UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE

SOFTVIEW LLC,

Plaintiff,

v.

APPLE INC.; AT&T MOBILITY LLC; DELL INC.; HTC CORP.; HTC AMERICA, INC.; HUAWEI TECHNOLOGIES CO., LTD.; FUTUREWEI TECHNOLOGIES, INC.; KYOCERA CORP.; KYOCERA WIRELESS CORP.; LG ELECTRONICS, INC.; LG ELECTRONICS USA, INC.; LG ELECTRONICS MOBILECOMM U.S.A., INC.; MOTOROLA MOBILITY INC.; SAMSUNG ELECTRONICS CO., LTD.; SAMSUNG ELECTRONICS AMERICA, INC.; SAMSUNG TELECOMMUNICATIONS AMERICA, LLC; and SONY ERICSSON MOBILE COMMUNICATIONS (USA) INC.,

Defendants.

Civil Action No. 10-389-LPS

CONSOLIDATED

DECLARATION OF CRAIG JOHNSON IN SUPPORT OF PLAINTIFF SOFTVIEW LLC'S OPENING CLAIM CONSTRUCTION BRIEF

OF COUNSEL:

Morgan Chu
Samuel K. Lu
Erin McCracken
Craig Johnson
IRELL & MANELLA LLP
1800 Avenue of the Stars, Suite 900
Los Angeles, CA 90067
(310) 277-1010

Steven L Caponi (#3483)
BLANK ROME LLC
1201 Market Street, Suite 800
Wilmington, DE 19801
(301) 425-6400
caponi@blankrome.com
Attorneys for Plaintiff SoftView LLC

Dated: September 21, 2012

EXHIBIT
Petitioner - Motorola
PX 1023

I, Craig Johnson, declare as follows:

1. I am an attorney with the law firm Irell & Manella LLP, counsel for plaintiff SoftView LLC in the above-captioned matter. I am a member of the State Bar of California and have been admitted to practice *pro hac vice* before this Court. I have personal knowledge of the facts set forth in this Declaration and, if called as a witness, could and would testify competently to such facts under oath.

2. Attached hereto as Exhibit 1 is a copy of U.S. Patent No. 7,461,353.

3. Attached hereto as Exhibit 2 is a copy of U.S. Patent No. 7,831,926.

4. Attached hereto as Exhibit 3 are excerpts from the prosecution history of U.S. Patent No. 7,461,353.

5. Attached hereto as Exhibit 4 are excerpts from the prosecution history of U.S. Patent No. 7,831,926.

6. Attached hereto as Exhibit 5 are excerpts from Merriam-Webster's Collegiate Dictionary, Tenth Edition (Merriam-Webster, Inc. 2001).

7. Attached hereto as Exhibit 6 are excerpts from the Microsoft Computer Dictionary, Fifth Edition (Microsoft Press, 2002).

8. Attached hereto as Exhibit 7 is a copy of United States Patent and Trademark Office Order Granting Request for Inter Partes Reexamination of U.S. Patent No. 7,461,353, Application/Control Number: 95/000,634, dated August 8, 2011.

9. Attached hereto as Exhibit 8 is a copy of Apple's Statement Identifying The Claims It Will Assert At Trial, filed on May 7, 2012 in *Apple Inc. v. Samsung Electronics Co., Ltd.*, No. C-11-01846 (N.D. Cal.).

10. Attached hereto as Exhibit 9 is a copy of *Accessibility-iPad-Vision*, APPLE.COM, http://www.apple.com/accessibility/ipad/vision.html (last visited Sept. 20, 2012).

11. Attached hereto as Exhibit 10 is an excerpt of the August 3, 2012 testimony of Scott Forstall in *Apple Inc. v. Samsung Electronics Co., Ltd.* et al. No. C-11-01846 (N.D. Cal.).

12. Attached hereto as Exhibit 11 are excerpts from Newton's Telecom Dictionary, Thirteenth Edition (Telecom Books and FlatIron Pubilshing, 1998).

13. Attached hereto as Exhibit 12 are Figures 4C and 4D from the specifications of the patents-in-suit, with color highlighting added.

14. Attached hereto as Exhibit 13 is Figure 7A from the specifications of the patents-in-suit, with color highlighting added.

15. Attached hereto as Exhibit 14 are Figures 1A and 1B from the specification of the patents-in-suit, with color highlighting added.

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Executed this 21st day of September 2012 in Los Angeles, California.

Craig Johnson

Exhibit 1

US007461353B2

(12) **United States Patent**
Rohrabaugh et al.

(10) **Patent No.: US 7,461,353 B2**
(45) **Date of Patent: *Dec. 2, 2008**

(54) **SCALABLE DISPLAY OF INTERNET CONTENT ON MOBILE DEVICES**

(75) Inventors: **Gary B. Rohrabaugh**, Bellingham, WA (US); **Scott A. Sherman**, Bellingham, WA (US)

(73) Assignee: **Gary Rohrabaugh**, Bellingham, WA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 376 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **11/045,757**

(22) Filed: **Jan. 28, 2005**

(65) **Prior Publication Data**

US 2005/0131887 A1 Jun. 16, 2005

Related U.S. Application Data

(60) Division of application No. 09/878,097, filed on Jun. 8, 2001, now Pat. No. 7,210,099, which is a continuation-in-part of application No. 09/828,511, filed on Apr. 7, 2001, now abandoned.

(60) Provisional application No. 60/217,345, filed on Jul. 11, 2000, provisional application No. 60/211,019, filed on Jun. 12, 2000.

(51) **Int. Cl.**
G06F 17/00 (2006.01)

(52) **U.S. Cl.** **715/815**; 715/760; 715/234; 715/239; 715/249

(58) **Field of Classification Search** 715/517, 715/700–864

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,966,135 A	10/1999	Roy et al.

(Continued)

FOREIGN PATENT DOCUMENTS

WO	PCT/US01/40920	12/2001

(Continued)

OTHER PUBLICATIONS

http://www.w3.org/TR/1999/WD-SVG-19990211/—Pub Feb. 11, 1999 by W3C pp. 1-7 (A complet details can be down load at by the provides URL above).*

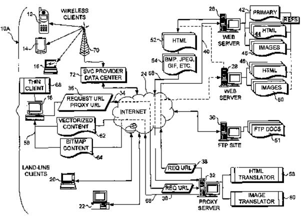
(Continued)

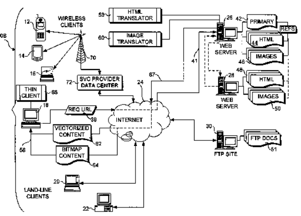
Primary Examiner—Doug Hutton
Assistant Examiner—Quoc A Tran
(74) *Attorney, Agent, or Firm*—Law Office of R. Alan Burnett; R. Alan Burnett

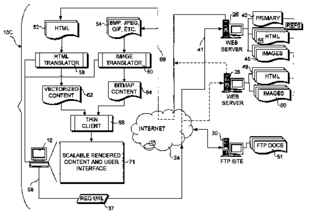
(57) **ABSTRACT**

Mobile devices enabled to support resolution-independent scalable display of Internet (Web) content to allow Web pages to be scaled (zoomed) and panned for better viewing on smaller screen sizes. The mobile devices employ software-based processing of original Web content, including HTML-based content, XML, cascade style sheets, etc. to generate scalable content. The scalable content and/or data derived therefrom are then employed to enable the Web content to be rapidly rendered, zoomed, and panned. Moreover, the rendered displays provide substantially the same or identical layout as the original Web page, enabling users to easily navigate to selected content and features on familiar Web pages. Display lists may also be employed to provide further enhancements in rendering speed. Additionally, hardware-based programmed logic may be employed to facilitate various operations.

319 Claims, 22 Drawing Sheets







U.S. PATENT DOCUMENTS

6,011,905	A	1/2000	Huttenlocher et al.	
6,057,854	A	5/2000	Davis, Jr. et al.	
6,185,625	B1	2/2001	Tso et al.	
6,300,947	B1	10/2001	Kanevsky	
6,421,733	B1	7/2002	Tso et al.	
6,466,203	B2	10/2002	Van Ee	
6,546,397	B1	4/2003	Rempell	
6,615,212	B1	9/2003	Dutta et al.	
6,642,925	B2 *	11/2003	Roy et al.	345/427
6,674,445	B1 *	1/2004	Chithambaram et al.	345/619
6,857,102	B1	2/2005	Bickmore et al.	
6,886,034	B2 *	4/2005	Blumberg	709/217
7,219,309	B2	5/2007	Kaasila et al.	
2001/0047428	A1	11/2001	Hunter	
2002/0112237	A1	8/2002	Ketts	
2004/0049598	A1	3/2004	Tucker et al.	
2005/0144256	A1	6/2005	Blumberg	

FOREIGN PATENT DOCUMENTS

WO	PCT/US01/40920	6/2002

OTHER PUBLICATIONS

Combined Printout of HTML Help Files for Opera 3.60 (HTML pages printed to PDF docs and combined), Available on May 12, 1999.

Screenshots compilations of Web pages using Opera 3.60, including: www.Altavista.com—May 1, 1999 www.cnet.com—May 8, 1999 www.excite.com—May 8, 1999 www.nytimes.com—Apr. 22, 1999 www.softsource.com—Apr. 17, 1999 www.utexas.edu—Apr. 27, 1999 www.uspto.gov—May 5, 1999 www.yahoo.com—May 8, 1999.

Benjamin B. Bederson et al., Pad++: A Zoomable Graphical Sketchpad For Exploring Alternate Interface Physics, Sep. 19, 1995, http://www.cs.unm.edu/pad++.

Benjamin B. Bederson et al., A Zooming Web Browser, SPIE 1996, http://www.cs.umd.edu/hcil/jazz/learn/papers/spie-96-webbrowser.pdf.

Specification for Simple Vector Format (SVF) v1.1 Jan. 16, 1995.

Specification for Simple Vector Format (SVF) v2.0 Dec. 6, 2000, http://www.svf.org/spec.html.

Changes to SVF, (Date unknown).

Scalable Vector Graphics (SVG) Specification, W3C Working Draft Feb. 11, 1999 WD-SVG-19990211, http://www.w3.org/TR/1999/WD-SVG-19990211/ (HTML format—initial page).

Scalable Vector Graphics (SVG) Specification, W3C Working Draft Apr. 12, 1999 WD-SVG-19990412, http://www.w3.org/TR/1999/WD-SVG-19990412/ (HTML format—initial page).

Scalable Vector Graphics (SVG) 1.0 Specification, W3C Candidate Recommendation Nov. 2, 2000.

Introduction to SVG, part of WD-SVG-19990211, http://www.w3.org/TR/1999/ WD-SVG-19990211/intro. html#Document . . . .

Steve Mulder, Sneak Peak at SVG, Mar. 4, 1999, http://www.webmonkey.com/99/10/index3a.html?tw=eg1999102.

Janus Boye, SVG Brings Fast Vector Graphics to Web, Jul. 29, 1999 http://www.irt.org/articles/js176/.

Vector Markup Language (VML), World Wide Web Consortium Note May 13, 1998, NOTE-VML-19980513, http://www.w3.org/TR/1998/NOTE-VML_19980513.

Precision Graphics Markup Language (PGML), World Wide Web Consortium NOTE Apr. 10, 1998, NOTE-PGML-19980410, http://www.w3.org/TR/1998/NOTE-PGML-19980410.

Rick Graham, Mobile SVG at BitFlash Inc., May 2001, http://www.w3.org/Talks/2002/1007-DI-Helsinki/bitflash/index.html.

U.S. Appl. No. 60/296,327 Available on PAIR.

U.S. Appl. No. 60/288,287, filed May 2, 2001. Available on PAIR—electronic copy previously filed via IDS submission of Sep. 18, 2007.

U.S. Appl. No. 60/296,275, filed Jun. 5, 2001. Available on PAIR—electronic copy previously filed via IDS submission of Sep. 18, 2007.

U.S. Appl. No. 60/296,237, filed Jun. 5, 2001. Available on PAIR—electronic copy previously filed via IDS submission of Sep. 18, 2007.

U.S. Appl. No. 60/296,274, filed Jun. 5, 2001. Available on PAIR—electronic copy previously filed via IDS submission of Sep. 18, 2007.

U.S. Appl. No. 60/296,284, filed Jun. 5, 2001. Available on PAIR—electronic copy previously filed via IDS submission of Sep. 18, 2007.

U.S. Appl. No. 60/296,231, filed Jun. 5, 2001. Available on PAIR—electronic copy previously filed via IDS submission of Sep. 18, 2007.

U.S. Appl. No. 60/296,224, filed Jun. 5, 2001. Available on PAIR—electronic copy previously filed via IDS submission of Sep. 18, 2007.

U.S. Appl. No. 60/296,426, filed Jun. 5, 2001. Available on PAIR—electronic copy previously filed via IDS submission of Sep. 18, 2007.

U.S. Appl. No. 60/296,273, filed Jun. 5, 2001. Available on PAIR—electronic copy previously filed via IDS submission of Sep. 18, 2007.

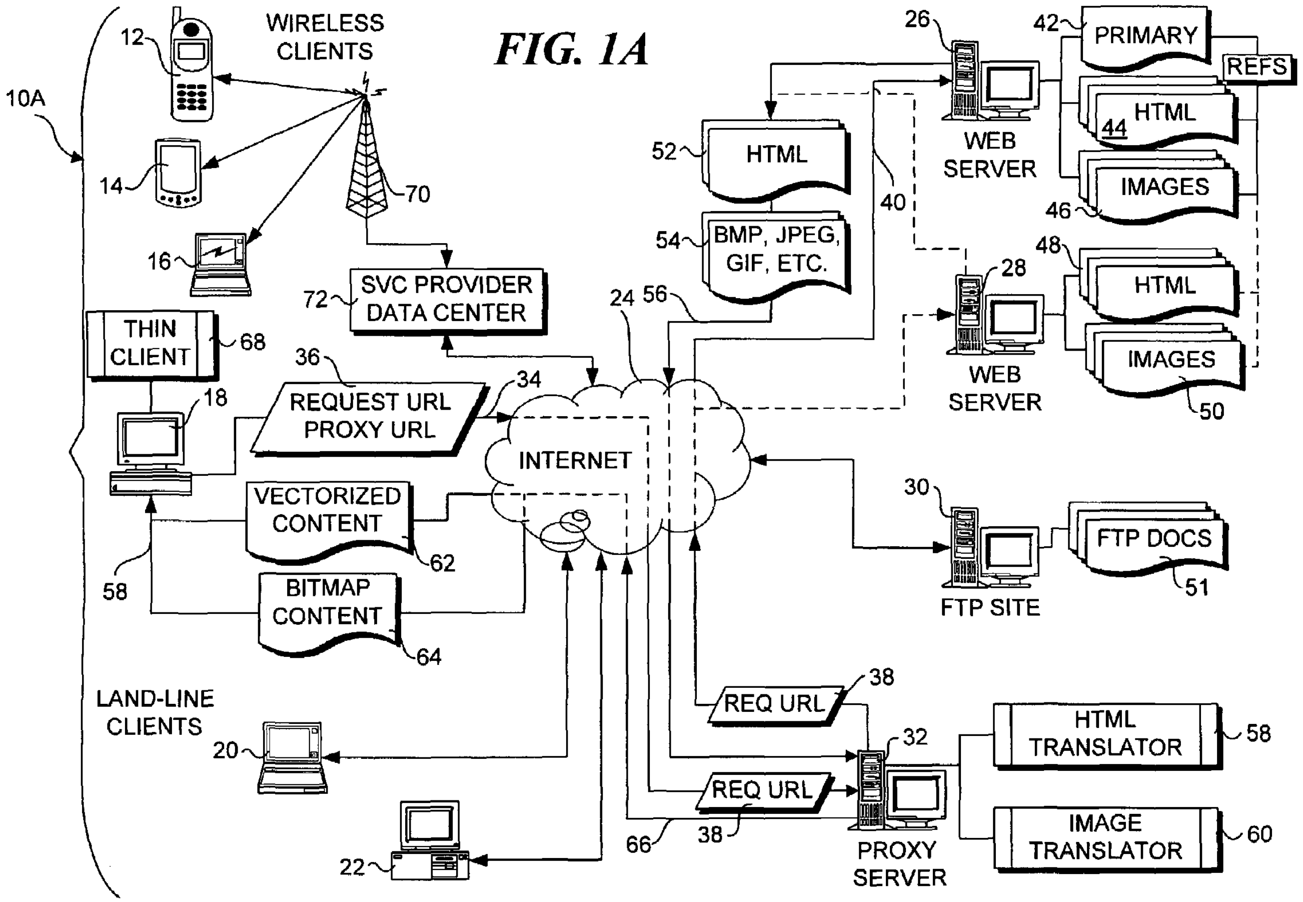
U.S. Appl. No. 60/296,283, filed Jun. 5, 2001. Available on PAIR—electronic copy previously filed via IDS submission of Sep. 18, 2007.

U.S. Appl. No. 60/296,281, filed Jun. 5, 2001. Available on PAIR—electronic copy previously filed via IDS submission of Sep. 18, 2007.

U.S. Appl. No. 60/296,327, filed Jun. 5, 2001. Available on PAIR—electronic copy previously filed via IDS submission of Sep. 18, 2007.

* cited by examiner

FIG. 1A


10A
12
WIRELESS CLIENTS
14
16
70
72
SVC PROVIDER DATA CENTER
THIN CLIENT
68
18
36
REQUEST URL PROXY URL
34
INTERNET
24
56
VECTORIZED CONTENT
62
BITMAP CONTENT
64
58
LAND-LINE CLIENTS
20
22
66
38
REQ URL
REQ URL
38
32
PROXY SERVER
HTML TRANSLATOR
58
IMAGE TRANSLATOR
60
52
HTML
54
BMP, JPEG, GIF, ETC.
40
26
WEB SERVER
42
PRIMARY
REFS
HTML
44
IMAGES
46
48
HTML
28
WEB SERVER
IMAGES
50
30
FTP SITE
FTP DOCS
51

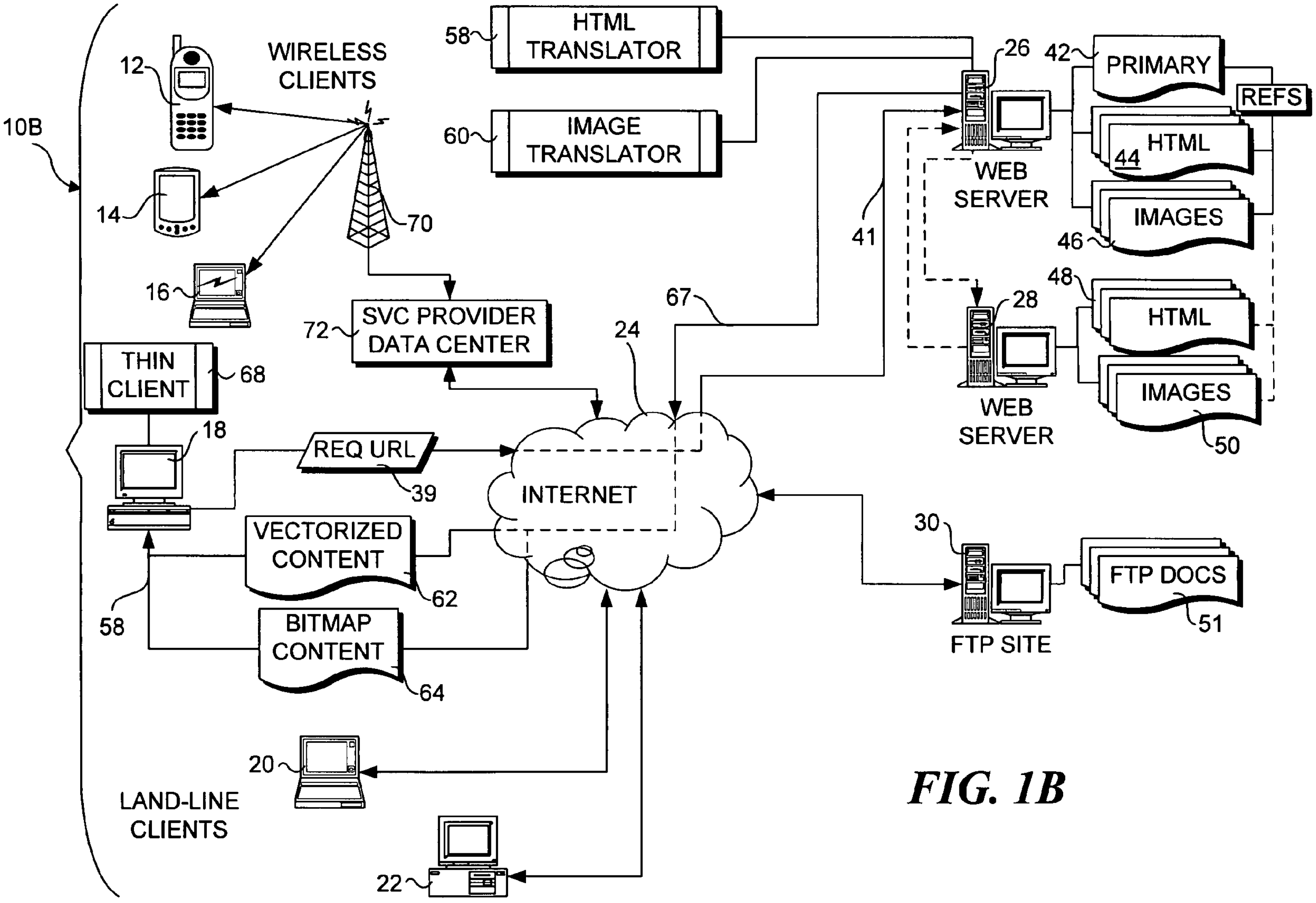

10B
12
WIRELESS
CLIENTS
14
16
70
58
HTML
TRANSLATOR
60
IMAGE
TRANSLATOR
72
SVC PROVIDER
DATA CENTER
THIN
CLIENT
68
18
REQ URL
39
INTERNET
24
67
41
VECTORIZED
CONTENT
62
58
BITMAP
CONTENT
64
20
LAND-LINE
CLIENTS
22
26
WEB
SERVER
42
PRIMARY
REFS
HTML
44
IMAGES
46
48
HTML
IMAGES
50
28
WEB
SERVER
30
FTP SITE
FTP DOCS
51

FIG. 1B

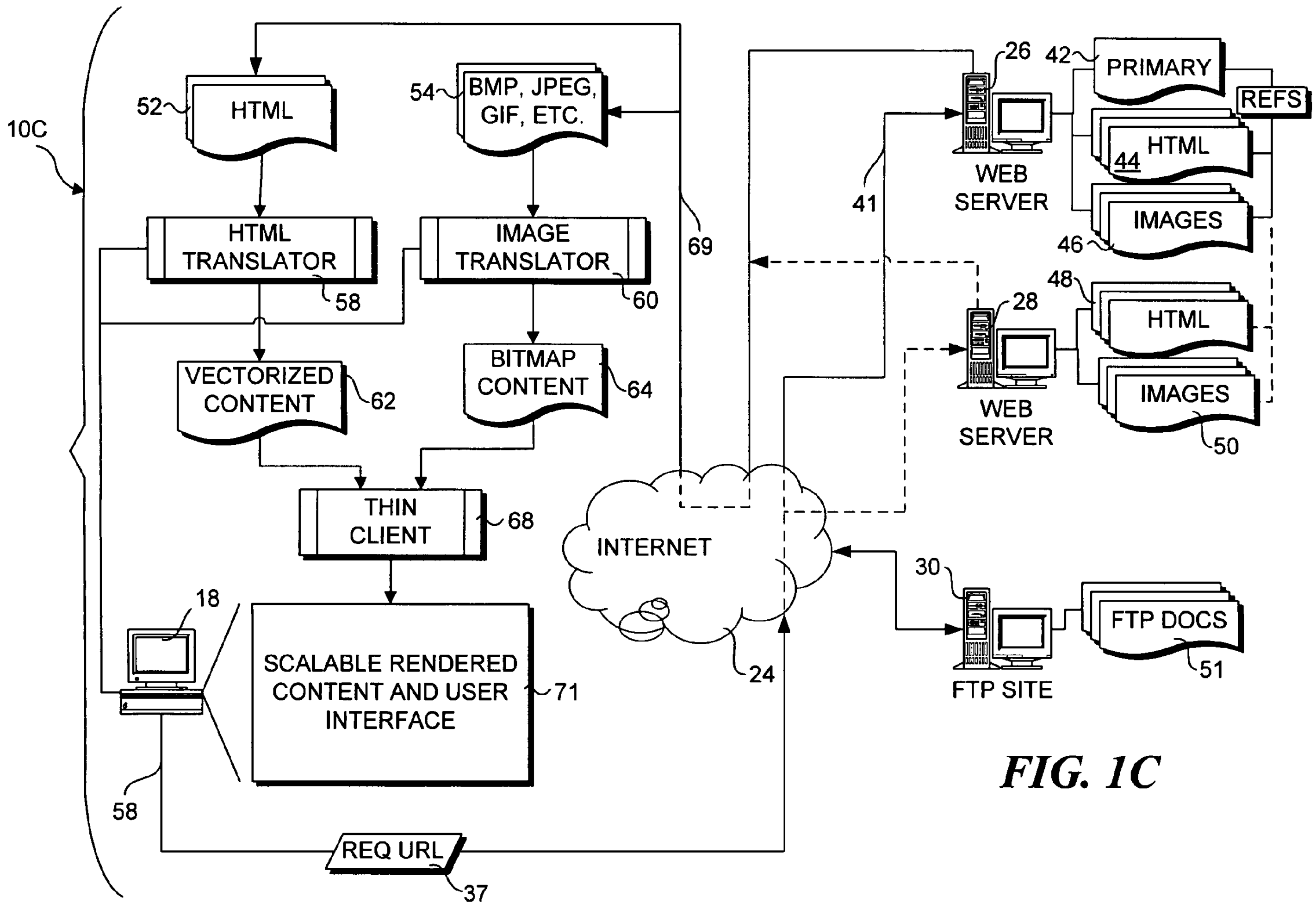

10C
52
HTML
54
BMP, JPEG, GIF, ETC.
HTML TRANSLATOR
58
IMAGE TRANSLATOR
60
69
VECTORIZED CONTENT
62
BITMAP CONTENT
64
THIN CLIENT
68
18
SCALABLE RENDERED CONTENT AND USER INTERFACE
71
58
REQ URL
37
INTERNET
24
41
26
WEB SERVER
42
PRIMARY
REFS
HTML
44
IMAGES
46
48
28
HTML
WEB SERVER
IMAGES
50
30
FTP SITE
FTP DOCS
51

