include an instruction specifying that the frame of a displayed window (i.e., specific content) can be resized and requires the display of a sizing pointer when the pointer is

over the boarders of a window. This instruction indicates the location of cursor image data, where the operating system would locate the appropriate pointer graphics file and cause the user terminal to display the appropriate cursor on the user's terminal when required. As a result, it would have been obvious to one skilled in the art at the time the invention was made to provide the location of a pointer graphic file when executing the apparatus of Schneider ('897) in a distributed environment because it enables the access of information for response to events.

as per claims 2 and 30 [As discussed above with respects to claim 1, the system of Schneider discloses that it is operable in a networked environment. Accordingly, a computer designed to give independent computing power responds to a user's request for information by providing the information on the user's terminal, see discussion independent claim 1, supra.].

as per claims 3, 9, 31, 37 and 67 [the pointer graphic, in the case of resizing, moving or editing a window, has a shape and appearance that provides an indication to the user that the pointer graphic pertains to areas of a "displayed window" (i.e., specific content).]

as per claim 10-13 and 56, [pointer graphics and pointer graphic files can be stored in local memory such as a hard or floppy drive or CD-ROM of a computer as well as communicated over a network, see Schneider ('897), col. 9, lines 3-21].

as per claims 14, 24, 38 and 68-69, [as described in claim 1, information can be distributed remotely by a server, such as an HTML page].

as per claims 16 and 40, [see discussion claim 50].

as per claims 17 and 41, [see discussion claim 1].

as per claims 18 and 42, [the system of Schneider will not change the pointer image if it is not found and provides the user with the ability to locate a pointer graphic using find object function for loading into the pointer graphics file set, see Schneider ('897), col. 7, lines 24-30].

as per claims 19-22 and 43-46, [the techniques and advantages of providing animation and audio are old and well know in the art. Accordingly, it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide audio and animation code in a pointer graphics file.].

as per claims 23, 47 and 55, [the instructions for handling windowing events includes instructions which controls how long a pointer will be displayed, wherein the duration of time a pointer is positioned within an area of a window determine how long a pointer will be displayed, see Schneider ('897), col. 5, lines 65-col. 6, lines 1-65].

as per claims 27, 48, and 54 [the techniques and advantages of calculating frequency data is old and well known in the art. Accordingly, it would have been obvious to a developer creating the system of Schneider ('897) to maintain a log of the pointer of most interest to user in-order to provide and make those pointers available as well as to identify the pointer that are of least interest so that they can be eliminated. This would provide an efficient use of systems resources where memory space would not be consumed by pointer graphics files that are rarely used].

as per claim 28, [the find object function of Schneider ('897 enables communication with a plurality of remote sites where the pointer graphics file can be located, see Schneider ('897),col. 7, lines 24-30].

as per claims 29 and 49, receiving a request at said at least one server to provide specified content information to said user terminal [receiving a request at a computer system designed to give independent computing power to a group of user to execute the pointer graphic manager program file or other applications on an end-user terminal, where execution generates specific content information to be used and displayed on a user's terminal, see Schneider ('897), col. 3, lines 58-63; col. 9, lines 7-18; col. 7, lines 24-30];

providing said specified content information to said user terminal in response to said request, said specified content information including at least one cursor display instruction and at least one indication of cursor image data corresponding to a specified image [providing application information to a user's terminal in response to a user's request to launch an application, the application information including includes instructions to be used by the operating system regarding the handling of events, such as the display of content and what actions can be performed on the displayed content. The instruction providing an implication of the pointer graphic to be used corresponding to an event];

transforming said initial cursor image displayed on said video monitor of said user terminal into the shape and appearance of said specific image in response to said cursor display instruction data [changing a pointer displayed into the shape and appearance of the pointer graphic modified by the functions of the pointer graphic manager program file, see Schneider ('897), col. 8, lines 52-col. 9, lines 1-67; abstract].

as per claims 47 and 55, [the instructions for handling windowing events includes instructions which controls how long a pointer will be displayed, wherein the duration of time a pointer is positioned within an area of a window determine how long a pointer will be displayed, see Schneider ('897), col. 5, lines 65-col. 6, lines 1-65].

as per claims 50, an Internet browser configured to modify a cursor image [software (e.g., the operating system, programs and various device drivers), wherein the pointer graphic manager establishes pointer choices available to the user through modification of the operating systems pointers and is applicable to any graphical user interface that typically uses pointer graphics, see Schneider, col. 5, lines 47-55; col. 8, lines 55-60.]

means for receiving specified content information from a remote server, said specified content information comprising a information to be displayed on said remote user's terminal and at least one cursor display instruction, wherein said cursor display instruction indicates cursor image data corresponding to said specific image [means for receiving application information where information is provided to be displayed on a user's terminal, such as windows, objects, and data, and instructions to be used by the operating system regarding the handling of events, such as the display of content and what actions can be performed on the displayed content. The instruction providing an implication of the pointer graphic to be used corresponding to an event];

means for recognizing said cursor display instruction in connection with processing said information to be displayed on said display [The operating system of a user's terminal recognizing

the instructions and responding to events regarding the display of a cursor in connection with executing an application];

means for executing a cursor display code, responsive to said cursor display instruction, said cursor display code being operable to modify said cursor image to said specific image [means for executing pointer graphic file, where the graphic file is executed when an event occurs, the pointer graphics file operable to modify a cursor image to the cursor image required by the event, see Schneider ('897), col. 5, line 55-col. 6, line 9].

as per claims 51-52, [pointer graphics files can be delivered to a computer in many forms including communication over a network. Accordingly, a pointer graphic file which is not stored on a user's terminal can be located on a predefined server for access and loading, see Schneider ('897),col. 7, lines 24-30].

as per claims 57-60, [see discussion, claims 19-22 and 43-46].

as per claim 61, [pointer graphic associated with a transmitted pointer graphic file corresponds to a portion of the data displayed on the monitor of the users].

Claims 4-8, 15, 32-36, 39, 42-46 and 62-66 are rejected under 35 U.S.C. 103(a) as being unpatentable over Schneider (US Patent # 5,710,897) in view of Judson (US Patent # 5,572,643).

The combination of Schneider ('897) and Judson ('643) teaches the applicant's invention as claimed in **claims 4-8, 15, 32-36 and 62-66** as follows:

as per claims 4, 15, 32, and 62, Schneider ('897) teaches that a pointer graphic is provided for various states, events or commands (e.g., functions) of an application window. See

Schneider ('897), abstract; col. 5, lines 65-col. 6, lines 1-10. For example, Schneider ('897) provides a wait pointer that is invoked when a displayed window of an application is busy. See Schneider ('897), col. 6, lines 13-17. Moreover, as discussed above with respects to claims 1, 29 and 50, a window is specific content information. Schneider ('897) also discloses that the pointers may be edited or modified as desired by the user and stored for later use. See Schneider ('897), id. However, Schneider is silent on providing a pointer graphic for when the system is in the mode of displaying advertising material. Judson ('643), provides for the display of advertising material during the downtime (i.e., when the system is busy) of downloading a hypertext document on a web page. See Judson ('643), abstract. One having ordinary skill in the art would recognize that the downtime and downloading of advertising material that occurs in the apparatus of Judson ('643) is an event and thus the pointer could be provided that changed to represent this event. One having ordinary skill in the art would also recognize that each pointer graphic depicted in the system of Schneider ('897) relates to the current event or mode of operation of the system. As a result, it would have been obvious to one having ordinary skill in the art at the time the invention was made to provide a customized pointer set using the system of Schneider ('897) for the event of displaying advertising material in the apparatus of Judson ('643), wherein the pointer graphic relates to the advertising material, because it would provide the user with an option of selecting a pointer that best suites there individual taste and sense of context interest (i.e., whether they are more interested in knowing that the system is busy or that the system is displaying an advertisement).

as per claims 5-8, 33-36 and 63-66, [the advertising material can include advertisements messages, logos, etc., see Judson ('643), abstract.].

7. Claims 25-26, 53 and 70-71 are rejected under 35 U.S.C. 103(a) as being unpatentable over Schneider (US Patent # 5,710,897) in view of Gosling et al., "The Java Language Environment- A White Paper" (hereinafter "Java").

Like applicant's claims 25-26 and 70-71, Schneider ('897) discloses an application that allows a user to modify a cursor at the operating system level. However, it does not disclose that the platform utilizes a virtual machine or that the specific content information includes HTML tags. Java teaches that the use of a virtual machine, where application instructions a provided in HTML tags, to extend the natural capabilities of a platform. One having ordinary skill in the art would recognize that the natural capabilities of a platform is a desirable feature. As a result, it would have been obvious to one skill in the art at the time the invention was made to provide the apparatus of Schneider ('897) on a platform using a virtual machine because it would allow a user to utilize applications which are not supported by the platform.

As per claim 53, [see java, security.].

Response to Arguments

8. Applicant's arguments filed on 12/17/98 have been fully considered but they are not deemed persuasive.

Applicant argues, with respects to claim 1 and 29, that the system of Schneider ('897) does not teach modifying a pointer depending on specific content information. The examiner disagrees with the applicant. The applied reference teaches that the system of Schneider modifies pointers depending on events that are possible. A variety of pointers are provided to user that are associated with windows of an application. An application window is specific content information that is displayed to a user because it is information that is specific to an application and displayed. Moreover, the pointer graphic displayed to the user relates to the window, since it is an image which provides the user with an indication of what operations are permissible on a particular area of the window's boarder or what state the window is in. Accordingly, the system of Schneider teaches associating a cursor image with specific content information, and thus claims 1, 29, and 50 stand rejected as obvious.

Claims depending from claims 1, 29 and 50 also stand rejected for the aforementioned reasons.

Conclusion

9. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

10. Response to this action should be mailed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231. If applicant desires to fax a response, (703) 308-9051 may be used for formal communications or (703) 305-9724 for informal or draft communications. Please label "PROPOSED" OR "DRAFT" for informal facsimile communications. For after final responses, please label "AFTER FINAL" or "EXPEDITED PROCEDURE' on the document. Hand delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington VA., Sixth Floor (Receptionist).

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chadwick A. Jackson, whose telephone number is (703) 308-9572. The examiner can normally be reached Mon-Thu from 7:30 a.m. - 6:00 p.m. ET. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Kim, can be reached at (703) 305-3821

12. Any inquiry of a general nature or relating to the status of this application or proceedings should be directed to the group receptionist whose telephone number is (703) 305-3900.

Chadwick A. Jackson

February 1, 1999

RAYMOND J. BAYERL
PRIMARY EXAMINER
ART UNIT 2773

Notice of References Cited

Application No.	Applicant(s)		
08/882,580	Rosen et al.		
Examiner	Group Art Unit	Page 1 of 1	
C. Jackson	2773		

U.S. PATENT DOCUMENTS

*		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS
	A					
	B					
	C					
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

*		DOCUMENT NO.	DATE	COUNTRY	NAME	CLASS	SUBCLASS
	N						
	O						
	P						
	Q						
	R						
	S						
	T						

NON-PATENT DOCUMENTS

*		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	U	The Java Language Environment – A White Paper Gosling et al., May 1995 pages 6-64	May 1995
	V		
	W		
	X		

* A copy of this reference is not being furnished with this Office action.
(See Manual of Patent Examining Procedure, Section 707.05(a).)

U.S. Patent and Trademark Office
PTO-892 (Rev. 9-96)

Part of Paper No. 11

*U.S. GPO: 1997-417-381/62700

Other Prior Art

According to the information contained in form PTO-1449 or PTO-892, there are one or more other prior art/non-patent literature documents missing from the original file history record obtained from the United States Patent and Trademark Office. Upon your request we will attempt to obtain these documents from alternative resources. Please note that additional charges will apply for this service.

This page is not part of the official USPTO record. It has been determined that content identified on this document is missing from the original file history record.

The Java Language Environment
A White Paper

James Gosling
Henry McGilton



Sun Microsystems Computer Company
A Sun Microsystems, Inc. Business
2550 Garcia Avenue
Mountain View, CA 94043 U.S.A.
415 960-1300 FAX 415 969-9131

May 1995

© 1995 Sun Microsystems, Inc. All rights reserved.
2550 Garcia Avenue, Mountain View, California 94043-1100 U.S.A.

This ALPHA quality release and related documentation are protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX® and Berkeley 4.3 BSD systems, licensed from UNIX System Laboratories, Inc., a wholly owned subsidiary of Novell, Inc., and the University of California, respectively. Third-party font software in this product is protected by copyright and licensed from Sun's font suppliers.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7013 (c)(1)(ii) and FAR 52.227-19.

The release described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

TRADEMARKS
Sun, the Sun logo, Sun Microsystems, Solaris, HotJava, and Java are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and certain other countries. The "Duke" character is a trademark of Sun Microsystems, Inc., and Copyright (c) 1992-1995 Sun Microsystems, Inc. All Rights Reserved. UNIX is a registered trademark in the United States and other countries, exclusively licensed through X/Open Company, Ltd. OPEN LOOK is a registered trademark of Novell, Inc. All other product names mentioned herein are the trademarks of their respective owners.

All SPARC trademarks, including the SCD Compliant Logo, are trademarks or registered trademarks of SPARC International, Inc. SPARCstation, SPARCserver, SPARCengine, SPARCstorage, SPARCware, SPARCcenter, SPARCclassic, SPARCcluster, SPARCdesign, SPARC811, SPARCprinter, UltraSPARC, microSPARC, SPARCworks, and SPARCompiler are licensed exclusively to Sun Microsystems, Inc. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK® and Sun™ Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

X Window System is a trademark of the X Consortium.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS PUBLICATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THE PUBLICATION. SUN MICROSYSTEMS, INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS PUBLICATION AT ANY TIME.



Please
Recycle

Contents

1. Introduction 6

2. Simple, Object Oriented, and Familiar 14

   2.1 Main Features of the Java Language 16

   2.2 Object Oriented 21

   2.3 Features Removed from C and C++ 28

   2.4 Summary 31

3. Architecture Neutral, Portable, and Robust 32

   3.1 Architecture Neutral 32

   3.2 Portable 33

   3.3 Robust 34

   3.4 Summary 36

4. Interpreted, Dynamic, Secure, and Threaded 38

   4.1 Dynamic Loading and Binding 39

4.2 Security in the Java Environment 41
4.3 Multithreading 44
4.4 Java Class Libraries 47
4.5 Performance 47
4.6 The Java Language Compared 49
4.7 Summary 50
5. The HotJava World-Wide Web Browser 52
5.1 The Evolution of Cyberspace 53
5.2 First Generation Browsers 53
5.3 The HotJava Browser—A New Concept in Web Browsers 54
5.4 The Essential Difference 55
5.5 Dynamic Content 55
5.6 Dynamic Types 57
5.7 Dynamic Protocols 58
5.8 Freedom to Innovate 59
5.9 Implementation Details 60
5.10 Security 61
5.11 Summary 62
6. Further Reading 64

Introduction 1

The Next Stage of the Known,
Or a Completely New Paradigm?

Taiichi Sakaiya—*The Knowledge-Value Revolution*

The Software Developer's Burden

Imagine you're a C or C++ software application developer. You've been at this for quite a while and your job doesn't seem to be getting any easier. These past few years you've seen the growth of multiple incompatible hardware architectures, each architecture supporting multiple incompatible operating systems, and each platform operating with one or more incompatible graphical user interfaces. And now you're supposed to cope with all this and make the applications work in a distributed client-server environment. The growth of the Internet, the World-Wide Web, and "electronic commerce" have introduced new dimensions of complexity into the development process.

1

The tools you work with to develop applications don't appear to be helping you a whole lot. You're still coping with the same old problems, and the use of object-oriented techniques seem to have added new problems without solving the old ones. You keep saying to yourself and your friends, "There *has* to be a better way!"

There Has To Be A Better Way

Now there *is* a better way—it's the Java programming language environment from Sun Microsystems. Imagine, if you will, this development world...

- Your programming language is *object-oriented* yet it's still dead *simple.*
- Your development cycle is *much* faster because the Java language is *interpreted.* The old compile-link-load-test-and-crash cycle has gone the way of the zumbooruk*—now you just compile and run. When you're satisfied with your application, you can obtain maximum performance by using the built in *just-in-time compiler* to compile the Java intermediate code to native machine code.
- Your applications are *portable* across multiple platforms. Write your applications once, and you never need to port them—they will run without modification on multiple operating systems and hardware architectures.
- Your applications are *robust* because the Java run-time system manages memory for you—you don't have dangling pointers and memory leaks and trashing of memory because of bad pointers, because there are no pointers.
- Your interactive graphical applications have *high performance* because multiple concurrent threads of activity in your application are supported by the *multithreading* built into the language.
- Your applications are *dynamically adaptable* to changing environments because you can dynamically load code modules from anywhere in the network.
- Your end users can trust that your applications are *secure,* even though they're downloading code from all over the Internet, because the Java run-time system has built-in protection against viruses and tampering.

---

* A small cannon designed to be mounted on and fired from the back of a camel—implies obsolescence.

1

The Better Way Is Here

You don't need to dream about these features to make developer life easier. They're here now in the form of the Java Programming Environment, otherwise known as "the Java language"—a *portable, interpreted, high performance, simple, object-oriented* programming language and supporting run-time environment developed at Sun Microsystems. The rest of this introductory chapter takes a brief look at the main design goals of the Java language system. The remainder of this paper examines the features of the Java language in more detail. At the end of this paper you'll find a chapter on the *HotJava browser*—an innovative World-Wide web browser, a major application written using the Java language environment. The HotJava browser is unique in its ability to download and execute Java code fragments on the fly from anywhere on the Internet, and do so in a secure manner.

Beginnings of the Java Language Project

The Java language was designed to meet the challenges of application development in the context of heterogeneous network-wide distributed environments. Paramount among these challenges is the secure delivery of applications that consume the minimum of system resources, can run on any hardware and software platform, and can be dynamically extended.

The Java language originated as part of a research project to develop advanced software for a wide variety of networked devices and embedded systems. The goal was to develop a small, reliable, portable, distributed, real-time operating environment. When the project was started, C++ was the language of choice. But over time the difficulties encountered with C++ grew to the point where the problems could best be addressed by creating an entirely new language environment. Design and architecture decisions drew from a variety of languages such as Eiffel, SmallTalk, Objective C, and Cedar/Mesa. The result was a language environment that has proven ideal for development of secure, distributed, network-based end-user applications in environments ranging from networked embedded devices to the World-Wide Web and the desktop.

Design Goals of the Java Language

The design requirements of Java can be summed up as a set of characteristics—to *live*, to *survive*, and to *flourish*:

1

- The massive growth of the Internet and the World-Wide Web* leads us to a completely new way of looking at development and distribution of software. To *live* in the world of electronic commerce and distribution, the Java language must support *secure, high-performance,* and highly *robust* application development on *multiple platforms* in *heterogeneous, distributed networks.*
- Operating on multiple platforms in heterogeneous networks invalidates the traditional schemes of binary distribution, release, upgrade, patch, and so on. To *survive* in this jungle, the Java language must be *architecture-neutral, portable,* and *dynamically adaptable.*
- To ensure you can *flourish* within your software development environment, the Java language system that emerged to meet these needs is *simple,* so it can be easily programmed by most developers, *familiar,* so that current developers can easily learn the Java language, *object oriented,* to fit into distributed client-server applications, *multithreaded,* for high performance in applications that need to perform multiple concurrent activities, and *interpreted,* for maximum portability and dynamic capabilities.

All in all, the above requirements comprise quite a collection of buzzwords, so let's briefly examine some of the features and their respective benefits before getting into details in the rest of this paper.

Simple, Object Oriented, and Familiar

A primary goal of the Java language was a *simple* language that could be programmed without extensive programmer training and which would be roughly attuned to current software practice. The fundamental concepts of the Java language can be grasped quickly—programmers can be productive from the start.

The Java language was designed as an *object-oriented* language from the ground up. Object-oriented technology has finally found its way into the programming mainstream after a gestation period of thirty years. The needs of distributed, client-server based systems coincide with the packaged, message-passing paradigms of object-based software. To function within increasingly complex,

* 15 percent per month by some estimates.

network-based environments, programming systems must adopt object-oriented concepts. The Java language provides a clean and efficient object-based development environment.

Programmers starting with the Java language have access to existing libraries of tested objects that provide functionality ranging from basic data types through I/O interfaces and network interfaces to graphical user interface toolkits. Many of these libraries can be extended (subclassed) to provide new behavior.

Even though C++ was rejected as an implementation language, keeping the language *familiar* resulted in the Java language looking as close to C++ as possible, while removing the unnecessary complexities of C++. Making the Java language a minimal subset of C++ while simultaneously retaining its "look and feel" means that programmers can migrate to the Java language easily and be productive quickly—the Java language learning curve can be as low as a couple of days.

Robust and Secure

The Java language is designed for creating highly *reliable* software. Emphasis is on extensive compile-time checking, and a second level of run-time checking. Java language features guide programmers into reliable programming habits. The Java memory management model—basically, no pointers or pointer arithmetic—eliminates entire classes of programming errors that bedevil C and C++ programmers. You can develop Java code with confidence that the system will find errors quickly and that major problems won't slip out into production code.

The Java language was designed to operate in distributed environments. With *security* features designed into the language and run-time system, the Java language enables construction of tamper-free programs. In the networked environment, Java programs are secure from intrusion by unauthorized code attempting to get behind the scenes and create viruses or invade file systems.

Architecture Neutral and Portable

The Java language was designed to support applications operating in networked environments, operating on a variety of hardware architectures, and running a variety of operating systems and language environments. The Java language compiler generates byte codes—an architecture-neutral intermediate format used to transport code efficiently to multiple hardware

and software platforms. Because of the interpreted nature of the Java language, you don't have a binary distribution and versioning problem—the same Java language byte codes will run on any platform.

Architecture neutrality is just one part of a truly *portable* system. The Java language takes portability a stage further by being strict in its definition of the basic language. The Java language puts a stake in the ground and specifies sizes of basic data types and the behavior of the arithmetic operators. You programs are always the same on every platform—there are no data type incompatibilities across hardware and software architectures.

The Java virtual machine is based on a well-defined porting layer, primarily based on the POSIX interface standard—an industry standard definition of a portable system interface. Porting to new architectures is a relatively straightforward task.

High Performance

Performance is always a consideration, and the Java language achieves superior performance by adopting a scheme by which the interpreter can run at full speed without needing to check the run-time environment. The *automatic garbage collector* runs as a low-priority background thread, ensuring a high probability that memory is available when required, leading to better performance. Applications requiring large amounts of compute power can be designed such that compute-intensive sections can be rewritten in native machine code as required and interfaced with the Java language environment. In general, users perceive that interactive applications perform adequately even though they're interpreted. When you need the full speed of the host hardware for compute-intensive applications, the Java language system provides an on-the-fly "just-in-time" compiler that will compile the Java language byte codes to machine code at run time. This innovative feature provides native code performance levels without compromising the portability of applications.

Interpreted, Threaded, and Dynamic

The Java *interpreter* can execute Java byte codes directly on any machine to which the interpreter and run-time system have been ported. In an interpreted environment such as the Java language system, the "link" phase of a program is simple, incremental, and lightweight. You gain a major benefit of much faster development cycles—prototyping, experimenting, and rapid development are the normal case.

Modern network-based applications such as the HotJava World-Wide Web browser typically need to do several things at the same time. A user operating the HotJava browser can run several animations concurrently while downloading an image and scrolling the page. The ability to perform multiple tasks concurrently is provided by the Java language's *multithreading* capability, which provides the means to build applications with many concurrent threads of activity. Multithreading thus results in a high degree of interactivity for the end user.

The Java language supports multithreading at the language level with the addition of sophisticated synchronization primitives, at the language library level with the `Thread` class, and at the run-time level with monitor and condition lock primitives. At the library level, the Java language's high-level system libraries have been written to be *thread-safe*—the functionality provided by the libraries is available to multiple concurrent threads of execution.

While the Java compiler is strict in its compile-time static checking, the language and run-time system are *dynamic* in their linking stages. Classes are linked as needed. New code modules can be linked in on demand from a variety of sources, even across a network. In the case of the HotJava web browser and similar applications, interactive executable code can be loaded from anywhere, enabling transparent updating of applications. The result is on-line services that constantly evolve, remaining innovative and fresh, drawing more customers, and spurring the growth of electronic commerce on the Internet.

The Java Environment—A New Approach to Distributed Computing

Taken individually, the characteristics discussed above can be found in a variety of software development environments. What's completely new is the manner in which the Java language and run-time system have combined them to produce an flexible and powerful programming system.

The Java language environment results in software applications that are *portable*, highly *secure*, and *high performance*. With the Java language to develop your software, your job as a software developer is much easier—you focus your full attention on the end goal of shipping innovative products on time, based on the solid foundation of the Java environment.

1

Simple, Object Oriented, and Familiar

2

You know you've achieved perfection in design,
Not when you have nothing more to add,
But when you have nothing more to take away.

Antoine de Saint Exupery.

In *The Rolling Stones*, Robert A. Heinlein comments:

> *Every technology goes through three stages: first a crudely simple and quite unsatisfactory gadget; second, an enormously complicated group of gadgets designed to overcome the shortcomings of the original and achieving thereby somewhat satisfactory performance through extremely complex compromise; third, a final proper design therefrom.*

Heinlein's comment could well describe the evolution of the C language over the course of its long lifetime. This chapter discusses the three primary design features of the Java language, namely, *simple, object-oriented,* and *familiar.* The end of this chapter contains a discussion on those features that were eliminated from C and C++ in the evolution towards the Java language.

2

Simple

Based on the design methodology known as KISS*, *simplicity* is one of the Java language's overriding design goals from a programming standpoint. Simplicity and removal (from C++-based languages) of many dubious "features" keep the Java language relatively small and reduce the programmer's burden in producing reliable applications. To this end, the Java language design team examined many aspects of the "modern" C and C++ languages† to determine features that could be eliminated in the context of modern object-oriented programming.

Object Oriented

To stay abreast of modern software development practices, the Java language was designed to be *object oriented* from the ground up. The benefits of object technology are slowly becoming realized as more organizations move their applications to the distributed client-server model. The Java language goes beyond C++ in both extending the object models and removing the major complexities of C++. With the exception of its primitive data types, everything in the Java language is an object.

Familiar

Another major design goal was that the Java language look *familiar* to the majority of programmers in the personal computer and workstation arenas. A large fraction of today's system and application programmers are familiar with C and C++. Thus, the Java language "looks like" C++. Programmers familiar with C, Objective-C, C++, Eiffel, Ada, and related languages should find their Java language learning curve is quite short—on the order of a couple of days or so.

To illustrate the simple, object-oriented, and familiar aspects of the Java language, we follow the tradition of a long line of illustrious (and some not so illustrious) programming books by showing you the `HelloWorld` program. It's about the simplest program you can write that actually does something. Here's `HelloWorld` in the Java language.

* Keep It Small and Simple

† Now enjoying their silver anniversaries

```
class HelloWorld {
    static public void main(String args[]) {
        System.out.println("Hello world!");
    }
}
```

This example declares a *class* named `HelloWorld`. Classes are discussed in the section on object-oriented programming, but in general we assume the reader is familiar with object technology and understands the basic notions of classes, objects, instance variables, and methods. Within the `HelloWorld` class, we declare a single *method* called `main` which in turn contains a single *method invocation* to display the string "Hello world!" on the standard output. The statement that prints "Hello world!" does so by invoking the `println` method of the `out` object, which is a class variable of the `System` class. That's all there is to `HelloWorld`.

2.1 *Main Features of the Java Language*

Primitive Data Types

Other than the primitive data types discussed here, everything in the Java language is an object. Even the primitive data types can be wrapped inside language-supplied objects as required. The Java language follows C and C++ fairly closely in its set of basic data types, with a couple of minor exceptions. There are only three groups of primitive data types—numeric types, Boolean types, and arrays.

Integer numeric types are 8-bit `byte`, 16-bit `short`, 32-bit `int`, and 64-bit `long`. There are no `unsigned` types.

Real numeric types are 32-bit `float` and 64-bit `double`. Real numeric types and their arithmetic operations are as defined by the IEEE 754 specification.

Character data is a minor departure from traditional C. The Java language uses the Unicode character set standard, so the traditional C `char` data type was expanded from eight to sixteen bits. If you write a declaration such as

```
char  myChar = 'Q';
```

you get a Unicode (16-bit unsigned character) type that's initialized to the Unicode value of the character Q.

The *Boolean* is a primitive type in the Java language, tacitly ratifying existing C and C++ programming practice. A `boolean` variable assumes the values `true` or `false`. A Java `boolean` is a distinct data type—unlike common C practices, a `boolean` can not be converted to a numeric type by casting.

Arithmetic and Relational Operators

All the familiar C and C++ operators apply in the Java language. Because the Java language lacks `unsigned` data types, the Java language adds the `>>>` operator to mean an unsigned (logical) right shift. The Java language also uses the `+` operator for string concatenation, discussed below.

Arrays

Arrays in the Java language are first-class language objects. A Java language array is a real object with a run-time representation. You can declare and allocate arrays of any type. To obtain multi-dimensional arrays, you allocate arrays of arrays. You declare an array of, say, `Object` with a declaration like this:

```
Object  myArray[];
```

This code declares that `myArray` is an uninitialized array of `Object`. This variable has no storage as yet. At some future point you must allocate the amount of storage you need, say:

```
myArray = new Object[10];
```

to allocate an array of 10 `Objects`. To get the length of an array, add `.length` to the array name—`myArray.length` would return the number of elements in `myArray`. Access to elements of `myArray` can be performed via the normal C indexing, but all array accesses are checked to ensure that their indexes are within the range of the array. An exception will be generated if the index is outside the bounds of the array.

The C notion of a pointer to an array of memory elements has gone, and with it, the arbitrary pointer arithmetic that leads to unreliable code in C. No longer can you "walk off the end" of an array, possibly trashing memory and leading to the famous "delayed crash" syndrome, where a memory access violation today manifests itself hours or days later. Programmers can be confident that Java array checking will help lead to more robust and reliable code.

Strings

Strings are objects, not arrays of characters as in C. String objects are instances of the String class. There are actually two kinds of String objects. The String class is for read-only (immutable) objects. The StringBuffer class is for mutable String objects.

Although Strings are objects, the Java language compiler "knows" about Strings as a special syntactic feature to accommodate the syntactic convenience that C programmers have enjoyed with C-style strings*. The Java language compiler understands that a string of characters enclosed in double quote signs is to be instantiated as a String object. Thus, the declaration:

```
String hello = "Hello world!";
```

instantiates an object of the String class behind the scenes, and initializes it with the character string `"Hello world!"`.

The Java language has extended the meaning of the + operator to indicate *string concatenation*. Thus you can write statements like:

```
System.out.println("There are " + num + " characters in the file");
```

This code fragment concatenates the string `"There are "` with the result of converting the numeric value num to a string, and concatenates that with the string `" characters in the file"`. Then it prints the result of those concatenations on the standard output.

String objects supply a `length` accessor method to obtain the number of characters in the string.

Multi Level Break

The Java language has no `goto` statement. To `break` or `continue` multiply nested loop or switch constructs, you use labels to label loop and switch constructs, and then `break` or `continue` to the label. Here's a small fragment of code from the Java language's built-in `String` class.

```
test:  for (int i = fromIndex; i + max1 <= max2; i++) {
    if (charAt(i) == c0) {
```

* Which are really just an array of characters.

```
        for (int k = 1; k<maxl; k++)
            if (charAt(i+k) != str.charAt(k)) {
                continue test;
```

The `continue` statement that leads to `test` is nested two levels deep inside `for` statements. This Java language feature leads to considerable simplification of programming time, and major reduction in future maintenance efforts.

Memory Management and Garbage Collection

C and C++ programmers are by now accustomed to the problems of explicitly managing memory—allocating memory, freeing memory, and keeping track of what memory can be freed when. Explicit memory management has proved to be a fruitful source of bugs, crashes, memory leaks, and poor performance.

The Java language completely removes the memory management load from the programmer. C-style pointers, pointer arithmetic, `malloc`, and `free` do not exist. *Automatic garbage collection* is an integral part of the Java language and run-time system. Once you have allocated an object, the run-time system keeps track of the object's status and automatically reclaims memory when objects are no longer in use, freeing memory for future use.

The Java language's memory management model is based on *objects* and *references* to objects. Because the Java language has no pointers, all references to allocated storage—which in practice means references to objects—are through symbolic "handles". The Java memory manager keeps track of references to objects. When an object has no more references, the object is a candidate for garbage collection. While the Java language has a `new` operator to allocate space for objects, there is no explicit `free` function.

The Java language's memory allocation model and automatic garbage collection make your programming task easier, eliminate entire classes of bugs, and in general provide better performance than you'd obtain through explicit memory management. Here's a code fragment that illustrates when garbage collection happens. It's an example from the on-line Java language programmer's guide:

```
class ReverseString {
    public static String reverseIt(String source) {
        int i, len = source.length();
        StringBuffer dest = new StringBuffer(len);

        for (i = (len - 1); i >= 0; i--) {
```

```
            dest.appendChar(source.charAt(i));
        }
        return dest.toString();
    }
}
```

The variable `dest` is used as a temporary object reference during the execution of the `reverseIt` method. When `dest` goes out of scope (the `reverseIt` method returns), the reference to that object has gone away and it's then a candidate for garbage collection.

The Background Garbage Collector

A common criticism of garbage collection schemes is that they tend to fire up at inappropriate times, creating poor interactive response for the user.

The Java language run-time system largely solves this problem by running the garbage collector in a low priority *thread*. This, by the way, is just one of many examples of the synergy one obtains from the Java language's integrated *multithreading* capabilities—an otherwise intractable problem is solved in a simple and elegant fashion.

The typical user of the typical interactive application has many natural pauses where the user is contemplating the scene in front of them or thinking of what to do next. The Java language run-time system can take advantage of these idle periods and run the garbage collector when no other threads are competing for CPU cycles The garbage collector gathers and compacts unused memory, thereby improving the probability that adequate memory resources are available when needed during periods of heavy interactive activity.

Integrated Thread Synchronization

The Java language supports *multithreading*, both at the language (syntactic) level and via support from its run-time system and `Thread` objects. While other systems have provided facilities for multithreading (usually via "lightweight process" libraries), building multithreading support into the language provides the programmer with a much more powerful tool for easily creating thread-safe multithreaded classes. Multithreading is discussed in more detail in a separate chapter.

2

2.2 Object Oriented

My Object All Sublime
I Will Achieve in Time

Gilbert and Sullivan—*The Mikado*

The Java language is *object oriented*. It draws on the best concepts and features of previous object-oriented languages, primarily Eiffel, SmallTalk, C++, and Objective-C.

Unfortunately, "object oriented" remains misunderstood, over-marketed, or takes on the trappings of a religion. The cynic's view of object-oriented programming is that it's just a new way* to organize your source code. While there may be some merit to this view, it doesn't tell the whole story, because you can achieve results with object-oriented programming that you can't with procedural techniques.

Object Technology in the Java Language

To be truly considered "object oriented", a programming language should support at a minimum three characteristics:

- *Encapsulation*—to implement information hiding and modularity
- *Inheritance*—code re-use and code organization
- *Dynamic binding*—for maximum flexibility while a program is executing.

The Java language meets these requirements nicely, and adds considerable run-time support to make your software development job easier.

At its simplest, object technology is a collection of analysis, design, and programming methodologies that focus design on *modelling* the characteristics and behavior of objects in the real world. True, this definition appears to be somewhat circular, so let's try to break out into clear air.

* Or not even a new way, depending on your time perspective.

Objects

What are objects? They're software programming *models* of objects of everyday life. In your everyday life, you're surrounded by objects—cars, coffee machines, ducks, trees, and so on. These objects have *state* and *behavior*. You can model everyday objects with software constructs called objects. Software objects can also be defined by their *state* and their *behavior*.

In your everyday transportation needs, a *car* can be modelled by an object. A car has *state* (how fast it's going, in which direction, its fuel consumption, and so on) and *behavior* (starts, stops, turns, slides, and runs into trees).

In your daily interactions with the financial markets, a *call option* can also be modelled by an object. A call option has *state* (current price, strike price, expiration date), and *behavior* (changes value, moves into and out of the money, or expires worthless[*]).

After your call options have expired worthless, you repair to the cafe for a cup of good hot coffee. The *espresso machine* can be modelled as an object. It has *state* (water temperature, amount of coffee in the hopper) and it has *behavior* (emits steam, makes noise, and brews a perfect cup of *Java*).

An important characteristic that distinguishes objects from procedures or functions is that an object can have a lifetime greater than that of the object that created it. This aspect of objects is subtle and mostly overlooked. In the distributed client-server world, this means that objects can be created in one place, passed around networks, and stored elsewhere to be retrieved at later stages for future work.

The Basics of Objects



In the implementation of an object, its *state* is defined by its *instance variables*. Instance variables are variables local to the object. Unless explicitly made public or made available to other "friendly" classes, they are inaccessible from outside the object. An object's *behavior* is defined by its *methods*. Methods manipulate the instance variables to create new state, and can also create new objects. This picture is a commonly used graphical representation of an object. The diagram illustrates the conceptual structure of a software object—it's kind of like a cell, with an outer membrane that's its interface to the world, and an inner nucleus that's protected by the outer membrane.

[*] More often than not.

An object's *instance variables* (data) are packaged, or encapsulated, with the object, hidden safely inside the nucleus, safe from access by other objects. The instance variables are surrounded by the object's *methods*. With certain well-defined exceptions, the object's methods are the only means by which other objects can access or alter its instance variables. In Java, classes can declare their instance variables to be public, in which cases the instance variables are globally accessible—this topic is covered in later in *Access Specifiers*.

Classes

A *class* defines the *instance variables* and *methods* of an object. A class in and of itself is not an object. A class is a *template* that defines how an object will look and behave when the object is created or *instantiated*. You can instantiate many objects from one class definition, just as you can construct many houses all the same from a single architect's drawing. The only way you can get concrete objects is by instantiating a previously defined class. Here's the declaration for a very simple class called Point

```
class Point {
    public float  x;
    public float  y;

    Point() {
        x = 0.0;
        y = 0.0;
    }
}
```

Methods with the same name as the class as in the code fragment above are called *constructors*. When you create (instantiate) an object of this class, the constructor method is invoked to perform any initialization that's needed—in this case, to set the instance variables to an initial state.

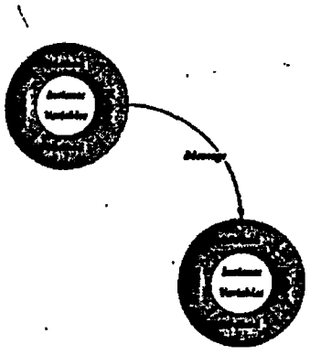
Any other object can now create an instance of Point with a line of code like this:

```
Point  myPoint;

myPoint = new Point();
```

Now, you can access the variables of this Point object by referring to the names of the variables, qualified with the name of the object:

```
myPoint.x = 10.0;
myPoint.y = 25.7;.
```



Methods and Messaging

If an object wants another object to do some work on its behalf, then in the parlance of object-oriented programming, the first object must send a *message* to the other object. An object sends a message to another object by invoking one of its *methods*, which look similar to functions in C and C++.

In general, methods operate only on the instance variables of a specific *instance* of the object. However, an object can declare its instance variables to be `public`, in which case other objects can access the instance variables directly without going through method invocations.

Using the message passing paradigms of object-oriented programming, you can build entire networks and webs of objects that pass messages between them to change state. This programming technique is one of the best ways to create complex simulations of real-world systems.

Constructors and Finalizers

When you declare a class in the Java language, you can declare optional *contructors* that perform initialization when you instantiate objects from that class, and you can declare an optional *finalizer* that will perform necessary tear down actions when the garbage collector is about to free the object. This minimal code fragment illustrates a *constructor*.

```
public final class Integer extends Number {
    private int value

    public Integer(int value){
        this.value = value;
    }
}
```

This example declares a class called `Integer`. There is in fact an `Integer` class in the Java language foundation—it's an object wrapper when you need to encapsulate an integer value inside an object. The public method `Integer` is the *constructor* for this class. You would access it in code via a declaration of the form

```
Integer  myIntegerObject = new Integer(123);
```

to create a new object of the Integer class initialized to the value 123.

By the way, what is the variable `this` in the above example? It is a special name that refers to this instance of the class—from inside an object, you need a way to refer to the object itself. That's what the `this` variable achieves.

This code fragment illustrates a finalize method in a class.

```
/**
  * Close the stream when garbage is collected.
*/
protected void finalize() {
    try {
        close();
    } catch (Exception e) {
    }
}
```

This `finalize` method will be invoked when the object is about to be garbage collected. In the particular code fragment, the `finalize` method merely closes an I/O file stream that was used by the object to ensure that the file descriptor for the stream is closed.

Subclassing

Subclassing is the mechanism by which new and enhanced objects can be defined in terms of existing objects. One example: a zebra is a horse with stripes. If you wish to create a zebra object, you notice that a zebra is kind of like a horse, only with stripes. In object oriented terms, you'd create a new class called Zebra, which is a *subclass* of the Horse class. In Java language terms, you'd do something like this:

```
class Zebra extends Horse {
    Your new instance variables and new methods go here
}
```

The definition of `Horse`, wherever it is, would define all the methods to describe the *behavior* of a horse—eat, neigh, trot, gallop, buck, and so on. The only method you need to override is the method for drawing the hide. You gain the benefit of already written code that does all the work—you don't have to re-invent the wheel, or in this case, the hoof. The `extends` keyword tells the Java compiler that Zebra is a subclass of Horse. Zebra is said to be a *derived class*—it's derived from Horse, which is called a *base class*.

Subclassing enables you to use existing code that's already been developed and, much more important, tested, for a more generic case. You override the parts of the class you need for your specific behavior. Thus, sub-classing gains

you re-use of existing code—you save on design, development, and testing. The Java supporting run-time system provides several libraries of utility functions that are tested and are also *thread-safe*.

Access Control

When you declare a new class in the Java language, you can indicate the level of access permitted to its instance variables and methods. The Java language provides four levels of access specifiers. Three of the levels must be explicitly specified if you wish to use them. They are `public`, `protected`, and `private`.

The fourth level doesn't have a name—it's often called "friendly" and is the access level you obtain if you don't specify otherwise. The "friendly" access level indicates that your instance variables and methods are accessible to all objects within the same package, but inaccessible to objects outside the package.

`public` methods and instance variables are available to any other class anywhere. `protected` means that instance variables and methods so designated are accessible only to subclasses of that class, and nowhere else. `private` methods and instance variables are accessible only from within the class in which they're declared—they're not available even to their subclasses.

Static and Final Methods and Variables

The Java language follows conventions from other object-oriented languages in providing for *class methods* and *class variables*. Normally, variables you declare in a class definition are *instance variables*—there will be one of those variables in every separate object that's created (instantiated) from the class. A class variable, on the other hand, is a variable that's local to the class itself. There's only copy of the variable and it's shared by every object you instantiate from the class. Here's a short code fragment.

```
class Rectangle extends Object {
    static final int version = 2;
    static final int revision = 0;
}
```

The Rectangle class declares two `static` variables to define the version and revision level of this class. Now, every instance of Rectangle that you create from this class will share these same variables. Notice they're also defined as `final` because you want them to be constants.

Class methods are methods you declare that operate only on instance variables of the class. Class methods can't access instance variables, nor can they invoke instance methods. Like class variables, you declare class methods by defining them as `static`.

Abstract Methods

Abstract methods are a powerful construct in the object-oriented paradigm. To understand abstract methods, we look at the notion of an *abstract superclass*. This is a class in which you define methods that aren't actually implemented by that class—they only provide place-holders such that subsequent subclasses can override those methods and supply their actual implementation.

This all sounds wonderfully, well, *abstract*, so why would you need an abstract superclass? Let's look at a *concrete* example, no pun intended.

Suppose you are creating a drawing application. The initial cut of your application can draw rectangles, lines, circles, polygons, and so on. Furthermore, you have a series of operations you can perform on the shapes—move, reshape, rotate, fill color, and so on. You *could* make each of these graphic shapes a separate class—you'd have a Rectangle class, a Line class, and so on. Each class needs instance variables to define its position, size, color, rotation and so on, which in turn dictates methods to set and get at those variables.

At this point, you realize you can collect all the instance variables into a single abstract superclass, and implement most of the methods to manipulate the variables in that abstract superclass. The skeleton of your abstract superclass might look something like this.

```
class Graphic {
    Point p;                        //  position
    float fillcolor;                //  gray shade of interior
    float linecolor;                //  gray shade of outline
    . . .;                          //  more instance variables

                                    //  more methods
    abstract void drawmyself()      //  do nothing method
}
```

Now, you can't instantiate the Graphic class. You can only instantiate a subclass of it. When you implement the Rectangle class, you'd subclass Graphic. Within Rectangle, you'd provide a concrete implementation of the

`drawmyself` method that draws a rectangle. You can continue in this way adding new shapes that are subclasses of Graphic, and all you ever need to implement is the method to draw the shape. You gain the benefit of re-using all the code that was defined inside the abstract superclass.

Packages

Packages are a Java language construct that gather collections of related classes into a single container. For example, all the Java language I/O system code is collected into a single package. The primary benefit of packages is organizing many class definitions into a single compilation unit. The secondary benefit from the programmer's viewpoint is that the "friendly" instance variables and methods are available to all classes within the same package, but not to classes defined outside the package.

As an example, you could create a rectangle geometry package containing the definitions of a point class and a rectangle class, plus all the operations you'd want to perform on points and rectangles. Within the package, you could have rectangles accessing the "friendly" instance variables of the point class directly, thereby improving the overall performance of the objects. But other classes outside the package would be able to access the internal state of point objects and rectangle objects only by invoking the `public` methods of the classes.

2.3 *Features Removed from C and C++*

The previous sections concentrated on features of the Java language. This section discusses features removed from C and C++ in the design of the Java language. The first step was to *eliminate redundancy* from C and C++. In many ways, the C language evolved into a collection of overlapping features, providing too many ways to say the same thing, while in many cases not providing needed features. C++, even in an attempt to add "classes in C"* merely added more redundancy while retaining the inherent problems of C.

* Or even just add some class to C.

Typedefs, Defines, and the Preprocessor

Source code written in the Java language is *simple*. There is no *preprocessor*, no `#define` and related capabilities, no `typedef`, and absent those features, no longer any need for *header files*. Instead of header files, Java language *interfaces* provide the definitions of other classes and their methods.

A major problem with C and C++ is the amount of context you need to understand another programmer's code—you have to read all related header files, all related `#defines`, and all related `typedefs` before you can even begin to analyze a program. In a sense, programming in this style essentially results in every programmer inventing a new programming language that's incomprehensible to anybody other than its creator and defeats the goals of good programming practices.

You can obtain the effects of `#define` by using constants. You obtain the effects of `typedef` by declaring classes—after all, a class effectively declares a new type. You don't need header files because the Java language compiler compiles class definitions into a binary form that retains all the type information through to link time.

By removing all this baggage, the Java language becomes remarkably *context free*. Programmers can read and understand code and, more importantly, modify and re-use code much faster and easier.

Structures and Unions

The Java language has no structures or unions as complex data types. You don't need structures and unions when you have classes—you can achieve the same effect simply by using instance variables of a class. This code fragment declares Point and Rectangle classes. In C you'd define these as structures. In the Java language, you simply declare a class. You can make the instance variables as private or as public as you wish, depending on how much you wish to hide the details of the implementation.

```
class Point extends Object {
    float  x;
    float  y;
      methods to access the instance variables
}
```

```
class Rectangle extends Object {
    Point  lowerLeft;
    Point  upperRight;
      methods to access the instance variables
}
```

Functions

The Java language has no *functions*. Object-oriented programming supersedes functional and procedural styles. Mixing the two styles just leads to confusion and dilutes the purity of an object-oriented language. Anything you can do with a function, you can do just as well by defining a class and creating methods for that class. By eliminating functions, the Java language has immensely simplified your job as a programmer—you work *only* with classes and methods.

Multiple Inheritance and Interfaces

The Java language discards *multiple inheritance* and all the problems it generates. The desirable features of multiple inheritance are provided by *interfaces*—conceptually similar to Objective C protocols.

An interface is not a definition of an object. Rather, it's a definition of a set of methods that one or more objects will implement. An important issue of interfaces is that they declare methods only—in general, no instance variables other than `final` variables (which are constants) may be defined in interfaces.

Goto Statements Gone

The Java language has no `goto` statement*. Studies illustrated that `goto` is (mis)used more often than not simply "because it's there". Eliminating `goto` led to a simplification of the language—there are no rules about the effects of `goto`ing into the middle of a `for` statement, for example. As mentioned above, multi-level `break` and `continue` remove the need for `goto` statements.

Operator Overloading

The Java language has no C++ style *operator overloading*.

---

* `goto` is still a reserved word—it just doesn't do anything.

Automatic Coercions

The Java language prohibits C and C++ style *automatic coercions*. If you wish to coerce a data element of one type to a data type that would result in loss of precision, you must do so explicitly by using a cast. Consider this code fragment:

```
int   myInt;
float  myFloat = 3.14159;
myInt = myFloat;
```

The assignment of `myFloat` to `myInt` would result in a compiler warning of possible loss of precision and that you should use a cast. Thus, you should rewrite the code fragments as:

```
int   myInt;
float  myFloat = 3.14159;
myInt = (int)myFloat;
```

Pointers

Most studies agree that *pointers* are one of the primary features that enable programmers to put bugs into their code. Given that structures are gone, and arrays and strings are objects, the need for pointers to these constructs goes away. Thus the Java language has no pointers. Any task that would require arrays, structures, and pointers in C can be much more easily and reliably performed in the Java language by declaring objects and arrays of objects.

2.4 Summary

To sum up this chapter, you've by now gained the understanding that the Java language is:

- *Simple*—the number of language constructs you need to understand to get your job done is minimal.
- *Object-oriented*—you can develop software using up-to-date software development practices.
- *Familiar*—the Java language "looks like" C and C++ while discarding the overwhelming complexities of those languages.

Architecture Neutral, Portable, and Robust 3

With the phenomenal growth of networks, today's developers must "think distributed". Applications—and even parts of applications—must be able to migrate easily to a wide variety of computer systems, a wide variety of hardware architectures, and a wide variety of operating system architectures. They must operate with a plethora of graphical user interfaces. Clearly, if applications must be able to execute anywhere on the network without *a priori* knowledge of the hardware and software platform on which it will run, the binary distribution problem quickly becomes unmanageable. To solve this problem, software must become *architecture-neutral* and *portable*. Finally, reliability is at a high premium in the distributed world—code from anywhere on the network should work robustly without low probabilities of creating "crashes" in applications importing code fragments.

3.1 *Architecture Neutral*

The solution that the Java language system adopts to solve the binary distribution problem is a "binary code format" that's independent of hardware architectures and operating system and window system interfaces. This file format is *architecture-neutral*. If the Java run-time system is made available on a given hardware and software platform, Java language applications can then execute on many different processors and operating system architectures without the need to port them.

Byte Codes

The Java language compiler doesn't generate "machine code" in the sense of native hardware instructions. Rather, the Java language compiler generates *byte codes*—a high-level, machine-independent "machine code" for a hypothetical machine that is implemented by the Java interpreter and run-time system. One of the early examples of this approach was the UCSD P System, which was immensely popular in the middle 1970s and early 1980s.

The architecture-neutral approach is useful not only for network-based applications, but also for single system software distribution. In today's personal computer market, application writers have to produce versions of their applications that are compatible with the IBM PC, Apple Macintosh, and 57 flavors of workstation architectures in the fragmented UNIX marketplace. With the PC market (through Windows/NT) diversifying into many CPU architectures, and Apple moving full steam from the 68000 to the PowerPC, production of software to run on all platforms becomes almost impossible. Using the Java language, the same version of your application can run on all platforms.

The Java language byte codes are designed to be both easy to interpret on any machine and easy to dynamically translate into native machine code if required.

3.2 Portable

The primary benefit of using the interpreted byte code approach is that compiled Java language programs are *portable* to any system on which the Java interpreter and run-time system have been implemented.

The architecture-neutral aspect discussed above is one major step towards being portable, but there's more to it than that. C and C++ both suffer from the defect of designating many fundamental data types as "implementation dependent". Programmers labor to ensure programs are portable across architectures by programming to a lowest common denominator.

The Java language eliminates this issue by defining standard behavior that will apply to the data types across all platforms. The sizes of the Java language's primitive data types are specified, as is the behavior of arithmetic on them. Here are the data types:

`byte`	8-bit two's complement
`short`	16-bit two's complement
`int`	32-bit two's complement
`long`	64-bit two's complement
`float`	32-bit IEEE 754 floating point
`double`	64-bit IEEE 754 floating point
`char`	16-bit Unicode character

Note that the Java language has no `unsigned` data types.

The data types and sizes described above are standard across all implementations of the Java language environment. These choices are reasonable given current microprocessor architectures because essentially all of the central processor architectures in use today share these characteristics That is, most modern processors can support two's complement arithmetic in 8-bit to 64-bit integer formats, and most support single- and double-precision floating point.

The libraries that are part of the Java language run-time system define portable interfaces. For example, there is an abstract Window class and implementations of it for UNIX, Windows, and Macintosh.

The Java language environment itself is readily portable to new architectures and operating systems. The Java compiler is itself written in the Java language. The Java run-time system is written in ANSI C with a clean portability boundary which is essentially POSIX-compliant. There are no "implementation dependent" notes in the Java language specification.

3.3 *Robust*

The Java language is intended for developing software that must be *robust*, highly *reliable*, and *secure*, in a variety of ways. The Java language places a lot of emphasis on early checking for possible problems, later dynamic (run-time) checking, and eliminating error prone situations.

Strict Compile Time Checking

The Java compiler employs extensive and stringent compile-time checking so that syntax-related errors can be detected early, before a program is deployed into service.

One of the advantages of a strongly typed language (like C++) is that it allows extensive compile-time checking so bugs can be found early. Unfortunately, C++ inherits a number of loopholes in this checking from C, which is relatively lax (the major issue is method/procedure declarations). The Java language *requires* declarations and does not support C-style implicit declarations.

Many of the stringent compile-time checks are carried over to the run-time, to check consistency at run-time, and to provide greater flexibility. The linker understands the type system and repeats many of the type checks done by the compiler, to guard against version mismatch problems.

The single biggest difference between the Java language and C/C++ is that the Java language's memory model eliminates the possibility of overwriting memory and corrupting data. Instead of pointer arithmetic, the Java language has true arrays, which means that the interpreter can check array and string indexes. In addition, a programmer can't write code that turns an arbitrary integer into a pointer by casting.

Garbage collection makes the programmer's job vastly easier. With the burden of memory management taken off the programmer's shoulders, storage allocation errors go away.

While the Java language doesn't pretend to completely remove the software quality assurance problem, removal of entire classes of programming errors considerably eases the job of testing and quality assurance.

A Major Benefit—Fast and Fearless Prototyping

Very dynamic languages like Lisp, TCL, and SmallTalk are often used for *prototyping*. One of the reasons for their success at this is that they are very robust—you don't have to worry about freeing or corrupting memory.

Programmers can be relatively fearless about dealing with memory when programming in the Java language because they don't have to worry about it getting messed up.

Another reason commonly given that languages like Lisp, TCL, and SmallTalk are good for prototyping is that they don't require you to pin down decisions early on—these languages are semantically rich.

The Java language has exactly the opposite property—it forces you to make explicit choices. Along with these choices come a lot of assistance—you can write method invocations and, if you get something wrong, you get told about it at compile time. You don't have to worry about method invocation error.

3.4 Summary

The Java language—an architecture-neutral and portable programming language—provides an attractive and simple solution to the problem of distributing your applications across heterogeneous network-based computing platforms. In addition, the simplicity and robustness of the underlying Java language results in higher quality reliable applications in which users can have a high level of confidence.

3

Interpreted, Dynamic, Secure, and Threaded 4

Programmers using "traditional" software development tools have become inured to the artificial edit-compile-link-load-throw-the-application-off-the-cliff-let-it-crash-and-start-all-over-again style of current development practice.

Additionally, keeping track of what must be recompiled when a declaration changes somewhere else is straining the capabilities of development tools—even fancy "make"-style tools such as are found on UNIX systems. This development approach bogs down as applications grow into the hundreds of thousands of lines of code sizes.

Better methods of fast and fearless prototyping and development are needed. The Java language environment is one of those better ways, because it's *interpreted* and *dynamic*.

As discussed in the previous chapter on architecture-neutrality, the Java compiler generates *byte codes* for the Java Virtual Machine*. The notion of virtual interpreted machines is not new. But the Java language brings the concepts into the realm of secure, distributed, network-based systems.

The Java language virtual machine is a strictly defined virtual machine for which an *interpreter* must be available for each hardware architecture and operating system on which you wish to run Java language applications. Once

* One of the ancestors of the virtual machine concept was the UCSD P System, developed by Kenneth Bowles at the University of San Diego in the late 1970s.

you have the Java language interpreter and run-time support available on a given hardware and operating system platform, you can run any Java language application from anywhere.

The notion of a separate "link" phase after compilation is pretty well absent from the Java environment. Linking, which is actually the process of loading new classes by the *Class Loader*, is a more incremental and lightweight process. The concomitant speedup in your development cycle means that your development process can be much more rapid and exploratory, and because of the robust nature of the Java language and run-time system, you will catch bugs at a much earlier phase of the cycle.

4.1 Dynamic Loading and Binding

The Java language's portable and interpreted nature produces a highly *dynamic* and *dynamically extensible* system. The Java language was designed to adapt to evolving environments. Classes are linked in as required and can be downloaded from across networks. Incoming code is verified before being passed to the interpreter for execution.