FIG. 1C

FIG. 2A

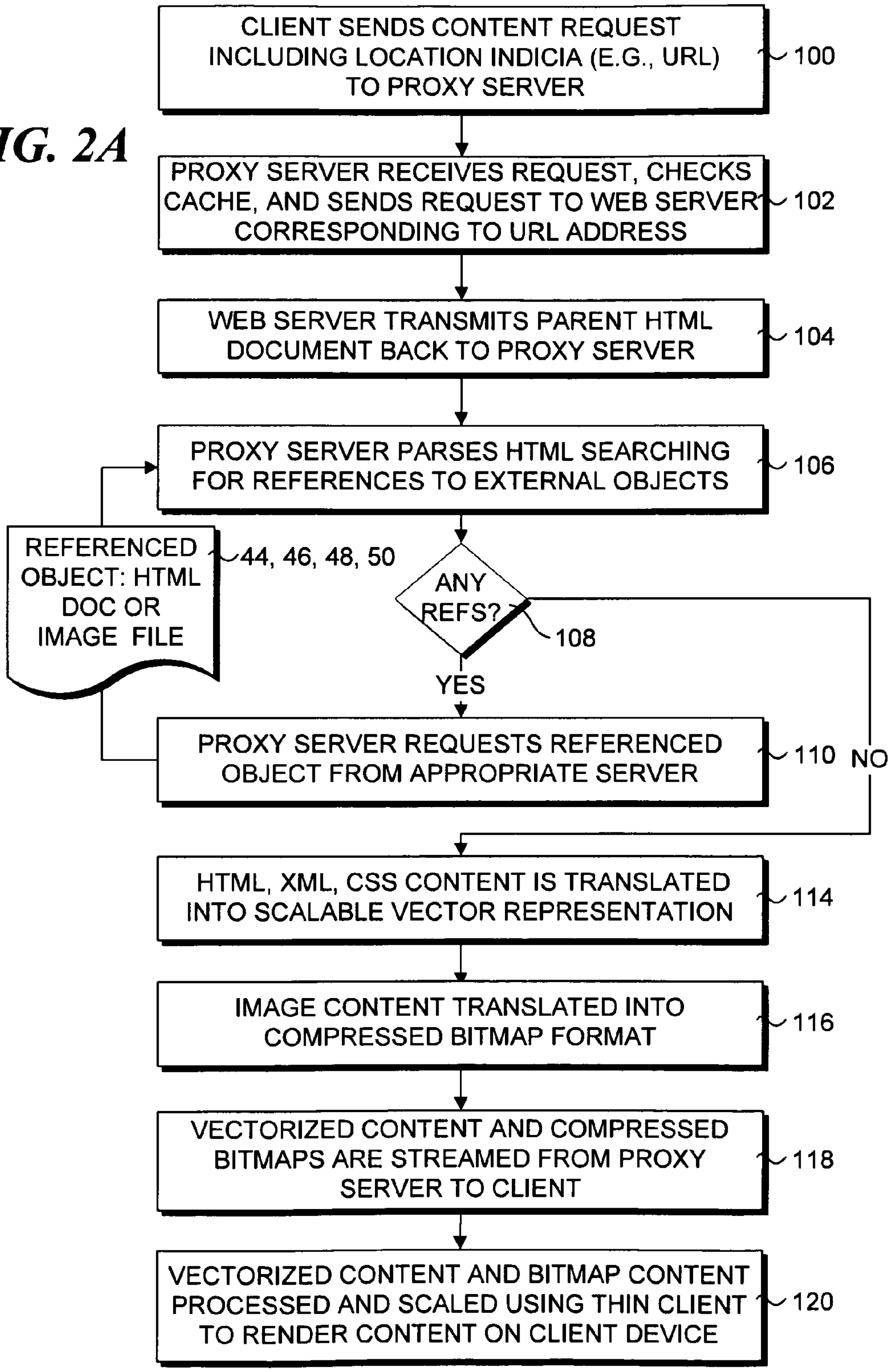

CLIENT SENDS CONTENT REQUEST INCLUDING LOCATION INDICIA (E.G., URL) TO PROXY SERVER
100
PROXY SERVER RECEIVES REQUEST, CHECKS CACHE, AND SENDS REQUEST TO WEB SERVER CORRESPONDING TO URL ADDRESS
102
WEB SERVER TRANSMITS PARENT HTML DOCUMENT BACK TO PROXY SERVER
104
PROXY SERVER PARSES HTML SEARCHING FOR REFERENCES TO EXTERNAL OBJECTS
106
REFERENCED OBJECT: HTML DOC OR IMAGE FILE
44, 46, 48, 50
ANY REFS?
108
YES
NO
PROXY SERVER REQUESTS REFERENCED OBJECT FROM APPROPRIATE SERVER
110
HTML, XML, CSS CONTENT IS TRANSLATED INTO SCALABLE VECTOR REPRESENTATION
114
IMAGE CONTENT TRANSLATED INTO COMPRESSED BITMAP FORMAT
116
VECTORIZED CONTENT AND COMPRESSED BITMAPS ARE STREAMED FROM PROXY SERVER TO CLIENT
118
VECTORIZED CONTENT AND BITMAP CONTENT PROCESSED AND SCALED USING THIN CLIENT TO RENDER CONTENT ON CLIENT DEVICE
120

FIG. 2B

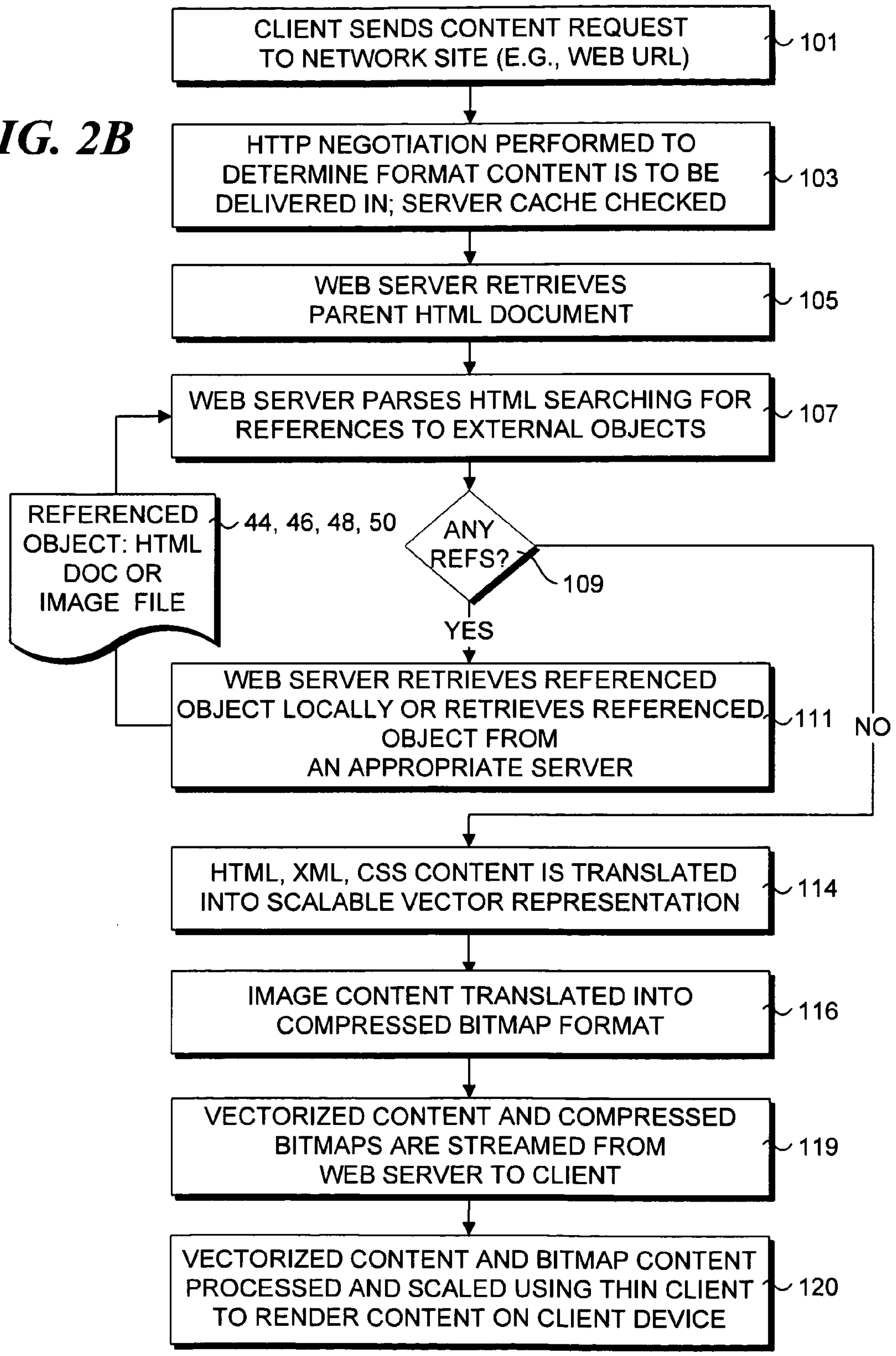
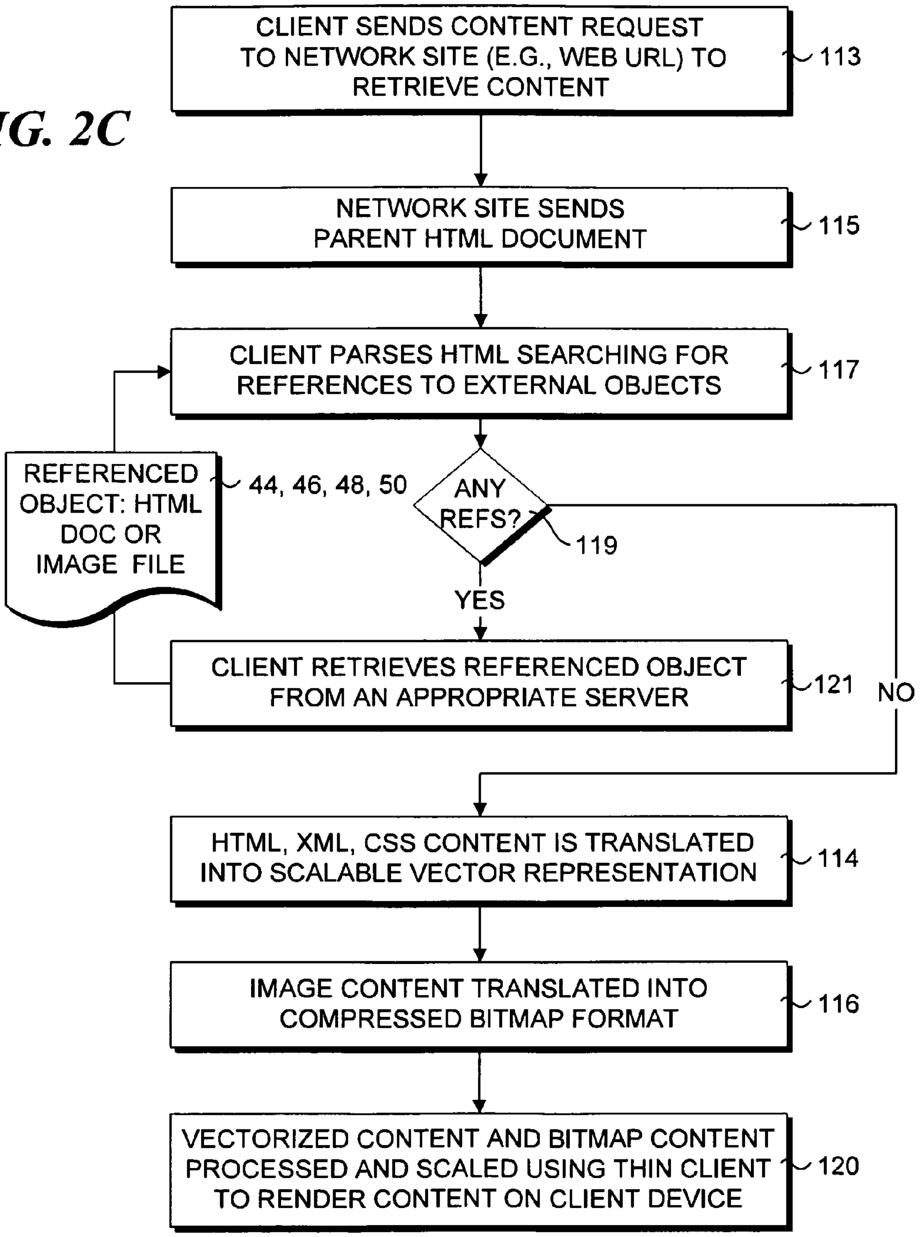

CLIENT SENDS CONTENT REQUEST TO NETWORK SITE (E.G., WEB URL)
101
HTTP NEGOTIATION PERFORMED TO DETERMINE FORMAT CONTENT IS TO BE DELIVERED IN; SERVER CACHE CHECKED
103
WEB SERVER RETRIEVES PARENT HTML DOCUMENT
105
WEB SERVER PARSES HTML SEARCHING FOR REFERENCES TO EXTERNAL OBJECTS
107
REFERENCED OBJECT: HTML DOC OR IMAGE FILE
44, 46, 48, 50
ANY REFS?
109
YES
NO
WEB SERVER RETRIEVES REFERENCED OBJECT LOCALLY OR RETRIEVES REFERENCED OBJECT FROM AN APPROPRIATE SERVER
111
HTML, XML, CSS CONTENT IS TRANSLATED INTO SCALABLE VECTOR REPRESENTATION
114
IMAGE CONTENT TRANSLATED INTO COMPRESSED BITMAP FORMAT
116
VECTORIZED CONTENT AND COMPRESSED BITMAPS ARE STREAMED FROM WEB SERVER TO CLIENT
119
VECTORIZED CONTENT AND BITMAP CONTENT PROCESSED AND SCALED USING THIN CLIENT TO RENDER CONTENT ON CLIENT DEVICE
120

FIG. 2C


CLIENT SENDS CONTENT REQUEST TO NETWORK SITE (E.G., WEB URL) TO RETRIEVE CONTENT
113
NETWORK SITE SENDS PARENT HTML DOCUMENT
115
CLIENT PARSES HTML SEARCHING FOR REFERENCES TO EXTERNAL OBJECTS
117
REFERENCED OBJECT: HTML DOC OR IMAGE FILE
44, 46, 48, 50
ANY REFS?
119
YES
NO
CLIENT RETRIEVES REFERENCED OBJECT FROM AN APPROPRIATE SERVER
121
HTML, XML, CSS CONTENT IS TRANSLATED INTO SCALABLE VECTOR REPRESENTATION
114
IMAGE CONTENT TRANSLATED INTO COMPRESSED BITMAP FORMAT
116
VECTORIZED CONTENT AND BITMAP CONTENT PROCESSED AND SCALED USING THIN CLIENT TO RENDER CONTENT ON CLIENT DEVICE
120

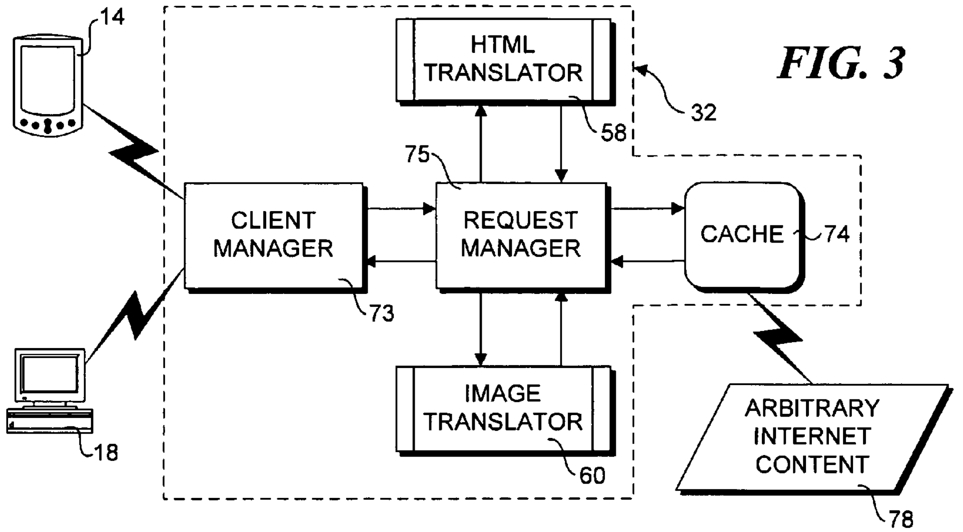

14
HTML TRANSLATOR
32
58
75
CLIENT MANAGER
REQUEST MANAGER
CACHE
74
73
IMAGE TRANSLATOR
ARBITRARY INTERNET CONTENT
18
60
78

FIG. 3

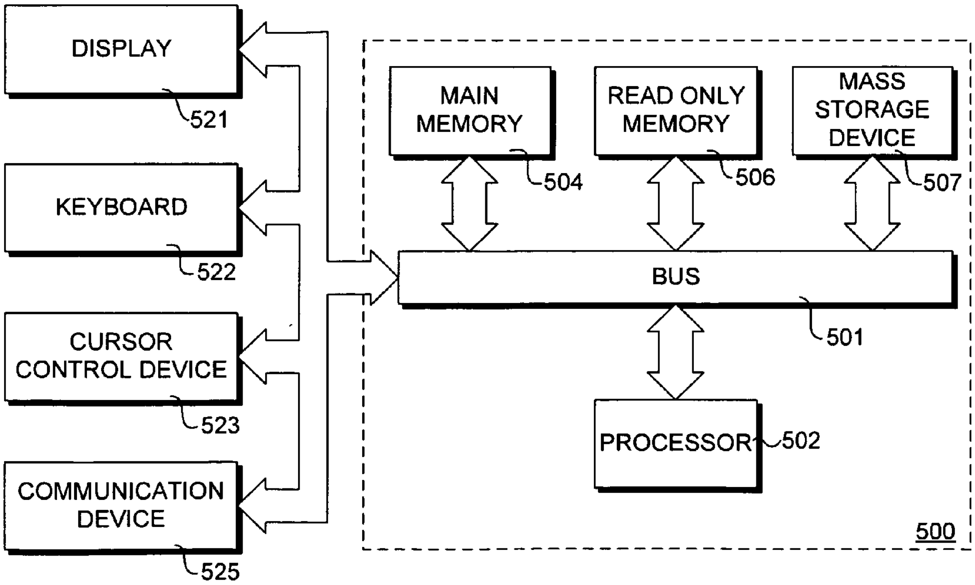
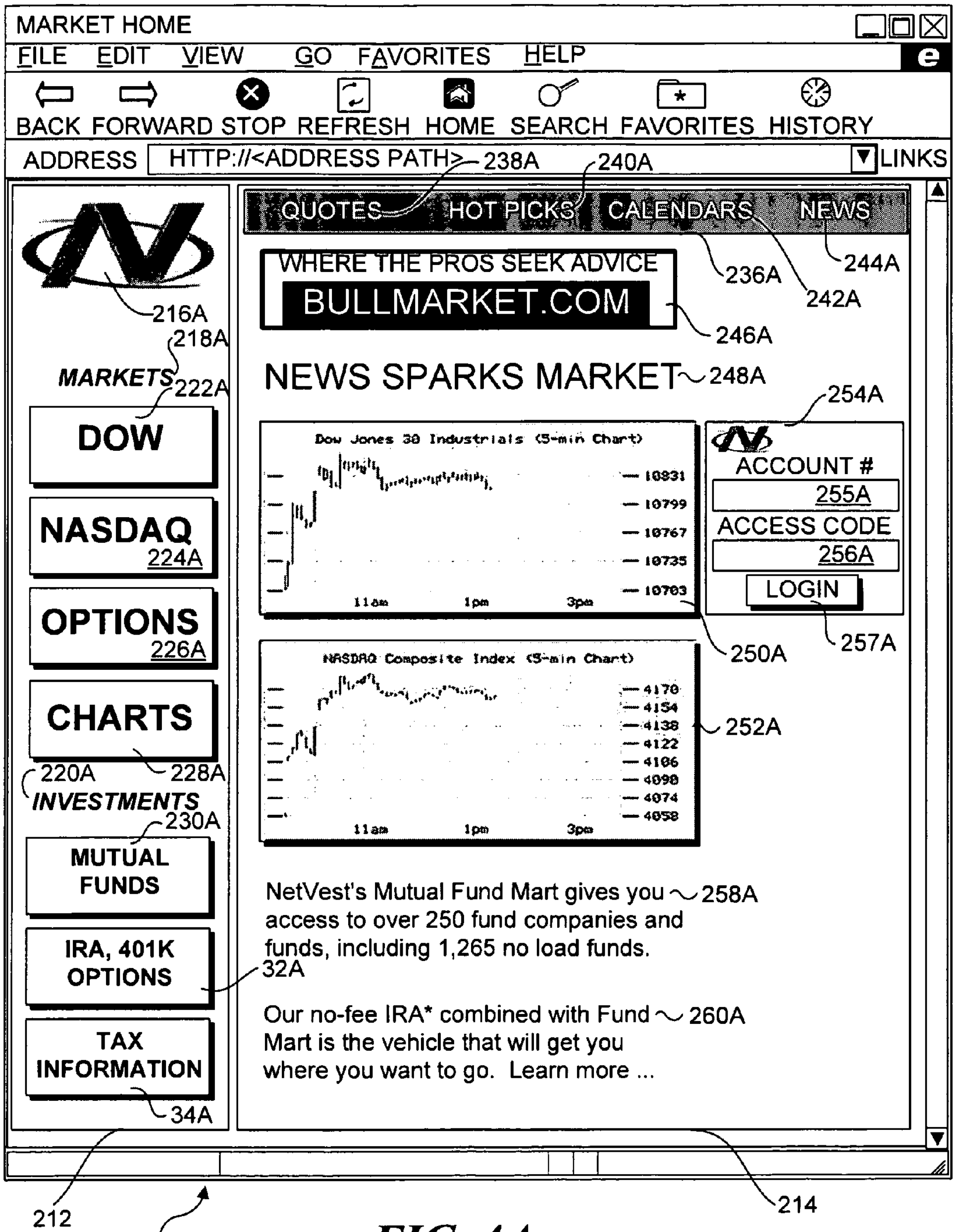

DISPLAY
521
KEYBOARD
522
CURSOR CONTROL DEVICE
523
COMMUNICATION DEVICE
525
MAIN MEMORY
504
READ ONLY MEMORY
506
MASS STORAGE DEVICE
507
BUS
501
PROCESSOR
502
500

FIG. 10



NEWS SPARKS MARKET

NetVest's Mutual Fund Mart gives you access to over 250 fund companies and funds, including 1,265 no load funds.

Our no-fee IRA* combined with Fund Mart is the vehicle that will get you where you want to go. Learn more ...