Object-oriented programming has become accepted as a means to solve at least a part of the "software crisis," by assisting encapsulation of data and corresponding procedures, and encouraging the re-use of code. Most programmers doing object-oriented development today adopted C++ as their language of choice. But C++ suffers from a somewhat serious problem that impedes its widespread use in the production and distribution of "software ICs." This defect is known as the *fragile superclass problem*.

The Fragile Superclass Problem

This problem arises as a side-effect of the way that C++ is usually implemented. Any time you add a new method or a new instance variable to a class, any and all classes that reference that class will require a re-compilation, or they'll break. Keeping track of the dependencies between class definitions and their clients has proved to be a fruitful source of programming error in C++, even with the help of "make"-like utilities. The fragile superclass issue is sometimes also referred to as the "constant re-compilation problem." You *can* avoid these problems in C++, but with extraordinary difficulty, and doing so effectively means not using any of the language's OO features directly. By avoiding the object-oriented features of C++, developers defeat the goal of re-usable "software ICs."

Solving the Fragile Superclass Problem

The Java language solves the fragile superclass problem in several stages. The Java compiler doesn't compile references down to numeric values—instead, it passes symbolic reference information through to the byte code verifier and the interpreter. The Java interpreter performs final name resolution once, when classes are being linked. Once the name is resolved, the reference is rewritten as a numeric offset, enabling the Java interpreter to run at full speed.

Finally, the storage layout of objects is not determined by the compiler. The layout of objects in memory is deferred to run time and determined by the interpreter. Updated classes with new instance variables or methods can be linked in without affecting existing code.

At the small expense of a name lookup the first time any name is encountered, the Java language eliminates the fragile superclass problem. Java programmers can use object-oriented programming techniques in a much more straightforward fashion without the constant recompilation burden engendered by C++. Libraries can freely add new methods and instance variables without any effect on their clients. Your life as a programmer is simpler.

Interfaces

The Java language implements the concept of an *interface*—a concept borrowed from Objective C* and similar to a class. An interface is simply a specification of a collection of methods that an object responds to. An interface does not include any instance variables or implementations. You can import and use multiple interfaces in a flexible manner, thus providing the benefits of multiple inheritance without the inherent difficulties created by the usual rigid class inheritance structure.

Run Time Representations

Classes in the Java language have a run-time representation. There is a class named `Class`, instances of which contain run-time class definitions. If you're handed an object, you can find out what class it belongs to. In a C or C++

* It's similar in concept to Objective C's *protocols*.

program, you may be handed a pointer to an object, but if you don't know what type of object it is, you have no way to find out. In the Java language, finding out based on the run-time type information is straightforward.

It is also possible to look up the definition of a class given a string containing its name. This means that you can compute a data type name and have it easily dynamically-linked into the running system.

4.2 *Security in the Java Environment*

Security commands a high premium in the growing use of the Internet for products and services ranging from electronic distribution of software and multimedia content, to "digital cash". The area of security with which we're concerned here is how the Java compiler and run-time system restrict application programmers from creating subversive code.

The Java language compiler and run-time system implement several layers of defense against potentially incorrect code. One of the Java compiler's primary lines of defense is its memory allocation and reference model. Simply put, Java does not have "pointers" in the traditional C and C++ sense—memory cells that contain the addresses of other memory cells.

Memory layout decisions are not made by the compiler, as they are in C and C++. Rather, memory layout is deferred to run-time, and will potentially differ depending on the characteristics of the hardware and software platforms on which the Java language system is executing. The Java interpreter references memory via symbolic "handles" that are resolved to real memory addresses at run time. Java programmers can't forge pointers to memory, because the memory allocation and referencing model is completely opaque to the programmer and controlled entirely by the underlying run-time system.

Very late binding of structures to memory means that programmers can't infer the physical memory layout of a class by looking at its declaration. By removing the C/C++ memory layout and pointer models, the Java language has eliminated the programmer's ability to get behind the scenes and manufacture pointers to memory. These features must be viewed as positive benefits rather than a restriction on the programmer, because they ultimately lead to more reliable and secure applications.

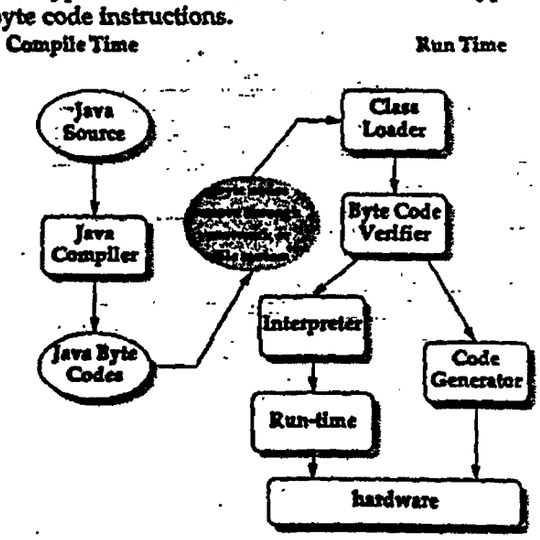
The Byte Code Verification Process

What about the concept of a "hostile compiler"? Although the Java compiler ensures that Java source code doesn't violate the safety rules, when an application such as the HotJava web browser imports a code fragment from anywhere, it doesn't actually know if code fragments follow the Java language rules for safety—the code may not have been produced by a known-to-be trustworthy Java compiler. In such a case, how is the Java run-time system on your machine to trust the incoming byte code stream? The answer is simple—it doesn't trust the incoming code, but subjects it to *byte code verification*.

The tests range from simple verification that the format of a code fragment is correct, to passing through a simple theorem prover to establish that the code fragment plays by the rules—that it doesn't forge pointers, it doesn't violate access restrictions, and it accesses objects as what they are (for example, that "InputStream" objects are always used as "InputStreams" and never as anything else). A language that is safe, plus run-time verification of generated code, establishes a base set of guarantees that interfaces cannot be violated.

The Byte Code Verifier

The last phase of the byte code loader is the *verifier*. It traverses the byte codes, constructs the type state information, and verifies the types of the parameters to all the byte code instructions.



The illustration shows the flow of data and control from Java language source code through the Java compiler, to the byte code verifier and hence on to the Java interpreter. The important issue is that the Java class loader and the byte code verifier make no assumptions about the primary source of the byte code stream—the code may have come from the local system, or it may have travelled halfway around the planet. The byte code verifier acts as a sort of gatekeeper. The byte code verifier ensures that the code passed to the Java interpreter is in a fit state to be executed and can run without fear of breaking the Java interpreter. Imported code is not allowed to execute by any means until after it has passed the verifier's tests. Once the verifier is done, a number of important properties are known:

- There are no operand stack overflows or underflows
- The types of the parameters of all byte code instructions are known to always be correct
- No illegal data conversions are done, like converting integers to pointers
- Object field accesses are known to be legal—private or public or protected

While all this checking appears excruciatingly detailed, by the time the byte code verifier has done its work, the Java interpreter can proceed knowing that the code will run securely. Knowing these properties makes the Java interpreter much faster, because it doesn't have to check anything. There are no operand type checks and no stack overflow checks. The interpreter can thus function at full speed without compromising reliability.

Security Checks in the Class Loader

After incoming code has been vetted and determined clean by the byte code verifier, the next line of defense is the Java *class loader*. The environment seen by a thread of execution running Java byte codes can be visualized as a set of classes partitioned into separate *name spaces*. There is one name space for classes that come from the local file system, and a separate name space for each network source.

When a class is imported from across the network it is placed into the private name space associated with its origin. When a class references another class, it is first looked for in the name space for the local system (built-in classes), then in the name space of the referencing class. There is no way that an imported class can "spoof" a built-in class. Built-in classes can never accidentally

reference classes in imported name spaces—they can only reference such classes explicitly. Similarly, classes imported from different places are separated from each other.

Security in the Java Networking Package

Java's networking package provides the interfaces to handle the various network protocols (FTP, HTTP, Telnet, and so on). This is your front line of defense at the network interface level. The networking package can be set up with configurable levels of paranoia. You can

- Disallow all network accesses
- Allow all network accesses
- Allow network accesses to only the hosts from which the code was imported
- Allow network accesses only outside the firewall if the code came from outside

4.3 *Multithreading*

Sophisticated computer users become impatient with the do-one-thing-at-a-time mindset of the average personal computer. Users perceive that their world is full of multiple events all happening at once, and they like to have their computers work the same way.

Unfortunately, writing programs that deal with many things happening at once can be *much* more difficult than writing in the conventional single-threaded C and C++ style. You *can* write multithreaded applications in languages such as C and C++, but the level of difficulty goes up by orders of magnitude, and even then there are no assurances that vendors' libraries are "thread safe".

The major problem with explicitly programmed thread support is that you can never be quite sure you have acquired the locks you need and released them again at the right time. If you return from a method prematurely, for instance, or if an exception is raised, for another instance, your lock has not been released—deadlock is the usual result.

Java Supports Threads at the Language Level

Built in support for *threads* provides Java programmers with a powerful tool to improve perceived interactive performance of graphical applications. If your application needs to run animations and play music while scrolling the page and downloading a text file from a server, *multithreading* is the way to obtain fast, lightweight concurrency within a single process space. Threads are sometimes also called lightweight processes or execution contexts.

Threads are an essential keystone of the Java language. Threads are a class from which you can instantiate new Thread objects. The Thread class provides a rich collection of methods to start the thread, run the thread, stop the thread, and make enquiries about the thread's status.

Java thread support includes a sophisticated set of *synchronization primitives* based on the widely used *monitor* and *condition variable* paradigm introduced twenty years ago by C.A.R. Hoare and implemented in a production setting in Xerox PARC's Cedar/Mesa system. Integrating support for threads into the language makes them much easier to use and more robust. Much of the style of the Java language's integration of threads was modelled after Cedar and Mesa.

Java's threads are pre-emptive. If your applications are likely to be compute-intensive, you might consider how to give up control periodically to give other threads a chance to run. This will ensure better interactive response for graphical applications.

Integrated Thread Synchronization

The Java language supports multithreading at the language (syntactic) level and via support from its run-time system and thread objects. At the language level, methods within a class that are declared `synchronized` do not run concurrently. Such methods run under control of *monitors* to ensure that variables remain in a consistent state. Every class and instantiated object has its own monitor that comes into play if required.

Here are a couple of code fragments from the sorting demonstration in the HotJava web browser. The main points of interest are the two methods `stop` and `startSort`, which share a common variable called `kicker` (it kicks off the sort thread):

```
public synchronized void stop() {
    if (kicker != null) {
        kicker.stop();
```

There are literally hundreds of programming languages available for developers to write programs to solve problems in specific areas. The programming languages cover a spectrum ranging across fully interpreted languages such as *UNIX Shells, awk, TCL, Perl,* and so on, all the way to "programming to the bare metal" languages like C and C++.

Languages at the level of Shells and TCL, for example, are fully interpreted high level languages. They deal with "objects" (at least in the sense they can be said to deal with objects at all) at the system level—their objects are files and processes rather than data structures. Some of these languages are suitable for very fast prototyping—you can develop your ideas quickly, try out new approaches, and discard non-working approaches without investing enormous amounts of time in the process. Scripting languages are also highly portable. Their primary drawback is performance—they are generally much slower than either native machine code or interpreted byte codes. This tradeoff may well be reasonable if the run time of such a program is reasonably short and you use the program on an itinerant basis.

At the lowest level are compiled languages such as C and C++, in which you can develop large-scale programming projects that will deliver high performance. The high performance comes at a cost, however. Drawbacks include the high cost of debugging non-robust memory management systems and difficult to implement and use multithreading capabilities. And of course when you use C++, you have the perennial fragile superclass issue. Last but definitely not least, the binary distribution problem of compiled code becomes unmanageable in the context of heterogeneous platforms all over the Internet.

The Java language environment creates an extremely attractive middle ground between the very high-level and portable but slow scripting languages and the very low level and fast but non-portable and unreliable compiled languages. The Java language fits somewhere in the middle of this space. In addition to being extremely simple to program, highly portable and architecture neutral, the Java language provides a level of performance that's entirely adequate for all but the most compute-intensive applications.

4.7 Summary

From the discussion above, you can see that the Java language provides *high performance* while its *interpreted* nature makes it the ideal development platform for fast and fearless prototyping. From the previous chapters, you've seen that the Java language is extremely *simple* and *object oriented*. The language is *secure*

to survive in the network-based environment. The *architecture-neutral* and *portable* aspects of the Java language make it the ideal development language to meet the challenges of distributing *dynamically extensible* software across networks.

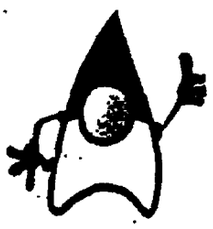
Now We Move On to the HotJava World-Wide Web Browser

These first four chapters have been your introduction to the Java language environment. You've learned about the capabilities of the Java language environment and its clear benefits to develop software for the distributed world. It's time to move on to the next chapter and take a look at the HotJava World-Wide Web browser—a major end-user application developed to make use of the dynamic features of the Java language environment.

The HotJava World-Wide Web Browser 5

It's a *jungle* out there,
So drink your Java

T-shirt caption from *Printer's Inc Cafe*, Palo Alto, California



The HotJava browser is a new World-Wide Web browser built entirely in the Java programming language. The HotJava browser is the first major end-user application created using the Java language and run-time system as a base. The HotJava browser not only showcases the powerful features of the Java language environment, it also provides an ideal platform for distributing Java programs across the most complex, distributed, heterogeneous network in the world—the Internet. The HotJava browser and its already rapidly growing Web population of Java language programs called *applets* (mini-applications), are the most compelling demonstration of the dynamic capabilities of the Java language environment.

The HotJava browser includes many innovative features and capabilities above and beyond the first generation of static Web browsers. The HotJava browser is *extensible*—its foremost feature is its ability to download Java programs (applets) from anywhere, even across networks, and execute them on the user's machine. It builds on the network browsing techniques established by Mosaic and other Web browsers and expands them by adding dynamic behavior that transforms static documents into dynamic applications. The

5

HotJava browser brings a much needed measure of *interactivity* to the concept of the Web browser, transforming the existing static data display of current generation Web browsers into a new dynamic, animation-oriented, interactive viewing system for hypertext. Content developers candistribute their applications across the Internet with the click of a button.

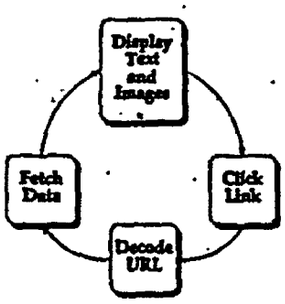
5.1 The Evolution of Cyberspace

The *Internet* has evolved into an amorphous ocean of data stored in many formats on a multiplicity of network hosts. Over time, various data storage and transmission protocols have evolved to impose some order on this chaos. One of the fastest growing areas of the net—the one we're primarily interested in here—is the World-Wide Web (WWW), which uses a hypertext-based markup system to enable users to navigate their way across the oceans of data.

Web browsers combine the functions of fetching data with figuring out what the data is and displaying it if possible. One of the most prevalent file formats broswers deal with is HyperText Markup Language, or HTML— a markup language that embeds simple text formatting commands within text to be formatted. The main key to the hypertext concept is HTML's use of *links* to other HTML pages either on the same host or elsewhere on the Internet.

A user in search of gold mining data, for instance, can follow links across the net from Mountain View, California, to the University of the Witwatersrand, South Africa, and arrive back at commercial data providers in Montreal, Canada, all within the context of tracing links in hypertext "pages."

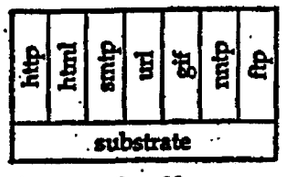
5.2 First Generation Browsers



What we could call the "first-generation" Web browsers—exemplified by NCSA Mosaic and Netscape Navigator—provide an illusion of being interactive. By using the (somewhat limited) language of HTML these browsers provide hypertext *links* on which you can click. The browser goes off across the network to fetch the data associated with that link, downloads the data, and displays it on your local screen. As we said, this is an illusion of interactivity.

This illustration depicts roughly the "interactive" flow of control in the first-generation Web browsers. As you see, it's not really interactive—it's just a fancy data fetching and display utility.

The HotJava browser brings a new twist to the concept of client-server computing. The general view of client-server computing is a big centralized server that clients connect to for a long time and from which they access data and applications. It is roughly a star with a big server in the middle and clients arrayed around it. The new model exemplified by the World-Wide Web is a wide-spread Web with short-lived connections between clients and many servers. The controlling intelligence shifts from the server to the client and the answer to "who's in charge?" shifts from the server to the client.



A conventional browser: a monolithic chunk, all bound tightly together.

The primary problem with the first-generation Web browsers is that they're built in a monolithic fashion with their awareness of every possible type of data, protocol, and behavior hard wired in order for them to navigate the Web. This means that every time a new data type, protocol, or behavior is invented, these browsers must be upgraded to be cognizant of the new situation. From the viewpoint of end users, this is an untenable position to be in. Users must continually be aware of what protocols exist, which browsers deal with those protocols, and which versions of which browsers are compatible with each other. Given the growth of the Internet, this situation is clearly out of control.

5.3 *The HotJava Browser—A New Concept in Web Browsers*

The HotJava browser solves the monolithic approach and moves the focus of interactivity away from the Web server and onto the Web *client*—that is, to the computer on which the user is browsing the Web. Because of its basis in the Java language system, a HotJava browser client can dynamically download segments of code that are executed right there on the client machine. Such Java-based "applets" (mini-applications) can provide full animation, play sound, and generally interact with the user in real time.

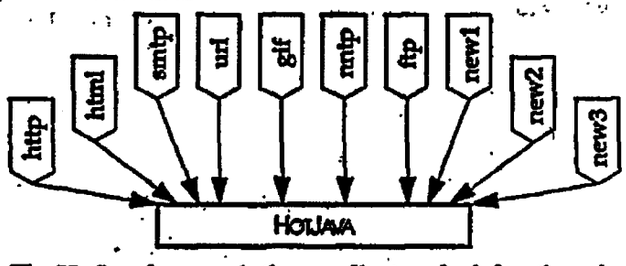
The HotJava browser removes the static limitations of the Mosaic generation of Web browsers with its ability to add arbitrary behavior to the browser. Using the HotJava browser you can add applications that range from interactive science experiments in educational material, to games and specialized shopping applications. You can implement interactive advertising, customized newspapers, and a host of application areas that haven't even been thought of yet. The capabilities of a Web browser whose behavior can be dynamically updated are open-ended.

Furthermore, the HotJava browser provides the means for users to access these applications in a new way. Software migrates transparently across the network as it's needed. You don't have to "install" software—it comes across the

network as you need it—perhaps after asking you to pay for it... Content developers for the World-Wide Web don't have to worry about whether or not some special piece of software is installed in a user's system—it just gets there automatically. This transparent acquiring of applications frees content developers from the boundaries of the fixed media types such as images and text and lets them do whatever they'd like.

5.4 The Essential Difference

The central difference between the HotJava browser and other browsers is that while these other browsers have knowledge of the Internet protocols hard-wired into them, the HotJava browser understands essentially none of them. What it *does* understand is to how find out about things it doesn't understand. The result of this lack of understanding is great flexibility and the ability to add new capabilities very easily.



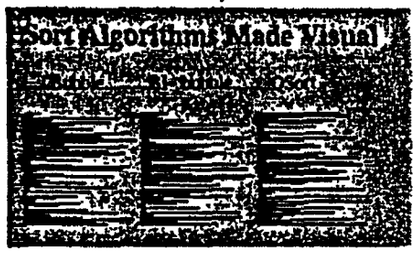
The HotJava browser is the coordinator of a federation of pieces, each with individual responsibility. New pieces can be added at any time. Pieces can be added from across the network, without needing to be concerned with what CPU architecture they were designed for and with reasonable confidence that they won't compromise the integrity of a user's system.

5.5 Dynamic Content

One of the most visible uses of the HotJava browser's ability to dynamically add to its capabilities is something we call *dynamic content*. For example, someone could write a Java language program following the HotJava API that implemented an interactive chemistry simulation. People browsing the net with the HotJava browser could easily get this simulation and interact with it,

rather than just having a static picture with some text. They can do this and be assured that the code that brings their chemistry experiment to life doesn't also contain malicious code that damages the system. Code that attempts to be malicious or which has bugs essentially can't breach the walls placed around it by the security and robustness features of the Java language.

For example, the following is a snapshot of the HotJava browser in use. Each diagram in the document represents a different sort algorithm. Each algorithm sorts an array of integers. Each horizontal line represents an integer: the length of the line corresponds to the value of the integer and the position of the line in the diagram corresponds to the position of the integer in the array.



In a book or HTML document, the author has to be content with these static illustrations. With the HotJava browser the author can enable the reader to click on the illustrations and see the algorithms animate:



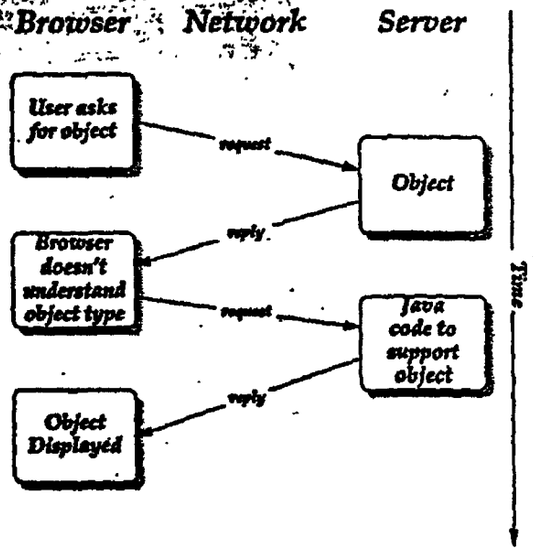


Using these dynamic facilities, content providers can define new types of data and behavior that meet the needs of their specific audiences, rather than being bound by a fixed set of objects.

5.6 Dynamic Types

The HotJava browser's dynamic behavior is also used for understanding different types of objects. For example, most Web browsers can understand a small set of image formats (typically GIF, X11 pixmap, and X11 bitmap). If they see some other type, they have no way to deal with it directly. The HotJava browser, on the other hand, can dynamically link the code from the host that has the image allowing it to display the new format. So, if someone invents a new compression algorithm, the inventor just has to make sure that a copy of the Java language code is installed on the server that contains the images they want to publish; they don't have to upgrade all the browsers in the world. The HotJava browser essentially upgrades itself on the fly when it sees this new type.

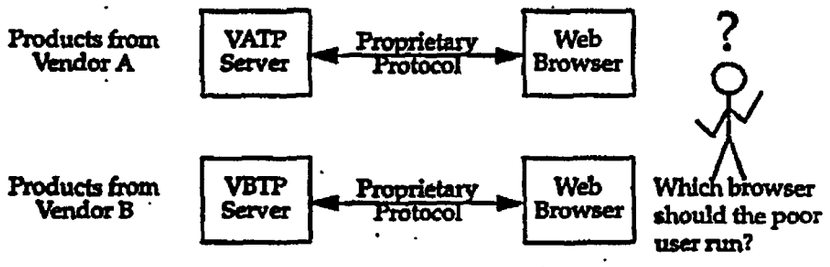
The following is an illustration of how HotJava negotiates with a server when it encounters an object of an unknown type:



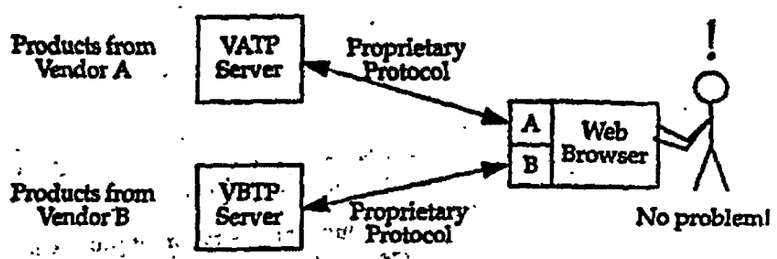
5.7 Dynamic Protocols

The protocols that Internet hosts use to communicate among themselves are key components of the net. For the World-Wide Web (WWW), HTTP is the most important of these communication protocols. In documents on the WWW a reference to a document is called a URL. The URL contains the name of the protocol, HTTP, that is used to find that document. Current Web browsers have the knowledge of HTTP built-in. Rather than having built-in protocol handlers, HotJava uses the protocol name to link in the appropriate handler. This allows new protocols to be incorporated dynamically.

The dynamic incorporation of protocols has special significance to how business is done on the Internet. Many vendors are providing new Web browsers and servers with added capabilities, such as billing and security. These capabilities most often take the form of new protocols. So each vendor comes up with their unique style of security (for example) and sells a server and browser that speak this new protocol. If a user wants to access data on multiple servers on which each has proprietary new protocols, the user needs multiple browsers. This is incredibly clumsy and defeats the synergistic cooperation that makes the World-Wide Web work.



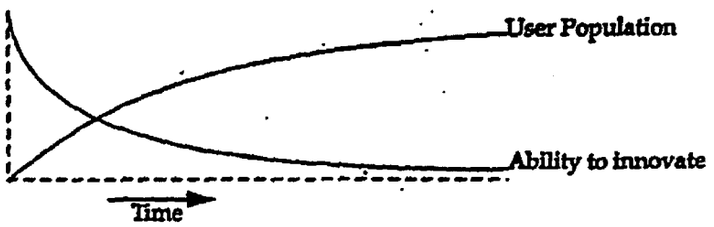
With the HotJava browser as a base, vendors can produce and sell exactly the piece that is their added value, and integrate smoothly with other vendors, creating a final result that is seamless and very convenient for the end user.



Protocol handlers get installed in a sequence similar to how content handlers get installed: The HotJava Browser is given a reference to an object (a URL). If the handler for that protocol is already loaded, it will be used. If not, the HotJava Browser will search first the local system and then the system that is the target of the URL.

5.8 *Freedom to Innovate*

Innovation on the Internet follows a pattern: initially: someone develops a technology. They're free to try all kinds of things since no one else is using the technology and there are no compatibility issues. Slowly, people start using it, and as they do, compatibility and interoperability concerns slow the pace of innovation. The Internet is now in a state where even simple changes that everyone agrees will have significant merit are very hard to make.



Within a community that uses the HotJava browser, individuals can experiment with new facilities while at the same time preserving compatibility and interoperability. Data can be published in new formats and distributed using new protocols and the implementations of these will be automatically and safely installed. There is no upgrade problem.

One need not be inventing new things to need these facilities. Almost all organizations need to be able to adapt to changing requirements. the HotJava browser's flexibility can greatly aid that. As new protocols and data types become important, they can be transparently incorporated.

5.9 *Implementation Details*

The basic structure of the HotJava browser is instructive. It is easiest understood from the operation of Mosaic:

Mosaic starts with a URL and fetches the object using the specified protocol. The *host* and *localinfo* fields are passed to the protocol handler. The result of this is a bag of bytes that contains the object that has been fetched. These bytes are inspected to determine the type of the data (for example, HTML document or JPEG image). From this type information, code to manipulate and view this data is invoked.

That's all there is to Mosaic. It's essentially very simple. But despite this, it is actually huge since it contains handlers for all of these data types. It's bundled together into one big monolithic lump.

In contrast, the HotJava browser is very small, since all of the protocol and data handlers are brought in from the outside. For example, when it calls the protocol handler, instead of having a table that has a fixed list of protocols that it understands, the HotJava browser instead uses this type string to derive a class name. The protocol handler for this type is dynamically linked in if it is missing. They can be linked in from the local system, or they can be linked in from definitions stored on the host where the URL was found, or anywhere else on the net that the browser suspects might be a good place to look. Similarly, the code to handle different types of data objects and different ways of viewing them.

What makes this federation of pieces and dynamic addition of capabilities possible is the Java language, the underlying programming language described in the earlier parts of this document, and the environment on which HotJava is built.

5.10 Security

Network security is of primary importance to Internet users, especially with the burgeoning growth of Internet commerce. Network-based applications must be able to defend themselves against a veritable gallimaufry of network viruses, worms, Trojan horses, and other forms of intruders. This section discusses the layers of defense provided by the Java language, run-time system, and the higher-level protocols of the HotJava browser itself.

One of the most important technical challenges in building a system like the HotJava browser is making it secure. Installing and running imported code fragments from across the network is an open invitation to all sorts of problems. On the one hand, such a facility provides great power that can be used to achieve very valuable ends, but on the other hand it can be subverted to become a breeding ground for computer viruses. The topic of safety is a very broad one and doesn't have a single answer. The HotJava browser has a series of facilities that layer and interlock to provide a fairly high degree of safety.

The First Layer—The Java Language Interpreter

The first layer of security in Java applications come from the ground rules of the Java language itself. These features have been described in detail in previous chapters in this paper.

When a code fragment is imported into HotJava it doesn't actually know whether or not the code fragment follows the Java language rules for safety, since it may not have been produced by the local Java language compiler. As described earlier, imported code fragments are subjected to a series of checks, starting with straightforward tests that the format of the code is correct and ending with a series of consistency checks by the Verifier.