FIG. 4A

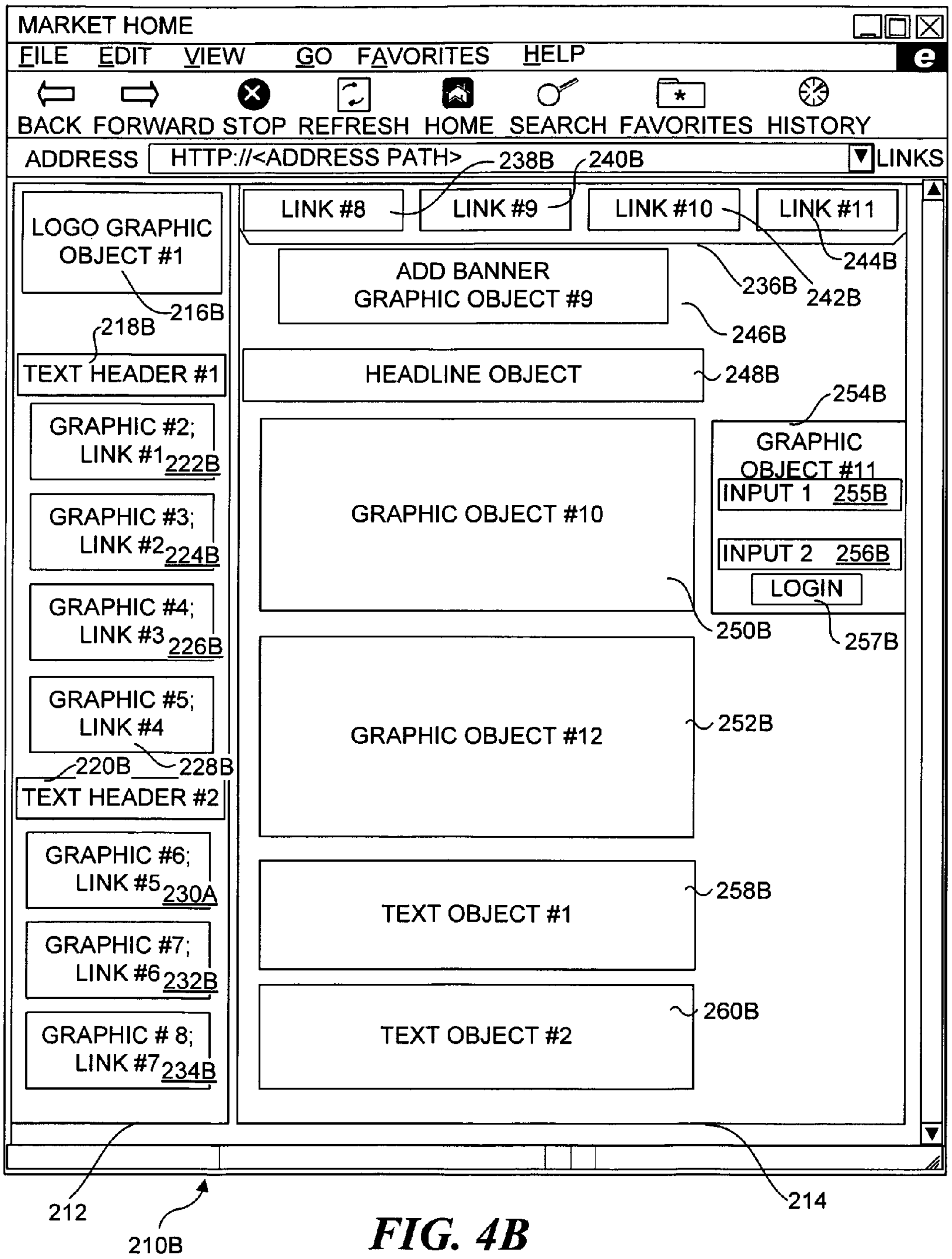

MARKET HOME
FILE EDIT VIEW GO FAVORITES HELP
BACK FORWARD STOP REFRESH HOME SEARCH FAVORITES HISTORY
ADDRESS
HTTP://<ADDRESS PATH>
LINKS
238B
240B
LINK #8
LINK #9
LINK #10
LINK #11
244B
236B
242B
LOGO GRAPHIC OBJECT #1
216B
218B
ADD BANNER GRAPHIC OBJECT #9
246B
TEXT HEADER #1
HEADLINE OBJECT
248B
254B
GRAPHIC #2; LINK #1 222B
GRAPHIC OBJECT #10
GRAPHIC OBJECT #11
INPUT 1 255B
INPUT 2 256B
LOGIN
GRAPHIC #3; LINK #2 224B
GRAPHIC #4; LINK #3 226B
250B
257B
GRAPHIC #5; LINK #4
GRAPHIC OBJECT #12
252B
220B
228B
TEXT HEADER #2
GRAPHIC #6; LINK #5 230A
TEXT OBJECT #1
258B
GRAPHIC #7; LINK #6 232B
GRAPHIC # 8; LINK #7 234B
TEXT OBJECT #2
260B
212
210B
214

FIG. 4B

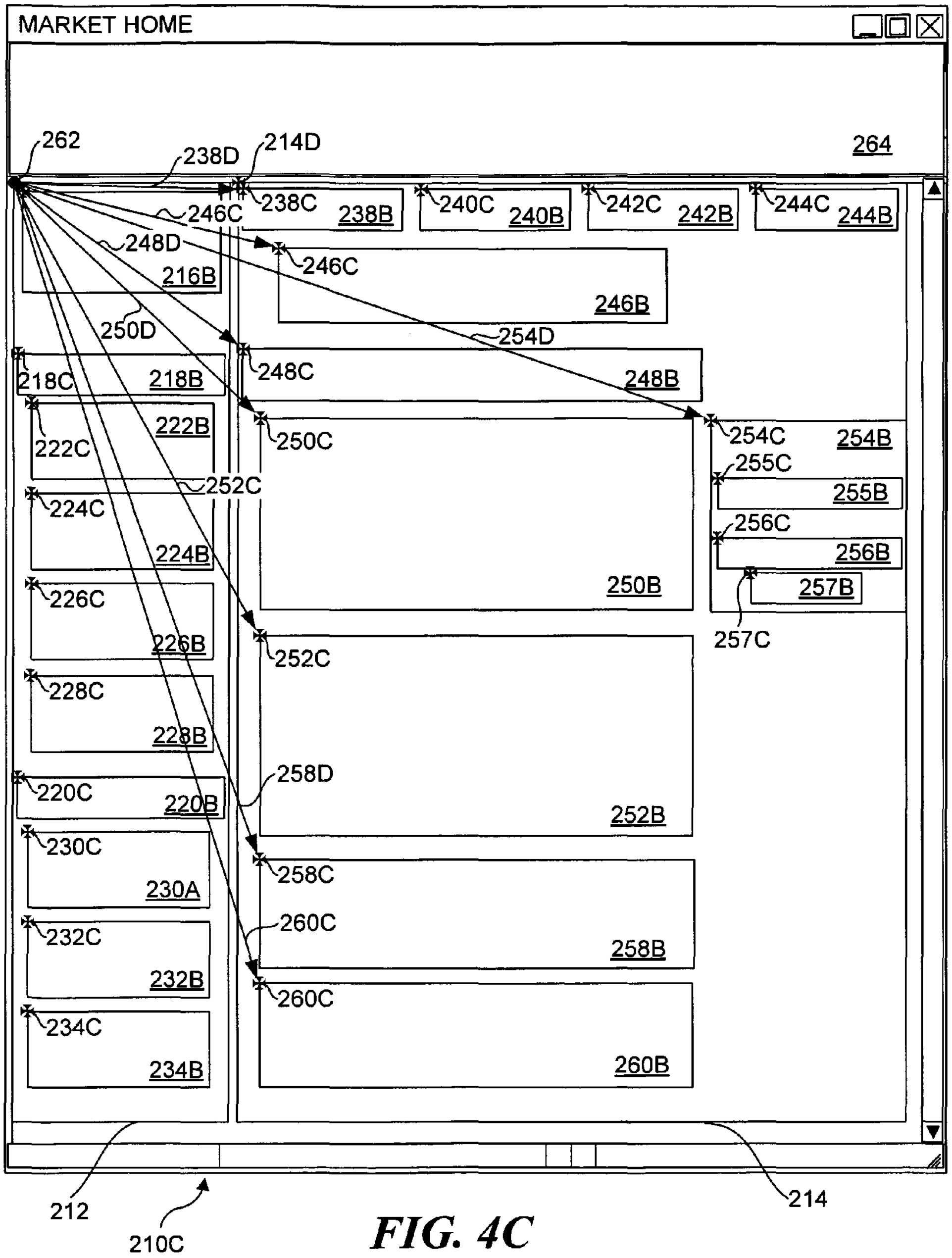

MARKET HOME
262
238D
214D
264
246C
238C
238B
240C
240B
242C
242B
244C
244B
248D
216B
246C
246B
250D
254D
218C
218B
248C
248B
222B
222C
250C
254C
254B
255C
255B
252C
224C
256C
256B
224B
257B
250B
226C
257C
226B
252C
228C
228B
258D
220C
220B
252B
230C
230A
258C
260C
232C
232B
258B
234C
260C
234B
260B
212
214
210C

FIG. 4C

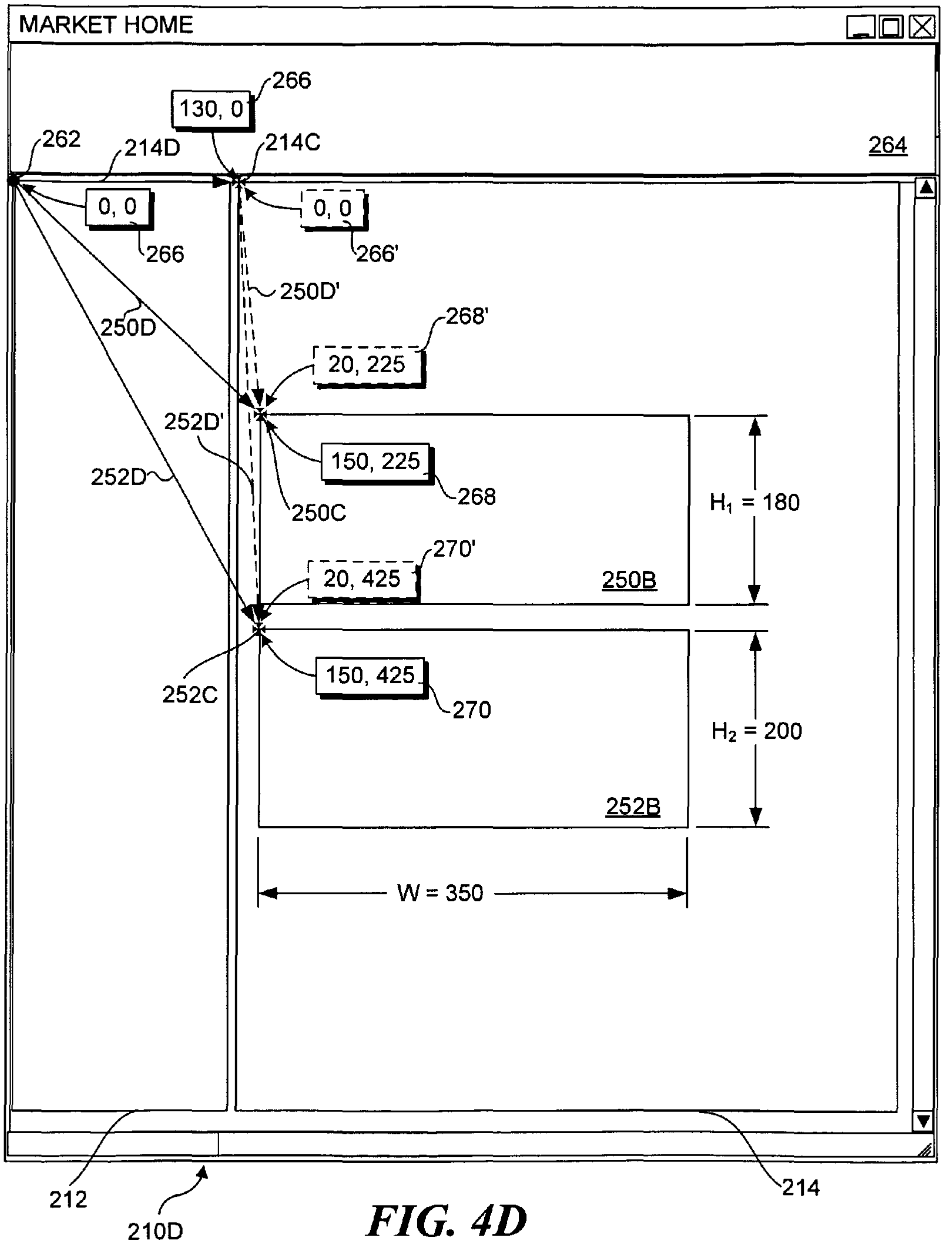
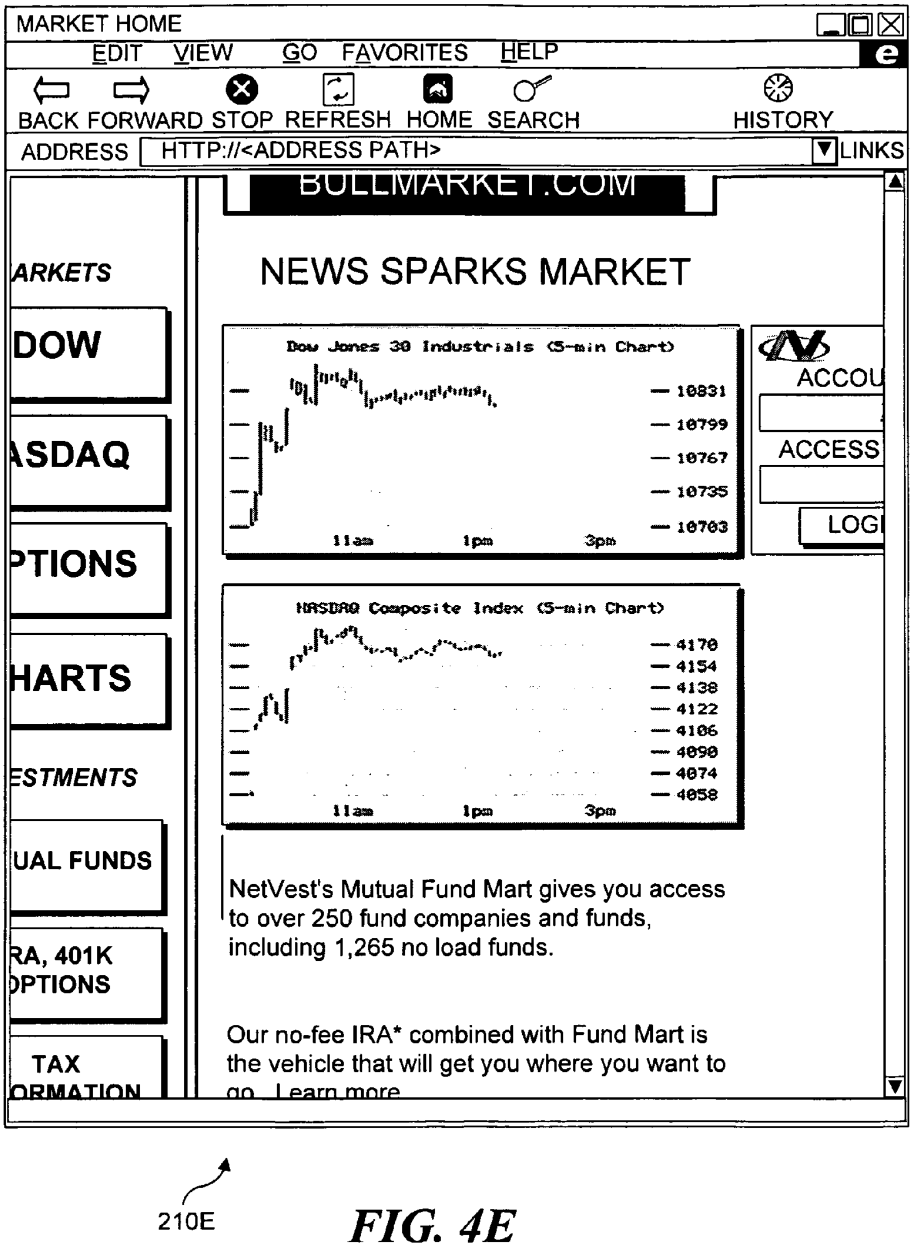

MARKET HOME
266
130, 0
262
214D
214C
264
0, 0
266
0, 0
266'
250D'
250D
268'
20, 225
252D'
150, 225
252D
268
250C
H1 = 180
270'
20, 425
250B
252C
150, 425
270
H2 = 200
252B
W = 350
212
210D
214

FIG. 4D



NetVest's Mutual Fund Mart gives you access to over 250 fund companies and funds, including 1,265 no load funds.

Our no-fee IRA* combined with Fund Mart is the vehicle that will get you where you want to go. Learn more

FIG. 4E

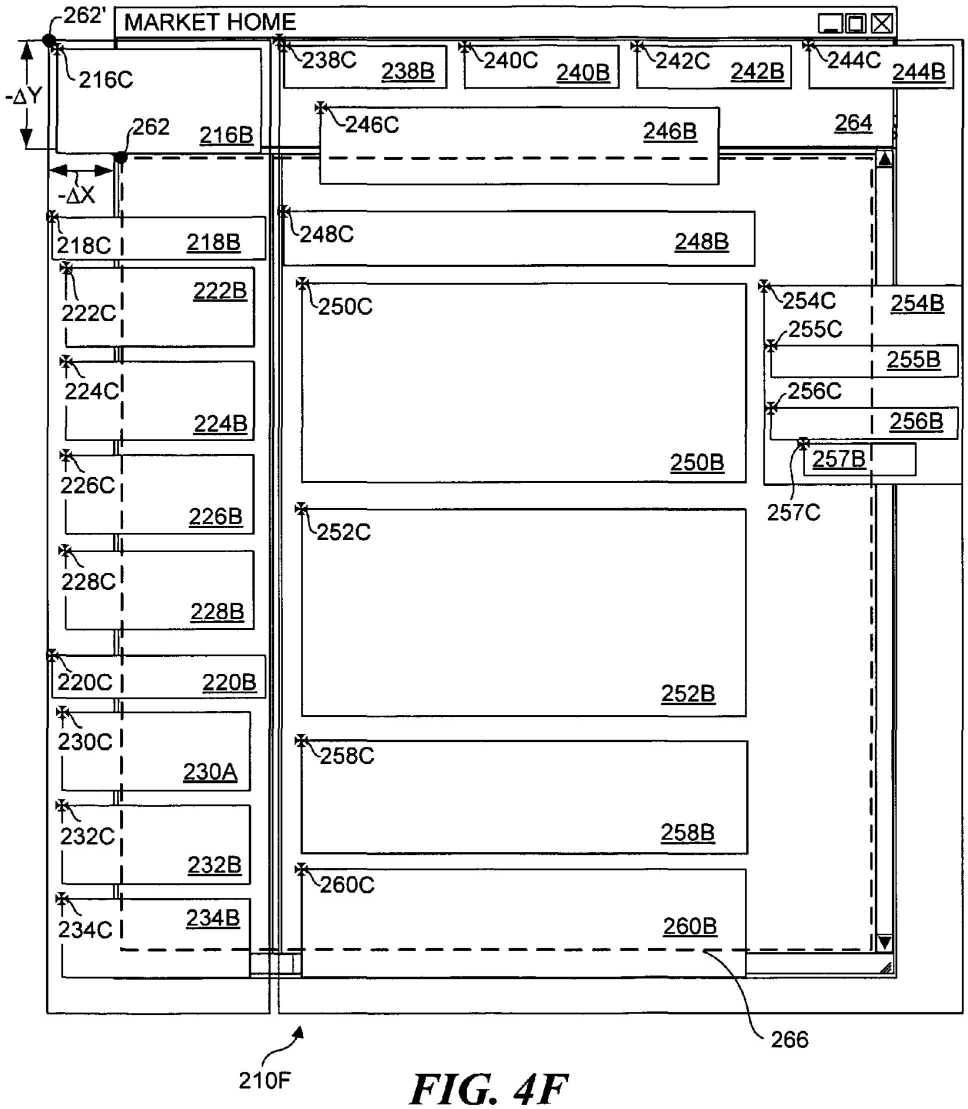

262'
MARKET HOME
216C
-ΔY
262
216B
-ΔX
238C
238B
240C
240B
242C
242B
244C
244B
246C
246B
264
218C
218B
248C
248B
222C
222B
250C
254C
254B
255C
255B
256C
256B
257B
257C
224C
224B
250B
226C
226B
252C
228C
228B
220C
220B
252B
230C
230A
258C
258B
232C
232B
260C
234C
234B
260B
266
210F

FIG. 4F

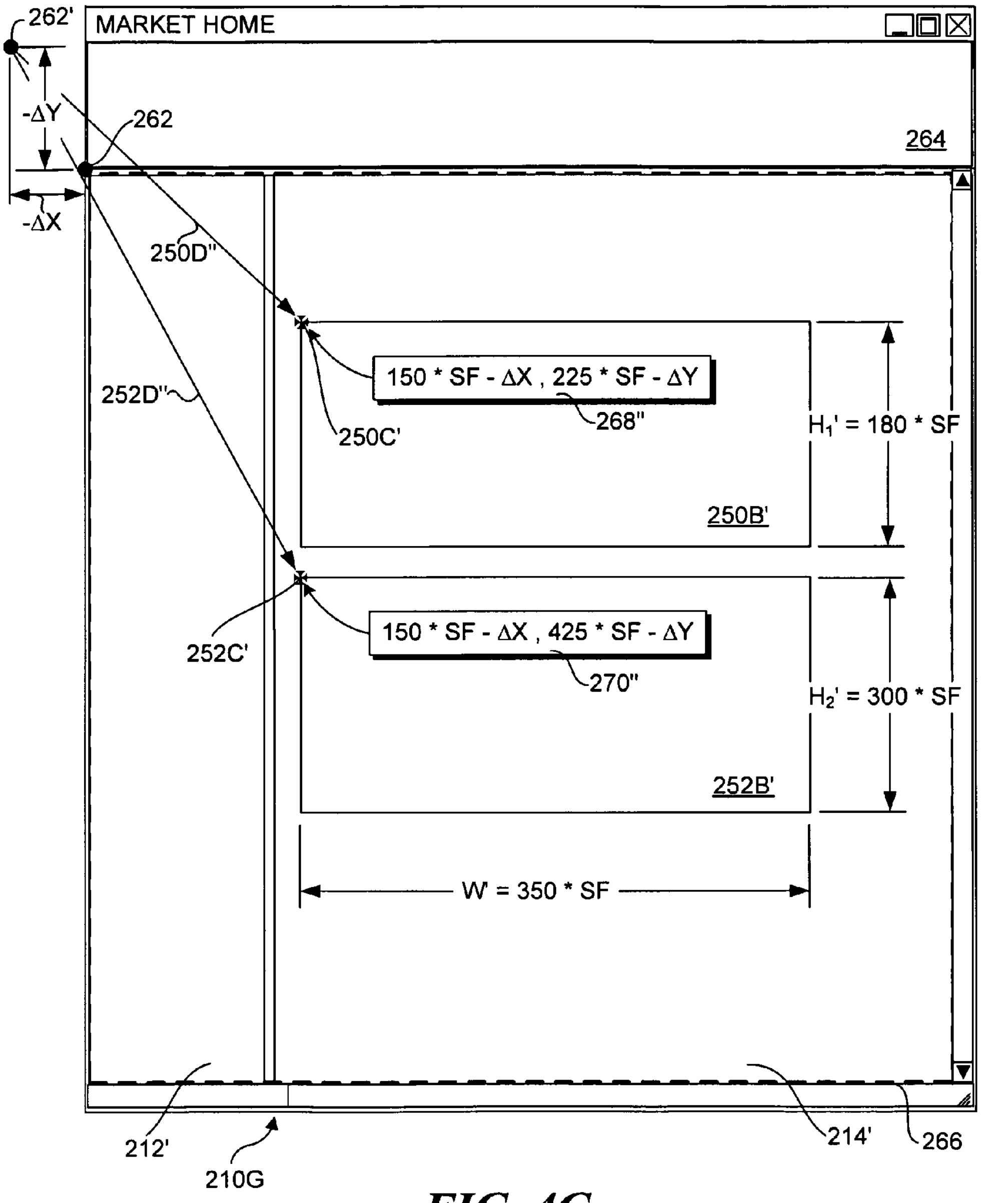

262'
MARKET HOME
-ΔY
262
264
-ΔX
250D"
150 * SF - ΔX , 225 * SF - ΔY
252D"
268"
250C'
H₁' = 180 * SF
250B'
150 * SF - ΔX , 425 * SF - ΔY
252C'
270"
H₂' = 300 * SF
252B'
W' = 350 * SF
212'
210G
214'
266

FIG. 4G

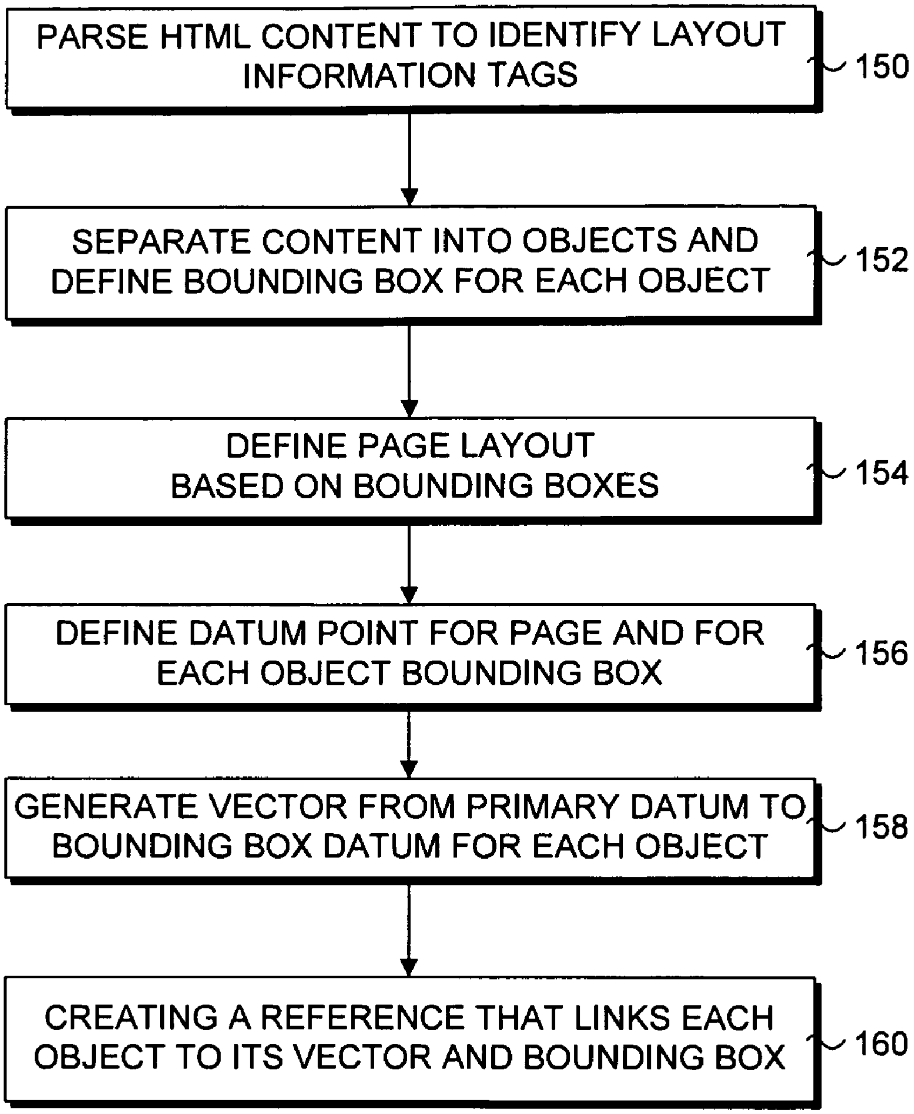

PARSE HTML CONTENT TO IDENTIFY LAYOUT INFORMATION TAGS
150
SEPARATE CONTENT INTO OBJECTS AND DEFINE BOUNDING BOX FOR EACH OBJECT
152
DEFINE PAGE LAYOUT BASED ON BOUNDING BOXES
154
DEFINE DATUM POINT FOR PAGE AND FOR EACH OBJECT BOUNDING BOX
156
GENERATE VECTOR FROM PRIMARY DATUM TO BOUNDING BOX DATUM FOR EACH OBJECT
158
CREATING A REFERENCE THAT LINKS EACH OBJECT TO ITS VECTOR AND BOUNDING BOX
160

FIG. 5

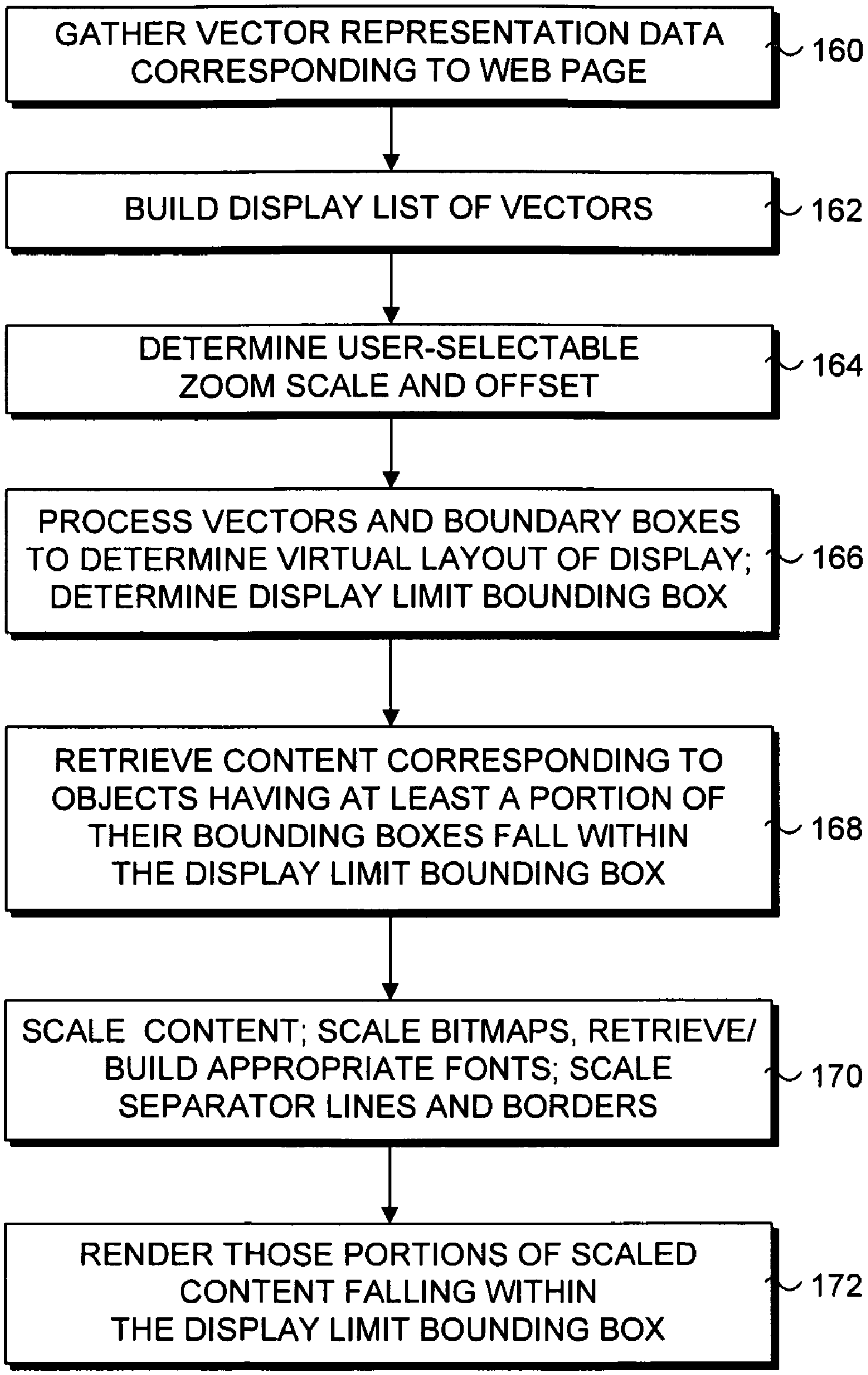

GATHER VECTOR REPRESENTATION DATA CORRESPONDING TO WEB PAGE
160
BUILD DISPLAY LIST OF VECTORS
162
DETERMINE USER-SELECTABLE ZOOM SCALE AND OFFSET
164
PROCESS VECTORS AND BOUNDARY BOXES TO DETERMINE VIRTUAL LAYOUT OF DISPLAY; DETERMINE DISPLAY LIMIT BOUNDING BOX
166
RETRIEVE CONTENT CORRESPONDING TO OBJECTS HAVING AT LEAST A PORTION OF THEIR BOUNDING BOXES FALL WITHIN THE DISPLAY LIMIT BOUNDING BOX
168
SCALE CONTENT; SCALE BITMAPS, RETRIEVE/ BUILD APPROPRIATE FONTS; SCALE SEPARATOR LINES AND BORDERS
170
RENDER THOSE PORTIONS OF SCALED CONTENT FALLING WITHIN THE DISPLAY LIMIT BOUNDING BOX
172

FIG. 6

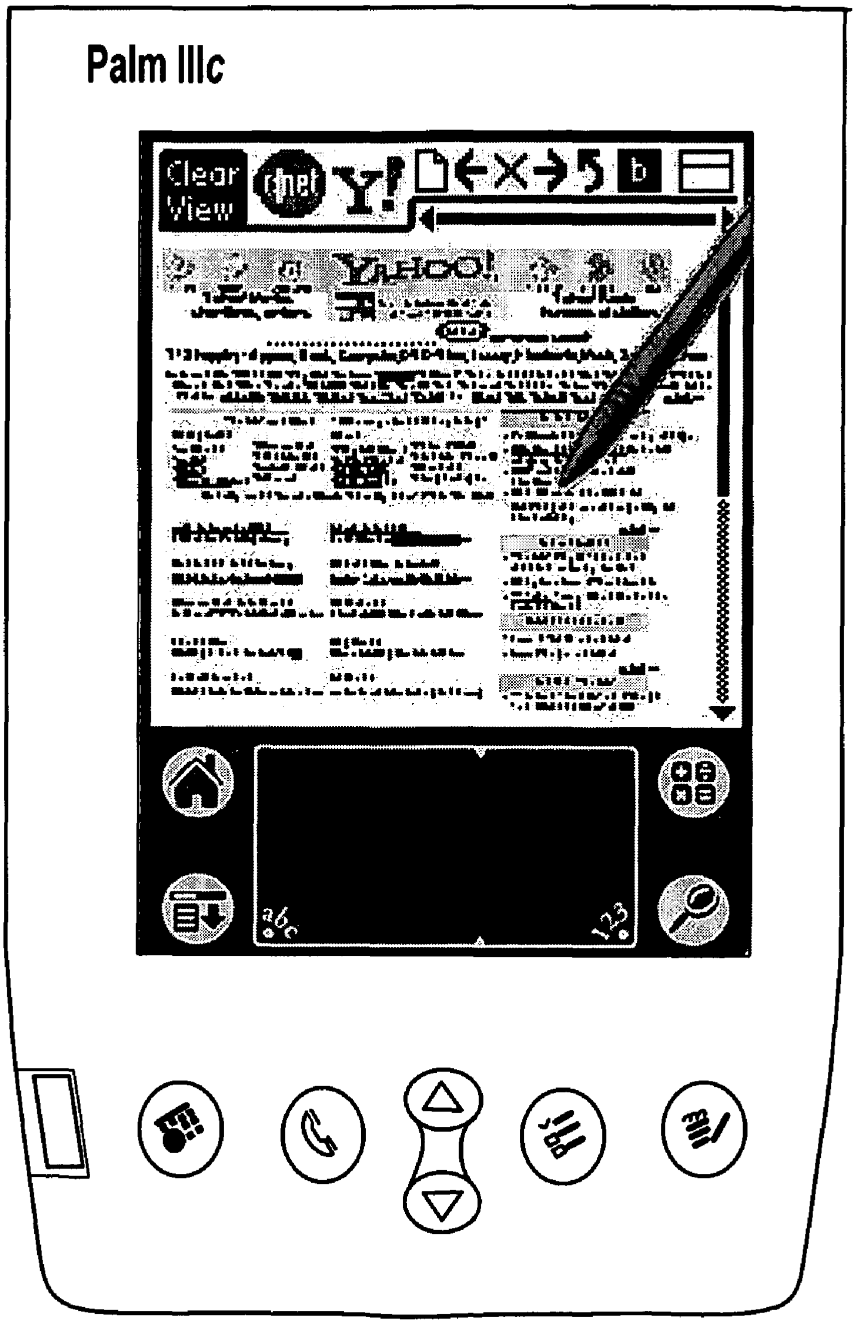
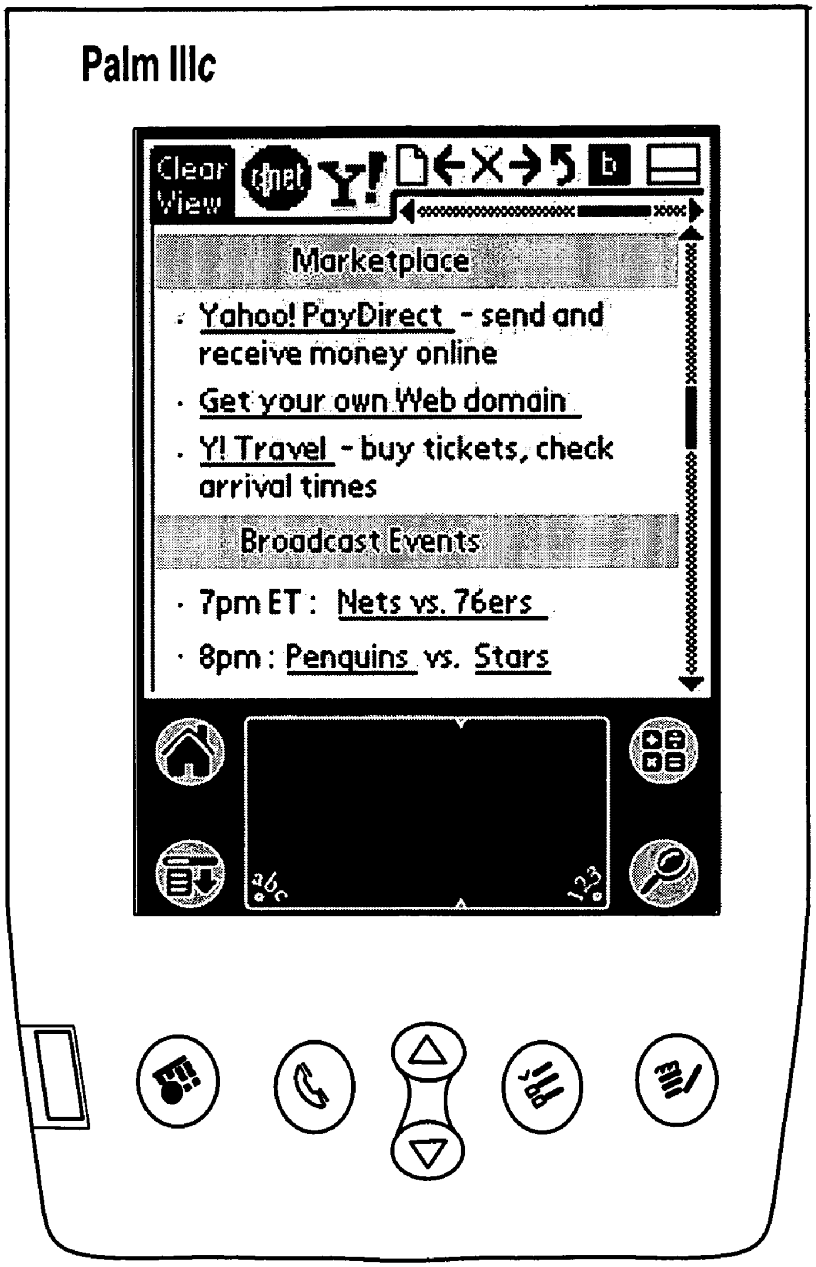


FIG. 7A



Marketplace

- Yahoo! PayDirect - send and receive money online
- Get your own Web domain
- Y! Travel - buy tickets, check arrival times

Broadcast Events

- 7pm ET : Nets vs. 76ers
- 8pm : Penguins vs. Stars

FIG. 7B

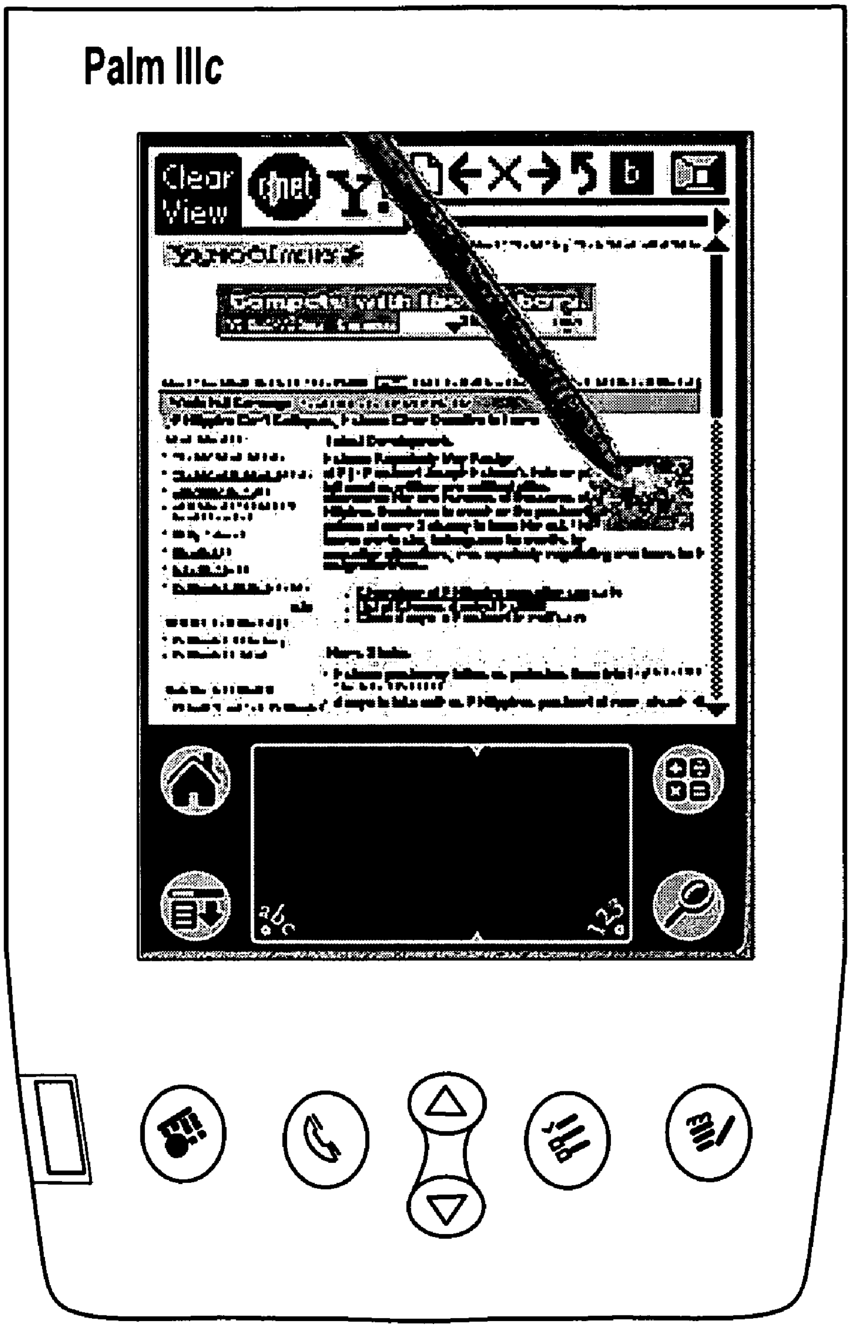


FIG. 8A

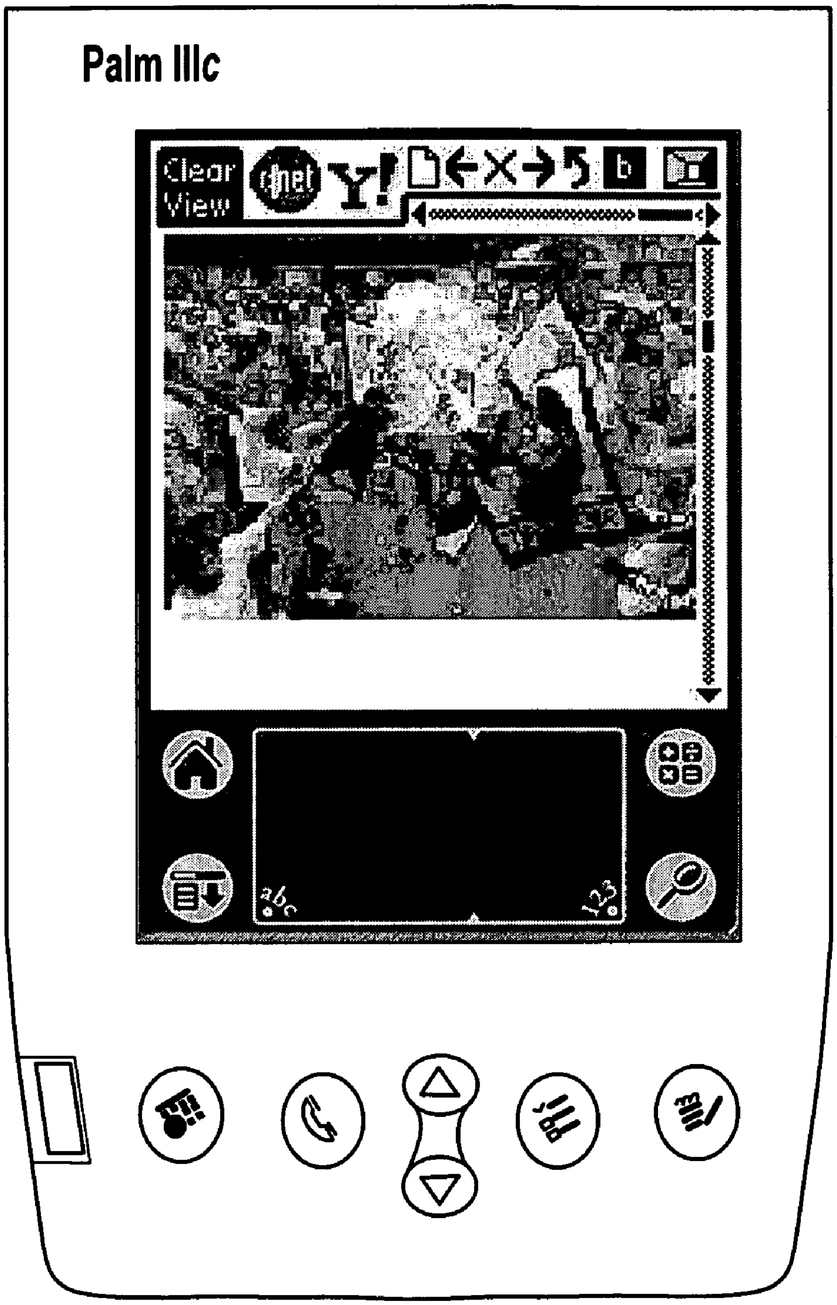


FIG. 8B

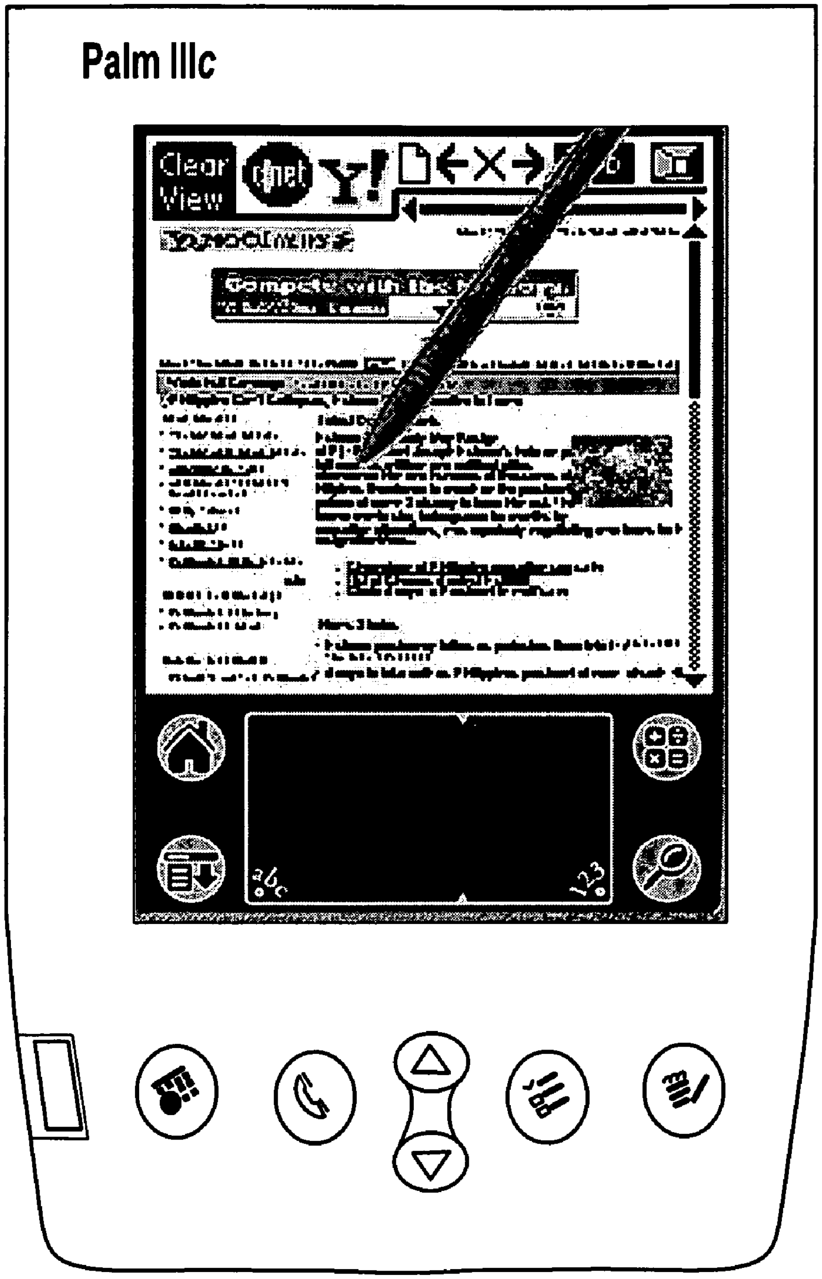
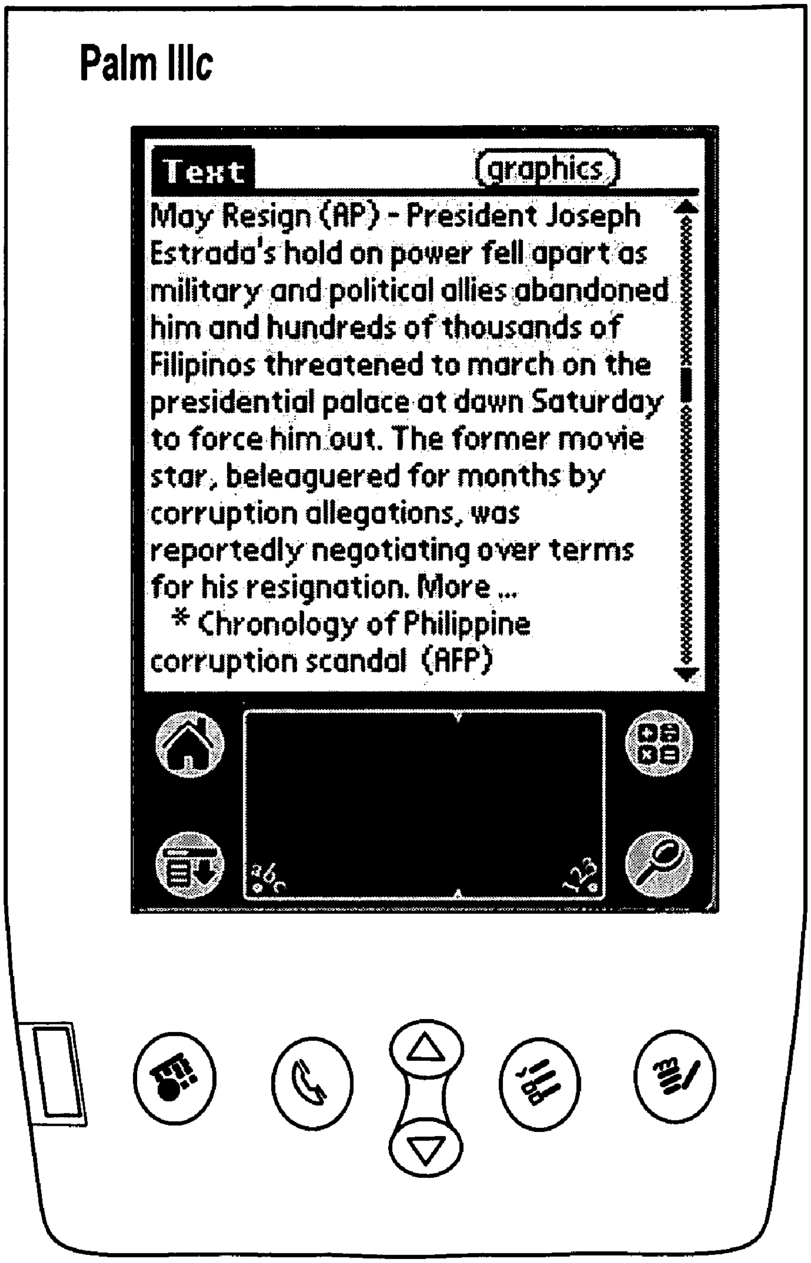


FIG. 9A



May Resign (AP) - President Joseph Estrada's hold on power fell apart as military and political allies abandoned him and hundreds of thousands of Filipinos threatened to march on the presidential palace at dawn Saturday to force him out. The former movie star, beleaguered for months by corruption allegations, was reportedly negotiating over terms for his resignation. More ...