The Next Layer—The Higher Level Protocols

Given this base set of guarantees that interfaces cannot be violated, higher level parts of the system implement their own protection mechanisms. For example, the file access primitives implement an access control list that controls read and write access to files by imported code (or code invoked by imported code). The defaults for these access control lists are very restrictive. If an attempt is made by a piece of imported code to access a file to which access has not been granted, a dialog box pops up to allow the user to decide whether or not to allow that specific access.

These access restrictions err on the conservative side, which makes constructing some very useful extensions impossible or awkward. We have a mechanism whereby public keys can be securely attached to code fragments that allows code with trusted public keys to have fewer restrictions. This mechanism isn't in the public release for legal reasons.

5.11 Summary

The HotJava Web browser—based upon the foundations of the Java language environment—brings a hitherto unrealized *dynamic* and *interactive* capability to the World-Wide Web. Dynamic *content*, dynamic *data types*, and dynamic *protocols* provide content creators with an entirely new tool that facilitates the burgeoning growth of electronic commerce and education. The advent of the dynamic and interactive capabilities provided by the HotJava Web browser brings the World-Wide Web to life, turning the Web into a new and powerful business and communication tool for all users.

5

Further Reading 6

I've got a little list.
I've got a little list.

Gilbert and Sullivan—*The Mikado*

The Java/HotJava Programmer's Guide

Sun Microsystems

`http://java.sun.com/progGuide/index.html`

This is the draft version of the Java/HotJava Programmer's Guide.

Pitfalls of Object-Oriented Development, by Bruce F. Webster
Published by M&T Books.

A collection of "traps to avoid" for people adopting object technology. Recommended reading—it alerts you to the problems you're likely to encounter and the solutions for them.

The Design and Evolution of C++, by Bjarne Stroustrop
Published by Addison Wesley

A detailed history of how we came to be where we are with C++.

NEXTSTEP Object-Oriented Programming and the Objective C Language. Addison Wesley Publishing Company, Reading, Massachusetts, 1993.

The book on Objective C. A good introduction to object-oriented programming concepts.

Discovering Smalltalk. By Wilf Lalonde. Benjamin Cummings, Redwood City, California, 1994.

An introduction to Smalltalk.

Eiffel: The Language. By Bertrand Meyer. Prentice-Hall, New York, 1992.

An introduction to the Eiffel language, written by its creator.

An Introduction to Object-Oriented Programming. By Timothy Budd. Addison Wesley Publishing Company, Reading, Massachusetts.

An introduction to the topic of object-oriented programming, as well as a comparison of C++, Objective C, SmallTalk, and Object Pascal.

Monitors; An Operating System Structuring Concept. By C. A. R. Hoare. Communications of the ACM, volume 17 number 10, 1974. Pages 549-557.

The original seminal paper on the concept of monitors as a means to synchronizing multiple concurrent tasks.

BEST COPY

Correct and Mail

09/97 FORM 9

BOX AF

PTO JC24 APR 12 1999 TRADEMARK OFFICE

Application No.: 08/882,580
Filed: June 25, 1997
Group Art Unit: 2773

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
Ten Post Office Square
Boston, Massachusetts 02109
Telephone: (617) 542-2290
Telecopier: (617) 451-0313

BOX AF
ASSISTANT COMMISSIONER FOR PATENTS
Washington, D.C. 20231

Date: April 8, 1999

Attorney
Docket No.: COMET-001XX

RECEIVED
APR 14 1999
Group 2700

Sir:

In re application of: James S. Rosen, et al.

Entitled: SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE

Transmitted herewith is an **amendment** in the above-identified application. The following checked items are applicable:

[ ] A Petition for Extension of Time for ___ month is hereby made, under §1.136(a); a check in the amount of _____ is enclosed per §1.17.

[X] **In the event a Petition for Extension of Time is required by this paper and not otherwise provided, such Petition is hereby made and authorization is provided herewith to charge Deposit Account No. 23-0804 for the cost of such extension.**

[ ] _____ is hereby appointed Associate Attorney by:
Registration No.:

_____
Attorney of Record:
Registration No.:

[ ] Other:

CLAIMS AFTER AMENDMENT:	MINUS PRIOR PAID CLAIMS:	EQUALS PRESENT EXTRA CLAIMS:	RATE:	ADDITIONAL FEE:
Independent	3 - 10	= 0	x $78.00 =	0
Total	69 - 132	= 0	x $18.00 =	0
[ ] Multiple Dependent Claims (1st presentation)			+ $260.00 =	0
				0
Small Entity filing, divide by 2. Previously submitted or verified statement must be attached (per §1.9, §1.27, §1.28)				0
				0

[X] No additional fee. [ ] The fee has been calculated above; a check in the amount of _____ is enclosed.

[X] **The Commissioner is hereby authorized to charge payment of any additional filing fees under §1.16 associated with this communication or credit any overpayment to Deposit Account No. 23-0804.**

I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: BOX AF, Assistant Commissioner for Patents, Washington, D.C. 20231 on April 8 1999.

David A. Dagg

Attorney of Record: David A. Dagg
Registration No.: 37,809

SUBMIT IN TRIPLICATE
DAD/cae/202706



[redacted] 2773

12/B
J. Douglas
4/15/99
(N.E.)

4/15/99

PATENT

RECEIVED
APR 14 1999
Group 2700

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application	:	James S. Rosen et al.
Application No.	:	08/882,580
Filed	:	June 25, 1997
For	:	SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR
Examiner	:	C. Jackson
Attorney's Docket	:	COMET-001XX

Group Art Unit: 2773

* *

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Box AF, Assistant Commissioner for Patents, Washington, D.C. 20231 on April 8, 1999.

By: David A. Dagg
David A. Dagg
Registration No. 37,809
Attorney for Applicant(s)

* *

AMENDMENT UNDER 37 C.F.R. § 1.116

BOX AF
Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

In response to the Final Office Action dated February 8, 1999, please amend the above identified patent application as follows:

In the Claims:

Please cancel claims 2, 30 and 72 through 132 without prejudice or dedication.

Please amend the Claims as follows:

1. (Twice Amended) A server system for modifying a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said system comprising:

cursor image data corresponding to said specific image;

cursor display code, said cursor display code operable to modify said cursor image; and

a first server computer for transmitting specified content information to said remote user terminal, said specified content information including at least one cursor display instruction indicating a location of said cursor image data, said cursor display instruction and said cursor display code operable to cause said user terminal to display a modified cursor image on said user's display in the shape and appearance of said specific image, wherein said specified content information is transmitted to said remote user terminal by said first server computer responsive to a request from said user terminal for said specified content information, and wherein said specified content information

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

further comprises information to be displayed on said display of said user's terminal, and wherein said cursor display code is operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image responsive to displaying of said information to be displayed on said display of said user's terminal.

3. (Twice Amended) The server system in accordance with claim 1 [2], wherein said specific image relates to at least a portion of said information to be displayed on said display of said user's terminal

29. (Twice Amended) A method for modifying an initial cursor image displayed on a display of a user terminal connected to at least one server, comprising:

receiving a request at said at least one server to provide specified content information to said user terminal;

providing said specified content information to said user terminal in response to said request, said specified content information including at least one cursor display instruction and at least one indication of cursor image data corresponding to a specific image; and

transforming said initial cursor image displayed on said

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

display of said user terminal into the shape and appearance of said specific image in response to said cursor display instruction, wherein said specified content information includes information that is to be displayed on said display of said user's terminal, and wherein said cursor display instruction indicates a cursor display code operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image responsive to displaying of said specified content information on said display of said user's terminal.

31. (Twice Amended) The method in accordance with claim 29 [30], wherein said transforming further comprises executing said cursor display code so as to display said specific cursor image while at least a portion of said information to be displayed on said display of said user's terminal.

32. (Twice Amended) The method in accordance with claim 29 [30], wherein said displaying of said specific image further comprises displaying advertising material related to at least a portion of said information to be displayed.

37. (Twice Amended) The method in accordance with claim 29 [30],

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

wherein said specific image has a shape and appearance relating to said information to be displayed.

50. (Twice Amended) An internet browser configured to modify a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said browser comprising:

means for receiving specified content information from a remote server, said specified content information comprising information to be displayed on said remote user's terminal and at least one cursor display instruction, wherein said cursor display instruction indicates cursor image data corresponding to said specific image;

means for recognizing said cursor display instruction in connection with processing said information to be displayed on said display; and

means for executing a cursor display code, responsive to said cursor display instruction and displaying of said information to be displayed on said display, said cursor display code being operable to modify said cursor image to said specific image.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

REMARKS

This amendment is filed responsive to the Examiner's Final Action dated February 8, 1999. All rejections and objections of the Examiner are respectfully traversed. Reconsideration is respectfully requested.

Claims 2, 30 and 72-132 have been cancelled.

Claims 1, 29 and 50 are independent.

Claims 1-71 are currently pending.

At paragraph 3 of the Office Action, the Examiner stated that claims 72-132 are directed to inventions that are independent or distinct from the invention of claims 1-71, and that claims 1-71 were constructively elected for prosecution in this case. Applicant hereby confirms the election of claims 1-71 for prosecution in this case, while reserving the right to pursue the remaining claims in divisional applications. Accordingly, claims 72-132 have been cancelled without prejudice.

At paragraphs 4-6 of the Office Action, the Examiner rejected claims 1-3, 9-14, 18-24, 27-31, 37-38, 43-52, 54-61 and 67-69 under 35 U.S.C. §103(a), citing U.S. Patent No. 5,710,897 of Schneider (Schneider hereinafter). Applicant respectfully traverses this rejection.

Nowhere in Schneider is there disclosed or suggested any

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

system or method for modifying a cursor image by transmitting content information to a remote user's terminal in response to a request for that content information, where the content information includes at least one cursor display instruction to modify the cursor image, and where the specified content information also includes information to be displayed on a display of a user's terminal, and where the cursor display code operates to process the cursor display instruction to modify the cursor image to the shape and appearance of said specific image ***responsive to displaying of the information to be displayed on the display of the user's terminal***, as is set forth in the present independent claims 1, 29 and 50. In distinct contrast, Schneider discloses a software system which changes system pointer images based on a current state of the user interface, without regard to what specific information is being displayed. The predetermined system pointer images of Schneider are associated with, and displayed in connection with individual user interface states. Specifically, Schneider teaches that each of the user modified pointer graphics represents ***a unique system operation***, and changes one from another depending on the location of the device pointer within the user interface. See Abstract. In further describing conditions under which the pointer graphics is modified, Schneider states in column 2 beginning at line 30 as follows:

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

> Since each pointer graphic represents a unique operation, the pointer graphics include a text pointer that is used only ***for text entry within a field in the graphical user interface***. Additionally, a wait pointer is included that is only used ***when the data processing system cannot accept any input***. Likewise, a size pointer, which may include one or more size pointers, is used ***when the device pointer moves over a particular corner of a sizable window*** displayed in the graphical user interface. Further, the plurality of pointer graphics includes a move pointer, which is used to indicate ***when a window can be moved***, and an illegal pointer, which is used ***when an object is placed on the device pointer over an unacceptable location or operation***. (emphasis added)

The emphasized text shows that the conditions under which the pointer graphic is modified in Schneider are completely unrelated to what information is being displayed, and are instead solely concerned with the relationship of the pointer to various aspects of the graphical user interface. Accordingly, the text pointer is displayed whenever the interface is capable of receiving text, independent of which specific text is currently being displayed, and the wait pointer is displayed when no further input can be received, regardless of any previously received information that may currently be displayed. Similarly, the size pointer is displayed whenever the pointer is positioned such that a window may be resized, regardless of what information is being displayed inside the window, the move pointer is displayed when the system is able to move an object in the display, without consideration of

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

what is being displayed in or on that object, and the illegal pointer becomes displayed in response to an object being moved over an illegal operation, whether or not information from within the object is currently being displayed. Thus it is clear that Schneider teaches a system for modifying system pointers based on the state of the graphical user interface with respect to the pointer position, and without responding in any way to the display of information received as part of any requested, specified content information.

In view of the foregoing, Applicants respectfully submit that the independent claims 1, 29 and 50 are patentably distinct over Schneider. With regard to claims 2-3, 9-14, 18-24, 27-31, 37-38, 43-52, 54-61 and 67-69, they each depend directly or indirectly from either claim 1, 29 or 50, and are believed to be patentably distinct over Schneider for at least the same reasons.

Further in paragraph 7 of the Office Action, the Examiner rejected claims 25-26, 53 and 70-71 under 35 U.S.C. 103(a), again citing Schneider, as well as "The Java Language Environment- A White Paper" by Gosling et al. ("Gosling" hereinafter). Applicant respectfully traverses this rejection.

Gosling discloses use of a virtual machine, where the virtual machine is capable of processing byte codes loaded from a server across the World Wide Web. Schneider is discussed with regard to

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

independent claims 1, 29 and 50 above. Nowhere in Gosling or Schneider, taken either independently or in combination, is there disclosed or suggested any system or method for modifying a cursor image by transmitting content information to a remote user's terminal in response to a request for that content information, where the content information includes at least one cursor display instruction to modify the cursor image, and where the specified content information also includes information to be displayed on a display of a user's terminal, and where the cursor display code operates to process the cursor display instruction to modify the cursor image to the shape and appearance of said specific image ***responsive to displaying of the information to be displayed on the display of the user's terminal***, as is set forth in the present independent claims 1, 29 and 50. Claims 25-26, 53 and 70-71 each depend directly or indirectly from claim 1, 29 or 50, and accordingly are believed to be patentable over Schneider and Gosling for at least the same reasons. Accordingly, reconsideration of all currently pending claims is respectfully requested.

As all pending claims are believed to be allowable, the application is believed to be in condition for allowance. Favorable action is respectfully requested.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Application No.: 08/882,580
Filed: June 25, 1997
Group Art Unit: 2773

The Examiner is encouraged to telephone the undersigned attorney to discuss any matter which would expedite allowance of the present application.

Respectfully submitted,

JAMES S. ROSEN ET AL.

By: David A. Dagg
David A. Dagg
Registration No. 37,809
Attorney for Applicant(s)

WEINGARTEN, SCHURGIN, GAGNEBIN
& HAYES LLP
Ten Post Office Square
Boston, MA 02109
Telephone: (617) 542-2290
Telecopier: (617) 451-0313

Dated: April 8 1999

DAD/cae
Enclosure
202618

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Transaction History Date 1999-04-23
Date information retrieved from USPTO Patent Application Information Retrieval (PAIR) system records at www.uspto.gov

BEST COPY

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

09/882,580

SERIAL NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NO.
[illegible]	[illegible]	[illegible]	[illegible]

[illegible]

EXAMINER	
[illegible]	
ART UNIT	PAPER NUMBER
[illegible]	#13

DATE MAILED: [illegible]

Below is a communication from the EXAMINER in charge of this application

COMMISSIONER OF PATENTS AND TRADEMARKS

ADVISORY ACTION

☒ THE PERIOD FOR RESPONSE:

~~is extended to run~~ _____ or continues to run 3 from the date of the final rejection

b) ☒ expires three months from the date of the final rejection or as of the mailing date of this Advisory Action, whichever is later. In no event however, will the statutory period for the response expire later than six months from the date of the final rejection.

Any extension of time must be obtained by filing a petition under 37 CFR 1 136(a), the proposed response and the appropriate fee. The date on which the response, the petition , and the fee have been filed is the date of the response and also the date for the purposes of determining the period of extension and the corresponding amount of the fee. Any extension fee pursuant to 37 CFR 1 17 will be calculated from the date of the originally set shortened statutory period for response or as set forth in b) above.

☐ Appellant's Brief is due in accordance with 37 CFR 1 192(a).

☒ Applicant's response to the final rejection, filed 4/12/99 has been considered with the following effect, but it is not deemed to place the application in condition for allowance:

1. ☐ The proposed amendments to the claim and /or specification will not be entered and the final rejection stands because:

   a. ☐ There is no convincing showing under 37 CFR 1.116(b) why the proposed amendment is necessary and was not earlier presented.

   b. ☐ They raise new issues that would require further consideration and/or search. (See Note).

   c. ☐ They raise the issue of new matter. (See Note).

   d. ☐ They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal.

   e. ☐ They present additional claims without cancelling a corresponding number of finally rejected claims.

   NOTE: _____

2. ☐ Newly proposed or amended claims _____ would be allowed if submitted in a separately filed amendment cancelling the non-allowable claims.

3. ☒ Upon the filing an appeal, the proposed amendment ☒ will be entered ☐ ~~will not be entered~~ and the status of the claims will be as follows:

   Claims allowed: _____
   Claims objected to: _____
   Claims rejected: 1, 3-29, 31-71

   However;

   ☐ Applicant's response has overcome the following rejection(s): _____

4. ☒ The affidavit, exhibit or request for reconsideration has been considered but does not overcome the rejection because The distinction argued by applicant is not present, b/c the interface of Schneider has content that drives the interface to the state when the pointer can be used, just as the advertisement in applicant's invention is in state.

5. ☐ The affidavit or exhibit will not be considered because applicant has not shown good and sufficient reasons why it was not earlier presented.

☐ The proposed drawing correction ☐ has ☐ has not been approved by the examiner.

☐ Other

RAYMOND J. BAYERL
PRIMARY EXAMINER
ART UNIT 2773

PTOL-303 (REV. 5-89)

*U.S GPO. 1997-417-381/62704

UNITED ST[illegible]'S DEPARTMENT OF COMMERCE
Patent and [illegible]rademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

09/882,580

APPLICATION NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NO.

EXAMINER	
ART UNIT	PAPER NUMBER
	#14

DATE MAILED:

INTERVIEW SUMMARY

All participants (applicant, applicant's representative, PTO personnel):

(1) David Dugg (Reg # 37,809) (3) Adam Solomon

(2) James Reisen (4) Tom Schmitter

Date of Interview 5/11/99

Type: ☐ Telephonic ☒ Personal (copy is given to ☐ applicant ☒ applicant's representative).

Exhibit shown or demonstration conducted: ☒ Yes ☐ No If yes, brief description: Demonstration of the applicant's invention.

Agreement ☐ was reached. ☒ was not reached.

Claim(s) discussed: 1, 29, 50

Identification of prior art discussed: Schneider (5,710,897)

Description of the general nature of what was agreed to if an agreement was reached, or any other comments:

How to recite limitations which would accurately describe the applicants invention as well as distinguish it from Schneider (5,710,897).

(A fuller description, if necessary, and a copy of the amendments, if available, which the examiner agreed would render the claims allowable must be attached. Also, where no copy of the amendments which would render the claims allowable is available, a summary thereof must be attached.)

1. ☐ It is not necessary for applicant to provide a separate record of the substance of the interview.

Unless the paragraph above has been checked to indicate to the contrary. A FORMAL WRITTEN RESPONSE TO THE LAST OFFICE ACTION IS NOT WAIVED AND MUST INCLUDE THE SUBSTANCE OF THE INTERVIEW. (See MPEP Section 713.04). If a response to the last Office action has are ready been filed, APPLICANT IS GIVEN ONE MONTH FROM THIS INTERVIEW DATE TO FILE A STATEMENT OF THE SUBSTANCE OF THE INTERVIEW.

2. ☐ Since the Examiner's interview summary above (including any attachments) reflects a complete response to each of the objections, rejections and requirements that may be present in the last Office action, and since the claims are now allowable, this completed form is considered to fulfill the response requirements of the last Office action. Applicant is not relieved from providing a separate record of the interview unless box 1 above is also checked.

Examiner Note: You must sign this form unless it is an attachment to another form.

FORM PTOL-413 (REV.1-96)

David C. Dugg

★ U.S GPO 1996-410-232/40051

FAX RECEIVED
JUN 08 1999
GROUP 2700

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEN POST OFFICE SQUARE
BOSTON, MASSACHUSETTS 02109

INTELLECTUAL PROPERTY LAW
PATENTS, TRADEMARKS AND COPYRIGHTS

TELEPHONE
(617) 542-2290
FACSIMILE
(617) 451-0313

OFFICIAL

JOSEPH WEINGARTEN (1919-1984)
STANLEY M. SCHURGIN
CHARLES L. GAGNEBIN III
PAUL J. HAYES
VICTOR B. LEBOVICI
DEAN GRAHAM BOSTOCK
EUGENE A. FEHER
BEVERLY E. HJORTH
HOLLIDAY C HEINE, PH.D.

GORDON R. MORIARTY
HOLMES W. ANDERSON
DAVID W. ROUILLE
RUSSELL W. BINNS, JR.
CHRISTOPHER J. LUTZ
JAMES F. THOMPSON
GWENDOLYN H. YIP
DAVID A. DAGG
PAUL J. CRONIN

FACSIMILE COVER SHEET

DATE: June 8, 1999

TO: Examiner C. Jackson — Fax No. Dialed: 703-308-6606

FROM: David A. Dagg — No. of pages transmitted (including this page):

Our File: COMET-001XX — Time:

Your Ref: 08/882,580 — Sent by: Rita Chalifoux

A confirmation copy of this transmission will not be mailed unless the following is checked: [ ]

MESSAGE

PLEASE DELIVER DIRECTLY TO:

EXAMINER Chadwick Jackson, Tel. (703) 308-9572.

GROUP ART UNIT 2773

FOR ENTRY

Transmitted herewith please find:

1. Continued Prosecution Application (CPA) (2) pages
2. Preliminary amendment (19) pages
3. Receipt for Facsimile Transmitted CPA (1) page

The Commissioner is hereby authorized to Charge Deposit Account No. 23-0804 for any additional filing fees associated with this communication or credit any overpayment.

THIS MESSAGE MAY CONTAIN CONFIDENTIAL OR PRIVILEGED INFORMATION INTENDED ONLY FOR THE PERSON(S) IDENTIFIED ABOVE. IF IT HAS BEEN RECEIVED AT ANY OTHER PLACE OR HAS NOT BEEN CLEARLY RECEIVED, PLEASE CALL THE ABOVE IDENTIFIED SENDING PARTY COLLECT FOR INSTRUCTIONS. DO NOT SHOW OR DISTRIBUTE THIS MESSAGE TO ANYONE OTHER THAN THE INTENDED RECIPIENT(S). THANK YOU.

DAD/rec/Enc./205974

BEST COPY

PTO/SB/28A (8/98)
Approved for use through 09/30/2000. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

If this RECEIPT is included with a request for a CPA filed by facsimile transmission, it will be date stamped and mailed to the ADDRESS in Item 1.

1. ADDRESS | *Applicant's Mailing Address for this receipt must be CLEARLY PRINTED or TYPED in the box below.*

Weingarten, Schurgin, Gagnebin
& Hayes LLP
Ten Post Office Square
Boston, MA 02109

RECEIPT FOR FACSIMILE TRANSMITTED CPA

NOTE: ***By this receipt, the PTO (a) acknowledges that a request for a CPA was filed by facsimile transmission on the date stamped below by the PTO and (b) verifies only that the application number provided by the applicant on this receipt is the same as the application number provided on the accompanying request for a CPA. This receipt CANNOT be used to acknowledge receipt of any paper(s) other than the request for a CPA.***

2. APPLICATION IDENTIFICATION:
(Provide at least enough information to identify the application)

a. For prior application

Application No: 08/882,580
Filing Date: June 25, 1997
Title: SERVER SYSTEM AND METHOD FOR MODIFYING.
Attorney Docket No: COMET-001XX
First Named Inventor: James S. Rosen

b. For instant CPA application

New Attorney Docket No: COMET-001XX
(if applicable)

The PTO date stamp, which appears in the box to the right, is an acknowledgement by the PTO of receipt of a request for a CPA filed by facsimile transmission on the date indicated below.

(THIS AREA FOR PTO DATE STAMP USE)

PTO HANDLING INSTRUCTIONS:

Please stamp area to the right with the date the complete transmission of the request for a CPA was received in the PTO and also include the PTO organization name that provided the date stamp (stamp may include both items). Verify that the application number provided by applicant on this receipt is the same as the application number provided by applicant on the request for a CPA accompanying this receipt. If there is an inconsistency between the application number provided on this receipt and the request for a CPA, strike through the inconsistent application number provided on this receipt and insert the correct application number, if possible. Then place in a window envelope and mail.

Burden Hour Statement: This form is estimated to take 0.4 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

#15
6-16-99
B. Hilliard

12/97 FORM 2 NOTE: PARENT WILL BE ABANDONED Sheet 1 of 2

REQUEST FORM FOR CONTINUED PROSECUTION APPLICATION (CPA) UNDER 37 CFR 1.53 d)

This application Atty Docket No.: COMET-001XX	Anticipated Classification Class: Subclass:
Prior application Atty Docket No.: COMET-001XX	First Named Inventor: James S. Rosen et al. Examiner: C. Jackson Art Unit: 2773

Express Mail No:
Date: June 8, 1999

BOX CPA
Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

This is a Request for filing a continued prosecution application under §1.53(d) of prior Application No. 08/882,580, filed June 25, 1997, entitled: SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE

[ ] This application is filed by fewer than all the inventors named in the prior application. 37 CFR 1.53(d)(4).

DELETE the following inventor(s) named in the prior nonprovisional application:

[X] A Petition for Extension of Time for 1 month(s) is hereby made, under provisions of §1.136(a). The Commissioner is hereby authorized to charge payment of this extension to Deposit Account No. 23-0804.

Small entity status:

[ ] A small entity statement is enclosed.
[X] A small entity statement was filed in the prior nonprovisional application and such status is still proper and desired.
[ ] Is no longer claimed.

1. [ ] Enter the unentered amendment previously filed on _____ per §1.116.
2. [X] A preliminary amendment is being transmitted herewith.

CLAIMS FILED:	MINUS BASE:	EXTRA CLAIMS:	RATE:	BASIC FEE:
				$760.00
Independent	9 - 3	= 6	x $78.00 =	468.00
Total	75 - 20	= 55	x $18.00 =	990.00
[ ] Multiple Dependent Claims (1st presentation)			+ $260.00 =	0.00
				$2,218.00
Small Entity filing, divide by 2. (Note: verified statement must be attached per §1.9, §1.27, §1.28.)				$1,109.00
				$1,109.00

[X] The filing fee has been calculated based on claims existing in the prior application as amended at 1 d: 2 above; The Commissioner is hereby authorized to charge payment of the total filing fee to Deposit Account No. 23-0804.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

SUBMIT IN TRIPLICATE

812/97 FORM 2 Sheet 2 of 2

REQUEST FORM FOR CONTINUED PROCESSING APPLICATION (CPA) UNDER 37 CFR 1.53(d) (CONTINUED)

Attorney
Docket No.: COMET-001X1

Date: June 8, 1999

[ ] Power of Attorney in the originally-filed application has been granted to one or more of the registered attorneys listed below The attorneys listed below not previously granted power in the originally-filed application, as well as _____, are hereby given associate power:
Registration No.:

Stanley M. Schurgin, Reg. No. 20,979
Charles L. Gagnebin III, Reg. No. 25,467
Paul J. Hayes, Reg. No. 28,307
Victor B. Lebovici, Reg. No. 30,864
Eugene A. Feher, Reg. No. 33,171
Beverly E. Hjorth, Reg. No. 32,033
Holliday C. Heine, Reg. No. 34,346
Gordon R. Moriarty, Reg. No. 38,973

By _____
Attorney of Record:
Registration No.:

[X] Other: Receipt for Facsimile Transmitted CPA

[X] In the event a Petition for Extension of Time under 37 CFR §1.17 is required by this paper and not otherwise provided, such Petition is hereby made and authorization is provided herewith to charge Deposit Account No. 23-0804 for the cost of such extension.

[X] The Commissioner is hereby authorized to charge payment of any additional filing fees under 37 CFR §1.16 associated with this communication or credit any overpayment to Deposit Account No. 23-0804.

Address all future communications to:
WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
Ten Post Office Square
Boston, Massachusetts 02109
Telephone: (617) 542-2290
Telecopier: (617) 451-0313

The filing of this CPA is a request to expressly abandon the prior application as of the filing date of the request for a CPA. It is understood that the filing of this CPA will be construed to include a waiver of confidentiality by the applicant under 35 U.S.C. 122 to the extent that any member of the public who is entitled under the provisions of 37 CFR 1.14 to access to, copies of, or information concerning, the prior application may be given similar access to, copies of, or similar information concerning, the other application or applications in the file jacket.

David A. Dagg
Attorney of Record: David A. Dagg
Registration No.: 37,809

DAD/rec
Enc.
204072

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

#16/C
6-16-99
BHilliard

FAX RECEIVED
JUN 0 8 1999
GROUP 2700

OFFICIAL
PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application	:	James S. Rosen et al.
Application No.	:	08/882,580
Filed	:	June 25, 1997
For	:	SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR
Examiner	:	C. Jackson
Attorney's Docket	:	COMET-001XX

Group Art Unit: 2773

* *

I hereby certify that this correspondence is being sent via Facsimile to Examiner C. Jackson in the United States Patent and Trademark Office at (703) 308-6606 on June 8, 1999.