* Chronology of Philippine corruption scandal (AFP)

FIG. 9B

SCALABLE DISPLAY OF INTERNET CONTENT ON MOBILE DEVICES

RELATED APPLICATIONS

The present application is a Divisional Application of U.S. application Ser. No. 09/878,097, filed Jun. 8, 2001, now U.S. Pat. No. 7,210,099, which is a Continuation-in-Part of U.S. Non-Provisional application Ser. No. 09/828,511, filed Apr. 7, 2001, now abandoned, entitled "RESOLUTION INDEPENDENT VECTOR DISPLAY OF INTERNET CONTENT," the benefit of the filing date of which is claimed under 35 U.S.C. § 120. This application further claims the benefit of the filing dates of U.S. Provisional Application No. 60/211,019, filed Jun. 12, 2000, entitled "METHOD AND SYSTEM FOR RESOLUTION INDEPENDENT DISPLAY OF HTML AND XML CONTENT" and U.S. Provisional Application No. 60/217,345, filed Jul. 11, 2000, entitled "METHOD AND SYSTEM FOR SELECTION, RETRIEVAL, AND CONVERSION OF COMPUTER CONTENT TO VECTOR FORMAT FOR RESOLUTION INDEPENDENT DISPLAY," under 35 U.S.C. § 119(e).

The present application is also related to U.S. application Ser. No. 11/735,477 filed on Apr. 15, 2007, U.S. application Ser. No. 11/735,482 filed on Apr. 15, 2007, U.S. application Ser. No. 11/738,486 filed on Apr. 21, 2007, and U.S. application Ser. No. 11/738,932 filed on Apr. 23, 2007, each of which are continuations of U.S. application Ser. No. 09/878,097, now U.S. Pat. No. 7,210,099.

COPYRIGHT NOTICE

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates generally to translation of Internet and World Wide Web content to scalable vector representation. More particularly, the invention relates to apparatus and methods for zoom enabling the display of content in an Internet information browser by retrieving and translating HyperText Markup Language (HTML), extensible Markup Language (XML), and other Internet content to vector representations of that content.

2. Description of the Related Art

Text only Internet information browsers began as a project at the CERN, European Organization for Nuclear Research, facility in Geneva Switzerland. From its inception the intent was to provide a mesh or web of access to data with a common user interface. Browsers moved from the academic environment when NCSA, the National Center for Supercomputing Applications at the University of Illinois in Urbana-Champaign developed Mosaic, an Internet information browser and World Wide Web client.

Internet content is stored in multiple file formats. These formats include HTML (Hyper Text Markup Language) and XML (extended Markup Language) as well as graphic file format GIF (Graphics Interchange Format) and JPEG (Joint Photographic Experts Group). These four file formats constitute the majority of Internet content. Font size and resizing display area for content can alter the size of the display of Internet content in existing browsers. The majority of Internet content displays as a flat single resolution with no browser support for zoom.

Much of the Internet content has been designed for display on desktop computers with a single target resolution. Even though HTML has the ability to adapt to changes in screen resolution, major Internet content providers have chosen to create their Web pages using fixed resolution structures, such as tables. This gives them the ability to control the look and feel of their Web sites. This fixed resolution approach has evolved to the point that the fixed resolution layout of Web pages has become the most common method to brand or uniquely identify Web sites. While this fixed resolution approach is good for site branding and product differentiation it does present a daunting technical problem for display of Internet content (designed for desktop computers) on small screen, low resolution, or different aspect ratio devices, such as cell phones and hand held computers.

BRIEF SUMMARY OF THE INVENTION

In accordance with aspects of the invention, mobile devices enabled to support resolution-independent scalable display of Internet (Web) content to allow Web pages to be scaled (zoomed) and panned for better viewing on smaller screen sizes are disclosed. The mobile devices employ novel processing of original Web content, including HTML-based content, XML, cascade style sheets, etc. to generate scalable content. The scalable content and/or data derived therefrom are then employed to enable the Web content to be rapidly rendered, zoomed, and panned. Moreover, the rendered displays provide substantially the same or identical layout as the original Web page, enabling users to easily navigate to selected content and features on familiar Web pages. Display lists may also be employed to provide further enhancements in rendering speed. Additionally, hardware-based programmed logic may also be employed to facilitate various operations.

According to further aspects, some mobile devices may employ touch-sensitive display screens that enable users to provide various inputs to control display of content within Web pages. Exemplary user inputs include tap-based inputs to selectively zoom in on columns, images, and paragraphs. Users can also define a window to zoom in on via the touch-sensitive display.

According to additional aspects of the invention, methods and software for enabling support for resolution-independent scalable display of Web content are provided. The methods and software enable users of various devices, from handheld devices with small screens, to desktop PC's and laptops, to very large screen devices, to view and interact with Web pages in a manner independent of the screen resolution of such device's built-in or associated display, while maintaining the look and feel of browsing such pages with a conventional desktop browser. Thus, users of various devices having different screen resolutions are enabled to browse Web pages from among literally billions of Web pages while providing a full Web browsing experience.

Other features of the present invention will be apparent from the accompanying drawings and from the detailed description that follows.

BRIEF DESCRIPTION OF THE DRAWINGS

The appended claims set forth the features of the invention with particularity. The invention, together with its advantages, may be best understood from the following detailed description taken in conjunction with the accompanying drawings of which:

FIG. **1**A is a block schematic diagram illustrating a first exemplary system infrastructure in accordance with the present invention in which content translation services are performed by a third-party proxy service that translates content requested from a client that is retrieved from one or more network resources into a scalable vector representation and delivers the translated content to the client;

FIG. **1**B is a block schematic diagram illustrating a second exemplary system infrastructure in which the translation of content is performed at a content provider's web site and delivered directly to the requesting client;

FIG. **1**C is a block schematic diagram illustrating a third exemplary system infrastructure in which content received from one or more network sources is translated into a scalable vector representation at the client;

FIG. **2**A is a flowchart illustrating how data is retrieved, processed and transferred in accordance with the system infrastructure of FIG. **1**A;

FIG. **2**B is a flowchart illustrating how data is retrieved, processed and transferred in accordance with the system infrastructure of FIG. **1**B;

FIG. **2**C is a flowchart illustrating how data is retrieved, processed and transferred in accordance with the system infrastructure of FIG. **1**C;

FIG. **3** is a block schematic diagram illustrating an exemplary architecture corresponding to the proxy server of FIG. **1**A;

FIG. **4**A is a representation of an exemplary web page has displayed on a conventional browser;

FIG. **4**B is a schematic diagram illustrates various objects that are generated based on the HTML code of the web page of FIG. **4**A;

FIG. **4**C is a schematic diagram illustrating a set of vectors and bounding boxes corresponding to the objects generated in FIG. **4**B;

FIG. **4**D is a schematic diagram illustrating how various vectors and bounding boxes may be defined in accordance with the invention;

FIG. **4**E is a representation of the web page of FIG. **4**A after it has been offset and scaled in accordance with the invention;

FIG. **4**F is a schematic diagram illustrating new datum points and bounding boxes corresponding to the scaled and offset web page;

FIG. **4**G is a schematic diagram illustrating new vectors and bounding box parameters for a pair of objects in the scaled and offset web page;

FIG. **5** is a flowchart illustrating the logic used by the invention when translating content into a scalable vector representation of that content;

FIG. **6** is a flowchart illustrating client-side operations that are performed to create a rendered display page based on the translated content the client receives and user-input;

FIGS. **7**A and **7**B are representations of a nominal and a zoomed in column view of an exemplary web page as they might appear on a Palm device;

FIGS. **8**A and **8**B are representation of nominal and zoomed in view of an exemplary graphic image as they might appear on the Palm device;

FIGS. **9**A and **9**B are representations of a nominal and zoomed in view of a text portion of a web page as they might appear on the Palm device; and

FIG. **10** illustrates an exemplary computer system that may be used for implementing various aspects of embodiments of the invention.

DETAILED DESCRIPTION OF THE INVENTION

Apparatus and methods are described for creating resolution independent vector display of Internet content to allow it to be scaled (zoomed) larger and smaller for better viewing or to fit any resolution or screen size. In addition, infrastructure and methods are provided for delivering such content to clients.

In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without some of these specific details. In other instances, well-known structures and devices are shown in block diagram form.

The present invention includes various operations, which will be described below. The operations of the present invention may be performed by hardware components or may be embodied in machine-executable instructions, which may be used to cause a general-purpose or special-purpose processor or logic circuits programmed with the instructions to perform the operations. Alternatively, the operations may be performed by a combination of hardware and software.

The present invention may be provided as a computer program product that may include one or more machine-readable mediums having stored thereon instructions, which may be used to program a computer (or other electronic devices) to perform a process according to the present invention. The machine-readable medium may include, but is not limited to, floppy diskettes, optical disks, CD-ROMs, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, magnetic or optical cards, flash memory, or other type of media/machine-readable medium suitable for storing electronic instructions. Moreover, the present invention may also be downloaded as a computer program product, wherein the program may be transferred from a remote computer (e.g., a server) to a requesting computer (e.g., a client) by way of data signals embodied in a carrier wave or other propagation medium via a communication link (e.g., a modem or network connection). Accordingly, herein, a carrier wave shall be regarded as comprising a machine-readable medium.

Client Overview

According to one embodiment, an ultra-thin client-side viewer provides the graphics, linking, caching, and function handling capabilities necessary for extending the web to almost any platform. It is designed as a lightweight browser (micro-browser) running directly on device operating systems. In alternative embodiments, the client-side viewer may be deployed as a standard browser plug-in, or Java applet for extending browser functionality. In one embodiment, the client-side viewer attains its small size and efficiency by taking advantage of the power of SVF (Simple Vector Format) to describe almost any current web content. SVF files can be handled with a tiny fraction of the client code required by normal web browsers because current browsers must interpret a large and growing number of file types and their idiosyncrasies. SVF was originally designed to handle a superset of the most commonly used file formats in the complex world of CAD. It can accommodate not only new graphical functions, but the storage and transfer of almost any foreseeable new functional capability. SVF has been under consideration by the W3C (World Wide Web Consortium) for adoption as a standard for vector content on the World Wide Web.

By working tightly with a server-side content translator, web content and functionality can be passed seamlessly to the end user platform without any degradation in the look or feel of the output. In addition, because the resulting file graphics are handled as vectors, the end user can control real time changes in the size of text and graphics as well as what portion of the file is viewable in the display. This "zoom and pan" capability, familiar to CAD and other vector content software users, adds dramatically to the usability of non-standard display sizes. For very small displays, real time zooming and panning allows the user to see graphics and text at sizes that make them easily readable, and then "back up" to view an entire page for context or pan in any direction for navigation. Because the client-side viewer manipulates vectors, there is no loss in quality as the display is zoomed. The graphics rendering engine within the client is so efficient that file manipulation happens in a fraction of a second. There is no perceptible wait for the user as the file is resized, or the window is repositioned. Content created for one display resolution now can be sized, real time, for any other display without degradation. Besides making small displays eminently usable, this technology extends web content into some surprising new arenas. For example, it enables normal desktop displays to be effective for individuals with visual impairment, or for content designed for 640×480 standard PC monitors to be shown without degradation on web billboards now appearing in cities like Seattle and San Francisco.

With a client of such extraordinary power packed in a tiny footprint, end user device manufacturers can free up valuable memory space for pre-fetching, caching and pre-loading content, dramatically improving performance for use in low bandwidth and portable applications. In the example of a wireless handheld device where expensive flash memory must be used instead of more cost effective bulk storage technology, the difference between consuming 10's of megabytes of flash memory with a standard browser versus running the client-side viewer described herein is dramatic.

Those "saved" megabytes of memory are now available for impressive interfaces, caching of often used content, and pre-fetching of intelligently selected linked files or pre-loading of content for targeted applications. For example, in a mapping application, the map tiles surrounding the viewed map could be downloaded and stored while the user was working with the initial tile, enabling an experience remarkably free from the current frustrations of waiting for a new map to be transferred for even the smallest change in magnification or coverage. If the user knows ahead of time what city they will visit on a business trip, maps and additional travel information in great detail could also be pre-loaded using a high bandwidth connection at home or in the office before heading out to shop or conduct business in the city. Additionally, SVF is a more efficient way to store web content. Resulting content files are reduced in size by anywhere from 20 to 80 percent over their source. SVF is also very compressible. With target file size reduction in the range of 90%, SVF files can take up as little as 1/10th the space of the web files in current use. This means that pre-translated content can be moved up to 10 times the rate of current web pages, and as much as 10 times as many pages, maps, stock charts, etc. can be stored for instant retrieval on the hand held platform as can be handled with current web technology.

When used on content created natively in SVF, additional capability can be extended to the client-side viewer.

Graphing the performance of stocks over time is only one use of SVF's ability to handle streams of data. Handling the output from financial systems, transactional systems, ERP packages, and CRM systems becomes easier and more flexible. Of course, systems integrators don't have to use these powerful capabilities to start with. If the target system provides web interfaces, these can be viewed, as designed, with no additional software to write, and no changes to the design or layout of the interface.

Server Overview

Enabling the client-side viewer to be so small and powerful is the server-side content translator. The server-side content translator rapidly translates Web content to SVF, compresses and encrypts the SVF results if desired, and transfers the vector formatted results to the client-side viewer. Alternatively, SVF files can be cached or stored in a file system for fetching and transfer at a later time. Pre-translated or cached content transfers are significantly faster as no conversion overhead is incurred, and file sizes are reduced using the more efficient SVF. Combine that with standard compression algorithms selectable for use with the client-side viewer for additional performance improvements.

During the translation process, and in the process of serving cached, pre-translated, or native SVF content, output files are "streamed" to the client-side viewer. Although this does not decrease the total time for file transfer, it can significantly improve the effective system performance for the end user. Content can be selectively streamed, with text and links coming through first, followed by graphic images and other content, for example. Should the user be accessing a link, rather than having interest in the entire file served, links can be selected early in the transfer and the next file download started immediately. In addition to streaming, the server-side content converter may also layer the content by type. This means that text can be put in one layer, links in another, GIF images in another, Javascript in another and so on. Layers can be turned on or off depending upon client capabilities, making files for less capable clients, or for users interested in a reduced functionality, higher transfer performance mode to be handled automatically.

All operational modes may be controlled through an administrative interface or accessible through a straightforward API (Application Program Interface). Furthermore, the system works with existing firewalls and within standard security protocols. In more secure modes, the server-side content converter and the client-side viewer may operate using Public/Private key authentication and encryption.

Exemplary System Infrastructures

In the following paragraphs, a description of three exemplary system infrastructures is provided. Schematic illustrations of these system infrastructures are shown in FIGS. **1**A, **1**B, and **1**C. It is noted that like-numbered components in these Figures perform substantially the same function. Therefore, any discussion of the functions of a component with reference to one or more of the infrastructures generally may apply to the other infrastructures as well, unless specifically noted otherwise.

A first of exemplary system infrastructure **10**A for implementing the invention is shown in FIG. **1**A. Infrastructure **10**A enables various clients, including wireless devices such as a cellular phone **12**, a wireless-enabled PDA **14**, and a wireless-enabled laptop computer **16**, as well as landline computers **18**, **20**, and **22**, to request content that is accessible via a network such as the Internet **24** to be retrieved from selected network resources, including web servers **26** and **28** and an FTP site **30**, wherein the content is translated into a scalable vector representation (e.g., SVF, also referred to herein as "vectorized content") through use of a proxy server **32** and sent to the requesting client. Upon being received by the client, the vectorized content is processed and rendered using a thin client to enable a user to view the content on the client device.

With reference to the flowchart of FIG. **2**A, the foregoing process is initiated by a client in a block **100**, wherein the client submits a request to proxy server **32** to retrieve and convert selected content. As depicted by a transfer path **34**, this comprises sending data **36**, which includes content network location indicia from which the content can be retrieved and proxy server network location information by which the content request may be delivered to over Internet **24** to proxy server **32**. Typically, it will be desired to retrieve a particular web page. Accordingly, the content network location indicia will comprise a URL (uniform resource locator) for the web page. Similarly, the proxy server network location information may also comprise a URL corresponding to a network access point for the proxy server. Optionally, the location information may comprise a network IP address for one or both of the content location and the proxy server location. If the content is to be retrieved from an Internet resource, the request will typically be sent using the HyperText Transfer Protocol (HTTP) over the TCP/IP transport.

Next, in a block **102**, the request is received by the proxy server and the proxy server checks its cache to see if it already has the request content in its cache. If it does, it sends this cached content back to the client. If it does not have the requested content cached, the proxy server sends out a request to retrieve the content from the network resource. For illustrative purposes, it will be assumed for the present example that the desired content comprises a web page that is stored on web server **26**. Typically, when the requested content comprises a web page, the content may be retrieved using conventional web content retrieval techniques, such as that employed by various modem browser clients, including Netscape Navigator and Internet Explorer. This generally comprises providing routing information, such as the URL for the web page (URL **38**) to routing services provided by Internet **24**, which routes the request to an appropriate network resource (e.g., web server **26**), as depicted by a transfer path **40**.

Typically, the URL will correspond to a web page whose content is stored by the web server in an HTML (HyperText Markup Language) document comprising HTML code and embedded text content, in addition to other optional content languages, that may contain references to other objects (e.g., HTML documents and graphic image files) stored locally to the server or stored on a remote server. For example, the HTML content corresponding to a single-frame web page is often stored in a single file, while multiple-frame web pages may comprise content that is stored in a single file or in multiple files. These files may be stored locally on the web server (e.g., on one of the server's hard disks), or on a local storage device connected to the web server via a local area network (LAN), such as a network attached storage (NAS) filer. Optionally, some of the web page's content may comprise one or more documents that are stored at remote locations that may be accessed via a WAN (wide area network) or the Internet.

HTML is a standardized language that describes the layout of content on a web page, and attributes of that content. This layout and attribute information is defined by sets of tags contained in HTML code corresponding to the page. The tags define various HTML layout and display information, including tables, paragraph boundaries, graphic image positions and bounding box sizes, typeface styles, sizes, and colors, borders, and other presentation attributes. A portion or all of a web page's text content may be contained in the parent HTML document corresponding to the URL. In addition to basic HTML, web page documents may contain XML (eXtensable markup language) code, as well as scripting language code, such as javascript. However, for simplicity, any documents containing web page content other than only graphic content that are discussed herein will be referred to as HTML documents.

In addition to HTML and other markup and scripting language content, it is very common for web pages to include graphical content. In general, graphical content is usually stored in an image file or files that are external from the parent HTML document for the web page. For example, the parent HTML document may contain one or more embedded image tags that reference the location where those images are stored. As before, the graphic images may be stored locally, or may be stored on remote servers that are accessed by the web server via a WAN, or the Internet. These files will typically comprise data stored in one of several well-known graphic formats, including bitmap files (BMP), GIF (Graphics Interchange Format) files, and JPEG (Joint Photographic Experts Group) files.

In response to receiving the request for content, web server **26** begins sending a parent HTML document **42** back to proxy server **32** in a block **104**. In a block **106**, the HTML content of the parent HTML document is parsed to search for references to external objects such as HTML frames and graphics. In a decision block **108**, a determination is made to whether any references are found. For each reference to an external object that is found, proxy server **32** requests to have the object retrieved from an appropriate network resource (e.g., a web server) in a block **110**, and data corresponding to the object is transmitted back to the proxy server, as depicted by locally accessible HTML documents **44** and graphic images **46**, as well as remotely accessible HTML documents **48** and graphic images **50**, which may be accessed via web server **28**. If the external object is a graphic image, there is no further processing of the object at this point. If the object is an HTML document, the functions provided by blocks **106** and **108** are repeated. Generally, this set of processing functions is repeated iteratively until all of the external objects are retrieved. However, as described below, there will be some instances in which certain objects will be retrieved at a later point in time. In addition to content stored on web servers that are accessed using HTTP, content may also be retrieved from various network sites using the File Transfer Protocol (FTP), such as FTP documents **51**, which are accessed via FTP server **30**.

In general, HTML documents and graphic files will be sent as packetized data streams using HTTP over one or more TCP/IP network connections, wherein the data streams will usually be asynchronous. Retrieval of HTML documents and graphic files corresponding to the embedded references will usually require additional transfer time. Furthermore, graphic content oftentimes comprises significantly larger file sizes than HTML content, leading to significant transfer times in some instances. For simplicity, the transfer of the various HTML documents and graphic files for the content request are depicted by HTML documents **52** and graphic documents **54**, which are transferred over a transfer path **56**.

When the HTML documents and graphic content are received by proxy server **32**, a scalable vector representation of the web page is generated in a block **114** by an HTML translator **58**. In brief, HTML translator **58** translates HTML, XML, and cascaded style sheet (CSS) layout content into a scalable vector representation, such as SVF. Details of the HTML translation process are contained below. In addition, the graphic images are converted into a compressed bitmap format in a block **116** by a graphics translator **60**. The vectorized content **62** and compressed bitmaps **64** are then streamed back to the client (i.e., computer **18**) in a block **118**, as depicted by a transfer path **66**. In one embodiment, the content portions are sent in separate streams using multiple connections. In another embodiment, the content portions are sent via a multiplexed stream using a single connection. As the vectorized content and compressed bitmap data are received by the client device, they are processed by a thin client **68** running on the client device, whereby a representation of the original web page content may be rendered on the client device's display screen at various user-selectable scaled resolutions and pan offsets in a block **120**, thereby enabling a user to more clearly see an overview or details in the web page. Further details of the client side processing are provided below.

As discussed above, wireless clients may also access the vectorized network (e.g., web site) content provided via proxy server **24**. The majority of this process is identical to that described above for land-line clients (e.g., computers **18**, **20**, and **22**), except for provisions required for sending data to and receiving data from wireless devices. In general, most wireless devices will access the Internet via a wireless service provider (i.e., a wireless telecommunications carrier) that is particular to that wireless device. Accordingly, a portion of the transmission path to and from proxy server **24** will comprise infrastructure provided by that service provider and/or shared with other service providers. For simplicity, this infrastructure is shown as a cellular tower **70** and a service provider data center **72**, although it will be understood by those skilled in the art that the connection path may comprise additional infrastructure components, including appropriate gateways and routers, that enable wireless devices to access proxy server **24**.

In some implementations, there will be no special formatting/protocol services that need to be performed by proxy service **24**—from the viewpoint of the proxy service, it will be immaterial whether the client is a land-based or wireless client; the special handling provisions for wireless devices will be handled entirely by the service providers infrastructure transparently at both ends of the communications path. In other instances, it may be desired or necessary to reformat the data content delivered to the wireless device at the proxy service. This will generally be dependent on the particular wireless protocol used, and what services are provided by the service provider for the wireless client.

Currently, in the United States, wireless clients generally access Internet **24** by using the Wireless Application Protocol (WAP). In Japan, the most popular access means is NTT DoCoMo's i-Mode wireless protocol. In addition to these wireless standards, new standards are anticipated to be in force in the near future, including NTT DoCoMo's FOMA (Freedom of Mobile Multimedia Access), which is transported over W-CDMA (Wideband Code Division Multiple Access), and CDMA-2000. For the purposes of the invention herein, it will be understood that those skilled in the mobile telecommunications arts will be knowledgeable about any particular format and/or transport protocol requirements that pertain to the particular protocol that is to be used.

A second exemplary system infrastructure **10**B for implementing the invention is shown in FIG. **1**B. As will be readily recognized, much of infrastructure **10**B is similar to infrastructure **10**A; however, rather than have a separate proxy server perform the proxy functions (retrieve and translate content), these functions are performed on machines operated by the web site in infrastructure **10**B.

The logic implemented by the invention when providing content to a client using infrastructure **10**B is illustrated in the flowchart of FIG. **2**B, wherein the process begins in a block **101** in which the client sends a content request **39** directly to the network site (e.g., web server **26**), as depicted by a transfer path **41**. In a block **103**, HTTP negotiations are performed to determine the format the content is to be delivered in. For example, the request may contain indicia identifying the type of content requested, such as an SVF MIME type (e.g., image/vnd.svf). This is to inform the web server that the request is for specially-formatted content rather than conventional content. The server first checks to see if it already has cached the requested content. If it has, it sends the content to the requesting client; otherwise, it retrieves the parent HTML document in a block **107**. It then performs processing steps in blocks **107**, **109**, and **111** to retrieve content referenced through embedded tags in a manner substantially similar to that discussed above with reference to respective blocks **106**, **108**, and **110**. The primary difference in this instance is that the web server does not receive requests from or send documents to a proxy server—rather, the content is retrieved and processed at the web server, wherein the retrieved content may be stored local to the web server or retrieved from a remote server in a manner similar to that described above.