By: David A Dagg
David A. Dagg
Registration No. 37,809
Attorney for Applicant(s)

* *

PRELIMINARY AMENDMENT

ATTN: BOX CPA
Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

This preliminary amendment is responsive to the Final Office Action dated February 8, 1999, in the above identified application for United States Patent. A request for a Continued Prosecution Application is being filed herewith. Prior to examination, please amend the application as follows:

Application No.: 08/882,580
Filed: June 25, 1997
Group Art Unit: 2773

In the Claims:

Please cancel claims 2, 30 and 72 through 132 without prejudice or dedication.

Please amend the Claims as indicated below:

1. (Twice Amended) A server system for modifying a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said system comprising:

cursor image data corresponding to said specific image;

cursor display code, said cursor display code operable to modify said cursor image; and

a first server computer for transmitting specified content information to said remote user terminal, said specified content information including at least one cursor display instruction indicating a location of said cursor image data, said cursor display instruction and said cursor display code operable to cause said user terminal to display a modified cursor image on said user's display in the shape and appearance of said specific image, wherein said specified content information is transmitted to said remote user terminal by said first server computer responsive to a request from said user terminal for said specified content information, and wherein said specified content information

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Application No.: 08/882,580
Filed: June 25, 1997
Group Art Unit: 2773

further comprises information to be displayed on said display of said user's terminal, said specific image including content corresponding to at least a portion of said information to be displayed on said display of said user's terminal, and wherein said cursor display code is operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image responsive to displaying of said at least a portion of said information to be displayed on said display of said user's terminal.

~~7~~ 8. (Twice Amended) The server system in accordance with claim 1 [2], wherein said specific image relates to at least a portion of said information to be displayed on said display of said user's terminal.

~~28~~ 29. (Twice Amended) A method for modifying an initial cursor image displayed on a display of a user terminal connected to at least one server, comprising:

receiving a request at said at least one server to provide specified content information to said user terminal;

providing said specified content information to said user terminal in response to said request, said specified content

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Application No.: 08/882,580
Filed: June 25, 1997
Group Art Unit: 2773

information including at least one cursor display instruction and at least one indication of cursor image data corresponding to a specific image; and

transforming said initial cursor image displayed on said display of said user terminal into the shape and appearance of said specific image in response to said cursor display instruction, wherein said specified content information includes information that is to be displayed on said display of said user's terminal, wherein said specific image includes content corresponding to at least a portion of said information to be displayed on said display of said user's terminal, and wherein said cursor display instruction indicates a cursor display code operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image responsive to displaying of said at least a portion of said information that is to be displayed on said display of said user's terminal.

31. (Twice Amended) The method in accordance with claim 29 [30], wherein said transforming further comprises executing said cursor display code so as to display said specific cursor image while at least a portion of said information to be displayed is displayed on said display of said user's terminal.

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Application No.: 08/882,590
Filed: June 25, 1997
Group Art Unit: 2773

30

32. (Twice Amended) The method in accordance with claim 29 [30], wherein said displaying of said specific image further comprises displaying advertising material related to at least a portion of said information to be displayed.

C4

35

37. (Twice Amended) The method in accordance with claim 29 [30], wherein said specific image has a shape and appearance relating to said information to be displayed.

C5

48

50. (Twice Amended) An internet browser computer program stored on a computer readable medium, said internet browser configured to modify a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said internet browser comprising:

program code operable to receive [means for receiving] specified content information from a remote server, said specified content information comprising information to be displayed on said remote user's terminal and at least one cursor display instruction, wherein said specific image includes content corresponding to at least a portion of said information to be displayed on said remote user's terminal, wherein said cursor display instruction indicates cursor image data corresponding to said specific image;

C6

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

program code operable to recognize [means for recognizing] said cursor display instruction in connection with processing said information to be displayed on said display; and

program code operable to execute [means for executing] a cursor display code, responsive to said cursor display instruction and displaying of said at least a portion of said information to be displayed on said display, said cursor display code being operable to modify said cursor image to said specific image.

C6

~~51~~ 49. (Twice Amended) The internet browser in accordance with claim ~~50~~ 48 further comprising program code operable [means] to retrieve said cursor image data from a prespecified server, when said cursor image data is not stored in said user terminal.

~~52~~ 50. (Amended) The internet browser in accordance with claim ~~50~~ 48 further comprising program code operable [means] to retrieve said cursor display code from a prespecified server, when said cursor display code is not stored in said user terminal.

C7

~~53~~ 51. (Twice Amended) The internet browser in accordance with claim ~~50~~ 48 further comprising program code operable [means] to determine whether cursor display instructions received by said user terminal were transmitted by an authorized server.

C8

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Application No.: 08/882,580
Filed: June 25, 1997
Group Art Unit: 2773

C8

52 ~~54~~. (Twice Amended) The internet browser in accordance with claim 48 ~~50~~ further comprising program code operable to transmit [means for transmitting] statistical information to a prespecified server so as to provide information relating to the usage of said specific image.

C9

54 ~~56~~. (Twice Amended) The internet browser in accordance with claim 53 ~~55~~ further comprising program code operable [means] to store said cursor display code in a memory located locally to said user terminal.

55 ~~57~~. (Twice Amended) The internet browser in accordance with claim 48 ~~50~~ further comprising program code operable [means] to provide audio clips corresponding to said display of said specific image.

C10

57 ~~59~~. (Twice Amended) The internet browser in accordance with claim 50 further comprising program code operable [means] to provide animated images corresponding to said specific image.

Please add the following new claims:

C11

70 ~~135~~. (New) A server system for modifying a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said system comprising:

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Application No.: 08/882,580
Filed: June 25, 1997
Group Art Unit: 2773

cursor image data corresponding to said specific image;

cursor display code, said cursor display code operable to modify said cursor image; and

a first server computer for transmitting specified content information to said remote user terminal, said specified content information including at least one cursor display instruction indicating a location of said cursor image data, said cursor display instruction and said cursor display code operable to cause said user terminal to display a modified cursor image on said user's display in the shape and appearance of said specific image, wherein said specified content information is transmitted to said remote user terminal by said first server computer responsive to a request from said user terminal for said specified content information, and wherein said specified content information further comprises information to be displayed on said display of said user's terminal, said specific image including content corresponding to at least a portion of said information to be displayed on said display of said user's terminal, and wherein said cursor display code is operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image responsive to movement of said cursor image over a display of said at least a portion of said information to be displayed on said display of said user's terminal.

C''

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Application No.: 08/882,530
Filed: June 25, 1997
Group Art Unit: 2773

71

~~134.~~ (New) A server system for modifying a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said system comprising:

cursor image data corresponding to said specific image;

cursor display code, said cursor display code operable to modify said cursor image; and

a first server computer for transmitting specified content information to said remote user terminal, said specified content information including at least one cursor display instruction indicating a location of said cursor image data, said cursor display instruction and said cursor display code operable to cause said user terminal to display a modified cursor image on said user's display in the shape and appearance of said specific image, wherein said specified content information is transmitted to said remote user terminal by said first server computer responsive to a request from said user terminal for said specified content information, and wherein said specified content information further comprises information to be displayed on said display of said user's terminal, said specific image including content corresponding to at least a portion of said information to be displayed on said display of said user's terminal, and wherein said cursor display code is operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image responsive to movement

C11

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Application No.: 08/882,580
Filed: June 25, 1997
Group Art Unit: 2773

of said cursor image over a specified location on said display of said user's terminal.

12

~~125.~~ (New) A method for modifying an initial cursor image displayed on a display of a user terminal connected to at least one server, comprising:

receiving a request at said at least one server to provide specified content information to said user terminal;

providing said specified content information to said user terminal in response to said request, said specified content information including at least one cursor display instruction and at least one indication of cursor image data corresponding to a specific image; and

C 11

transforming said initial cursor image displayed on said display of said user terminal into the shape and appearance of said specific image in response to said cursor display instruction, wherein said specified content information includes information that is to be displayed on said display of said user's terminal, wherein said specific image includes content corresponding to at least a portion of said information that is to be displayed on said display of said user's terminal, and wherein said cursor display instruction indicates a cursor display code operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542 2290
FAX (617) 451-0313

responsive to movement of said cursor image over a display of said at least a portion of said information to be displayed on said display of said user's terminal.

73

~~126.~~ (New) A method for modifying an initial cursor image displayed on a display of a user terminal connected to at least one server, comprising:

receiving a request at said at least one server to provide specified content information to said user terminal;

providing said specified content information to said user terminal in response to said request, said specified content information including at least one cursor display instruction and at least one indication of cursor image data corresponding to a specific image; and

11

transforming said initial cursor image displayed on said display of said user terminal into the shape and appearance of said specific image in response to said cursor display instruction, wherein said specified content information includes information that is to be displayed on said display of said user's terminal, wherein said specific image includes content corresponding to at least a portion of said information that is to be displayed on said display of said user's terminal, and wherein said cursor display instruction indicates a cursor display code operable to process said cursor display instruction to modify said cursor image to said cursor image in the shape and appearance of said specific image

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

responsive to movement of said cursor image over a specified location on said display of said user's terminal.

~~137.~~ 74 (New) An internet browser computer program stored on a computer readable medium, said internet browser configured to modify a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said internet browser comprising:

program code operable to receive specified content information from a remote server, said specified content information comprising information to be displayed on said remote user's terminal and at least one cursor display instruction, wherein said specific image includes content corresponding to at least a portion of said information to be displayed on said remote user's terminal, and wherein said cursor display instruction indicates cursor image data corresponding to said specific image;

program code operable to recognize said cursor display instruction in connection with processing said information to be displayed on said display; and

program code operable to execute a cursor display code, responsive to said cursor display instruction and movement of said cursor image over a display of said at least a portion of said information to on said remote user's terminal, said cursor display code being operable to modify said cursor image to said specific image.

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Application No.: 08/882,530
Filed: June 25, 1997
Group Art Unit: 2773

75

138. (New) An internet browser computer program stored on a computer readable medium, said internet browser configured to modify a cursor image to a specific image having a desired shape and appearance displayed on a display of a remote user's terminal, said internet browser comprising:

program code operable to receive specified content information from a remote server, said specified content information comprising information to be displayed on said remote user's terminal and at least one cursor display instruction, wherein said specific image includes content corresponding to at least a portion of said information to be displayed on said remote user's terminal, and wherein said cursor display instruction indicates cursor image data corresponding to said specific image;

program code operable to recognize said cursor display instruction in connection with processing said information to be displayed on said display; and

program code operable to execute a cursor display code, responsive to said cursor display instruction and movement of said cursor image over a specified location on said display, said cursor display code being operable to modify said cursor image to said specific image.

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Application No.: 08/882,580
Filed: June 25, 1997
Group Art Unit: 2773

REMARKS

This Preliminary Amendment is responsive to the Examiner's Final Action dated February 8, 1999. A request for a Continued Prosecution Application is being filed herewith. All rejections and objections of the Examiner are respectfully traversed. Reconsideration is respectfully requested.

Claims 2, 30 and 72-132 have been cancelled.

Claims 133-138 have been added.

Claims 1, 29, 50 and 133-138 are independent.

Claims 1-71 and 133-138 are currently pending.

At paragraph 3 of the Office Action, the Examiner stated that claims 72-132 are directed to inventions that are independent or distinct from the invention of claims 1-71, and that claims 1-71 were constructively elected for prosecution in this case. Applicants hereby confirm the election of claims 1-71 for prosecution in this case, while reserving the right to pursue the remaining claims in divisional applications. Accordingly, claims 72-132 have been cancelled without prejudice.

At paragraphs 4-6 of the Office Action, the Examiner rejected claims 1-3, 9-14, 18-24, 27-31, 37-38, 43-52, 54-61 and 67-69 under 35 U.S.C. §103(a), citing U.S. Patent No. 5,710,897 of Schneider (Schneider hereinafter). Applicant respectfully

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

traverses this rejection.

Nowhere in Schneider is there disclosed or suggested any system or method for modifying a cursor image by transmitting content information to a remote user's terminal in response to a request for that content information, where the content information includes at least one cursor display instruction to modify the cursor image, and where the specified content information also includes information to be displayed on a display of a user's terminal, and where the cursor display code operates to process the cursor display instruction to modify the cursor image to the shape and appearance of said specific image responsive to displaying of at least a portion of the information to be displayed on the display of the user's terminal, and wherein ***the specific image includes content corresponding to said at least a portion of said information to be displayed on said display of said user's terminal***, as is set forth in the present independent claims 1, 29 and 50.

In distinct contrast, Schneider discloses a software system which changes system pointer images based on a current state of the user interface, without regard to what specific information is being displayed. Schneider contains no hint or suggestion of even the desirability of including content information within a pointer image. There is, therefore, no relationship between specific

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

information being displayed in Schneider, and any pointer image. In clear contrast to the present system of claims 1, 29 and 50, the predetermined system pointer images of Schneider are associated with, and displayed in connection with, *corresponding user interface states*. Schneider specifically teaches various sets of user modified pointer graphics, in which each pointer graphic represents ***a unique system operation***. See Abstract. In further describing the specific substitute pointer graphics, Schneider states in column 2 beginning at line 30 as follows:

> Since each pointer graphic ***represents a unique operation***, the pointer graphics include a ***text pointer*** that is used only for text entry within a field in the graphical user interface. Additionally, a ***wait pointer*** is included that is only used when the data processing system cannot accept any input. Likewise, a ***size pointer***, . . . is used when the device pointer moves over a particular corner of a sizable window displayed in the graphical user interface. Further, the plurality of pointer graphics includes a ***move pointer***, which is used to indicate when a window can be moved, and an ***illegal pointer***, which is used when an object is placed . . . over an unacceptable location or operation. (emphasis added)

The emphasized text shows that the pointer graphics of Schneider include no content information, and are in no way related to what information is currently being displayed. Figures 3-5 and 9-15 of Schneider specifically show pointer images which reflect system operations, and which display no content

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

information. Specifically, the pointer images of Figures 3-5 and 9-15 of Schneider include various combinations of arrows and/or hand-shapes illustrating corresponding system states or operations. Thus it is clear that the substitute pointer images of Schneider include no content information, and are modified independently from the display of any particular content information.

In view of the foregoing, Applicants respectfully submit that the independent claims 1, 29 and 50 are patentably distinct over Schneider. With regard to claims 3, 9-14, 18-24, 27-29, 31, 37-38, 43-52, 54-61 and 67-69, they each depend directly or indirectly from either claim 1, 29 or 50, and are believed to be patentably distinct over Schneider for at least the same reasons.

Further in paragraph 7 of the Office Action, the Examiner rejected claims 25-26, 53 and 70-71 under 35 U.S.C. 103(a), again citing Schneider, as well as "The Java Language Environment- A White Paper" by Gosling et al. ("Gosling" hereinafter). Applicants respectfully traverse this rejection.

Gosling discloses use of a virtual machine, where the virtual machine is capable of processing byte codes loaded from a server across the World Wide Web. Schneider is discussed with regard to independent claims 1, 29 and 50 above. Nowhere in Gosling or Schneider, taken either independently or in combination, is there

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

disclosed or suggested any system or method for modifying a cursor image by transmitting content information to a remote user's terminal in response to a request for that content information, where the content information includes at least one cursor display instruction to modify the cursor image, and where the specified content information also includes information to be displayed on a display of a user's terminal, and where the cursor display code operates to process the cursor display instruction to modify the cursor image to the shape and appearance of said specific image responsive to displaying of at least a portion of the information to be displayed on the display of the user's terminal, and wherein ***the specific image includes content corresponding to said at least a portion of said information to be displayed on said display of said user's terminal***, as is set forth in the present independent claims 1, 29 and 50. Claims 25-26, 53 and 70-71 each depend directly or indirectly from claim 1, 29 or 50, and accordingly are believed to be patentable over Schneider and Gosling for at least the same reasons. Reconsideration of all currently pending claims is respectfully requested.

Newly added claims 133-138 are also believed to be patentably distinct over Schneider and Gosling for at least the same reasons as discussed above with regard to independent claims 1, 29, and 50.

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Application No.: 08/882,510
Filed: June 25, 1997
Group Art Unit: 2773

As all pending claims are believed to be allowable, the application is believed to be in condition for allowance. Favorable action is respectfully requested.

The Examiner is encouraged to telephone the undersigned attorney to discuss any matter which would expedite allowance of the present application.

Respectfully submitted,

JAMES S. ROSEN ET AL.

By: David A. Dagg
David A. Dagg
Registration No. 37,809
Attorney for Applicant(s)

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
Ten Post Office Square
Boston, MA 02109
Telephone: (617) 542-2290
Telecopier: (617) 451-0313

Dated: JUNE 8, 1999

DAD
Enclosure
205926

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

FAX RECEIVED
JUN 17 1999
GROUP 2700

OFFICIAL

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
TEN POST OFFICE SQUARE
BOSTON, MASSACHUSETTS 02109

INTELLECTUAL PROPERTY LAW
PATENTS, *TRADEMARKS* AND COPYRIGHTS

TELEPHONE
(617) 542-2290
FACSIMILE
(617) 451-0313

JOSEPH WEINGARTEN (1919-1984)
STANLEY M. SCHURGIN
CHARLES L. GAGNEBIN III
PAUL J. HAYES
VICTOR B. LEBOVICI
DEAN GRAHAM BOSTOCK
EUGENE A. FEHER
BEVERLY E. HJORTH
HOLLIDAY C. HEINE, PH.D.

GORDON R. MORIARTY
HOLMES W. ANDERSON
DAVID W. ROUILLE
RUSSELL W. BINNS, JR.
CHRISTOPHER J. LUTZ
JAMES F. THOMPSON
GWENDOLYN H. YIP
DAVID A. DAGG
PAUL J. CRONIN

FACSIMILE COVER SHEET

DATE: June 17, 1999

TO: Examiner C. Jackson — Fax No. Dialed: 703-308-6606

FROM: Victor B. Lebovici — No. of pages transmitted (including this page): 3

Our File: COMET-001XX — Time:

Your Ref: 08/882,580 — Sent by: Rita Chalifoux

A confirmation copy of this transmission will not be mailed unless the following is checked: [ ]

MESSAGE

PLEASE DELIVER DIRECTLY TO:

EXAMINER Chadwick Jackson, Tel. (703) 308-9572.

GROUP ART UNIT 2773

Enclosed please find a letter authorizing the Patent and Trademark Office to use Deposit Account No. 23-0804 for the CPA filed on June 8, 1999.

THIS MESSAGE MAY CONTAIN CONFIDENTIAL OR PRIVILEGED INFORMATION INTENDED ONLY FOR THE PERSON(S) IDENTIFIED ABOVE. IF IT HAS BEEN RECEIVED AT ANY OTHER PLACE OR HAS NOT BEEN CLEARLY RECEIVED, PLEASE CALL THE ABOVE IDENTIFIED SENDING PARTY COLLECT FOR INSTRUCTIONS. DO NOT SHOW OR DISTRIBUTE THIS MESSAGE TO ANYONE OTHER THAN THE INTENDED RECIPIENT(S). THANK YOU.

VBL/rac/Enc./

FAX RECEIVED
JUN 17 1999
GROUP 2700

OFFICIAL
PATENT

#17
6-18-99
BHilliard

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application	:	James S. Rosen et al.
Application No.	:	08/882,580
Filed	:	June 25, 1997
For	:	SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE
Examiner	:	C. Jackson
Attorney's Docket	:	COMET-001XX

Group Art Unit: 2773

* *

I hereby certify that this correspondence is being sent via Facsimile to Examiner C. Jackson in the United States Patent and Trademark Office at (703) 308-6606, on June 17, 1999.

By: Victor B. Lebovici
Victor B. Lebovici
Registration No. 30,864
Attorney for Applicant(s)

* *

L E T T E R

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

We have been informed by Examiner Chad Jackson that no fees have been charged to our deposit account with regard to the Continued Prosecution Application filed in the above identified case on June 8, 1999. Authorization is hereby granted to charge the filing fee of $1,109.00, as well as the one month extension

06/18/1999 BHILLIAR 00000001 08882580
01 FC:203 495.00 CH
02 FC:231 380.00 CH
03 FC:202 234.00 CH
04 FC:215 55.00 CH

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Application No.:08/882,580
Filed: June 25, 1997
Group Art Unit: 2773

fee of $55.00 required by that filing to Deposit Account No. 23-0804.

Respectfully submitted,

JAMES S. ROSEN ET AL.

By: _____
Victor B. Lebovici
Registration No. 30,864
Attorney for Applicant(s)

WEINGARTEN, SCHURGIN, GAGNEBIN
& HAYES LLP
Ten Post Office Square
Boston, MA 02109
Telephone: (617) 542-2290
Telecopier: (617) 451-0313

Dated: 6/17/99

VBL/rec/enc.
206566

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

BEST COPY

Transaction History Date 1999-06-21
Date information retrieved from USPTO Patent Application Information Retrieval (PAIR) system records at www.uspto.gov

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

09/882,580

SERIAL NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NO

EXAMINER	

ART UNIT	PAPER NUMBER
	18

DATE MAILED:

NOTICE OF ALLOWABILITY

PART I.

1. [X] This communication is responsive to 6/8/99 Filing
2. [X] All the claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice Of Allowance And Issue Fee Due or other appropriate communication will be sent in due course.
3. [X] The allowed claims are 1-71. and 133-138.
4. [ ] The drawings filed on _____ are acceptable
5. [ ] Acknowledgment is made of the claim for priority under 35 U.S.C. 119. The certified copy has [ ] been received [ ] not been received. [ ] been filed in parent application Serial No _____, filed on _____.
6. [ ] Note the attached Examiner's Amendment.
7. [ ] Note the attached Examiner Interview Summary Record, PTOL-413.
8. [X] Note the attached Examiner's Statement of Reasons for Allowance.
9. [X] Note the attached NOTICE OF REFERENCES CITED, PTO-892.
10. [ ] Note the attached INFORMATION DISCLOSURE CITATION, PTO-1449.

PART II.

A SHORTENED STATUTORY PERIOD FOR RESPONSE to comply with the requirements noted below is set to EXPIRE THREE MONTHS FROM THE "DATE MAILED" indicated on this form. Failure to timely comply will result in the ABANDONMENT of this application. Extensions of time may be obtained under the provisions of 37 CFR 1.136(a).

1. [ ] Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL APPLICATION, PTO-152, which discloses that the oath or declaration is deficient. A SUBSTITUTE OATH OR DECLARATION IS REQUIRED.
2. [X] APPLICANT MUST MAKE THE DRAWING CHANGES INDICATED BELOW IN THE MANNER SET FORTH ON THE REVERSE SIDE OF THIS PAPER.
   a. [X] Drawing informalities are indicated on the NOTICE RE PATENT DRAWINGS, PTO-948, attached hereto or to Paper No. 5 CORRECTION IS REQUIRED
   b. [ ] The proposed drawing correction filed on _____ has been approved by the examiner. CORRECTION IS REQUIRED.
   c. [ ] Approved drawing corrections are described by the examiner in the attached EXAMINER'S AMENDMENT CORRECTION IS REQUIRED.
   d. [X] Formal drawings are now REQUIRED.

Any response to this letter should include in the upper right hand corner, the following information from the NOTICE OF ALLOWANCE AND ISSUE FEE DUE: ISSUE BATCH NUMBER, DATE OF THE NOTICE OF ALLOWANCE, AND SERIAL NUMBER.

Attachments:

- _ Examiner's Amendment
- _ Examiner Interview Summary Record, PTOL-413
- X Reasons for Allowance
- X Notice of References Cited PTO-892
- _ Information Disclosure Citation, PTO-1449
- _ Notice of Informal Application, PTO-152
- _ Notice re Patent Drawings, PTO-948
- _ Listing of Bonded Draftsmen
- _ Other

RAYMOND J. BAYERL
PRIMARY EXAMINER
ART UNIT 2773

PTOL-37 (REV. 4-89) *

ATTACHMENT TO NOTICE OF ALLOWABILITY

1. This action is responsive to communications: Amendment, filed on 6/8/99.

2. The present title of the invention is "SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE" having claims 1-71 and 133-138 as amended. In the Amendment, filed on 6/8/99, claims 1, 3, 29, 31, 32, 37, 50-54, 56, 57 and 59 were amended, claims 2, 30 and 72-132 were canceled and claims 133-138 were added.

Reasons for Allowance

3. The following is an Examiner's statement of reasons for allowance:

The Examiner has carefully considered each of the independent claims 1, 29, 50 and 133-138 drawn to a "system" and "method" for "modifying a cursor image to a specific image on the display of a user's terminal." Each of the independent claims recite the limitation of "the specific image includes content corresponding to at least a portion of the information to be displayed on the display of a user's terminal." This particular type of system for modifying a cursor image to a specific image on the display of a user's terminal is neither taught nor suggested by the prior art made of record.

Lection et al. (US Patent # 5,801,698) teaches providing dynamic busy cursor information corresponding to busy cursor, where busy cursor reformatting program code performs the reformatting of a busy cursor with dynamic information and displays the reformatted busy cursor on the user's terminal. However, the dynamic information does not include content corresponding to at least a portion of information displayed on the display of the user's terminal. In contrast, the reformatting of the busy cursor with dynamic information is unrelated to the application which is providing the information displayed on the display of a user's terminal, and thus the applied reference teaches away from the Applicants invention as claimed in claims 1, 29 50 and 133-138.

Schneider (US Patent # 5,710,897) teaches a pointer graphic corresponding to a pointer graphic to be used by the operating system (see Schneider ('897), col. 5, lines 65- col. 6, lines 1-10; col. 7, lines 28-31; col. 8, lines 10-12, and 55-60); pointer graphic file set, the pointer graphic file set is operable to modify a displayed pointer graphic based on the occurrence of an event (see Schneider ('897), col. 5, line 55-col. 6, line 11); a personal computer designed to give users independent computing power, where information, such as applications, documents, files, etc., are transmitted to a remote user. See Schneider ('897), col. 3, lines 58-68).

Schneider ('897) suggests transmitting specified content information that includes instructions indicating the location of a cursor image file because the system of Schneider ('897) is operable on a networked environment, where an application that is executed on a network

includes instructions to be used by the operating system regarding the handling of events. This instruction indicates the location of cursor image data, where the operating system would locate the appropriate pointer graphics file and cause the user terminal to display the appropriate cursor on the user's terminal when required. However, Schneider ('897) neither suggests or teaches that a pointer graphic to be used by the operating system is a specific image that includes content corresponding to a portion of information to be displayed on the display of a user's terminal. Accordingly, the Applicants invention as claimed in claims 1, 29, 50 and 133-138 is not taught or suggested by the prior art.

Conclusion

4. Response to this action should be mailed to: Commissioner of Patents and Trademarks, Washington, D.C. 20231. If applicant desires to fax a response, (703) 308-9051 may be used for formal communications or (703) 305-9724 for informal or draft communications. Please label "PROPOSED" OR "DRAFT" for informal facsimile communications. For after final responses, please label "AFTER FINAL" or "EXPEDITED PROCEDURE' on the document. Hand delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington VA., Sixth Floor (Receptionist).

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chadwick A. Jackson, whose telephone number is (703) 308-9572. The examiner can normally be reached Mon-Thu from 7:30 a.m. - 6:00 p.m. ET. If attempts to reach

the examiner by telephone are unsuccessful, the examiner's supervisor, Matthew Kim, can be reached at (703) 305-3821.

6. Any inquiry of a general nature or relating to the status of this application or proceedings should be directed to the group receptionist whose telephone number is (703) 305-3900.

Chadwick A. Jackson

June 15, 1999

RAYMOND J. BAYERL
PRIMARY EXAMINER
ART UNIT 2773

Notice of References Cited

Application No.	Applicant(s)		
08/882,580	Rosen et al.		
Examiner	Group Art Unit	Page 1 of 1	
C. Jackson	2773		

U.S. PATENT DOCUMENTS

*		DOCUMENT NO.	DATE	NAME	CLASS	SUBCLASS
*	A	5,801,698	Sep. 1, 1998	Lection et al.	345	347
	B					
	C					
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

*		DOCUMENT NO.	DATE	COUNTRY	NAME	CLASS	SUBCLASS
	N						
	O						
	P						
	Q						
	R						
	S						
	T						

NON-PATENT DOCUMENTS

*		DOCUMENT (Including Author, Title, Source, and Pertinent Pages)	DATE
	U		
	V		
	W		
	X		

* A copy of this reference is not being furnished with this Office action
(See Manual of Patent Examining Procedure, Section 707.05(a).)

18

UNITED STATES DEPA. .'ENT OF COMMERCE
Patent and Trademark Office

NOTICE OF ALLOWANCE AND ISSUE FEE DUE

LM61/0621
[illegible]
[illegible]
[illegible]

APPLICATION NO.	FILING DATE	TOTAL CLAIMS	EXAMINER AND GROUP ART UNIT	DATE MAILED
[illegible]	[illegible]	075	JACKSON, C 2773	06/21/9[illegible]
First Named Applicant ROSEN,		35 USC 154(b) term ext. =	0 days.	

TITLE OF INVENTION: [illegible]SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE

ATTY'S DOCKET NO.	CLASS-SUBCLASS	BATCH NO.	APPLN. TYPE	SMALL ENTITY	FEE DUE	DATE DUE
[illegible] 758-002	345-349.000	U73	UTILITY	YES	$605.00	09/21/9[illegible]

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED.

THE ISSUE FEE MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED.

HOW TO RESPOND TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.
If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is changed, pay twice the amount of the FEE DUE shown above and notify the Patent and Trademark Office of the change in status, or
B. If the status is the same, pay the FEE DUE shown above.

If the SMALL ENTITY is shown as NO:

A. Pay FEE DUE shown above, or

B. File verified statement of Small Entity Status before, or with, payment of 1/2 the FEE DUE shown above.

II. Part B-Issue Fee Transmittal should be completed and returned to the Patent and Trademark Office (PTO) with your ISSUE FEE. Even if the ISSUE FEE has already been paid by charge to deposit account, Part B Issue Fee Transmittal should be completed and returned. If you are charging the ISSUE FEE to your deposit account, section "4b" of Part B-Issue Fee Transmittal should be completed and an extra copy of the form should be submitted.

III. All communications regarding this application must give application number and batch number.
Please direct all communications prior to issuance to Box ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PATENT AND TRADEMARK OFFICE COPY

95 (REV. 10-96) Approved for use through 06/30/99. (0651-0033)

*U.S. GPO 1998-437-639/60023

BEST COPY

Application No.: 08/882,580
Filed: June 25, 1997
Group Art Unit: 2773

#B

09/97 FORM 9

WEINGARTEN, SCHURGIN, GAGNEBIN & HAYES LLP
Ten Post Office Square
Boston, Massachusetts 02109
Telephone: (617) 542-2290
Telecopier: (617) 451-0313

OIPE JC84 JUL 22 1999 PATENT & TRADEMARK OFFICE

Date: July 20, 1999

ASSISTANT COMMISSIONER FOR PATENTS
Washington, D.C. 20231

Attorney
Docket No.: COMET-001XX

RECEIVED
JUL 29 1999
Publishing Division
10

Sir:

AUG 02 1999

In re application of: James S. Rosen et al.

Entitled: SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE

Transmitted herewith is an **amendment** in the above-identified application. The following checked items are applicable:

[ ] A Petition for Extension of Time for ___ month is hereby made, under §1.136(a); a check in the amount of _____ is enclosed per §1.17.

[X] **In the event a Petition for Extension of Time is required by this paper and not otherwise provided, such Petition is hereby made and authorization is provided herewith to charge Deposit Account No. 23-0804 for the cost of such extension.**

CLAIMS AFTER AMENDMENT:	MINUS PRIOR PAID CLAIMS:	EQUALS PRESENT EXTRA CLAIMS:	RATE:	ADDITIONAL FEE:
Independent	9 - 9	=	x $78.00 =	$ 0.00
Total	197 - 75	= 122	x $18.00 =	$ 2196.00
[ ] Multiple Dependent Claims (1st presentation)			+ $260.00 =	$ 0.00
[redacted]				$ 2196.00
Small Entity filing, divide by 2. Previously submitted or verified statement must be attached (per §1.9, §1.27, §1.28)				1098.00
[redacted]				$ 1098.00

[ ] No additional fee. [X] The fee has been calculated above; a check in the amount of $1098.00 is enclosed.

[X] **The Commissioner is hereby authorized to charge payment of any additional filing fees under §1.16 associated with this communication or credit any overpayment to Deposit Account No. 23-0804.**

I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on 7/20/99.

Attorney of Record: Victor B. Lebovici
Registration No.: 30,864

SUBMIT IN TRIPLICATE

DAD/kdm/208411



8-9-99
B. Hilliard

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application	:	James S. Rosen et al.
Application No.	:	08/882,580
Filed	:	June 25, 1997
For	:	SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR
Examiner	:	C. Jackson
Attorney's Docket	:	COMET-001XX

Group Art Unit: 2773

* *

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on July 20, 1999.

By: David A. Dagg
David A. Dagg
Registration No. 37,809
Attorney for Applicant(s)

* *

AMENDMENT UNDER 37 C.F.R. 1.312

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Pursuant to 37 C.F.R. 1.312, Applicant hereby respectfully requests entry of this Amendment after the Notice of Allowance dated June 21, 1999. This Amendment adds dependent claims to several allowed independent claims. As further described in the remarks below, all claims added herein correspond to currently allowed dependent claims which depend from allowed independent claims.

07/28/1999 TLUU11 00000044 08882580 1098.00 OP
01 FC:203

In the Claims:

Please add the following dependent claims:

139. (New) The server system in accordance with claim 133, wherein said specific image relates to at least a portion of said information to be displayed on said display of said remote user's terminal.

140. (New) The server system in accordance with claim 139, wherein said specific image comprises advertising material related to at least a portion of said information to be displayed on said display of said user's terminal.

141. (New) The server system in accordance with claim 140, wherein said advertising material further comprises a brand logo.

142. (New) The server system in accordance with claim 140, wherein said advertising material further comprises a corporate mascot.

143. (New) The server system in accordance with claim 140, wherein said advertising material further comprises images of a good or a service corresponding to said information to be displayed on said display of said user's terminal.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

144. (New) The server system in accordance with claim 140, wherein said advertising material further comprises messages relating to said information to be displayed on said display of said user's terminal.

145. (New) The server system in accordance with claim 139, wherein said specific image has a shape and appearance corresponding to said information to be displayed on said display of said user's terminal.

146. (New) The server system in accordance with claim 133, wherein at least a portion of said cursor image data and said cursor display code are disposed locally to said first server computer.

147. (New) The server system in accordance with claim 133, wherein at least a portion of said cursor image data and said cursor display code are disposed within a second server computer located remotely to said first server computer.

148. (New) The server system in accordance with claim 133, wherein at least a portion of said cursor image data and said cursor display code are disposed locally to said user's terminal.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

149. (New) The server system in accordance with claim 147 wherein said first server computer in response to a request from said user terminal transmits information stored in said second server computer to said user terminal.

150. (New) The server system in accordance with claim 133, wherein said specified content information is transmitted in the form of HTML files that define a web page.

151. (New) The server system in accordance with claim 150, wherein said cursor image data includes at least in part an advertisement for goods or services contained in said web page.

152. (New) The server system in accordance with claim 133, wherein said user terminal includes a browser application responsive to said cursor display instruction, said browser application executing said cursor display code using parameters defined in said cursor display instruction.

153. (New) The server system in accordance with claim 133, said cursor display instruction further comprising an image identifier indicating said cursor image data corresponding to said specific image.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

154. (New) The server system in accordance with claim 133 wherein said first server computer transmits said cursor image data in response to a request received from said remote user terminal indicating that a copy of said cursor image data is not stored in said remote user terminal.

155. (New) The server system in accordance with claim 133 wherein said cursor display instruction further comprises an image identifier that corresponds to a graphic animation sequence.

156. (New) The server system in accordance with claim 155, wherein said cursor display instruction further comprises instructions operable to modify said specific image to display said graphic animation sequence.

157. (New) The server system in accordance with claim 133, wherein said cursor display instruction further comprises an audio identifier that corresponds to an audio information sequence.

158. (New) The server system in accordance with claim 157, wherein said cursor display instruction further comprises instructions operable to play an audio clip corresponding to said audio information sequence.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

159. (New) The server system in accordance with claim 133, wherein said cursor display instruction further comprises information that controls a duration of time said specific image is displayed on said display of said remote user's terminal.

160. (New) The server system in accordance with claim 1, wherein said specified content information is transmitted in the form of one or more hypertext objects.

161. (New) The server system in accordance with claim 133, wherein said specified content information includes instructions executable by a virtual machine on said user terminal.

162. (New) The server system in accordance with claim 133, wherein said specified content information includes HTML tags recognized by said cursor display code.

163. (New) The server system in accordance with claim 133, wherein said cursor display code generates usage data for calculating usage statistics of said specific image.

164. (New) The server system in accordance with claim 133, wherein said cursor display instruction transmitted to said remote user

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0311

terminal initiates communication with a plurality of server systems for obtaining additional cursor image data.

165. (New) The server system in accordance with claim 134, wherein said specific image relates to at least a portion of said information to be displayed on said display of said remote user's terminal.

166. (New) The server system in accordance with claim 165, wherein said specific image comprises advertising material related to at least a portion of said information to be displayed on said display of said user's terminal.

167. (New) The server system in accordance with claim 166, wherein said advertising material further comprises a brand logo.

168. (New) The server system in accordance with claim 166, wherein said advertising material further comprises a corporate mascot.

169. (New) The server system in accordance with claim 166, wherein said advertising material further comprises images of a good or a service corresponding to said information to be displayed on said display of said user's terminal.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

170. (New) The server system in accordance with claim 166, wherein said advertising material further comprises messages relating to said information to be displayed on said display of said user's terminal.

171. (New) The server system in accordance with claim 165, wherein said specific image has a shape and appearance corresponding to said information to be displayed on said display of said user's terminal.

172. (New) The server system in accordance with claim 134, wherein at least a portion of said cursor image data and said cursor display code are disposed locally to said first server computer.

173. (New) The server system in accordance with claim 134, wherein at least a portion of said cursor image data and said cursor display code are disposed within a second server computer located remotely to said first server computer.

174. (New) The server system in accordance with claim 134, wherein at least a portion of said cursor image data and said cursor display code are disposed locally to said user's terminal.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542 2290
FAX (617) 451 0313

175. (New) The server system in accordance with claim 173 wherein said first server computer in response to a request from said user terminal transmits information stored in said second server computer to said user terminal.

176. (New) The server system in accordance with claim 134, wherein said specified content information is transmitted in the form of HTML files that define a web page.

177. (New) The server system in accordance with claim 176, wherein said cursor image data includes at least in part an advertisement for goods or services contained in said web page.

178. (New) The server system in accordance with claim 134, wherein said user terminal includes a browser application responsive to said cursor display instruction, said browser application executing said cursor display code using parameters defined in said cursor display instruction.

179. (New) The server system in accordance with claim 134, said cursor display instruction further comprising an image identifier indicating said cursor image data corresponding to said specific image.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

180. (New) The server system in accordance with claim 134 wherein said first server computer transmits said cursor image data in response to a request received from said remote user terminal indicating that a copy of said cursor image data is not stored in said remote user terminal.

181. (New) The server system in accordance with claim 134 wherein said cursor display instruction further comprises an image identifier that corresponds to a graphic animation sequence.

182. (New) The server system in accordance with claim 181, wherein said cursor display instruction further comprises instructions operable to modify said specific image to display said graphic animation sequence.

183. (New) The server system in accordance with claim 134, wherein said cursor display instruction further comprises an audio identifier that corresponds to an audio information sequence.

184. (New) The server system in accordance with claim 183, wherein said cursor display instruction further comprises instructions operable to play an audio clip corresponding to said audio information sequence.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

185. (New) The server system in accordance with claim 134, wherein said cursor display instruction further comprises information that controls a duration of time said specific image is displayed on said display of said remote user's terminal.

186. (New) The server system in accordance with claim 134, wherein said specified content information is transmitted in the form of one or more hypertext objects.

187. (New) The server system in accordance with claim 134, wherein said specified content information includes instructions executable by a virtual machine on said user terminal.

188. (New) The server system in accordance with claim 134, wherein said specified content information includes HTML tags recognized by said cursor display code.

189. (New) The server system in accordance with claim 134, wherein said cursor display code generates usage data for calculating usage statistics of said specific image.

190. (New) The server system in accordance with claim 134, wherein said cursor display instruction transmitted to said remote user

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451 0313

terminal initiates communication with a plurality of server systems for obtaining additional cursor image data.

191. (New) The method in accordance with claim 135, wherein said transforming further comprises executing said cursor display code so as to display said specific cursor image while at least a portion of said information to be displayed is displayed on said display of said user's terminal.

192. (New) The method in accordance with claim 135, wherein said displaying of said specific image further comprises displaying advertising material related to at least a portion of said information to be displayed.

193. (New) The method in accordance with claim 192, wherein said advertising material further comprises a brand logo.

194. (New) The method in accordance with claim 192, wherein said advertising material further comprises a corporate mascot.

195. (New) The method in accordance with claim 192, wherein said advertising material further comprises images of a good or a service corresponding to said information to be displayed.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

196. (New) The method in accordance with claim 192, wherein said advertising material further comprises messages relating to said information to be displayed.

197. (New) The method in accordance with claim 135, wherein said specific image has a shape and appearance relating to said information to be displayed.

198. (New) The method in accordance with claim 135, wherein said specified content information further comprises HTML files that define a web page.

199. (New) The method in accordance with claim 198, wherein said cursor image data corresponds to an advertisement for goods or services contained in said web page.

200. (New) The method in accordance with claim 135, wherein said transforming further comprises:

employing a browser application <u>including said cursor display code</u> responsive to said cursor display instruction; and

executing said cursor display code by employing parameters defined in said cursor display instruction.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

201. (New) The method in accordance with claim 135 wherein said specified content information comprises an image identifier that corresponds to the location of data representing said specific image.

202. (New) The method in accordance with claim 135, further comprising transmitting said cursor image data in response to a request received from said remote user terminal indicating that a copy of said cursor image data is not stored in said remote user terminal.

203. (New) The method in accordance with claim 135, wherein said cursor display instructions further comprises an image identifier that corresponds to a graphic animation sequence.

204. (New) The method in accordance with claim 203, further comprising modifying said remote user terminal's cursor to display said graphic animation sequence.

205. (New) The server system in accordance with claim 135, wherein said cursor display instruction further comprises an audio identifier that corresponds to an audio information sequence.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

206. (New) The server system in accordance with claim 205, further comprising playing an audio clip corresponding to said audio information sequence responsive to displaying said specific image.

207. (New) The method in accordance with claim 135 further comprising controlling a duration of time said specific image is displayed on said remote user's display.

208. (New) The method in accordance with claim 135, further comprising providing usage data by said cursor display code for calculating usage statistics of said specific image, responsive to said cursor display instruction.

209. (New) A computer storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform the method steps in claim 135.

210. (New) The method in accordance with claim 136, wherein said transforming further comprises executing said cursor display code so as to display said specific cursor image while at least a portion of said information to be displayed is displayed on said display of said user's terminal.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

211. (New) The method in accordance with claim 136, wherein said displaying of said specific image further comprises displaying advertising material related to at least a portion of said information to be displayed.

212. (New) The method in accordance with claim 211, wherein said advertising material further comprises a brand logo.

213. (New) The method in accordance with claim 211, wherein said advertising material further comprises a corporate mascot.

214. (New) The method in accordance with claim 211, wherein said advertising material further comprises images of a good or a service corresponding to said information to be displayed.

215. (New) The method in accordance with claim 211, wherein said advertising material further comprises messages relating to said information to be displayed.

216. (New) The method in accordance with claim 136, wherein said specific image has a shape and appearance relating to said information to be displayed.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

217. (New) The method in accordance with claim 136, wherein said specified content information further comprises HTML files that define a web page.

218. (New) The method in accordance with claim 217, wherein said cursor image data corresponds to an advertisement for goods or services contained in said web page.

219. (New) The method in accordance with claim 136, wherein said transforming further comprises:

employing a browser application including said cursor display code responsive to said cursor display instruction; and

executing said cursor display code by employing parameters defined in said cursor display instruction.

220. (New) The method in accordance with claim 136 wherein said specified content information comprises an image identifier that corresponds to the location of data representing said specific image.

221. (New) The method in accordance with claim 136, further comprising transmitting said cursor image data in response to a request received from said remote user terminal indicating that a

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542 2290
FAX (617) 451-0313

copy of said cursor image data is not stored in said remote user terminal.

222. (New) The method in accordance with claim 136, wherein said cursor display instructions further comprises an image identifier that corresponds to a graphic animation sequence.

223. (New) The method in accordance with claim 222, further comprising modifying said remote user terminal's cursor to display said graphic animation sequence.

224. (New) The server system in accordance with claim 136, wherein said cursor display instruction further comprises an audio identifier that corresponds to an audio information sequence.

225. (New) The server system in accordance with claim 224, further comprising playing an audio clip corresponding to said audio information sequence responsive to displaying said specific image.

226. (New) The method in accordance with claim 136 further comprising controlling a duration of time said specific image is displayed on said remote user's display.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

227. (New) The method in accordance with claim 136, further comprising providing usage data by said cursor display code for calculating usage statistics of said specific image, responsive to said cursor display instruction.

228. (New) A computer storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform the method steps in claim 136.

229. (New) The internet browser in accordance with claim 137 further comprising program code operable to retrieve said cursor image data from a prespecified server, when said cursor image data is not stored in said user terminal.

230. (New) The internet browser in accordance with claim 137 further comprising program code operable to retrieve said cursor display code from a prespecified server, when said cursor display code is not stored in said user terminal.

231. (New) The internet browser in accordance with claim 137 further comprising program code operable to determine whether cursor display instructions received by said user terminal were transmitted by an authorized server.

WRINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

232. (New) The internet browser in accordance with claim 137 further comprising program code operable to transmit statistical information to a prespecified server so as to provide information relating to the usage of said specific image.

233. (New) The internet browser in accordance with claim 137, wherein said specific image reverts back to its original shape and appearance after a prespecified duration.

234. (New) The internet browser in accordance with claim 232 further comprising program code operable to store said cursor display code in a memory located locally to said user terminal.

235. (New) The internet browser in accordance with claim 137 further comprising program code operable to provide audio clips corresponding to said display of said specific image.

236. (New) The internet browser in accordance with claim 235 wherein information relating to said audio clips is contained within said cursor display instruction.

237. (New) The internet browser in accordance with claim 137 further comprising program code operable to provide animated

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

images corresponding to said specific image.

238. (New) The internet browser in accordance with claim 237 wherein information relating to said animated images are contained within said cursor display instruction.

239. (New) The internet browser in accordance with claim 137, wherein said specific image corresponds to at least a portion of said information to be displayed on said display of said user's terminal.

240. (New) The internet browser in accordance with claim 137, wherein said specific image comprises advertising material related to at least a portion of said information to be displayed on said display of said user's terminal.

241. (New) The internet browser in accordance with claim 240, wherein said advertising material further comprises a brand logo.

242. (New) The internet browser in accordance with claim 240, wherein said advertising material further comprises a corporate mascot.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

243. (New) The internet browser in accordance with claim 240, wherein said advertising material further comprises images of a good or a service corresponding to said information to be displayed on said display of said user's terminal.

244. (New) The internet browser in accordance with claim 240, wherein said advertising material further comprises messages relating to said information to be displayed on said display of said user's terminal.

245. (New) The internet browser in accordance with claim 137, wherein said specific image has a shape and appearance related to said information to be displayed on said display of said display of said user's terminal.

246. (New) The internet browser in accordance with claim 137, wherein said specified content information is transmitted in the form of at least one HTML file that defines a web page.

247. (New) The internet browser in accordance with claim 137, wherein said specified content information is transmitted in the form of one or more hypertext objects.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542 2290
FAX (617) 451 0313

248. (New) The internet browser in accordance with claim 137, wherein said specified information content is executable at least in part by a virtual machine on said user terminal.

249. (New) The internet browser in accordance with claim 137, wherein said specified information content includes at least one instruction in an interpreted programming language.

250. (New) The internet browser in accordance with claim 138 further comprising program code operable to retrieve said cursor image data from a prespecified server, when said cursor image data is not stored in said user terminal.

251. (New) The internet browser in accordance with claim 138 further comprising program code operable to retrieve said cursor display code from a prespecified server, when said cursor display code is not stored in said user terminal.

252. (New) The internet browser in accordance with claim 138 further comprising program code operable to determine whether cursor display instructions received by said user terminal were transmitted by an authorized server.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542 2290
FAX (617) 451-0313

253. (New) The internet browser in accordance with claim 138 further comprising program code operable to transmit statistical information to a prespecified server so as to provide information relating to the usage of said specific image.

254. (New) The internet browser in accordance with claim 138, wherein said specific image reverts back to its original shape and appearance after a prespecified duration.

255. (New) The internet browser in accordance with claim 253 further comprising program code operable to store said cursor display code in a memory located locally to said user terminal.

256. (New) The internet browser in accordance with claim 138 further comprising program code operable to provide audio clips corresponding to said display of said specific image.

257. (New) The internet browser in accordance with claim 255 wherein information relating to said audio clips is contained within said cursor display instruction.

258. (New) The internet browser in accordance with claim 138 further comprising program code operable to provide animated

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

images corresponding to said specific image.

259. (New) The internet browser in accordance with claim 258 wherein information relating to said animated images are contained within said cursor display instruction.

260. (New) The internet browser in accordance with claim 138, wherein said specific image corresponds to at least a portion of said information to be displayed on said display of said user's terminal.

261. (New) The internet browser in accordance with claim 138, wherein said specific image comprises advertising material related to at least a portion of said information to be displayed on said display of said user's terminal.

262. (New) The internet browser in accordance with claim 261, wherein said advertising material further comprises a brand logo.

263. (New) The internet browser in accordance with claim 261, wherein said advertising material further comprises a corporate mascot.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

264. (New) The internet browser in accordance with claim 261, wherein said advertising material further comprises images of a good or a service corresponding to said information to be displayed on said display of said user's terminal.

265. (New) The internet browser in accordance with claim 261, wherein said advertising material further comprises messages relating to said information to be displayed on said display of said user's terminal.

266. (New) The internet browser in accordance with claim 138, wherein said specific image has a shape and appearance related to said information to be displayed on said display of said display of said user's terminal.

267. (New) The internet browser in accordance with claim 138, wherein said specified content information is transmitted in the form of at least one HTML file that defines a web page.

268. (New) The internet browser in accordance with claim 138, wherein said specified content information is transmitted in the form of one or more hypertext objects.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

269. (New) The internet browser in accordance with claim 138, wherein said specified information content is executable at least in part by a virtual machine on said user terminal.

270. (New) The internet browser in accordance with claim 138, wherein said specified information content includes at least one instruction in an interpreted programming language.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

REMARKS

Pursuant to 37 C.F.R. 1.312, Applicant hereby respectfully requests entry of this Amendment after the Notice of Allowance dated June 21, 1999. The dependent claims added by this Amendment all correspond to currently allowed dependent claims which depend from allowed independent claims, as follows:

NEW DEPENDENT CLAIMS	CORRESPONDING ALLOWED CLAIMS
139-164	3-28
165-190	3-28
191-209	31-49
210-228	31-49
229-249	51-71
250-270	51-71

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Application No.: 08/882,580
Filed: June 25, 1997
Group Art Unit: 2773

The Examiner is encouraged to telephone the undersigned Attorney with regard to any matter relating to this Amendment.

Respectfully submitted,

JAMES S. ROSEN ET AL.

By: David A. Dagg
David A. Dagg
Registration No. 37,809
Attorney for Applicant(s)

WEINGARTEN, SCHURGIN, GAGNEBIN
& HAYES LLP
Ten Post Office Square
Boston, MA 02109
Telephone: (617) 542-2290
Telecopier: (617) 451-0313

Dated: July 20, 1999

DAD
Enclosure
207864

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Transaction History Date 1999-08-25
Date information retrieved from USPTO Patent Application Information Retrieval (PAIR) system records at www.uspto.gov

BEST COPY

UNITED STATES DEPARTMENT OF COMMERCE
Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231

APPLICATION NUMBER	FILING DATE	FIRST NAMED APPLICANT	ATTORNEY DOCKET NO
08/882,580	06/25/97	ROSEN	658-002

LM61/0825
WEINGARTEN SCHURGIN GAGNEBIN & HAYES LLP
TEN POST OFFICE SQUARE
ROSTON MA 02109

EXAMINER	
JACKSON, C	
ART UNIT	**PAPER NUMBER**
2773	20

DATE MAILED: 08/25/99

Response to Rule 312 Communication

☐ The petition filed on _____ under 37 CFR 1.312(b) is granted. The paper has been forwarded to the examiner for consideration on the merits.

_____

_____
Director,
Patent Examining Group _____

☐ The amendment filed on 07/22/99 under 37 CFR 1.312 has been considered, and has been:

☐ entered.

☐ entered as directed to matters of form not affecting the scope of the invention (Order 3311).

☒ disapproved. See explanation below.

☐ entered in part. See explanation below.

The amendment require a substantial amount of additional work on the part of the office and effect the scope of the claims, which

Matthew M. Kim
Supervisory Patent Examiner
Technology Center 2700

FORM PTOL-271 (REV. 1-96)

OIPE JC86 SEP 20 1999 PATENT & TRADEMARK OFFICE

RECEIVED SEP 29 1999 Publishing Division

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Notice of Allowance dated: June 21, 1999
Issue Batch Number: U73

In re application	:	James S. Rosen, et al.
Application No	:	08/882,580
Filed	:	June 25, 1997
For	:	SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE
Examiner	:	C. Jackson
Attorney's Docket	:	COMET-001XX

Group Art Unit: 2773

* *

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Box Issue Fee, Assistant Commissioner for Patents, Washington, D.C. 20231 on SEPTEMBER 16, 1999.

By: David A. Dagg
David A. Dagg
Registration No. 37,809
Attorney for Applicants

* *

LETTER TO CHIEF DRAFTSMAN
RE: SUBMISSION OF FORMAL DRAWINGS

Box Issue Fee
Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Pursuant to the Notice dated June 21, 1999 submitted herewith in the above-identified application are formal drawings.

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

Application No.: 08/882,580
Filed: June 25, 1997
Group Art Unit: 2773

Kindly substitute these drawings for the ones currently on file as appropriate and charge any costs associated therewith to Patent and Trademark Office Account No. 23-0804. Triplicate copies of this transmittal letter are enclosed.

Respectfully submitted,

James S. Rosen, et al.