As before, the retrieved HTML documents are translated into scalable vector representations by HTML translator **58** in a block **114**, while the graphic images are translated into a compressed bitmap format by image translator **60** in a block **116**, as depicted by vectorized content **62** and bitmap content **64**. The vectorized content and bitmap content are then streamed from the web server to the client in a block **119**, as depicted by a transfer path **67**. Upon arriving at the client, the vectorized content and bitmap content are processed, scaled, and rendered on the client in a block **120**.

A third exemplary system infrastructure **10**C for implementing the invention is shown in FIG. **1**C. In this configuration, the proxy functions are performed at the client. As shown by a block **113** in FIG. **2**C, the process for providing vectorized content to a client in accordance with infrastructure **10**C begins in a block **113**, in which the client sends a content request **37** to a network site, such as web server **26**, via Internet **24**. In response, the network site retrieves the parent HTML document and sends it to the requesting client in a block **115**. In a manner similar to that discussed above with reference to blocks **106**, **108**, and **110** of FIG. **1**A, the client first parses the parent HTML document searching for embedded references to external objects and retrieves these objects, whereupon the embedded reference search is performed on the newly retrieved document until all of the content corresponding to the original content request has been retrieved. This content is depicted by HTML documents **52** and image files **54**, which are sent from the network site to the client via a transfer path **69**. At this point, the client performs translations on the HTML content and the graphic image content that are substantially similar to that performed by the proxy server in FIG. **1**A or at the web site in FIG. **1**B, as provided by blocks **114** and **116**. The vectorized and image content is then processed and scaled by thin client **68** in a block **120**, as depicted by device output **71**.

Attention now is focused on the functionality provided by proxy server **24** in system infrastructure **10**A of FIG. **1**. Fundamentally, the proxy server functions as a proxy. It accepts requests for content from client devices as full URLs using standard HTTP mechanisms carried over a multiplexed TCP connection. Standard HTTP content negotiations features specify the formats in which content is to be delivered (SVF, bitmap, and possibly others, which can be handed off to cooperating client-side display software). As described in further details below, in some embodiments the proxy server appears for the client as a normal proxy (that is, the client knows it is retrieving content via the proxy), while in other embodiments the proxy is transparent to the client.

The proxy server responds to client content requests by delivering content in one of the requested formats, by retrieving the content in an appropriate format from its cache, or from an upstream content source (again using standard HTTP content negotiation features), or by translating upstream content from a supported original format to SVF or the client bitmap format.

Requests from the server installation to its cache and from the cache to upstream content sources are made in HTTP carried over TCP using simple straightforward Web content requests. For example, requests from clients to the proxy server comprise HTTP proxy requests (e.g., "GET http://www/xyz.com/some_page.html HTTP/1.0 . . . ") carried over TCP or over a lightweight multiplexing protocol over TCP. The multiplexing protocol allows the server to push image thumbnails to the client before the SVF stream is available, as well as offering a channel for control and status information, more simultaneous channels than the client operating system may support, and a mechanism for prioritizing information flow from server to client under loose client control. In addition to HTTP requests, the proxy server architecture supports other user-level protocols, such as FTP and Gopher.

Details of some of the primary components of the proxy server architecture are shown in FIG. **3**. Internally, the proxy server comprises a suite of coordinated processes connecting to upstream content through an HTTP cache **74**. In one embodiment all functions except caching are performed in a single process, wherein multiple threads are used to effect asynchronous I/O. Separate processes communicated via persistent multiplexed connections carried over the most efficient reliable transport available (e.g., Unix sockets over single processor and symmetric multiprocessor (SMP) computers; TCP sockets between separate computers). All processes are capable of servicing multiple requests simultaneously. No process maintains client state outside the context of a single request, so all components can be repeated and load balanced across multiple CPU's of an SMP computer or across separate computers on a LAN.

The various content translators used by the proxy server accept (via HTTP PUT) or request (driven by HTTP proxy GET/POST) content in supported, but client-unsupported, formats; and return (via HTTP PUT or GET/POST response) one or more representations of that content in a client-supported format. In the embodiments illustrated in FIG. **1**A-C, two translators are used: HTML translator **58** and image translator **60**. Future content types may be accommodated by new translators, by extending existing translators to cover the new content types, or by extending the client's capabilities. Standard HTTP content negotiation mechanisms are used to inform the proxy server of the client's capabilities and expectations on each request.

Managers at the proxy server coordinate the operations of other components. Two managers are presently defined; a client manager **73** that handles client proxy requests, and a request manager **75** that handles unproxied HTTP requests from other services. The managers accept requests, attempt to service them from HTTP cache **74**, and drive HTML translator **58** and image translator **60** when content does not match the clients' requirements. Managers also handle translator requests for inline content (e.g., image dimensions for page layout), and push translated content into HTTP cache **74**. Additionally, the client manager coordinates delivery of primary and inlined content, and provides process and status information to the clients.

As discussed above, HTML translator **58** creates a scalable vector representation of the original HTML content of a requested web page. In order to better explain how translation of HTML content is performed, one embodiment of a translation process is described below as applied to an exemplary web page. In addition, details of conventional web page client and server-side processing are provided so as to clarify how web content is laid out during a pre-rendering process on the client.

FIG. **4** shows a representation of a web page **210** served from an exemplary stock brokerage Internet web site as it would appear when rendered on a modern Internet browser, such as Microsoft's Internet Explorer or Netscape's Navigator. Web page **210** is exemplary of many web pages that implement frames, and includes two adjacent frames **212** and **214**. A logo graphic object **216**A is displayed at the top of frame **212**, which additionally includes a "MARKETS" text header **218**A, an "INVESTMENTS" text header **220**A, and a plurality of links with overlaying graphic objects, including a "DOW" link **222**A, a "NASDAQ" link **224**A, an "OPTIONS" link **226**A, a "CHARTS" link **228**A, a "MUTUAL FUNDS" link **230**A, a "IRA, 401K OPTIONS" link **232**A, and a "TAX INFORMATION" link **234**.

A horizontal group of links **236** is disposed at the top of frame **214**, and includes a "QUOTES" link **238**A, a "HOT PICKS" link **240**A, a "CALENDARS" link **242**A, and a "NEWS" link **244**A. An advertisement banner **246**A is displayed just below the horizontal group of links and just above a "NEWS SPARKS MARKET" headline **248**A. Frame **214** also includes a pair of graphic image objects, including a DOW chart **250**A and a NASDAQ chart **252**A. A set of user input objects is disposed adjacent to DOW chart **250**A within a graphic object **254**A, including an "ACCOUNT #" input box **255**A, an "ACCESS CODE" input box **256**A, and a "LOGIN" button **257**A. In addition to the foregoing objects, frame **214** also includes text objects **258**A and **260**A.

An HTML listing corresponding to web page **210** is presented below as LISTING 1. Note that LISTING 1 sometimes refers to object descriptions and link paths rather than the text or path location of actual objects for simplicity, and that other elements commonly found in HTML pages, such as META entries, are omitted for clarity.

Listing 1

```
<html>
<head><title>”MARKET HOME”</title></head>

<body bgcolor=”#FFFFFF” link=”0033CC” vlink=”0033CC”>

<frameset cols=”25%,75% frameborder=0 border=0>
<frame>
<align=left><align=top>
<img src=”/directory path/logo.gif” align = left border=“0” height=”80” width=”100”>
```

-continued

```
<br><br>
<t3>TEXT HEADER #1 align=left</t3><br>

<table width=”90%” border=0 cellspacing=10 cellpadding=0 bgcolor=“#000000”
  align=center>
    <tr>
        <a href=”URL or path for LINK #5” <img src=”/directory
        path/GRAPHIC#2” height=”50” width =”150></a>
    <tr>
        <a href=”URL or path for LINK #6” <img src=”/directory
        path/GRAPHIC#3” height=”50” width =”150></a>
    <tr>
        <a href=”URL or path for LINK #7” <img src=”/directory
        path/GRAPHIC#4” height=”50” width =”150></a>
    <tr>
        <a href=”URL or path for LINK #8” <img src=”/directory
        path/GRAPHIC#5” height=”50” width =”150></a>
</table>
    <br>
        <t3>TEXT HEADER #1 align=left</t3>
    <br>
<table width=”90%” border=0 cellspacing=10 cellpadding=0 bgcolor=“#000000”
align=center>
    <tr>
        <a href=”URL or path for LINK #9” <img src=”/directory
        path/GRAPHIC#6” height=”50” width =”150></a>
    <tr>
        <a href=”URL or path for LINK #10” <img src=”/directory
        path/GRAPHIC#7” height=”50” width =”150></a>
    <tr>
        <a href=”URL or path for LINK #11” <img src=”/directory
        path/GRAPHIC#8” height=”50” width =”150></a>

</table>
</frame>

<frame>

<table>
    <tr>
        <table width=“100%” border=0 cellspacing=15 cellpadding=0
        bgcolor=“#000000” align=center>
      <tr>
        <td><a href=”URL or path for link#1”> alt=”QUOTES”</a>
        <td><a href=”URL or path for link#2”> alt=”HOT PICKS”</a>
        <td><a href=”URL or path for link#3> alt=”CALENDERS”</a>
        <td><a href=”URL or path for link#4>alt=”NEWS”</a>
</table><br>
    <br>
        <img src=”URL for GRAPHIC #9” align=center
        border=”0” height=”80” width=”325”>
    <br><t1>HEADLINE TEXT>/t1>
    <table>
      <Colgroup span=”2”>
        <Col width = “400” align=“center”>
        <Col width = “200” align=”center”>
        <tr><td>
            <img src=”/directory path/GRAPHIC #10” align = center
            border=“0” height=”180” width=”350”>
        <td>
    /* INPUT FOR ACCOUNT NUMBER AND ACCESS CODE */
    <SCRIPT LANGUAGE =”Javascript”>
    <!---
          [Javascript variable declarations]
          [Javascript functions to enable login] ---!>
    </SCRIPT>
      <table>
        <td>
          <img src=”/directory path/GRAPHIC #11” align = center>
          <table width=”150” height=”25”>
          <td>
          <font size=−2 face=“arial,helvetica,verdana”>Account#</font>
          <tr><input type=text name=“USERID” maxlength=9 size=20>
          <tr><font size=−2 face=“arial, helvetica”>Access Code:</font>
          <tr><input type=password name=“PASSWORD” maxlength=10 size=20
              onKeyDown=“SuppressEnterBell(event)”
              onKeyPress=“SuppressEnterBell(event)”
              onKeyUp=“SubmitOnEnter(event)”>
          <br>&amp;nbsp;
```

-continued

```
89.           <br><input type=“button” value=“Login”
90.           OnClick=“ProcessForm( )”>&amp;nbsp;&amp;nbsp;<input type=“reset”>
91.           <br>&amp;nbsp;
92.           </td>
93.           </table>
94.       </table>
95.   <tr>
96.           <img src=”/directory path/GRAPHIC #12” border=“0
97.           height=”200” width=”350”>
98.   <tr>
99.               <p>TEXT FOR TEXT OBJECT #1</p><br>
100.              <p>TEXT FOR TEXT OBJECT #2</p>
101.      </table>
102.    </frame>
103.  </frameset>
104.  </html>
```

Web page documents comprise HTML code that is parsed, interpreted, and rendered by a browser. An HTML document comprises a plurality of HTML “markup” elements (tags) with corresponding attributes, that are used to describe the layout and formatting of various objects, including plain text and graphic objects, embedded between tag pairs. Exemplary elements include text tags (e.g., <b></b> for bolding text), links (e.g., <a href=“URL”></a>), formatting (e.g., <p></p> for creating a new paragraph, graphical (e.g., <img src=“name”>), wherein “name” defines an absolute or relative location at where an image is stored, tables (e.g., <table></table>) creates a table, and forms (e.g., <form></form> creates all forms).

As of Netscape Navigator **3**.**0** (and other later browsers), web pages could include frames. When using frames, the display page is divided into multiple framed areas. Framing enables a single display page to include source code from several HTML documents (one for each frame) or optionally, enables a single document to include more complicated grouping of contents whereby different content groups are contained in separate frames. Frames are commonly found on the web pages at sites that display a great deal of text and graphical content, such as MSN.com, ESPN.com, and USAToday.com.

With reference to the flowchart of FIG. **5**, the process for translating the HTML content into a scalable vector representation proceeds as follows. The process is initiated when the proxy server receives the HTML corresponding to the parent document (and frame documents, if appropriate), whereupon a pre-rendering parsing of the HTML is performed to determine where to place the various objects on the display page in a block **150**. For example, elements such as tables, column definitions, graphic images, paragraphs and line breaks are identified. If frames are included, each frame is examined in the sequential order it appears in the HTML document, or the order in which the HTML documents corresponding to the frames in a frameset are downloaded to the browser. During further processing, the actual objects are rendered in their respective positions. Some of these objects are rendered almost immediately, such as plain text, while other objects, such as graphic objects, must first be retrieved prior to being fully-rendered. With respect to tables, there are some instances in which all of the objects corresponding to the cells in the table must be retrieved prior to rendering any of the table, while a well-designed table can be rendered incrementally. For example, by using Column grouping, the format of the corresponding table can be quickly determined by the browser. In some instances, one or more bitmaps may actually need to be fetched before the page layout can be determined.

Next, in a block **152**, the content is separated into objects based on logical groupings of content portions and a page layout is built using bounding boxes that are produced for each object. As the primary HTML document is parsed, logical groupings of content will emerge. For instance, text content contained within paragraph tags <p></p> forms a logical grouping of text content. In essence, a logical grouping means the content should appear together as a logical group, such as within a substantially rectangular outline, in the rendered page. Other logical groupings include frames, table content, row content, single line entries such as headlines and headers, and user-interface objects, as well as graphic layout objects, such as separator bars, and graphic images. In addition to logically grouping content into objects, a “bounding box” is defined for each object. In general, the bounding box defines an outlined shape within which the content (text or graphic image) will appear. In most instances, the bounding box will be substantially rectangular in shape. However, bounding boxes comprising more complex shapes may also be produced.

In further detail, the following explains how objects corresponding to graphic images are produced. In HTML, objects comprising graphic content are identified by an <img src=“/local directory path/graphic image file” (for a local graphic image) or “URL” (for a remote graphic image)> or <object> or other tags. In the foregoing tag, local graphic images are typically stored on the same server as the web page, or another computer that is local to the site's server, and generally are located through a local directory path (absolute or relative to the location of the present page) that points to the graphic image file. Remote images are those images that are stored on servers at sites that are remote to the web server. For example, with reference to LISTING 1, when the parser encounters line 9, the browser identifies that data comprising a graphic image corresponding to logo graphic object **1** will be arriving (or may have already been received), and the displayed image is to have a height of 80 pixels and a width of 100 pixels. The location of each object on a display page will be dependent on previous HTML layout elements, such as tables, paragraphs, line breaks, and other graphic objects. The size and location of the other graphic objects (i.e., graphic objects #**2**-**12**) on the page are determined in a similar manner. The HTML code for these objects are shown in lines **16**, **19**, **22**, **25**, **34**, **37**, **40**, **59**, **67**, **78** and **96**, respectively. As identified in the HTML code, data corresponding to graphic objects #**9** (advertisement banner **46**A) is forwarded to the browser from an external site (as indicated by the URL to GRAPHIC #**9**), while graphic objects **1**-**8** and **10**-**12** are sent from the web site the parent HTML document is sent from.

In a similar manner, the foregoing technique is applied to the HTML code in the primary document to identify other types of objects as well. In addition to parsing the primary HTML document, similar processing is performed on referenced documents, such as documents that include frame content that is defined and stored separate from the primary HTML document.

A representation of the results of the functions performed in block **152** are shown in FIG. **4**B. In the Figure, objects corresponding to the original content of FIG. **4**A are shown with an appended "B" that is added to each object's root reference number, wherein the root reference number for an object is that same as the logically grouped content in FIG. **4**A that it corresponds to, e.g., an object **248**B is generated for "NEWS SPARKS MARKET" headline **248**A, etc.

Next, in a block **154**, the page layout is defined based on the bounding boxes. In actuality, generation of the page layout information is performed in conjunction with defining the boundary boxes for the objects, wherein the location of a given object is based on the location of other related (e.g., if within a table) or non-related objects corresponding to HTML content that have been previously parsed. For example, the location of a given paragraph will depend on the other content for the page that are listed prior to the definition for the paragraph in the primary HTML document or referenced document, if applicable. As the HTML content of the primary and any referenced HTML documents are parsed, the page layout is generated based on the various HTML tags and the content embedded between tag pairs and/or referenced by a tag pair statement (e.g., graphic images).

As will be recognized by those skilled in the art, the functions performed in blocks **150**, **152**, and **154** are commonly performed by conventional browsers during a pre-rendering process. In some browsers, these functions are performed by the Mozilla rendering engine, which comprises open source software that is readily available for use by developers. At present, the software for the Mozilla rendering engine may be accessed via the Internet at www.mozilla.org. Accordingly, in one embodiment, the present invention uses core functionality provided by the Mozilla rendering engine source code to perform the functions of block **150**, **152**, and **154**.

At this point, the present invention deviates substantially from the prior art by using the various object layout data generated during the pre-rendering process to generate a scalable vector representation of the original page content. First, in a block **156**, a datum point is defined for the page and the bounding box for each object. For example, as shown in FIG. **4**C, a rendered page datum **262** is defined to be coincident with the upper left hand corner of the display frame of the rendered page for the web page. Generally, any point on the page may be used as the page datum—the only requirement is that the page datum that is selected is used consistently throughout the process. The use of the upper left hand corner of the display frame is advantageous since the location of the first object encountered in the HTML code for a page is located relative to this corner.

In general, the datum points for each object may also be located any place on the object, as long as the object datum points are used in a predictable manner. For example, as depicted in FIG. **4**C, various datum points for corresponding objects are defined to be coincident with the upper left hand corner of the bounding box for that object, wherein the object's datum point shares the root reference number of the object with an appended "C."

Once the page's datum point and an object's datum point are known, a vector between these points is generated for each object in a block **158**. With reference to FIG. **4**D, in one embodiment, wherein the page datum point corresponds to the upper left and corner of the display frame and is assigned an XY value 266 of 0,0, the vector for a given object may be stored as the XY value of the datum point of that object relative to 0,0, such as a value of 150, 225 (ref. num. **268**) for a vector **250**D pointing to an object datum **250**C, and a value of 150, 425 (ref. num. **270**) for a vector **252**D pointing to an object datum **252**C. In another embodiment, each vector may be stored as XY data relative to a 0,0 datum point corresponding to the upper left hand corner of the frame the object belongs to. For example, a vector **250**D' from a frame datum **214**D to object datum **250**C is stored as 20, 200 (ref. num. **268'**), while a vector **252**D from frame datum **214**D to object datum **252**C is stored as 20, 425. In this embodiment, offset information for each frame relative to a known datum will also be stored, as depicted by a vector **214**D.

The scalable vector representation is completed in a block **160**, wherein a reference is created for each object that includes or links an object's content and attributes, such as object type (e.g., text, image), object typeface, and boundary box parameters, to the object's vector. For example, object **250**B is a graphic image having a vector **250**D and a bounding box that is 180 pixels high and 350 pixels wide, while object **252**B is a graphic image having a vector **252**D and a bounding box that includes a height of 200 pixels and a width of 350 pixels. This enables client-side operations to be performed that only initially consider the vectors, wherein if it is determined that a vector's endpoint (and/or the bounding box corresponding to the object the vector points to) would appear off of a display, there is no need to retrieve the content and attribute data linked to the vector. This concept is explained in further detail in the following section.

It is noted that a portion of the display content produced on a client device will never contain any rendered content, as this portion is reserved for the browser's user interface. In WINDOWS™ environments, this portion will include the browser's window frame, as well as the pulldown and icon menus provided in the browser's user interface, which are depicted by a box **264** in the Figures herein.

Client-Side Software and Processing

As discussed above, the present invention supports a wide variety of clients, including land-based clients and wireless clients. Each client requires some client-side software that enables the scalable vector content data provided to it to be rendered at a user-selectable scale factor and offset on the client's display, such as a monitor or built-in LCD screen.

By enabling original content from a web site to be displaced in such a resolution-independent manner, users will be able to view content in a manner that did not previously exist, greatly enhancing the user experience. For example, in some implementations the client may be a personal computer (PC). Using a least-common denominator approach, many web pages are designed for a smaller resolution (for example 640×480 pixels, a minimum resolution commonly supported by nearly all PC's, including legacy PC's) than the resolution provided by the video output capabilities available with many of today's PC's, such as 1024×768 pixels, 1280×1024 pixels, and even 1600×1200 pixels. As a result, when these web pages are displayed on a high-resolution display, they occupy only a portion of the display, making portions of the pages, especially those portions containing small text, difficult to read. By enabling users to selectively magnify the entire page, these design flaws are easily overcome. Alternatively, the client may be a small device, such as a hand held computer or a cell phone, which has a smaller display resolution than common Web pages are designed for. As explained below, through use of the invention's scalable vector representation and client-side processing, users are enabled to view the entire content of billions of existing Web pages using handheld devices in a simple and reasonable way.

In one embodiment, the client software may be a plug-in to a Web browser, such as Netscape Navigator or Microsoft Internet Explorer. Such a plug-in might have the browser download the data and display it in a sub-window of the browser. Alternatively, the client software may be a Java applet running in a browser. As another option, the client software may be a stand-alone program that interfaces with the proxy server or proxy software directly. The client software may bypass the proxy when requesting information that won't be translated to vectors, such as bitmaps.

With reference to FIG. **6**, client-side processing proceeds in the following manner. In a block **160**, the vector representation data (i.e., vectorized HTML content and compressed bitmap content) for the web page is gathered at the client. Typically, this data will be stored in a cache at the client as it is being received, and the client simply retrieved the data from the cache. In a block **162**, a display list of vectors is built. This process is well known in the CAD arts, and is enabling rapid zooming of vector-based objects. In a block **164**, user selectable scale and offset (pan) values are determined. Based on various user interactions with the user-interface of the client, the user is enabled to control the zoom (size) and offset of the rendered page. For example, suppose the user provides zoom and offset inputs to produce a rendered page **210**E, as shown in FIG. **4**E. In this rendered page, the original origin is now off of the screen (the page image is shifted upward and toward the left—see FIG. **4**F), and the view has been scaled approximately 1.3 times.

Next, in a block **166**, the vectors and boundary boxes are processed based on the scale and offset, and a bounding box defining the limits of the display content is determined. The results of this step are shown in FIG. **4**F, while FIG. **4**G shows specific details one how the vectors and bounding boxes corresponding to image objects **250**B and **250**B (now **250**B' and **252**B', respectively) are processed. Logically, there are generally two ways to scale and offset the rendered content. In one embodiment, vectors and bounding boxes are mapped to a virtual display area in memory that has much greater resolution (e.g., 100,000×100,000 pixels) than any real display, and a virtual display limit bounding box is scaled and moved around over the virtual display area. Accordingly, during subsequent processing described below, objects falling within the display bounding box are rendered by reducing the scaling of those objects in the virtual display to how the objects will appear on the client device display relative to the virtual display bounding box. In the alternate, a fixed reference frame corresponding to the display resolution of the client device screen is maintained, wherein all vectors and bounding boxes are scaled and offset relative to the fixed reference frame. Each scheme has its advantages and disadvantages. One advantage of the second method is that the display bounding box is always maintained to have a size that matches the resolution of the content display area on the client device.

As shown in FIG. **4**G, respective offsets in X and Y, (−ΔX and −ΔY in the Figure) are applied to the starting point of each of the vectors. The vectors are then scaled by a scale factor "SF." The results of the new vectors are depicted by vectors **250**D" and **252**D". This produces a new datum for each object's bounding box that is relative to rendered page datum **262**, which remains fixed. As discussed above, only a portion of the display screen will actually be used to display content (as defined by a display limit bounding box **266** in this embodiment), while other portions of the screen, including box **264**, will comprise a generally fixed-size user interface. Accordingly, rendered page datum **262** is not located at the upper left hand corner of the display area, although it possibly could be located at this point when either the current user interface is inactive (i.e., the display portion of the user interface is temporary disabled) or the user interface is contained in other portions of the display.

This foregoing process establishes a starting point (the new datum) for where the content in each object's bounding box will be rendered. At this point, each object's bounding box is then drawn from its new datum using the scaling factor. For example, in the original web page **210**D (FIG. **4**D), bounding box **250**B had an X-axes datum of **150** pixels, a Y-axis datum of 225 pixels, and a height and width of 180×350 pixels. In contrast, after being offset and scaled, bounding box **250**B' has an X-axis datum of 150* SF−ΔX, a Y-axis datum of 225*SF−ΔY, and a height and width of 180*SF×350*SF.

Returning to the flowchart of FIG. **6**, once the vectors and bounding boxes are offset and scaled, content corresponding to objects having at least a portion of their bounding boxes falling within the display limit bounding box is retrieved from the client device's display list in a block **168**. For examples, as shown in FIG. **4**F, content corresponding to all of the objects except for those falling entirely outside of display limit bounding box **266** (objects **216**, **238**, **240**, **242** and **244**) is retrieved from the display list. That content is then scaled in a block **170**. For image content, this comprises decompressing and scaling the compressed bitmaps corresponding to those images. For text content, this comprises scaling the font (i.e., typeface) that the text content portions of the web page are written in the parent HTML document and any referenced documents. There are various techniques for typeface scaling that may be implemented here, depending on the available resources provided by the operating system of the client device. For example, for WINDOWS™ operating systems, many TRUETYPE™ fonts are available, which use a common scalable definition for each font, enabling those fonts to be scaled to just about any size. In other cases, such as current PDA (e.g., Palm Pilots) operating systems, there is no existing feature that supports scaling fonts. As a result, bitmapped fonts of different font sizes and styles may be used. In addition to scaling image and text content, other types of content, such as separator lines and borders may also be scaled by block **170**.

The process is completed in a block **172**, wherein those portions of the scaled content falling within the display limit bounding box are rendered on the client device's display.