By: David A. Dagg
David A. Dagg
Registration No. 37,809
Attorney for Applicants

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
Ten Post Office Square
Boston, Massachusetts 02109
Telephone: (617) 542-2290
Telecopier: (617) 451-0313

Date: September 16, 1999

Enclosure
211543

WEINGARTEN, SCHURGIN,
GAGNEBIN & HAYES LLP
TEL (617) 542-2290
FAX (617) 451-0313

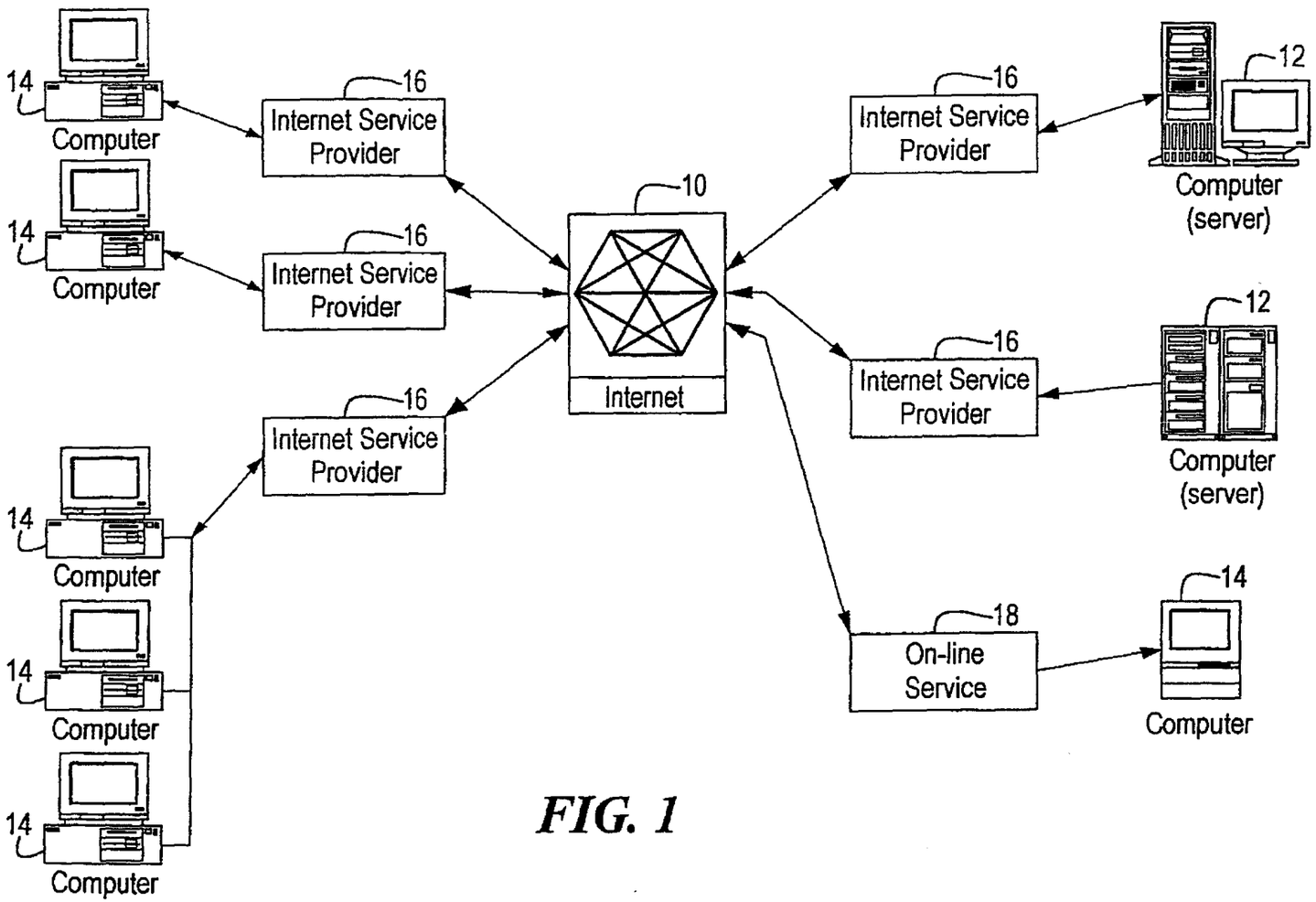

14
Computer
14
Computer
14
Computer
14
Computer
14
Computer
16
Internet Service Provider
16
Internet Service Provider
16
Internet Service Provider
10
Internet
16
Internet Service Provider
16
Internet Service Provider
18
On-line Service
12
Computer (server)
12
Computer (server)
14
Computer

FIG. 1

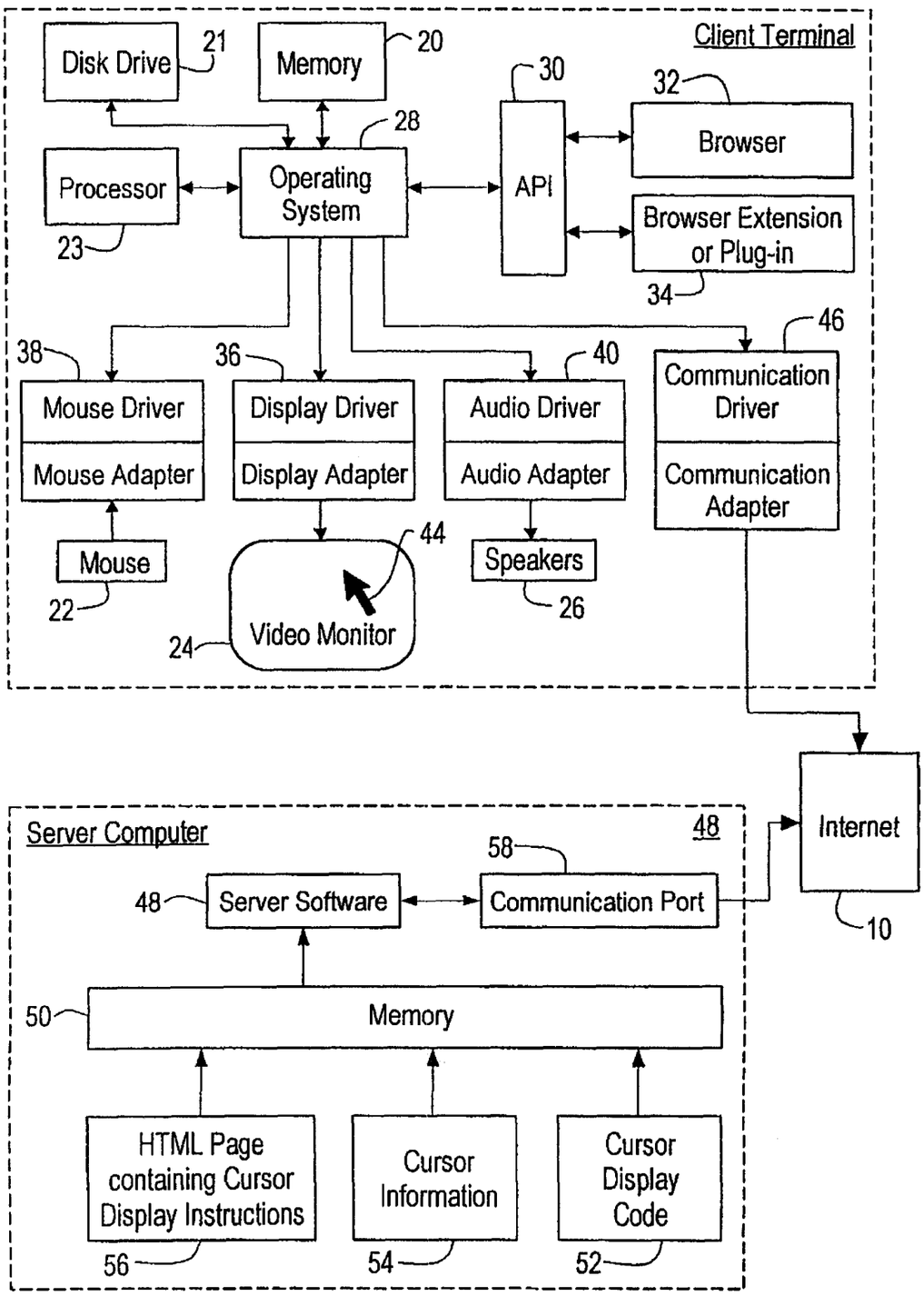

Client Terminal
Disk Drive
21
Memory
20
30
32
Browser
28
Processor
Operating System
API
Browser Extension or Plug-in
23
34
46
38
36
40
Mouse Driver
Mouse Adapter
Display Driver
Display Adapter
Audio Driver
Audio Adapter
Communication Driver
Communication Adapter
44
Mouse
Speakers
22
26
24
Video Monitor
Internet
10
Server Computer
48
58
48
Server Software
Communication Port
50
Memory
HTML Page containing Cursor Display Instructions
Cursor Information
Cursor Display Code
56
54
52

FIG. 2

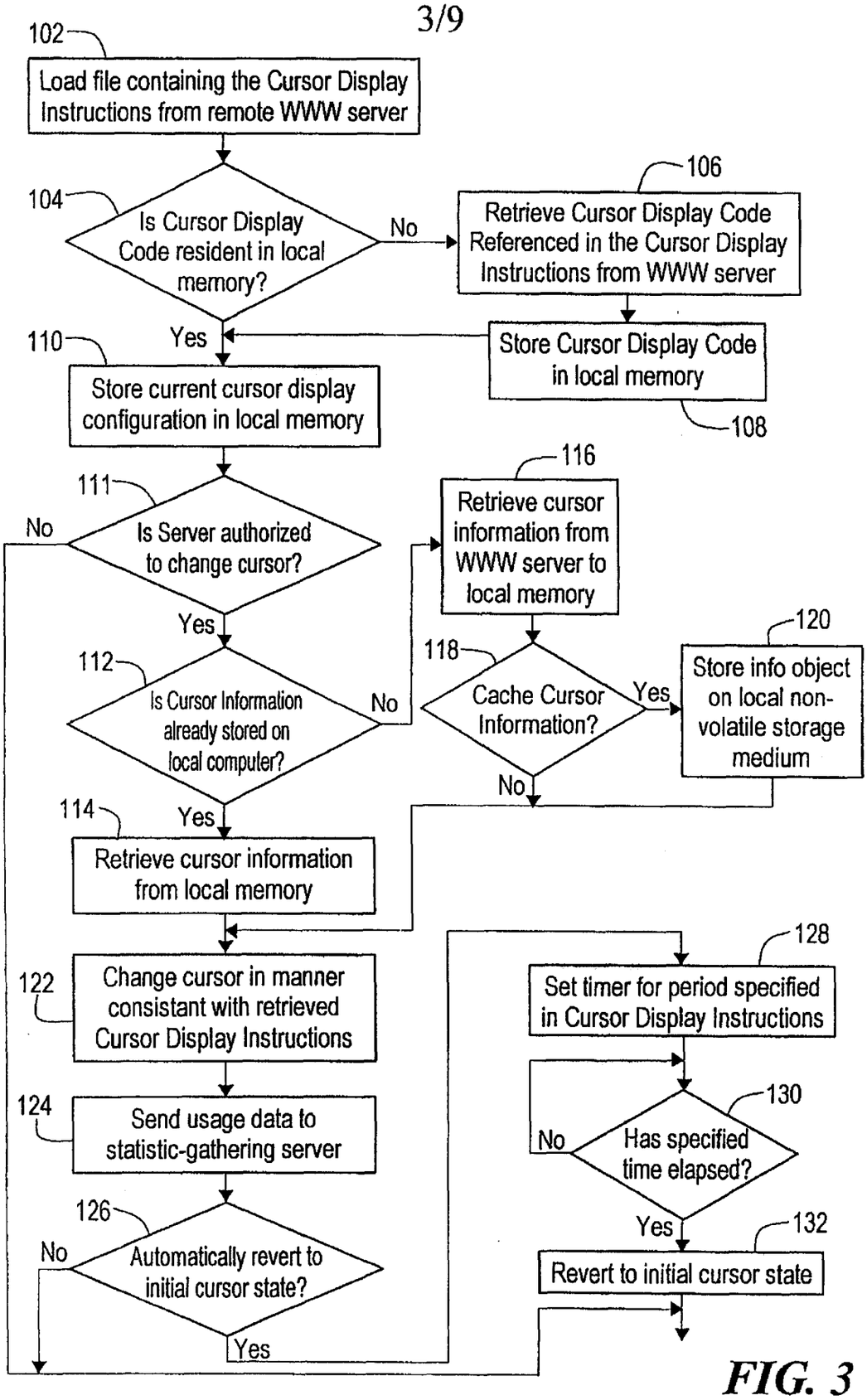

102
Load file containing the Cursor Display Instructions from remote WWW server
104
Is Cursor Display Code resident in local memory?
No
106
Retrieve Cursor Display Code Referenced in the Cursor Display Instructions from WWW server
Yes
Store Cursor Display Code in local memory
108
110
Store current cursor display configuration in local memory
111
Is Server authorized to change cursor?
No
Yes
112
Is Cursor Information already stored on local computer?
No
Yes
114
Retrieve cursor information from local memory
116
Retrieve cursor information from WWW server to local memory
118
Cache Cursor Information?
Yes
120
Store info object on local non-volatile storage medium
No
122
Change cursor in manner consistant with retrieved Cursor Display Instructions
124
Send usage data to statistic-gathering server
126
Automatically revert to initial cursor state?
No
Yes
128
Set timer for period specified in Cursor Display Instructions
130
Has specified time elapsed?
No
Yes
132
Revert to initial cursor state

FIG. 3

```
<OBJECT
        202.ID=cc1
        203.TYPE="application/x-oleobject"
        204.CLASSID="clsid:CB005660-D0C7-11cf-B7F6-00AA00A3F278"
        205.CODEBASE="http://cometsystems.com/controls/cc.cab#ver=4,70,
           0,1122"
        206.<PARAM NAME="CursorType" VALUE="1"
        207.<PARAM NAME="CursorImage"
           VALUE="http://cometsystems.com/library/images/acme.cur">
        208.<PARAM NAME="Counter" VALUE="http://
           cometsystems.com/accounting">
        209.<PARAM NAME="DisplayDuration" VALUE="5">
        210.<PARAM NAME="CacheCursor" VALUE="1">
        211.<PARAM NAME="ServerSignature" VALUE="54F5254A23BD988AB54">
        212.<PARAM NAME="DormantDelay" VALUE="600">
        213.<PARAM NAME="CursorTrajectoryMap" VALUE="http://
           cometsystems.com/maps/trajectory">
        214.<PARAM NAME="CursorPositionMap" VALUE="http://
           cometsystems.com/maps/position">
        215.<PARAM NAME="CursorVelocityMap"
           VALUE="http://cometsystems.com/maps/velocity">
        216.<PARAM NAME="CursorPositionMap" VALUE="http://
           cometsystems.com/maps/velocity">
        217.<PARAM NAME="CursorButtonMap" VALUE="http://
           cometsystems.com/maps/buttonstate">
        218.<PARAM NAME="ContentType" VALUE="5">
        219.<PARAM NAME="PriorityLevel" VALUE="1">
        220.<PARAM NAME="StreamBufferSize" VALUE="0">
        221.<PARAM NAME="SatelliteImage"
           VALUE="http://cometsystems.com/library/images/acmesat.bmp">
        222.<PARAM NAME="SatelliteXDisplacement" VALUE="-50">
        223.<PARAM NAME="SatelliteYDisplacement" VALUE="50">
        224.<PARAM NAME="ExtraDisplayParameters"
           VALUE="http://cometsystems.com/library/params/acme.prm">
</OBJECT>
```

FIG. 4

```
<script language="VBScript">
<!--\1`;
302.Sub window_onLoad()
303.    ccl.RememberCurrentCursor()
304.    ccl.SetNormalCursor("http://cometsystems.com/library/
           images/acme.cur")
305.end sub

306.Sub window_onUnload()
307.    ccl.Reset()
308.end sub
-->
</script>
```

FIG. 5

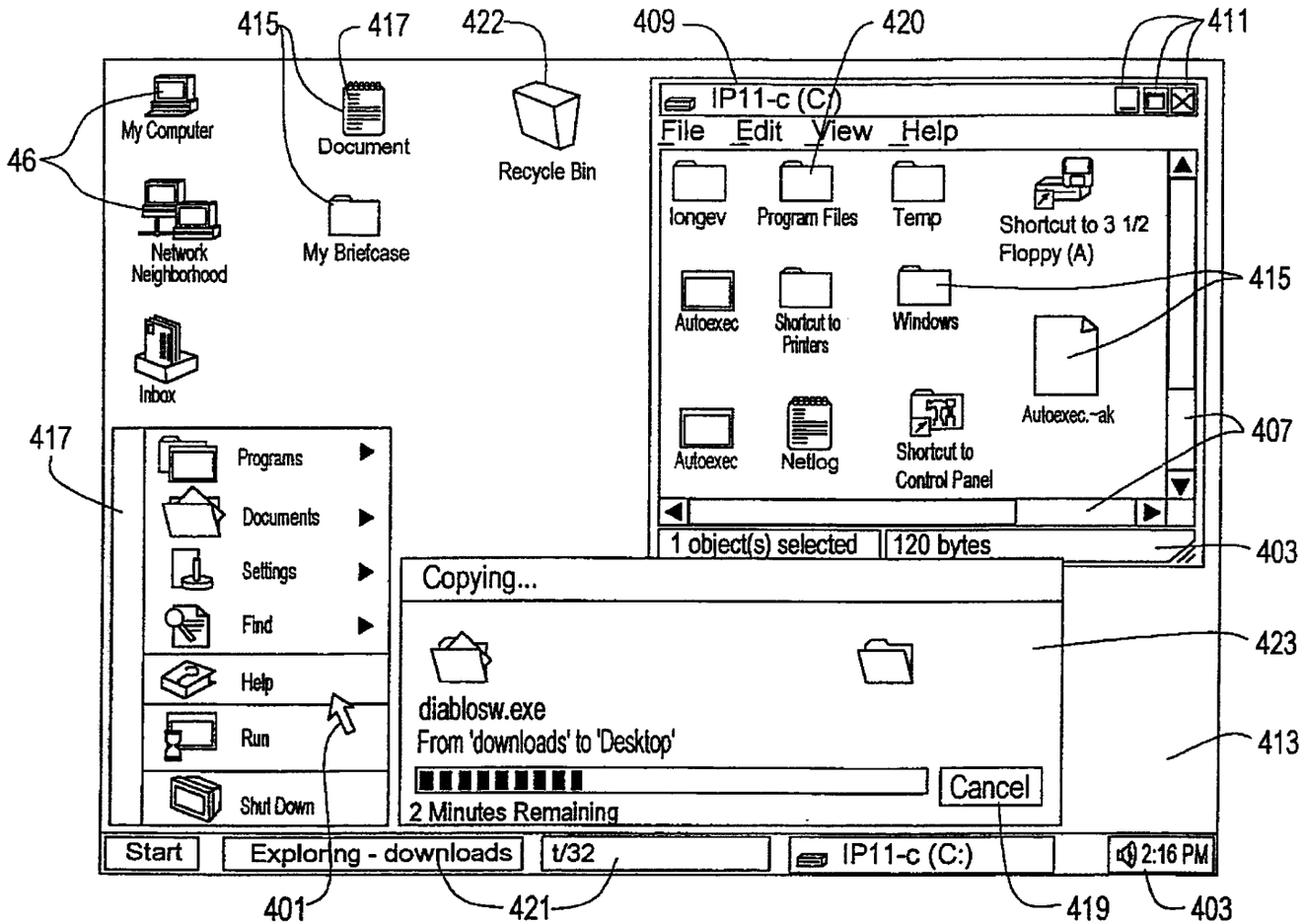

415
417
422
409
420
411
My Computer
Document
Recycle Bin
46
Network Neighborhood
My Briefcase
Inbox
IP11-c (C:)
File Edit View Help
longev
Program Files
Temp
Shortcut to 3 1/2 Floppy (A)
Autoexec
Shortcut to Printers
Windows
Autoexec.~ak
Autoexec
Netlog
Shortcut to Control Panel
1 object(s) selected
120 bytes
403
407
417
Programs
Documents
Settings
Find
Help
Run
Shut Down
Copying...
423
diablosw.exe
From 'downloads' to 'Desktop'
413
2 Minutes Remaining
Cancel
Start
Exploring - downloads
t/32
IP11-c (C:)
2:16 PM
401
421
419
403

FIG. 6

6/9

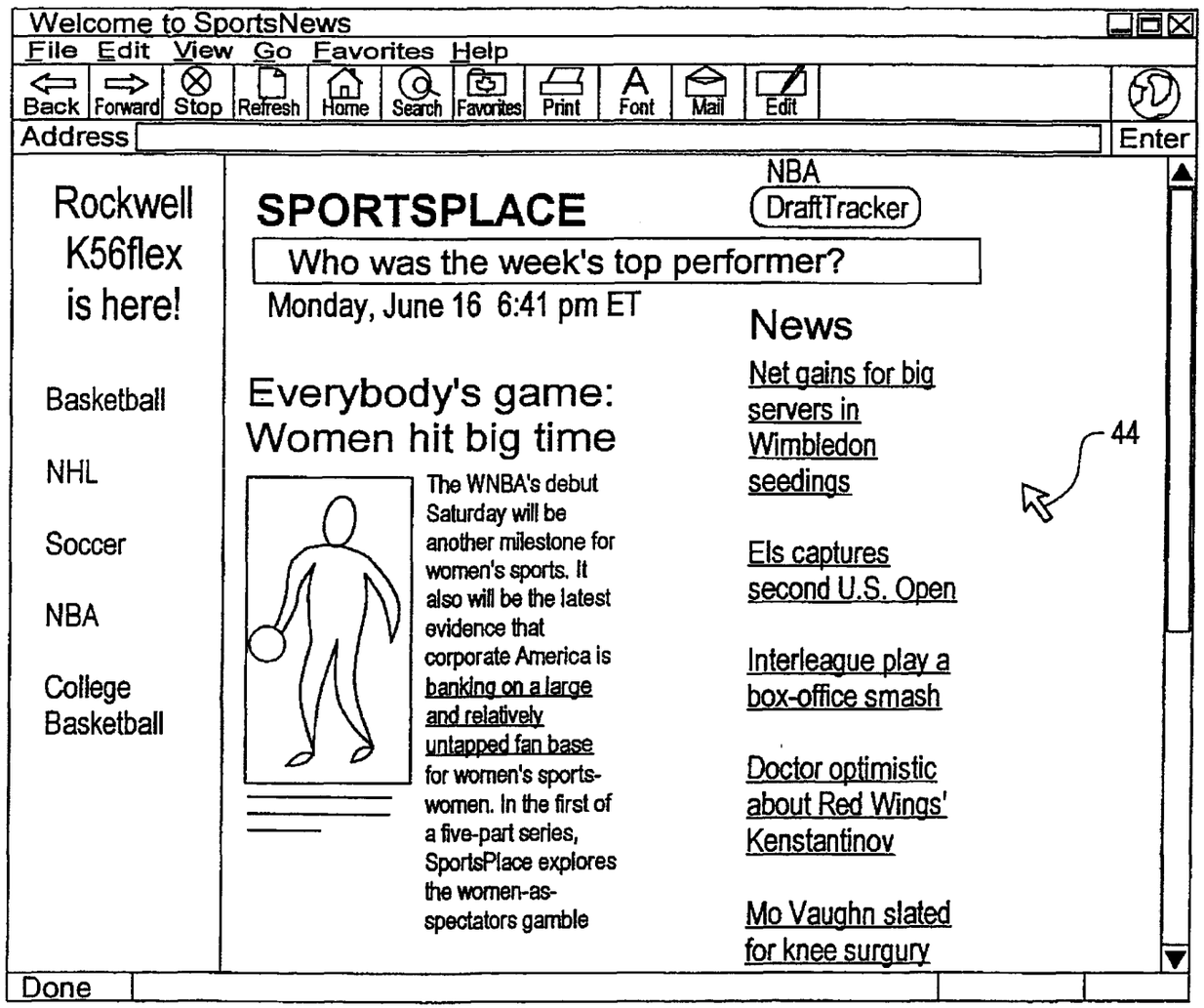

Welcome to SportsNews
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Font Mail Edit
Address
Enter
60
Rockwell K56flex is here!
Basketball
NHL
Soccer
NBA
College Basketball
SPORTSPLACE
NBA DraftTracker
Who was the week's top performer?
Monday, June 16 6:41 pm ET
Everybody's game: Women hit big time
The WNBA's debut Saturday will be another milestone for women's sports. It also will be the latest evidence that corporate America is banking on a large and relatively untapped fan base for women's sports-women. In the first of a five-part series, SportsPlace explores the women-as-spectators gamble
News
Net gains for big servers in Wimbledon seedings
44
Els captures second U.S. Open
Interleague play a box-office smash
Doctor optimistic about Red Wings' Kenstantinov
Mo Vaughn slated for knee surgury
Done

FIG. 7

7/9

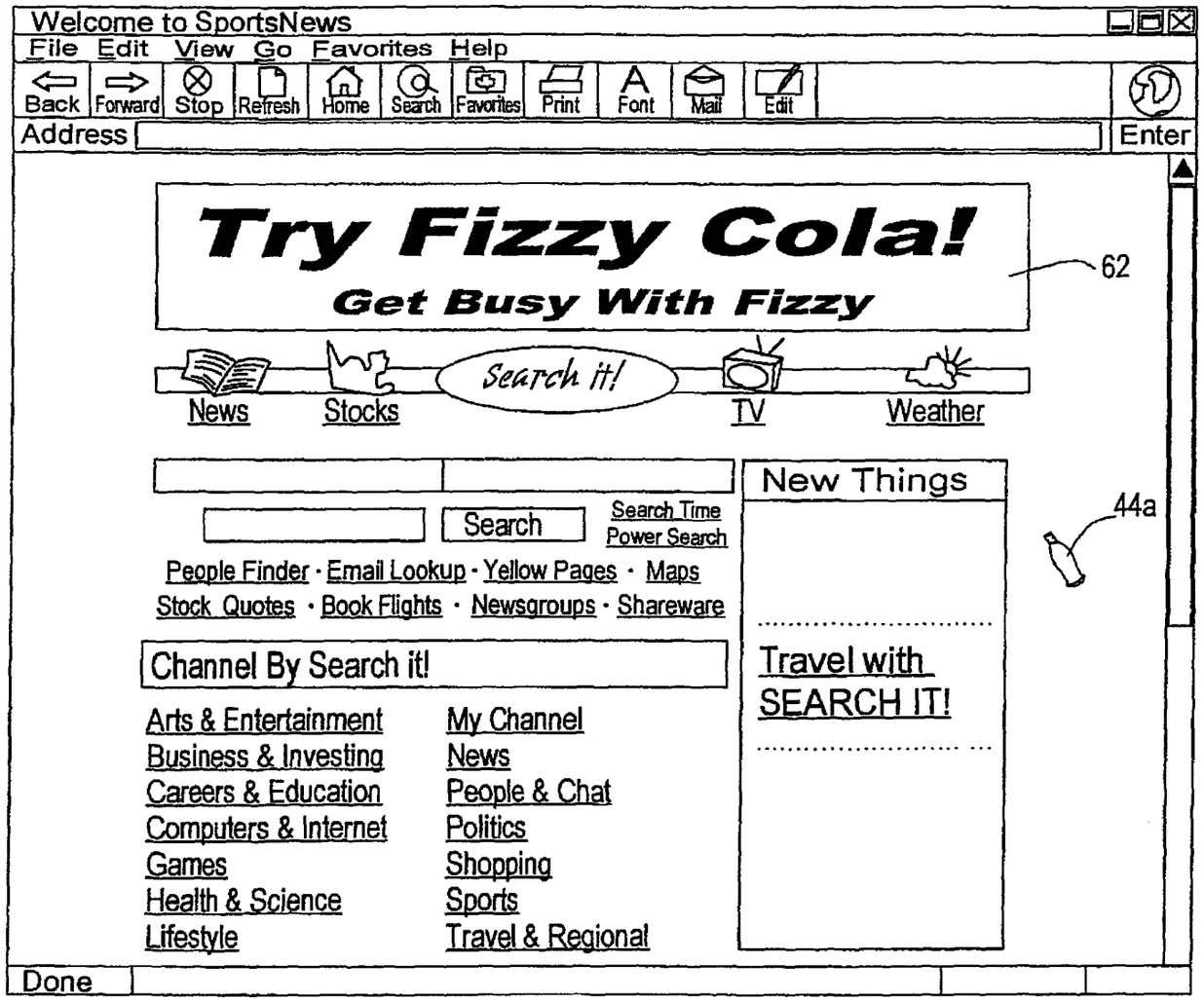

Welcome to SportsNews
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Font Mail Edit
Address
Enter
60a
Try Fizzy Cola!
Get Busy With Fizzy
62
News
Stocks
Search it!
TV
Weather
Search
Search Time
Power Search
People Finder · Email Lookup · Yellow Pages · Maps
Stock Quotes · Book Flights · Newsgroups · Shareware
Channel By Search it!
Arts & Entertainment
Business & Investing
Careers & Education
Computers & Internet
Games
Health & Science
Lifestyle
My Channel
News
People & Chat
Politics
Shopping
Sports
Travel & Regional
New Things
Travel with SEARCH IT!
44a
Done

FIG. 8

6/8

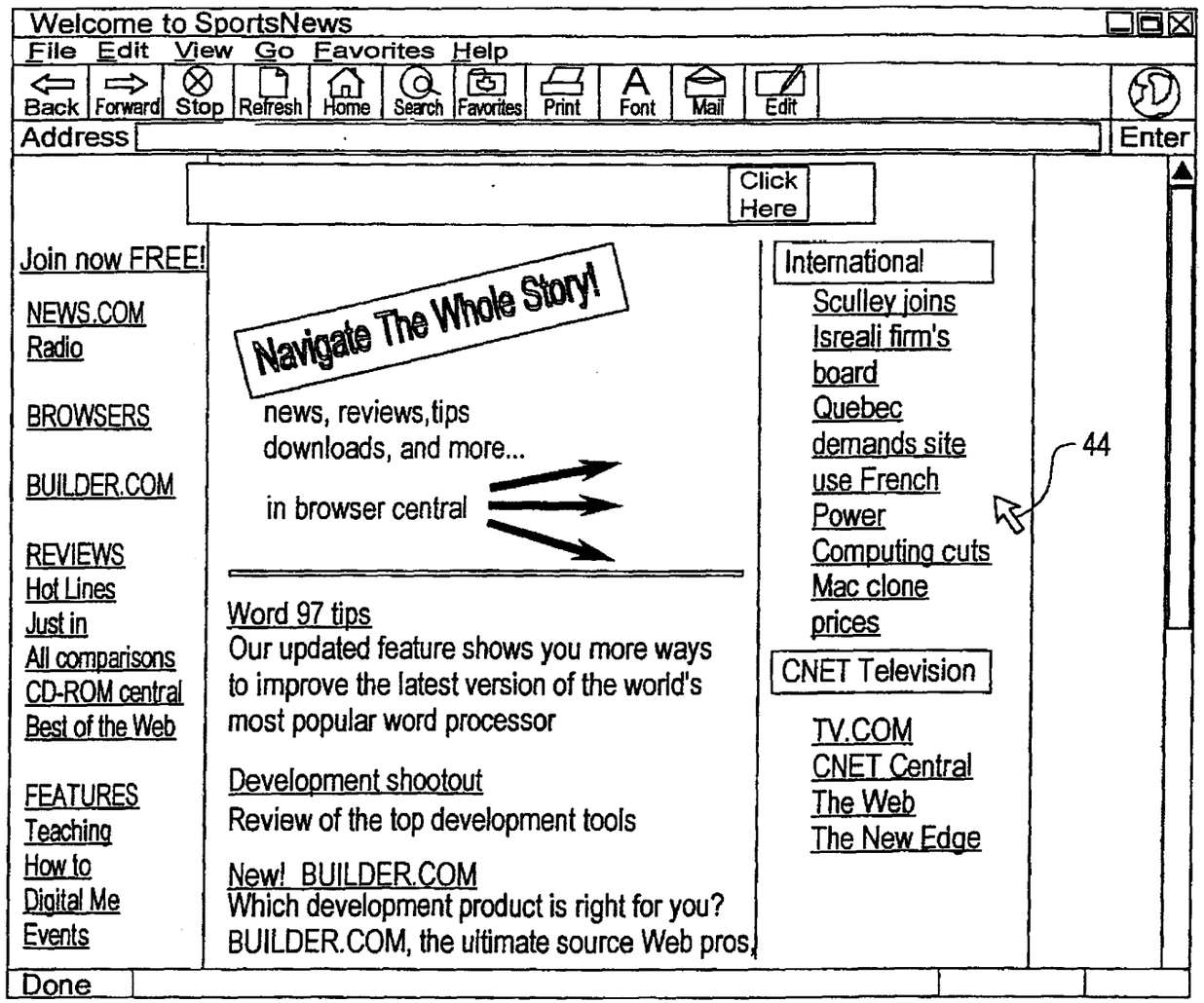

Welcome to SportsNews
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Font Mail Edit
Address
Enter
60b
Click Here
Join now FREE!
NEWS.COM
Radio
BROWSERS
BUILDER.COM
REVIEWS
Hot Lines
Just in
All comparisons
CD-ROM central
Best of the Web
FEATURES
Teaching
How to
Digital Me
Events
Navigate The Whole Story!
news, reviews,tips
downloads, and more...
in browser central
Word 97 tips
Our updated feature shows you more ways to improve the latest version of the world's most popular word processor
Development shootout
Review of the top development tools
New! BUILDER.COM
Which development product is right for you?
BUILDER.COM, the ultimate source Web pros,
International
Sculley joins
Isreali firm's
board
Quebec
demands site
use French
Power
Computing cuts
Mac clone
prices
44
CNET Television
TV.COM
CNET Central
The Web
The New Edge
Done

FIG. 9

6/6

BEST COPY

PART B—ISSUE FEE TRANSMITTAL

Complete and mail this form, together with applicable fees, to: **Box ISSUE FEE**
Assistant Commissioner for Patents
Washington, D.C. 20231

SEP 23 1999 PATENT AND TRADEMARK OFFICE

J4

MAILING INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE. Blocks 1 through 4 should be completed where appropriate. All further correspondence including the Issue Fee Receipt, the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Legibly mark-up with any corrections or use Block 1)

[illegible]

Note: The certificate of mailing below can only be used for domestic mailings of the Issue Fee Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing.

Certificate of Mailing

I hereby certify that this Issue Fee Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Box Issue Fee address above on the date indicated below.

Thelma E. Hawkins (Depositor's name)
(Signature)
September 21, 1999 (Date)

APPLICATION NO.	FILING DATE	TOTAL CLAIMS	EXAMINER AND GROUP ART UNIT	DATE MAILED
[illegible]	[illegible]	025	TADESSE, E [illegible]	[illegible]
First Named Applicant	[illegible]			

TITLE OF INVENTION [illegible]

ATTY'S DOCKET NO.	CLASS-SUBCLASS	BATCH NO.	APPLN. TYPE	SMALL ENTITY	FEE DUE	DATE DUE
COMET-001XX	[illegible]	[illegible]	UTILITY	YES	[illegible]	[illegible]

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363). Use of PTO form(s) and Customer Number are recommended, but not required.
   - ☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
   - ☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47) attached.

2. For printing on the patent front page, list (1) the names of up to 3 registered patent attorneys or agents OR, alternatively, (2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
   1 Weingarten, Schurgin, Gagnebin & Hayes LLP
   2 _____
   3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)
PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. Inclusion of assignee data is only appropriate when an assignment has been previously submitted to the PTO or is being submitted under separate cover. Completion of this form is NOT a substitute for filing an assignment.
(A) NAME OF ASSIGNEE
Comet Systems, Inc.
(B) RESIDENCE: (CITY & STATE OR COUNTRY)
New York, New York
Please check the appropriate assignee category indicated below (will not be printed on the patent)
☐ Individual ☒ corporation or other private group entity ☐ government

4a. The following fees are enclosed (make check payable to Commissioner of Patents and Trademarks):
☒ Issue Fee
☒ Advance Order - # of Copies 10

4b. The following fees or deficiency in these fees should be charged to:
DEPOSIT ACCOUNT NUMBER 23-0804
(ENCLOSE AN EXTRA COPY OF THIS FORM)
☒ Issue Fee
☒ Advance Order - # of Copies 10

The COMMISSIONER OF PATENTS AND TRADEMARKS IS requested to apply the Issue Fee to the application identified above.

(Authorized Signature) [signature] (Date) 9/21/99

NOTE: The Issue Fee will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the Patent and Trademark Office.

09/24/1999 STEFERR1 00000027 08882580

01 FC:242 605.00 OP
02 FC:561 30.00 OP

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending on the needs of the individual case. Any comments on the amount of time required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, D.C. 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND FEES AND THIS FORM TO: Box Issue Fee, Assistant Commissioner for Patents, Washington D.C. 20231

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

RECEIVED
SEP 28 1999
Publishing Division

14

TRANSMIT THIS FORM WITH FEE

File History Content Report

The following content is missing from the original file history record obtained from the United States Patent and Trademark Office. No additional information is available.

Document Date - 1999-11-30

Document Title - USPTO Grant

This page is not part of the official USPTO record. It has been determined that content identified on this document is missing from the original file history record.

OIPE JC#2
JAN 23 2004
PATENT & TRADEMARK OFFICE

09/1882,580

#21
1-3-04
B.J.H

Rev 09/03

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application	:	James S. Rosen, et al.
Patent No.	:	5,995,102
Issued	:	November 30, 1999
For	:	SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE
Attorney's Docket	:	COMET-001XX

* *

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on 1-21-2004.

By: _____
Victor B. Lebovici
Registration No. 30,864
Attorney for Applicants

* *

L E T T E R

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

It is desired to cite for the record the prior art listed on the enclosed Form PTO 1449.

Respectfully submitted,

JAMES S. ROSEN, ET AL.

By: _____
Victor B. Lebovici
Registration No. 30,864
Attorney for Applicants

WEINGARTEN, SCHURGIN,
GAGNEBIN & LEBOVICI LLP
Ten Post Office Square
Boston, Massachusetts 02109
Telephone: (617) 542-2290
Telecopier: (617) 451-0313

297135

WEINGARTEN, SCHURGIN,
GAGNEBIN & LEBOVICI LLP
TEL (617) 542-2290
FAX (617) 451-0313

Date: Janaury 21, 2004.
Page 1 of 1

U.S. DEPARTMENT OF COMMERCE (REV. 05/03) PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE CITATION *(Use several sheets if necessary)*	ATTY. DOCKET NO. COMET-001XX	PATENT NO 5,995,102
	APPLICANT: James Samuel Rosen, et al.	
	ISSUE DATE November 30, 1999	TC ART UNIT

OIPE JC100 JAN 23 2004 PATENT & TRADEMARK OFFICE

U.S. PATENT DOCUMENTS

EXAMINER INITIAL		DOCUMENT NUMBER	PUBLICATION/ ISSUE DATE	NAME	CLASS	SUBCLASS	FILING DATE
		US6,239,795	5/29/2001	Ulrich, et al.	345	333	
		US					
		US					
		US					
		US					
		US					
		US					

FOREIGN PATENT DOCUMENTS

		DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION YES	TRANSLATION NO

OTHER DOCUMENTS *(including Author, Title, Date, Pertinent Pages, etc.)*

		Java API Documentation 1.0.2, Sun Microsystems, Inc. 1996
		Using Microsoft Internet Explorer, Peter Kent, 1995, Que Corporation, Page 71

EXAMINER	DATE CONSIDERED

*EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

297134

Other Prior Art

According to the information contained in form PTO-1449 or PTO-892, there are one or more other prior art/non-patent literature documents missing from the original file history record obtained from the United States Patent and Trademark Office. Upon your request we will attempt to obtain these documents from alternative resources. Please note that additional charges will apply for this service.

This page is not part of the official USPTO record. It has been determined that content identified on this document is missing from the original file history record.

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 1997

Application or Docket Number

CLAIMS AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA
BASIC FEE		
TOTAL CLAIMS	71 minus 20 =	* 51
INDEPENDENT CLAIMS	3 minus 3 =	*
MULTIPLE DEPENDENT CLAIM PRESENT		

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐	RATE	FEE	OR	OTHER THAN SMALL ENTITY	RATE	FEE
		395.00	OR			790.00
	x$11=		OR		x$22=	1122
	x41=		OR		x82=	
	+135=		OR		+270=	
	TOTAL		OR		TOTAL	1912

CLAIMS AS AMENDED - PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 132	Minus	** 71	= 61
Independent	* ~~10~~	Minus	*** 3	= ~~7~~
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				

☒ SMALL ENTITY	RATE	ADDITIONAL FEE	OR	OTHER THAN SMALL ENTITY	RATE	ADDITIONAL FEE
	~~x$11=~~	$549	OR		18 x$22=	
	~~x41=~~	223	OR		~~x82=~~	
	+135=		OR		+270=	
	TOTAL ADDIT. FEE	822	OR		TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				

RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
x$11=		OR	x$22=	
x41=		OR	x82=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				

RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
x$11=		OR	x$22=	
x41=		OR	x82=	
+135=		OR	+270=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

FORM PTO-975 (Rev. 8/97) *U.S. Government Printing Office 1997 - 430-571/69194 Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

Form PTO 1130
(REV 2/94)

59391 U.S. PTO
08/882580
06/25/97

PACE DATA ENTRY CODING SHEET

U.S. DEPARTMENT OF COMMERCE
Patent and Trademark Office

1ST EXAMINER	DATE
2ND EXAMINER	DATE

APPLICATION NUMBER	TYPE APPL	FILING DATE MONTH DAY YEAR	SPECIAL HANDLING	GROUP ART UNIT	CLASS	SHEETS OF DRAWING

TOTAL CLAIMS	INDEPENDENT CLAIMS	SMALL ENTITY?	FILING FEE	FOREIGN LICENSE	ATTORNEY DOCKET NUMBER

CONTINUITY DATA

CONT CODE	STATUS CODE	PARENT APPLICATION SERIAL NUMBER	PCT APPLICATION SERIAL NUMBER	PARENT PATENT NUMBER	PARENT FILING DATE MONTH DAY YEAR
			PCT/		
			PCT/		
			PCT/		
			PCT/		
			PCT/		

PCT/FOREIGN APPLICATION DATA

FOREIGN PRIORITY CLAIMED	COUNTRY CODE	PCT/FOREIGN APPLICATION SERIAL NUMBER	FOREIGN FILING DATE MONTH DAY YEAR

Thomson Innovation Patent Export, 2017-05-19 05:29:44 -0500

Table of Contents

1. US5995102A Server system and method for modifying a cursor image

Family 1/1

1 record(s) per family

Record 1/1 US5995102A Server system and method for modifying a cursor image

Publication Number: US5995102A 19991130

Title: Server system and method for modifying a cursor image

Title - DWPI: Server altered cursor symbols for Internet advertising has web page loaded that contains cursor changing instructions causing download of plug-in that alters cursor shape

Priority Number: US1997882580A

Priority Date: 1997-06-25

Application Number: US1997882580A

Application Date: 1997-06-25

Publication Date: 1999-11-30

IPC Class Table:

IPC	Section	Class	Subclass	Class Group	Subgroup
G06F000300	G	G06	G06F	G06F0003	G06F000300
G06F0003033	G	G06	G06F	G06F0003	G06F0003033
G06F0003048	G	G06	G06F	G06F0003	G06F0003048
G06F000314	G	G06	G06F	G06F0003	G06F000314
G06F000900	G	G06	G06F	G06F0009	G06F000900
G06F001700	G	G06	G06F	G06F0017	G06F001700
G06F001730	G	G06	G06F	G06F0017	G06F001730
G06Q003002	G	G06	G06Q	G06Q0030	G06Q003002
G09G000508	G	G09	G09G	G09G0005	G09G000508

IPC Class Table - DWPI:

IPC - DWPI	Section - DWPI	Class - DWPI	Subclass - DWPI	Class Group - DWPI	Subgroup - DWPI
G06F001516	G	G06	G06F	G06F0015	G06F001516
G06F001700	G	G06	G06F	G06F0017	G06F001700

G06F001730	G	G06	G06F	G06F0017	G06F001730
G06F000300	G	G06	G06F	G06F0003	G06F000300
G06F0003033	G	G06	G06F	G06F0003	G06F0003033
G06F0003048	G	G06	G06F	G06F0003	G06F0003048
G06F00030481	G	G06	G06F	G06F0003	G06F00030481
G06F000314	G	G06	G06F	G06F0003	G06F000314
G06F000316	G	G06	G06F	G06F0003	G06F000316
G06F000900	G	G06	G06F	G06F0009	G06F000900
G06Q003000	G	G06	G06Q	G06Q0030	G06Q003000
G06Q003002	G	G06	G06Q	G06Q0030	G06Q003002
G06T001160	G	G06	G06T	G06T0011	G06T001160
G06T001380	G	G06	G06T	G06T0013	G06T001380
G09G000500	G	G09	G09G	G09G0005	G09G000500
G09G000508	G	G09	G09G	G09G0005	G09G000508

Assignee/Applicant: Comet Systems Inc.,New York,NY,US

JP F Terms:

JP FI Codes:

Assignee - Original: Comet Systems Inc.

Any CPC Table:

Type	Invention	Additional	Version	Office
Current	**G06Q 30/0263**	G06F 2203/04801	20130101	EP
Current	G06F 3/04812	G09G 5/08	20130101	EP
Current	G06F 3/14	G09G 2370/027	20130101	EP
Current	G06F 3/1454		20130101	EP
Current	G06F 3/167		20130101	EP
Current	G06F 17/30861		20130101	EP
Current	G06Q 30/02		20130101	EP
Current	G06Q 30/0277		20130101	EP
Current	G06T 11/60		20130101	EP
Current	G06T 13/80		20130101	EP

ECLA: G06Q00300277 | G06F00030481C | G06F000314 | G06F000314T | G06F001730W | G06Q003002 | S09G000508 | S09G037002N

Abstract:

A system for modifying a cursor image, as displayed on a video monitor of a remote terminal, to a specific image having a desired shape and appearance. The system stores cursor image data corresponding to the specific image, and a cursor display code. The cursor display code contains information in response to which the cursor image is modified to the specific image. A server computer transmits specified information to the remote terminal. The information includes at least one cursor display instruction. The cursor display instruction is operable to modify, in conjunction with the cursor information and the cursor image data, a cursor image displayed by a display of the remote terminal in the shape and appearance of the specific image.

Language of Publication: EN

INPADOC Legal Status Table:

Gazette Date	Code	INPADOC Legal Status Impact
2013-03-19	AS	-
Description: ASSIGNMENT LEXOS MEDIA IP, LLC, NEW YORK ASSIGNMENT OF ASSIGNORS INTEREST; ASSIGNOR:LEXOS MEDIA, INC.; REEL/FRAME:030044/0198 2013-01-15		
2011-04-27	FPAY	+
Description: FEE PAYMENT		
2010-01-25	AS	-
Description: ASSIGNMENT LEXOS MEDIA, INC., FLORIDA ASSIGNMENT OF ASSIGNORS INTEREST; ASSIGNOR:ALOT, INC.; REEL/FRAME:023832/0990 2009-10-30		
2009-10-01	AS	-
Description: ASSIGNMENT COMET SYSTEMS, INC., NEW YORK RELEASE BY SECURED PARTY; ASSIGNORS:PROSPECT STREET NYC DISCOVERY FUND, L.P.; PROSPECT STREET CO-INVESTMENT FUND, L.P.; BURTON, RICHARD; AND OTHERS; REEL/FRAME:023312/0694 1999-06-14		
2009-09-24	AS	-
Description: ASSIGNMENT ALOT, INC., NEW YORK CHANGE OF NAME; ASSIGNOR:MIVA DIRECT, INC.; REEL/FRAME:023273/0767 2009-06-03		
2009-09-24	AS	-
Description: ASSIGNMENT COMET SYSTEMS, INC., NEW YORK CHANGE OF NAME; ASSIGNOR:HALEY ACQUISITION CORP.; REEL/FRAME:023273/0639 2004-04-02		
2009-09-22	AS	-

Description: ASSIGNMENT HALEY ACQUISITION CORP., FLORIDA MERGER; ASSIGNOR:COMET SYSTEMS, INC.; REEL/FRAME:023263/0150 2004-03-22		
2007-04-05	FPAY	+
Description: FEE PAYMENT		
2007-04-04	AS	-
Description: ASSIGNMENT MIVA DIRECT, INC., NEW YORK ASSIGNMENT OF ASSIGNORS INTEREST; ASSIGNOR:COMET SYSTEMS, INC.; REEL/FRAME:019116/0255 2005-06-06		
2003-04-09	FPAY	+
Description: FEE PAYMENT		
1998-06-30	AS	-
Description: ASSIGNMENT BURTON, RICHARD, CALIFORNIA SECURITY INTEREST; ASSIGNOR:COMET SYSTEMS, INC.; REEL/FRAME:009295/0620 1998-06-04		
1998-06-30	AS	-
Description: ASSIGNMENT RHL VENTURES LLC AT C/O WIT CAPITAL, NEW YORK SECURITY INTEREST; ASSIGNOR:COMET SYSTEMS, INC.; REEL/FRAME:009295/0620 1998-06-04		
1998-06-30	AS	-
Description: ASSIGNMENT PROSPECT STREET NYC CO-INVESTMENT FUND, L.P., NEW SECURITY INTEREST; ASSIGNOR:COMET SYSTEMS, INC.; REEL/FRAME:009295/0620 1998-06-04		
1998-06-30	AS	-
Description: ASSIGNMENT PROSPECT STREET NYC DISCOVERY FUND, L.P., NEW YORK SECURITY INTEREST; ASSIGNOR:COMET SYSTEMS, INC.; REEL/FRAME:009295/0620 1998-06-04		
1998-06-30	AS	-
Description: ASSIGNMENT KLEIN, STEVE, NEW YORK SECURITY INTEREST; ASSIGNOR:COMET SYSTEMS, INC.; REEL/FRAME:009295/0620 1998-06-04		
1997-12-18	AS	-
Description: ASSIGNMENT PROSPECT STREET NYC CO-INVESTMENT FUND, L.P., NEW SECURITY AGREEMENT; ASSIGNOR:COMET SYSTEMS, INC.; REEL/FRAME:008885/0598 1997-12-04		
1997-12-18	AS	-
Description: ASSIGNMENT PROSPECT STREET NYC DISCOVERY FUND, L.P., NEW YORK SECURITY		

AGREEMENT; ASSIGNOR:COMET SYSTEMS, INC.; REEL/FRAME:008885/0598 1997-12-04		
1997-06-25	AS	-
Description: ASSIGNMENT COMET SYSTEMS, INC., NEW YORK ASSIGNMENT OF ASSIGNORS INTEREST; ASSIGNORS:ROSEN, JAMES S.; SCHMITTER, THOMAS A.; HALL, MARK; REEL/FRAME:008631/0209 1997-06-20		

Post-Issuance (US):

Reassignment (US) Table:

Assignee	Assignor	Date Signed	Reel/Frame	Date
LEXOS MEDIA IP LLC,NEW YORK,NY,US	LEXOS MEDIA, INC.	2013-01-15	030044/0198	2013-03-19
Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).				
Corresponent: KEVIN M. MOSS 1177 AVENUE OF THE AMERICAS KRAMER LEVIN NAFTALIS & FRANKEL LLP NEW YORK, NY 10036				
LEXOS MEDIA INC.,FORT MYERS,FL,US	ALOT, INC.	2009-10-30	023832/0990	2010-01-25
Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).				
Corresponent: DORSEY & WHITNEY LLP 250 PARK AVENUE NEW YORK, NY 10177				
ALOT INC.,NEW YORK,NY,US	MIVA DIRECT, INC.	2009-06-03	023273/0767	2009-09-24
Conveyance: CHANGE OF NAME (SEE DOCUMENT FOR DETAILS).				
Corresponent: WEINGARTEN SCHURGIN GAGNEBIN & LEBOVICI TEN POST OFFICE SQUARE BOSTON, MA 02109				
MIVA DIRECT INC.,NEW YORK,NY,US	COMET SYSTEMS, INC.	2005-06-06	019116/0255	2007-04-04
Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).				
Corresponent: WEINGARTEN, SCHURGIN, GAGNEBIN & LEBOVICI LLP TEN POST OFFICE SQUARE BOSTON, MASSACHUSETTS 02109				
COMET SYSTEMS INC.,NEW YORK,NY,US	HALEY ACQUISITION CORP.	2004-04-02	023273/0639	2009-09-24
Conveyance: CHANGE OF NAME (SEE DOCUMENT FOR DETAILS).				
Corresponent: WEINGARTEN SCHURGIN GAGNEBIN & LEBOVICI TEN POST OFFICE SQUARE BOSTON, MA 02109				

HALEY ACQUISITION CORP.,FORT MYERS,FL,US	COMET SYSTEMS, INC.	2004-03-22	023263/0150	2009-09-22
Conveyance: MERGER (SEE DOCUMENT FOR DETAILS).				
Corresponent: WEINGARTEN, SCHURGIN, GAGNEBIN & LEBOVIC TEN POST OFFICE SQUARE BOSTON, MA 02109				
COMET SYSTEMS INC.,NEW YORK,NY,US	PROSPECT STREET NYC DISCOVERY FUND, L.P.	1999-06-14	023312/0694	2009-10-01
	PROSPECT STREET CO-INVESTMENT FUND, L.P.	1999-06-14		
	BURTON, RICHARD	1999-06-14		
	RHL VENTURES L.L.C.	1999-06-14		
	KLEIN, STEPHEN D.	1999-06-14		
Conveyance: RELEASE BY SECURED PARTY				
Corresponent: WEINGARTEN, SCHURGIN, GAGNEBIN & LEBOVICI LLP TEN POST OFFICE SQUARE BOSTON, MASSACHUSETTS 02109				
PROSPECT STREET NYC DISCOVERY FUND L.P.,NEW YORK,NY,US	COMET SYSTEMS, INC.	1998-06-04	009295/0620	1998-06-30
PROSPECT STREET NYC CO-INVESTMENT FUND L.P.,NEW YORK,NY,US				
RHL VENTURES LLC AT C/O WIT CAPITAL,NEW YORK,NY,US				
KLEIN STEVE,NEW YORK,NY,US				
BURTON RICHARD,PALO ALTO,CA,US				
Conveyance: SECURITY INTEREST (SEE DOCUMENT FOR DETAILS). | SECURITY INTEREST (SEE DOCUMENT FOR DETAILS). | SECURITY INTEREST (SEE DOCUMENT FOR DETAILS). | SECURITY INTEREST (SEE DOCUMENT FOR DETAILS). | SECURITY INTEREST (SEE DOCUMENT FOR DETAILS).				
Corresponent: MORGAN, LEWIS & BOCKIUS LLP MR. MATTHEW T. BAILEY 1800 M STREET, N.W. WASHINGTON, D.C. 20036 | MORGAN, LEWIS & BOCKIUS LLP MR. MATTHEW T. BAILEY 1800 M STREET, N.W. WASHINGTON, D.C. 20036 | MORGAN, LEWIS & BOCKIUS LLP MR. MATTHEW T. BAILEY 1800 M STREET, N.W. WASHINGTON, D.C. 20036 | MORGAN, LEWIS & BOCKIUS LLP MR. MATTHEW T. BAILEY 1800 M STREET, N.W. WASHINGTON, D.C. 20036 | MORGAN, LEWIS & BOCKIUS LLP MR. MATTHEW T. BAILEY 1800 M STREET, N.W. WASHINGTON, D.C. 20036				
PROSPECT STREET NYC DISCOVERY FUND L.P.,NEW YORK,NY,US	COMET SYSTEMS, INC.	1997-12-04	008885/0598	1997-12-18

PROSPECT STREET NYC CO-INVESTMENT FUND L.P.,NEW YORK,NY,US	-	-	-	-
Conveyance: SECURITY AGREEMENT | SECURITY AGREEMENT				
Corresponent: MORGAN, LEWIS & BOCKIUS LLP MATTHEW T. BAILEY 1800 M STREET, N.W. WASHINGTON, D.C. 20036 | MORGAN, LEWIS & BOCKIUS LLP MATTHEW T. BAILEY 1800 M STREET, N.W. WASHINGTON, D.C. 20036				
COMET SYSTEMS INC.,NEW YORK,NY,US	ROSEN, JAMES S.	1997-06-20	008631/0209	1997-06-25
	SCHMITTER, THOMAS A.	1997-06-20	-	-
	HALL, MARK	1997-06-20	-	-
Conveyance: ASSIGNMENT OF ASSIGNORS INTEREST (SEE DOCUMENT FOR DETAILS).				
Corresponent: SOFER & HAROUN, LLP JOSEPH SOFER 342 MADISON AVENUE, SUITE 1921 NEW YORK, NY 10173-1907				

Maintenance Status (US):

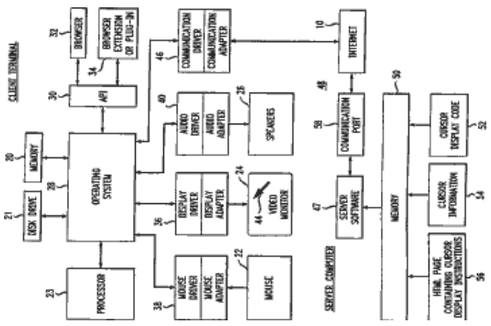
Litigation (US): 2017-04-28 2017 Lexos Media IP, LLC Boscov's Department Store, LLC E.D. Texas 2:17cv00373 | 2017-04-28 2017 Lexos Media IP, LLC Amerimark Direct, LLC E.D. Texas 2:17cv00372 | 2016-07-11 2016 Lexos Media IP, LLC Saks Incorporated E.D. Texas 2:16cv00751 | 2016-07-11 2016 Lexos Media IP, LLC Nordstrom, Inc. E.D. Texas 2:16cv00750 | 2016-07-11 2016 Lexos Media IP, LLC Musician's Friend, Inc. E.D. Texas 2:16cv00749 | 2016-07-11 2016 Lexos Media IP, LLC Costco Wholesale Corp. E.D. Texas 2:16cv00748 | 2016-07-11 2016 Lexos Media IP, LLC Victoria's Secret Stores Brand Management, Inc. E.D. Texas 2:16cv00752 | 2016-07-11 2016 Lexos Media IP, LLC Apmex, Inc. E.D. Texas 2:16cv00747 | 2015-12-23 2015 Lexos Media IP, LLC Recreational Equipment, Inc. E.D. Texas 2:15cv02107 | 2015-12-18 2015 Lexos Media IP, LLC Sears Brands, LLC E.D. Texas 2:15cv02100 | 2015-12-18 2015 Lexos Media IP, LLC Sears Brands, LLC E.D. Texas 2:15cv02098 | 2015-12-14 2015 Lexos Media IP, LLC Express, LLC E.D. Texas 2:15cv02073 | 2015-12-08 2015 Lexos Media IP, LLC Avon Products, Inc. E.D. Texas 2:15cv02052 | 2015-12-04 2015 Lexos Media IP, LLC The Home Depot USA, Inc Homer TLC, Inc. E.D. Texas 2:15cv02051 | 2012-10-26 2012 Lexos Media, Inc. Zynga, Inc. S.D. New York 1:12cv07994 | 2012-07-24 2012 Lexos Media, Inc. Zynga, Inc. M.D. Florida 2:12cv00395 | 2012-04-19 2012 Zynga Inc Lexos Media, Inc N.D. California 5:12cv01952

Opposition (EP):

License (EP):

EPO Procedural Status:

Front Page Drawing:



Assignee - Current US: LEXOS MEDIA IP LLC

THOMSON REUTERS

Copyright 2007-2017 THOMSON REUTERS

United States Patent and Trademark Office
Office of the Commissioner for Patents

SERVER SYSTEM AND METHOD FOR MODIFYING A CURSOR IMAGE

PATENT #	APPLICATION #	FILING DATE	ISSUE DATE
5995102	08882580	06/25/1997	11/30/1999

Payment Window Status

No maintenance fees are due.

WINDOW	STATUS	FEES
11.5 Year	Closed	Paid

Window	First Day to Pay	Surcharge Starts	Last Day to Pay	Status	Fees
3.5 Year	11/30/2002	05/31/2003	12/01/2003	Closed	Paid
7.5 Year	11/30/2006	05/31/2007	11/30/2007	Closed	Paid
11.5 Year	11/30/2010	06/01/2011	11/30/2011	Closed	Paid

Patent Holder Information

Customer # 207

Entity Status SMALL

Phone Number 6172263800

Address PRETI FLAHERTY BELIVEAU & PACHIOS LLP
60 State Street
Suite 1100
BOSTON, MA 02109
UNITED STATES