As discussed above, it is foreseen that the invention will be used with client devices having small, low resolution displays, such as PDAs and pocket PCs. Examples of various views of an exemplary web pages obtained from the YAHOO™ web site are shown in FIGS. **7**A-B, **8**A-B and **9**A-B. For instance, FIG. **7**A represents how the YAHOO™ home page might appear on a Palm IIIc color PDA.

In addition to directly scaling and offsetting content, the client user-interface software for PDA's provides additional functionality. For instance, a user may select to view a column (results represented in FIG. **7**B by tapping that column with a stylus, a shown in FIG. **7**A. Similarly, the user may select to zoom in on an image by tapping the image with the stylus, as shown in FIGS. **8**A and **8**B, or select to view a paragraph in an article by tapping on the paragraph, as shown in FIGS. **9**A and **9**B. It is noted that in some instances, the display of the paragraph may be reformatted to fit the characteristics of the display, rather than following the original format in the zoom-out view.

It is further noted that that different scaling factors can be applied to the X and Y axis so as to change the aspect ratio of the display. For example, a Web page may be designed to be displayed on a computer having a resolution of 800×600 pixels, or a 4X to 3Y aspect ratio. In this case, the display corresponds to a "landscape" layout, wherein there are more pixels along the X axis than along the Y axis. Conversely, many handheld devices display images having a "portrait" layout, wherein there are more pixels along the Y axis than the X axis. By enabling different scaling factors to be applied to the X and Y axes, the present invention enables the aspect ratio of a rendered display image to be adjusted to better fit the aspect ratio of the client device.

An Exemplary Computer Architecture

An exemplary machine in the form of a computer system **500** in which features of the present invention may be implemented will now be described with reference to FIG. **10**. Computer system **500** may represent a workstation, host, server, print server, or printer controller. Computer system **500** comprises a bus or other communication means **501** for communicating information, and a processing means such as processor **502** coupled with bus **501** for processing information. Computer system **500** further comprises a random access memory (RAM) or other dynamic storage device **504** (referred to as main memory), coupled to bus **501** for storing information and instructions to be executed by processor **502**. Main memory **504** also may be used for storing temporary variables or other intermediate information during execution of instructions by processor **502**. Computer system **500** also comprises a read only memory (ROM) and/or other static storage device **506** coupled to bus **501** for storing static information and instructions for processor **502**.

A data storage device **507** such as a magnetic disk or optical disc and its corresponding drive may also be coupled to bus **501** for storing information and instructions. Computer system **500** can also be coupled via bus **501** to a display device **521**, such as a cathode ray tube (CRT) or Liquid Crystal Display (LCD), for displaying information to an end user. Typically, an alphanumeric input device **522**, including alphanumeric and other keys, may be coupled to bus **501** for communicating information and/or command selections to processor **502**. Another type of user input device is cursor control **523**, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor **502** and for controlling cursor movement on display **521**.

A communication device **525** is also coupled to bus **501**. Depending upon the particular presentation environment implementation, the communication device **525** may include a modem, a network interface card, or other well-known interface devices, such as those used for coupling to Ethernet, token ring, or other types of physical attachment for purposes of providing a communication link to support a local or wide area network, for example. In any event, in this manner, the computer system **500** may be coupled to a number of clients and/or servers via a conventional network infrastructure, such as a company's Intranet and/or the Internet, for example.

Importantly, the present invention is not limited to having all of the routines located on the same computer system. Rather, individual objects, program elements, or portions thereof may be spread over a distributed network of computer systems. Additionally, it is appreciated that a lesser or more equipped computer system than the example described above may be desirable for certain implementations. Therefore, the configuration of computer system **500** will vary from implementation to implementation depending upon numerous factors, such as price constraints, performance requirements, and/or other circumstances. For example, according to one embodiment of the present invention, a cell phone or a hand held computer may comprise only a processor or a micro controller and a memory, such as a micro code ROM or RAM, for storing static or dynamically loaded instructions and/or data.

In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

What is claimed is:

1. A wireless device, comprising:

processing means;

wireless communications means, to facilitate wireless communication with a network that supports access to the Internet;

a display;

memory; and

storage means, in which a plurality of instructions are stored that when executed by the processing means enable the wireless device to perform operations including,

rendering a browser interface via which a user is enabled to request access to an original Web page, the Web page comprising HTML-based Web content having an original format defining an original width and height of the Web page and an original page layout, functionality, and design of content on the Web page;

in response to a user request to access the Web page,

retrieving the Web page via the wireless communication means, and translating at least a portion of the HTML-based Web content from its original format into scalable content that supports a scalable resolution-independent representation of the Web page that preserves the original page layout, functionality and design of the content defined by its original format when scaled and rendered; and

scaling the scalable content to render the Web page on the display such that a width of the Web page is rendered to fit across the display.

2. The wireless device of claim **1**, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a corresponding user interface input.

3. The wireless device of claim **2**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

4. The wireless device of claim **1**, wherein the Web page includes at least one hyperlink, and wherein execution of the instructions performs further operations comprising:

enabling the user to select the hyperlink; and, in response thereto,

retrieving and translating Web content associated with the hyperlink to produce additional scalable content; and

employing the additional scalable content to render the Web content associated with the hyperlink on the display.

5. The wireless device of claim **1**, wherein execution of the instructions performs further operations comprising:

parsing markup language code to determine the original page layout of display content within the Web page, wherein the original page layout defines a layout location for a plurality of objects, including at least one of text objects, graphic layout objects, or graphic image objects included in the Web page;

defining a primary datum corresponding to the original page layout; and, for each object,

defining an object datum corresponding to the layout location for the object;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to the vector that is generated.

6. The wireless device of claim **1**, wherein execution of the instructions performs further operations comprising enabling the Web page to be displayed at different resolutions by scaling the scalable content to re-render the display in response to associated user inputs,

wherein the original page layout, functionality, and design of the Web page content are preserved at each of the different resolutions.

7. The wireless device of claim **1**, wherein execution of the instructions performs further operations comprising returning the display of the Web page to a previous view in response to a corresponding user input.

8. The wireless device of claim **1**, wherein execution of the instructions performs further operations comprising enabling a user to pan a view of the Web page in response to a corresponding user input.

9. The wireless device of claim **8**, wherein execution of the instructions performs further operations comprising enabling the view of the Web page to be panned in real-time.

10. The wireless device of claim **1**, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein the scalable content is scaled when rendered so as to produce a display of the Web page having a different aspect ratio.

11. The wireless device of claim **1**, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a column of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is enlarged.

12. The wireless device of claim **11**, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

13. The wireless device of claim **11**, wherein the corresponding user input comprises tapping on the column via the display.

14. The wireless device of claim **11**, wherein the display is re-rendered such that content corresponding to the selected column is rendered to fit across the display.

15. The wireless device of claim **1**, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to zoom on an image via a corresponding user input, wherein in response thereto, the display is re-rendered such that the image is enlarged.

16. The wireless device of claim **15**, wherein the corresponding user input comprises tapping on the image via the display.

17. The wireless device of claim **15**, wherein the display is re-rendered such that the image is rendered to fit across the display.

18. The wireless device of claim **1**, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a paragraph of the Web content via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is enlarged.

19. The wireless device of claim **18**, wherein the content of the paragraph is reformatted to fit characteristics of the display when the display is re-rendered.

20. The wireless device of claim **18**, wherein the corresponding user input comprises tapping on the paragraph via the display.

21. The wireless device of claim **18**, wherein the display is re-rendered such that content corresponding to the selected paragraph is rendered to fit across the display.

22. The wireless device of claim **1**, wherein the Web page includes text, layout attributes, and images, and wherein execution of the instructions performs further operations comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

23. The wireless device of claim **1**, wherein execution of the instructions performs further operations comprising:

generating a vector-based display list associated with the scalable content; and

employing the display list to re-render the display at different scale factors to zoom the Web page.

24. The wireless device of claim **1**, wherein execution of the instructions performs further operations comprising:

parsing markup language code corresponding to the retrieved Web page to determine the original page layout of the content on the Web page;

logically grouping selected content into objects;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to a layout location datum for the object's associated display content;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to the vector that is generated.

25. The wireless device of claim **24**, wherein execution of the instructions performs further operations comprising:

generating a bounding box for each object, the bounding box representing a portion of a rendered display page occupied by the object's associated group of content.

26. The wireless device of claim **25**, wherein execution of the instructions performs further operations comprising:

mapping the object vectors and associated bounding boxes to a virtual display in memory.

27. The wireless device of claim **26**, wherein execution of the instructions performs further operations comprising:

enabling a user to view the Web page at a user-selectable zoom level and panned view by,

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and panned view corresponding to a rendered display of the Web page desired by a user; and

determining a virtual display bounding box for the virtual display associated with the first scale factor and offset;

identifying object bounding boxes having at least a portion falling within the virtual display bounding box; and,

for each of such object bounding boxes,

retrieving content associated with that object bounding box;

applying an appropriate scale factor to the content associated with that object bounding box to produce scaled content; and

rendering the portion of scaled content within the virtual display bounding box to render the content on the display.

28. The wireless device of claim **1**, wherein the scalable content includes scalable text content, and wherein execution of the instructions performs further operations comprising scaling a scalable font to render the scalable text content.

29. The wireless device of claim **1**, wherein at least a portion of the instructions comprise Java-based instructions.

30. The wireless device of claim **1**, wherein the device comprises a mobile phone.

31. The wireless device of claim **1**, wherein the device comprises one of a Personal Digital Assistant (PDA) or handheld computer.

32. The wireless device of claim **1**, wherein the network comprises a mobile service provider network.

33. The wireless device of claim **1**, wherein a portion of the scalable content comprises vector-based content.

34. The wireless device of claim **1**, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

35. The wireless device of claim **1**, wherein the device enables a user to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page.

36. A mobile hand-held device, comprising:

a processor,

a wireless communications device, to facilitate wireless communication with a network that supports access to the Internet;

a display; and

flash memory, operatively coupled to the processor, in which a plurality of instructions are stored that when executed by the processor enable the mobile hand-held device to perform operations including,

rendering a browser interface via which a user is enabled to request access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

retrieving the Web page via the wireless communications device, and processing HTML-based Web content to produce scalable content; and

employing at least one of the scalable content or data derived therefrom to,

render the Web page on the display; and

re-render the display in response to associated user inputs to enable the Web page to be browsed at various zoom levels and panned views while preserving the original page layout, functionality, and design of the Web page content at each zoom level and panned view.

37. The mobile hand-held device of claim **36**, wherein the device comprises a mobile phone.

38. The mobile hand-held device of claim **36**, wherein the device comprises one of a Personal Digital Assistant (PDA) or hand-held computer.

39. The mobile hand-held device of claim **36**, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input.

40. The mobile hand-held device of claim **39**, wherein the user interface input enables the user to define an area of a current view of the Web page on which to zoom in on.

41. The mobile hand-held device of claim **36**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

42. The mobile hand-held device of claim **36**, wherein the Web page includes at least one hyperlink, and wherein execution of the instructions performs further operations comprising:

enabling the user to select the hyperlink via the display; and, in response thereto,

retrieving and processing HMTL-based Web content associated with the hyperlink to produce additional scalable content; and

employing at least one of the additional scalable content or data derived therefrom to render the Web content associated with the hyperlink on the display.

43. The mobile hand-held device of claim **36**, wherein at least a portion of the scalable content comprises scalable vector-based content.

44. The mobile hand-held device of claim **36**, wherein execution of the instructions performs further operations comprising returning the display of the Web page to a previous view in response to a corresponding user input made via the display.

45. The mobile hand-held device of claim **36**, wherein execution of the instructions performs further operations comprising enabling a user to pan a view of the Web page in response to a corresponding user input made via the display.

46. The mobile hand-held device of claim **36**, wherein execution of the instructions performs further operations comprising enabling the panned view of the Web page to be panned in real-time.

47. The mobile hand-held device of claim **36**, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein said at least one of scalable content or data derived therefrom is scaled to render a display of the Web page having a different aspect ratio.

48. The mobile hand-held device of claim **36**, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a column of the Web content via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is rendered to fit across the display.

49. The mobile hand-held device of claim **48**, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

50. The mobile hand-held device of claim **48**, wherein the corresponding user input comprises tapping on the column via the display.

51. The mobile hand-held device of claim **36**, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to zoom on an image via a corresponding user input, wherein in response thereto, the display is re-rendered such that the image is rendered to fit across the display.

52. The mobile hand-held device of claim **51**, wherein the corresponding user input comprises tapping on the image via the display.

53. The mobile hand-held device of claim **36**, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a paragraph of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is rendered to fit across a display area of the display.

54. The mobile hand-held device of claim **53**, wherein the content of the paragraph is reformatted to fit characteristics of the display area when the display is re-rendered.

55. The mobile hand-held device of claim **53**, wherein the corresponding user input comprises tapping on the paragraph via the display.

56. The mobile hand-held device of claim **36**, wherein the Web page includes text, layout attributes, and images, and wherein execution of the instructions performs further operations comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

57. The mobile hand-held device of claim **36**, further comprising dynamic memory having at least a portion employed for rendering purposes, wherein execution of the instructions performs further operations comprising:

building a display list via use of the scalable content and rendering display list content on a virtual display in the dynamic memory; and

scaling the display list content to re-render the display of the Web page.

58. The mobile hand-held device of claim **36**, wherein execution of the instructions performs further operations comprising:

parsing HTML-based code corresponding to the retrieved Web page to determine the original page layout of the content on the Web page;

logically grouping selected content into objects;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to a layout location datum for the object's associated display content;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to the vector that is generated.

59. The mobile hand-held device of claim **58**, wherein execution of the instructions performs further operations comprising:

generating a bounding box for each object, the bounding box representing a portion of a rendered display page occupied by the object's associated group of content.

60. The mobile hand-held device of claim **59**, further comprising dynamic memory having at least a portion employed for rendering purposes, wherein execution of the instructions performs further operations comprising:

mapping the object vectors and associated bounding boxes to a virtual display in the dynamic memory.

61. The mobile hand-held device of claim **60**, wherein execution of the instructions performs further operations comprising:

enabling a user to view the Web page at a user-selectable zoom level and panned view by,

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and panned view corresponding to a rendered display of the Web page desired by a user;

determining a virtual display bounding box for the virtual display associated with the first scale factor and offset;

identifying object bounding boxes having at least a portion falling within the virtual display bounding box; and,

for each of such object bounding boxes,

retrieving content associated with that object bounding box;

applying an appropriate scale factor to the content associated with that object bounding box to produce scaled content; and

rendering the portion of scaled content within the virtual display bounding box to render the content on the display.

62. The mobile hand-held device of claim **36**, wherein the scalable content includes scalable text content, and wherein execution of the instructions performs further operations comprising scaling a scalable font to render the scalable text content.

63. The mobile hand-held device of claim **36**, wherein the original format of the Web page defines a height and width for the Web page, and wherein execution of the instructions performs further operations comprising:

determining an applicable scale factor to display at least one of the width and height of the Web page to fit across a browser display area of the display; and

employing the scale factor to render the browser display area.

64. The mobile hand-held device of claim **36**, wherein at least a portion of the instructions comprise Java-based instructions.

65. The mobile hand-held device of claim **36**, wherein a portion of the HTML-based Web content comprises XML code.

66. The mobile hand-held device of claim **36**, wherein a portion of the HTML-based Web content comprises cascaded style sheet data defining aspects of the Web page design that are preserved at each zoom level and panned view.

67. The mobile hand-held device of claim **36**, wherein the network comprises a Local Area Network or Wide Area Network.

68. The mobile hand-held device of claim **36**, wherein the device comprises one of a notebook computer or laptop computer.

69. The mobile device of claim **36**, wherein the device enables a user to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web a e content of each Web page.

70. A mobile device, comprising:

processing means;

wireless communications means, to facilitate wireless communication with a network that supports access to the Internet;

a display, to facilitate user input and display rendered content; and

storage means, in which a plurality of instructions are stored,

wherein, upon execution of the instructions by the processing means, the mobile device is enabled to perform operations, including,

rendering a browser interface via which a user is enabled to request access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

retrieving the Web page via the wireless communications means, and processing at least a portion of the HTML-based Web content to produce scalable content; and employing at least one of the scalable content or data derived therefrom to,

render the Web page on the display; and

re-render the display in response to associated user inputs made via the display to enable the Web page to be browsed at various zoom levels and panned views while preserving the original page layout, functionality, and design of the Web page content at each zoom level and panned view of the Web page.

71. The mobile device of claim **70**, wherein the processing means includes a general-purpose processor.

72. The mobile device of claim **70**, wherein the processing means includes a special-purpose processor.

73. The mobile device of claim **70**, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input.

74. The mobile device of claim **73**, wherein the user interface input enables the user to define an area of a current view of the Web page on which to zoom in on.

75. The mobile device of claim **70**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

76. The mobile device of claim **70**, wherein execution of the instructions performs further operations comprising enabling a user to pan a view of the Web content in response to a corresponding user interface input made via the display.

77. The mobile device of claim **70**, wherein execution of the instructions performs further operations comprising enabling the view of the Web content to be panned in real-time.

78. The mobile device of claim **70**, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to zoom on an image via a corresponding user input, wherein in response thereto, the display is re-rendered such that the image is rendered to fit across the display.

79. The mobile device of claim **78**, wherein the corresponding user input comprises tapping on the image via the display.

80. The mobile device of claim **70**, further comprising dynamic memory having at least a portion employed for rendering purposes, wherein execution of the instructions performs further operations comprising:

building a display list via use of the scalable content and rendering display list objects on a virtual display in the dynamic memory; and

scaling display list objects to re-render the display of the Web page.

81. The mobile device of claim **70**, wherein the network comprises a mobile service provider network.

82. The mobile device of claim **70**, wherein the device comprises a mobile phone.

83. The mobile device of claim **70**, wherein the device comprises one of a Personal Digital Assistant (PDA) or hand-held computer.

84. The mobile device of claim **70**, wherein a portion of the scalable content comprises vector-based content.

85. The mobile device of claim **70**, wherein the processing means includes logic circuitry programmed with a portion of the instructions.

86. The mobile device of claim **70**, wherein the device comprises one of a notebook computer or laptop computer.

87. A mobile hand-held device, comprising:

a processor,

a wireless communications interface, to facilitate wireless communication with a network that supports access to the Internet;

a display; and

non-volatile memory, operatively coupled to the processor, in which a plurality of instructions are stored that when executed by the processor enable the mobile hand-held device to perform operations including,

rendering a browser interface on the display via which a user is enabled to request access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

in response to a user request of the Web page,

retrieving the Web page via the wireless communications interface;

rendering the Web page such that a width of the Web page is rendered to fit across the display; and

re-rendering the display in response to associated user inputs to enable the Web page to be browsed at various zoom levels and panned views while preserving the original page layout, functionality, and design of the Web a e content at each zoom level and panned view.

88. The mobile hand-held device of claim **87**, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a column of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is enlarged.

89. The mobile hand-held device of claim **88**, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

90. The mobile hand-held device of claim **88**, wherein the corresponding user input comprises tapping on the column via the display.

91. The mobile hand-held device of claim **88**, wherein the display is re-rendered such that content corresponding to the selected column is rendered to fit across the display.

92. The mobile hand-held device of claim **87**, wherein the Web page includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to zoom on an image via a corresponding user input, wherein in response thereto, the display is re-rendered such that the image is enlarged.

93. The mobile hand-held device of claim **92**, wherein the corresponding user input comprises tapping on the image via the display.

94. The mobile hand-held device of claim **92**, wherein the display is re-rendered such that the image is rendered to fit across the display.

95. The mobile hand-held device of claim **87**, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a paragraph of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is enlarged.

96. The mobile hand-held device of claim **95**, wherein the content of the paragraph is reformatted to fit characteristics of the display when re-rendered.

97. The mobile hand-held device of claim **95**, wherein the corresponding user input comprises tapping on the paragraph via the display.

98. The mobile hand-held device of claim **95**, wherein the display is re-rendered such that content corresponding to the selected paragraph is rendered to fit across the display.

99. The mobile hand-held device of claim **87**, wherein execution of the instructions performs further operations comprising enabling a user to pan a view of the web page while in a zoomed state under which a portion of the web page is displayed.

100. The mobile hand-held device of claim **87**, wherein execution of the instructions performs further operations comprising returning the display of the Web page to a previous view in response to a corresponding user input.

101. The mobile hand-held device of claim **87**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

102. The mobile hand-held device of claim **87**, wherein execution of the instructions performs further operations comprising enabling a user to pan a view of the Web page in response to a corresponding user input made via the display.

103. The mobile hand-held device of claim **87**, wherein execution of the instructions performs further operations comprising enabling the view of the Web page to be panned in real-time.

104. The mobile hand-held device of claim **87**, wherein the Web page includes text, layout attributes, and images, and wherein execution of the instructions performs further operations comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

105. The mobile hand-held device of claim **87**, wherein a portion of the HTML-based Web content comprises XML code.

106. The mobile hand-held device of claim **87**, wherein a portion of the HTML-based Web content comprises cascaded style sheet data defining aspects of the Web page design that are preserved at each zoom level and panned view.

107. The mobile hand-held device of claim **87**, wherein the network comprises a mobile service provider network.

108. The mobile hand-held device of claim **87**, wherein the device comprises a mobile phone.

109. The mobile hand-held device of claim **87**, wherein the device comprises one of a Personal Digital Assistant (PDA) or hand-held computer.

110. The mobile device of claim **87**, wherein the network comprises a Local Area Network or a Wide Area Network.

111. The mobile hand-held device of claim **87**, wherein the device enables a user to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page.

112. A wireless device, comprising:

a processor;

a wireless communications interface, to facilitate wireless communication with a network that supports access to the Internet;

a display;

memory; and

a storage device, on which a plurality of instructions are stored that when executed by the processor enable the wireless device to perform operations including,

rendering a browser interface via which a user is enabled to request access to a Web page, the Web page comprising HTML-based Web content having an original format including HTML code defining an original page layout, functionality, and design of corresponding content on the Web page;

retrieving, via the wireless communications interface, and translating at least a portion of the HTML-based Web content into scalable content that supports a scalable resolution-independent representation of the Web page that preserves the original page layout, functionality and design of the content defined by its original format when scaled and rendered;

employing the scalable content to render the Web page on the display using a first scale factor; and

enabling the Web page to be displayed at a different resolution by scaling the scalable content using a second scale factor to re-render the display,

wherein the original page layout, functionality, and design of the Web page content are preserved under both the first and second scale factors.

113. The wireless device of claim **112**, wherein the display is re-rendered in real-time.

114. The wireless device of claim **112**, wherein the device comprises a hand-held device.

115. The wireless device of claim **112**, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

116. The wireless device of claim **112**, wherein execution of the instructions performs further operations comprising enabling a user to pan a view of the Web page in response to a corresponding user input.

117. The wireless device of claim **116**, wherein execution of the instructions performs further operations comprising enabling the view of the Web page to be panned in real-time.

118. A method, comprising:

rendering a browser interface on a display of a device via which a user is enabled to request access to a Web page, the Web page comprising HTML-based Web content having an original format defining an original width and height of the Web page and an original page layout, functionality, and design of content on the Web page;

in response to a user request to access the Web page,

retrieving the Web page via the device, and translating at least a portion of the HTML-based Web content from its original format into scalable content that supports a scalable resolution-independent representation of the Web page that preserves the original page layout, functionality and design of the content defined by its original format when scaled and rendered; and

scaling the scalable content to render the Web page on the display such that a width of the Web page is rendered to fit across the display.

119. The method of claim **118**, further comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a corresponding user interface input.

120. The method of claim **119**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

121. The method of claim **118**, wherein the Web page includes at least one hyperlink, the method further comprising:

enabling the user to select the hyperlink; and, in response thereto,

retrieving and translating Web content associated with the hyperlink to produce additional scalable content; and

employing the additional scalable content to render the Web content associated with the hyperlink on the display.

122. The method of claim **118**, further comprising:

parsing markup language code to determine the original page layout of display content within the Web page, wherein the original page layout defines a layout location for a plurality of objects, including at least one of text objects, graphic layout objects, or graphic image objects included in the Web page;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to the layout location for the object;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to the vector that is generated.

123. The method of claim **118**, further comprising enabling the Web page to be displayed at different resolutions by scaling the scalable content to re-render the display in response to associated user inputs,

wherein the original page layout, functionality, and design of the Web page content are preserved at each of the different resolutions.

124. The method of claim **118**, further comprising returning the display of the Web page to a previous view in response to a corresponding user input.

125. The method of claim **118**, further comprising enabling a user to pan a view of the Web page in response to a corresponding user input.

126. The method of claim **125**, further comprising enabling the view of the Web page to be panned in real-time.

127. The method of claim **118**, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein the scalable content is scaled when rendered so as to produce a display of the Web page having a different aspect ratio.

128. The method of claim **118**, further comprising enabling a user to zoom on a column of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is rendered to fit across the display.

129. The method of claim **128**, wherein the corresponding user input comprises tapping on the column via the display.

130. The method of claim **128**, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

131. The method of claim **118**, wherein the Web content includes at least one image, the method further comprising enabling a user to zoom on an image via a corresponding user input, wherein in response thereto, the display is re-rendered such that the image is rendered to fit across the display.

132. The method of claim **131**, wherein the corresponding user input comprises tapping on the image via the display.

133. The method of claim **118**, further comprising enabling a user to zoom on a paragraph of the Web content via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is rendered to fit across the display.

134. The method of claim **133**, wherein the corresponding user input comprises tapping on the paragraph via the display.

135. The method of claim **133**, wherein the content of the paragraph is reformatted to fit characteristics of the display when the display is re-rendered.

136. The method of claim **118**, wherein the Web page includes text, layout attributes, and images, the method further comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

137. The method of claim **118**, further comprising:

generating a vector-based display list associated with the scalable content; and

employing the display list to re-render the display at different scale factors to zoom the Web page.

138. The method of claim **118**, further comprising:

parsing markup language code corresponding to the retrieved Web page to determine the original page layout of the content on the Web page;

logically grouping selected content into objects;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to a layout location datum for the object's associated display content;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to the vector that is generated.

139. The method of claim **138**, further comprising:

generating a bounding box for each object, the bounding box representing a portion of a rendered display page occupied by the object's associated group of content.

140. The method of claim **139**, further comprising:

mapping the object vectors and associated bounding boxes to a virtual display in memory.

141. The method of claim **140**, further comprising:

enabling a user to view the Web page at a user-selectable zoom level and panned view by,

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and panned view corresponding to a rendered display of the Web page desired by a user; and

determining a virtual display bounding box for the virtual display associated with the first scale factor and offset;

identifying object bounding boxes having at least a portion falling within the virtual display bounding box; and,

for each of such object bounding boxes,

retrieving content associated with that object bounding box;

applying an appropriate scale factor to the content associated with that object bounding box to produce scaled content; and

rendering the portion of scaled content within the virtual display bounding box to render the content on the display.

142. The method of claim **118**, wherein the scalable content includes scalable text content, the method further comprising scaling a scalable font to render the scalable text content.

143. The method of claim **118**, wherein the method is facilitated, at least in part, via execution of Java-based instructions.

144. The method of claim **118**, wherein the device comprises a mobile phone.

145. The method of claim **118**, wherein the device comprises a hand-held device.

146. The method of claim **118**, further comprising accessing the Internet via a wireless connection to retrieve the Web page.

147. The method of claim **118**, wherein a portion of the scalable content comprises vector-based content.

148. The method of claim **118**, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

149. A method, comprising:

rendering a browser interface on a hand-held device via which a user is enabled to request access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

retrieving the Web page via the hand-held device, and processing HTML-based Web content to produce scalable content; and

employing at least one of the scalable content or data derived therefrom to,

render the Web page on a display of the hand-held device; and

re-render the display in response to associated user inputs to enable the Web page to be browsed at various zoom levels and panned views while preserving the original page layout, functionality, and design of the Web page content at each zoom level and panned view.

150. The method of claim **149**, wherein the hand-held device comprises a mobile phone.

151. The method of claim **149**, wherein the hand-held device comprises one of a Personal Digital Assistant (PDA) or hand-held computer.

152. The method of claim **149**, further comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input.

153. The method of claim **152**, wherein the user interface input enables the user to define an area of a current view of the Web page on which to zoom in on.

154. The method of claim **149**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

155. The method of claim **149**, wherein the Web page includes at least one hyperlink, the method further comprising:

enabling the user to select the hyperlink via the display; and, in response thereto,

retrieving and processing HMTL-based Web content associated with the hyperlink to produce additional scalable content; and

employing at least one of the additional scalable content or data derived therefrom to render the Web content associated with the hyperlink on the display.

156. The method of claim **149**, wherein at least a portion of the scalable content comprises scalable vector-based content.

157. The method of claim **149**, further comprising returning the display of the Web page to a previous view in response to a corresponding user input made via the display.

158. The method of claim **149**, further comprising enabling a user to pan a view of the Web page in response to a corresponding user input made via the display.

159. The method of claim **149**, further comprising enabling the view of the Web page to be panned in real-time.

160. The method of claim **149**, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein said at least one of scalable content or data derived therefrom is scaled to render a display of the Web page having a different aspect ratio.

161. The method of claim **149**, further comprising enabling a user to zoom on a column of the Web content via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is enlarged.

162. The method of claim **161**, wherein the corresponding user input comprises tapping on the column via the display.

163. The method of claim **161**, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

164. The method of claim **161**, wherein the display is re-rendered such that content corresponding to the selected column is rendered to fit across the display.

165. The method of claim **149**, wherein the Web content includes at least one image, the method further comprising enabling a user to zoom on an image via a corresponding user input, wherein in response thereto, the display is re-rendered such that the image is enlarged.

166. The method of claim **165**, wherein the corresponding user input comprises tapping on the image via the display.

167. The method of claim **165**, wherein the display is re-rendered such that the image is rendered to fit across the display.

168. The method of claim **149**, further comprising enabling a user to zoom on a paragraph of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is enlarged.

169. The method of claim **168**, wherein the content of the paragraph is reformatted to fit characteristics of the display area when the display is re-rendered.

170. The method of claim **168**, wherein the display is re-rendered such that content corresponding to the selected paragraph is rendered to fit across the display.

171. The method of claim **168**, wherein the corresponding user input comprises tapping on the paragraph via the display.

172. The method of claim **149**, wherein the Web page includes text, layout attributes, and images, the method further comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

173. The method of claim **149**, wherein the hand-held device includes dynamic memory having at least a portion employed for rendering purposes, the method further comprising:

building a display list via use of the scalable content and rendering display list content on a virtual display in dynamic memory; and

scaling the display list content to re-render the display of the Web page.

174. The method of claim **149**, further comprising:

parsing HTML-based code corresponding to the retrieved Web page to determine the original page layout of the content on the Web page;

logically grouping selected content into objects;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to a layout location datum for the object's associated display content;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to the vector that is generated.

175. The method of claim **174**, further comprising:

generating a bounding box for each object, the bounding box representing a portion of a rendered display page occupied by the object's associated group of content.

176. The method of claim **175**, wherein the hand-held device includes dynamic memory having at least a portion employed for rendering purposes, the method further comprising:

mapping the object vectors and associated bounding boxes to a virtual display in the dynamic memory.

177. The method of claim **176**, further comprising:

enabling a user to view the Web page at a user-selectable zoom level and panned view by,

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and panned view corresponding to a rendered display of the Web page desired by a user;

determining a virtual display bounding box for the virtual display associated with the first scale factor and offset;

identifying object bounding boxes having at least a portion falling within the virtual display bounding box; and,

for each of such object bounding boxes,

retrieving content associated with that object bounding box;

applying an appropriate scale factor to the content associated with that object bounding box to produce scaled content; and

rendering the portion of scaled content within the virtual display bounding box to render the content on the display.

178. The method of claim **149**, wherein the scalable content includes scalable text content, the method further comprising scaling a scalable font to render the scalable text content.

179. The method of claim **149**, wherein the original format of the Web page defines a height and width for the Web page, the method further comprising:

determining an applicable scale factor to display at least one of the width and height of the Web page across a browser display area of the display; and

employing the scale factor to render the browser display area.

180. The method of claim **149**, wherein the method is facilitated, at least in part, via execution of Java-based instructions.

181. The method of claim **149**, wherein a portion of the HTML-based Web content comprises XML code.

182. The method of claim **149**, wherein a portion of the HTML-based Web content comprises cascaded style sheet data defining aspects of the Web page design that are preserved at each zoom level and panned view.

183. The method of claim **149**, wherein a portion of the scalable content comprises vector-based content.

184. The method of claim **149**, further comprising enabling a user to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page.

185. A method, comprising:

rendering a browser interface on a display of a device via which a user is enabled to request access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

in response to a user request of the Web page via the browser interface,

retrieving the Web page via the device;

rendering the Web page via the device such that a full width of the Web page is rendered on the display; and

re-rendering the Web page in response to associated user inputs to the hand-held device to enable the Web page to be browsed at various zoom levels and panned views while preserving the original page layout, functionality, and design of the Web page content at each zoom level and panned view of the Web page,

wherein the method enables a user of the device to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page.

186. The method of claim **185**, further comprising enabling a user to zoom on a column of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is enlarged.

187. The method of claim **186**, wherein the corresponding user input comprises tapping on the column via the display.

188. The method of claim **186**, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

189. The method of claim **186**, wherein the display is re-rendered such that content corresponding to the selected column is rendered to fit across the display.

190. The method of claim **185**, wherein the Web page includes at least one image, the method further comprising enabling a user to zoom on an image via a corresponding user input, wherein in response thereto, the display is re-rendered such that the image is enlarged.

191. The method of claim **190**, wherein the corresponding user input comprises tapping on the image via the display.

192. The method of claim **190**, wherein the display is re-rendered such that the image is rendered to fit across the display.

193. The method of claim **185**, further comprising enabling a user to zoom on a paragraph of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is enlarged.

194. The method of claim **193**, wherein the corresponding user input comprises tapping on the paragraph via the display.

195. The method of claim **193**, wherein the content of the paragraph is reformatted to fit characteristics of the display when re-rendered.

196. The method of claim **193**, wherein the display is re-rendered such that content corresponding to the selected paragraph is rendered to fit across the display.

197. The method of claim **185**, further comprising enabling a user to pan a view of the web page while in a zoomed state under which a portion of the web page is displayed in response to a user input made via the display.

198. The method of claim **185**, further comprising returning the display of the Web page to a previous view in response to a corresponding user input.

199. The method of claim **185**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

200. The method of claim **185**, further comprising enabling a user to pan a view of the Web page in response to a corresponding user input made via the display.

201. The method of claim **185**, further comprising enabling the view of the Web page to be panned in real-time.

202. The method of claim **185**, wherein the Web page includes text, layout attributes, and images, the method further comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

203. The method of claim **185**, wherein a portion of the HTML-based Web content comprises XML code.

204. The method of claim **185**, wherein a portion of the HTML-based Web content comprises cascaded style sheet data defining aspects of the Web page design that are preserved at each zoom level and panned view.

205. The method of claim **185**, wherein the Web page is retrieved via a wireless connection to one of a mobile service provider network, local area network, or wide area network.

206. The method of claim **185**, wherein the device comprises a mobile phone.

207. The method of claim **185**, wherein the device comprises a hand-held device.

208. The method of claim **185**, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

209. A method, comprising:

rendering a browser interface on a display via which a user of a device is enabled to request access to a Web page, the Web page comprising HTML-based Web content having an original format including HTML code defining an original page layout, functionality, and design of corresponding content on the Web page;

retrieving the Web page, via the device, and translating at least a portion of the HTML-based Web content into scalable content that supports a scalable resolution-independent representation of the Web page that preserves the original page layout, functionality and design of the content defined by its original format when scaled and rendered; and

employing the scalable content to render the Web page on the display using a first scale factor; and

enabling the Web page to be displayed at a different resolution by scaling the scalable content using a second scale factor to re-render the display,

wherein the original page layout, functionality, and design of the Web page content are preserved under both the first and second scale factors.

210. The method of claim **209**, wherein the display is re-rendered in real-time.

211. The method of claim **209**, wherein the device comprises a hand-held device.

212. The method of claim **209**, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

213. The method of claim **209**, further comprising enabling a user to pan a view of the Web page in response to a corresponding user input.

214. The method of claim **213**, further comprising enabling the view of the Web page to be panned in real-time.

215. The method of claim **209**, further comprising enabling a user to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page.

216. A machine-readable medium having a plurality of instructions tangibly stored thereon, which when executed enable a device to perform operations comprising:

rendering a browser interface via which a user is enabled to request access to a Web page hosted by an Internet Web site, the Web page comprising HTML-based Web content having an original format defining an original width and height of the Web page and an original page layout, functionality, and design of content on the Web page;

retrieving the Web page via the wireless communication means, and translating at least a portion of the HTML-based Web content from its original format into scalable content that supports a scalable resolution-independent representation of the Web page that preserves the original page layout, functionality and design of the content defined by its original format when scaled and rendered; and

scaling the scalable content to render the Web page on the display such that a width of the Web page is rendered to fit across the display.

217. The machine-readable medium of claim **216**, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a corresponding user interface input.

218. The machine-readable medium of claim **217**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

219. The machine-readable medium of claim **216**, wherein the Web page includes at least one hyperlink, and wherein execution of the instructions performs further operations comprising:

enabling the user to select the hyperlink; and, in response thereto,

retrieving and translating Web content associated with the hyperlink to produce additional scalable content; and

employing the additional scalable content to render the Web content associated with the hyperlink on the display.

220. The machine-readable medium of claim **216**, wherein execution of the instructions performs further operations comprising:

parsing markup language code to determine the original page layout of display content within the Web page, wherein the original page layout defines a layout location for a plurality of objects, including at least one of text objects, graphic layout objects, or graphic image objects included in the Web page;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to the layout location for the object;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to the vector that is generated.

221. The machine-readable medium of claim **216**, wherein execution of the instructions performs further operations comprising enabling the Web page to be displayed at different resolutions by scaling the scalable content to re-render the display in response to associated user inputs,

wherein the original page layout, functionality, and design of the Web page content are preserved at each of the different resolutions.

222. The machine-readable medium of claim **216**, wherein execution of the instructions performs further operations comprising returning the display of the Web page to a previous view in response to a corresponding user input.

223. The machine-readable medium of claim **216**, wherein execution of the instructions performs further operations comprising enabling a user to pan a view of the Web page in response to a corresponding user input.

224. The machine-readable medium of claim **223**, wherein execution of the instructions performs further operations comprising enabling the view of the Web page to be panned in real-time.

225. The machine-readable medium of claim **216**, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein the scalable content is scaled when rendered so as to produce a display of the Web page having a different aspect ratio.

226. The machine-readable medium of claim **216**, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a column of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is enlarged.

227. The machine-readable medium of claim **226**, wherein the corresponding user input comprises tapping on the column via the display.

228. The machine-readable medium of claim **226**, wherein the display is re-rendered such that content corresponding to the selected column is rendered to fit across the display.

229. The machine-readable medium of claim **226**, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

230. The machine-readable medium of claim **216**, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to zoom on an image via a corresponding user input, wherein in response thereto, the display is re-rendered such that the image is enlarged.

231. The machine-readable medium of claim **230**, wherein the corresponding user input comprises tapping on the image via the display.

232. The machine-readable medium of claim **230**, wherein the display is re-rendered such that the image is rendered to fit across the display.

233. The machine-readable medium of claim **216**, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a paragraph of the Web content via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is enlarged.

234. The machine-readable medium of claim **233**, wherein the corresponding user input comprises tapping on the paragraph via the display.

235. The machine-readable medium of claim **233**, wherein the content of the paragraph is reformatted to fit characteristics of the display when the display is re-rendered.

236. The machine-readable medium of claim **233**, wherein the display is re-rendered such that content corresponding to the selected paragraph is rendered to fit across the display.

237. The machine-readable medium of claim **216**, wherein the Web page includes text, layout attributes, and images, and wherein execution of the instructions performs further operations comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

238. The machine-readable medium of claim **216**, wherein execution of the instructions performs further operations comprising:

generating a vector-based display list associated with the scalable content; and

employing the display list to re-render the display at different scale factors to zoom the Web page.

239. The machine-readable medium of claim **216**, wherein execution of the instructions performs further operations comprising:

parsing markup language code corresponding to the retrieved Web page to determine the original page layout of the content on the Web page;

logically grouping selected content into objects;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to a layout location datum for the object's associated display content;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to the vector that is generated.

240. The machine-readable medium of claim **239**, wherein execution of the instructions performs further operations comprising:

generating a bounding box for each object, the bounding box representing a portion of a rendered display page occupied by the object's associated group of content.

241. The machine-readable medium of claim **240**, wherein execution of the instructions performs further operations comprising:

mapping the object vectors and associated bounding boxes to a virtual display in memory.

242. The machine-readable medium of claim **241**, wherein execution of the instructions performs further operations comprising:

enabling a user to view the Web page at a user-selectable zoom level and panned view by,

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and panned view corresponding to a rendered display of the Web page desired by a user; and

determining a virtual display bounding box for the virtual display associated with the first scale factor and offset;

identifying object bounding boxes having at least a portion falling within the virtual display bounding box; and,

for each of such object bounding boxes,

retrieving content associated with that object bounding box;

applying an appropriate scale factor to the content associated with that object bounding box to produce scaled content; and

rendering the portion of scaled content within the virtual display bounding box to render the content on the display.

243. The machine-readable medium of claim **216**, wherein the scalable content includes scalable text content, and wherein execution of the instructions performs further operations comprising scaling a scalable font to render the scalable text content.

244. The machine-readable medium of claim **216**, wherein at least a portion of the instructions comprise Java-based instructions.

245. The machine-readable medium of claim **216**, wherein the device comprises a mobile phone.

246. The machine-readable medium of claim **216**, wherein the device comprises a hand-held device.

247. The machine-readable medium of claim **216**, wherein the Web page is accessed via a mobile service provider network.

248. The machine-readable medium of claim **216**, wherein a portion of the scalable content comprises vector-based content.

249. The machine-readable medium of claim **216**, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

250. The machine-readable medium of claim **216**, wherein the instructions are embodied as a Web browser.

251. The machine-readable medium of claim **216**, wherein execution of the instructions enables a user to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page.

252. A machine-readable medium having a plurality of instructions comprising a Web browser tangibly stored thereon, which when executed enable a device to perform operations comprising:

rendering a browser interface on a display associated with the device via which a user is enabled to request access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

retrieving the Web page and processing HTML-based Web content to produce scalable content; and

employing at least one of the scalable content or data derived therefrom to,

render the Web page on the display; and

re-render the display in response to associated user inputs to enable the Web page to be browsed at various zoom levels and panned views while preserving the original page layout, functionality, and design of the Web page content at each zoom level and panned view of the Web page,

wherein the Web browser enables a user of the device to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page at each zoom level and panned view.

253. The machine-readable medium of claim **252**, wherein the device comprises a mobile phone.

254. The machine-readable medium of claim **252**, wherein the device comprises a hand-held device.

255. The machine-readable medium of claim **252**, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input.

256. The machine-readable medium of claim **255**, wherein the user interface input enables the user to define an area of a current view of the Web page on which to zoom in on.

257. The machine-readable medium of claim **252**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

258. The machine-readable medium of claim **252**, wherein the Web page includes at least one hyperlink, and wherein execution of the instructions performs further operations comprising:

enabling the user to select the hyperlink; and, in response thereto,

retrieving and processing HMTL-based Web content associated with the hyperlink to produce additional scalable content; and

employing at least one of the additional scalable content or data derived therefrom to render the Web content associated with the hyperlink on the display.

259. The machine-readable medium of claim **252**, wherein at least a portion of the scalable content comprises scalable vector-based content.

260. The machine-readable medium of claim **252**, wherein execution of the instructions performs further operations comprising returning the display of the Web page to a previous view in response to a corresponding user input made via the display.

261. The machine-readable medium of claim **252**, wherein execution of the instructions performs further operations comprising enabling a user to pan a view of the Web page in response to a corresponding user input made via the display.

262. The machine-readable medium of claim **252**, wherein execution of the instructions performs further operations comprising enabling the view of the Web page to be panned in real-time.

263. The machine-readable medium of claim **252**, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein said at least one of scalable content or data derived therefrom is scaled to render a display of the Web page having a different aspect ratio.

264. The machine-readable medium of claim **252**, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a column of the Web content via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is rendered to fit across the display.

265. The machine-readable medium of claim **264**, wherein the corresponding user input comprises tapping on the column via the display.

266. The machine-readable medium of claim **264**, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

267. The machine-readable medium of claim **252**, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to zoom on an image via a corresponding user input, wherein in response thereto, the display is re-rendered such that the image is rendered to fit across the display.

268. The machine-readable medium of claim **267**, wherein the corresponding user input comprises tapping on the image via the display.

269. The machine-readable medium of claim **252**, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a paragraph of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is rendered to fit across a browser display area of the display.

270. The machine-readable medium of claim **269**, wherein the corresponding user input comprises tapping on the paragraph via the display.

271. The machine-readable medium of claim **269**, wherein the content of the paragraph is reformatted to fit characteristics of the display area when the display is re-rendered.

272. The machine-readable medium of claim **252**, wherein the Web page includes text, layout attributes, and images, and wherein execution of the instructions performs further operations comprising:

receiving content corresponding to the text and layout attributes via a first connection; and receiving content corresponding to at least one image via a second connection.

273. The machine-readable medium of claim **252**, wherein the device includes dynamic memory having at least a portion employed for rendering purposes, and wherein execution of the instructions performs further operations comprising:

building a display list via use of the scalable content and rendering display list content on a virtual display in the dynamic memory; and

scaling the display list content to re-render the display of the Web page.

274. The machine-readable medium of claim **252**, wherein execution of the instructions performs further operations comprising:

parsing HTML-based code corresponding to the retrieved Web page to determine the original page layout of the content on the Web page;

logically grouping selected content into objects;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to a layout location datum for the object's associated display content;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to the vector that is generated.

275. The machine-readable medium of claim **274**, wherein execution of the instructions performs further operations comprising:

generating a bounding box for each object, the bounding box representing a portion of a rendered display page occupied by the object's associated group of content.

276. The machine-readable medium of claim **275**, wherein the device includes dynamic memory having at least a portion employed for rendering purposes, and wherein execution of the instructions performs further operations comprising:

mapping the object vectors and associated bounding boxes to a virtual display in the dynamic memory.

277. The machine-readable medium of claim **276**, wherein execution of the instructions performs further operations comprising:

enabling a user to view the Web page at a user-selectable zoom level and panned view by,

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and panned view corresponding to a rendered display of the Web page desired by a user;

determining a virtual display bounding box for the virtual display associated with the first scale factor and offset;

identifying object bounding boxes having at least a portion falling within the virtual display bounding box; and,

for each of such object bounding boxes,

retrieving content associated with that object bounding box;

applying an appropriate scale factor to the content associated with that object bounding box to produce scaled content; and

rendering the portion of scaled content within the virtual display bounding box to render the content on the display.

278. The machine-readable medium of claim **252**, wherein the scalable content includes scalable text content, and wherein execution of the instructions performs further operations comprising scaling a scalable font to render the scalable text content.

279. The machine-readable medium of claim **252**, wherein the original format of the Web page defines a height and width for the Web page, and wherein execution of the instructions performs further operations comprising:

determining an applicable scale factor to display at least one of the width and height of the Web page across a browser display area of the display; and

employing the scale factor to render the browser display area.

280. The machine-readable medium of claim **252**, wherein at least a portion of the instructions comprise Java-based instructions.

281. The machine-readable medium of claim **252**, wherein a portion of the HTML-based Web content comprises XML code.

282. The machine-readable medium of claim **252**, wherein a portion of the HTML-based Web content comprises cascaded style sheet data defining aspects of the Web page design that are preserved at each zoom level and panned view.

283. The machine-readable medium of claim **252**, wherein a portion of the scalable content comprises vector-based content.

284. The machine-readable medium of claim **252**, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

285. A machine-readable medium having a plurality of instructions tangibly stored thereon, which when executed enable a wireless device to perform operations comprising:

rendering a browser interface on a display of the wireless device via which a user is enabled to request access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

in response to a user request of the Web page,

retrieving the Web page via the wireless device;

rendering the Web page such that a width of the Web page is rendered to fit across the display; and

re-rendering the Web page in response to associated user inputs to enable the Web page to be browsed at various zoom levels and panned views while preserving the original page layout, functionality, and design of the Web page content at each zoom level and panned view.

286. The machine-readable medium of claim **285**, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a column of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is enlarged.

287. The machine-readable medium of claim **286**, wherein the corresponding user input comprises tapping on the column via the display.

288. The machine-readable medium of claim **286**, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

289. The machine-readable medium of claim **286**, wherein the display is re-rendered such that content corresponding to the selected column is rendered to fit across the display.

290. The machine-readable medium of claim **285**, wherein the Web page includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to zoom on an image via a corresponding user input, wherein in response thereto, the display is re-rendered such that the image is enlarged.

291. The machine-readable medium of claim **290**, wherein the corresponding user input comprises tapping on the image via the display.

292. The machine-readable medium of claim **290**, wherein the display is re-rendered such that the image is rendered to fit across the display.

293. The machine-readable medium of claim **285**, wherein execution of the instructions performs further operations comprising enabling a user to zoom on a paragraph of the Web page via a corresponding user input, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is enlarged.

294. The machine-readable medium of claim **293**, wherein the corresponding user input comprises tapping on the paragraph via the display.

295. The machine-readable medium of claim **293**, wherein the content of the paragraph is reformatted to fit characteristics of the display when re-rendered.

296. The machine-readable medium of claim **293**, wherein the display is re-rendered such that content corresponding to the selected paragraph is rendered to fit across the display.

297. The machine-readable medium of claim **285**, wherein execution of the instructions performs further operations comprising enabling a user to pan a display of the web page while in a zoomed state under which a portion of the web page is displayed.

298. The machine-readable medium of claim **285**, wherein execution of the instructions performs further operations comprising returning the display of the Web page to a previous view in response to a corresponding user input.

299. The machine-readable medium of claim **285**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

300. The machine-readable medium of claim **285**, wherein execution of the instructions performs further operations comprising enabling a user to pan a view of the Web page in response to a corresponding user input made via the display.

301. The machine-readable medium of claim **285**, wherein execution of the instructions performs further operations comprising enabling the view of the Web page to be panned in real-time.

302. The machine-readable medium of claim **285**, wherein the Web page includes text, layout attributes, and images, and wherein execution of the instructions performs further operations comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

303. The machine-readable medium of claim **285**, wherein a portion of the HTML-based Web content comprises XML code.

304. The machine-readable medium of claim **285**, wherein a portion of the HTML-based Web content comprises cascaded style sheet data defining aspects of the Web page design that are preserved at each zoom level and panned view.

305. The machine-readable medium of claim **285**, wherein the wireless device is configured to connect to a mobile service provider network and retrieve the Web page via the mobile service provider network.

306. The machine-readable medium of claim **285**, wherein the wireless device comprises a mobile phone.

307. The machine-readable medium of claim **285**, wherein the wireless device comprises a hand-held device.

308. The machine-readable medium of claim **285**, wherein the wireless device comprises one of a notebook computer or laptop computer.

309. The machine-readable medium of claim **285**, wherein the instructions are embodied as a Web browser.

310. The machine-readable medium of claim **285**, wherein execution of the instructions enables a user to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page.

311. A machine-readable medium having a plurality of instructions comprising a Web browser stored thereon, which when executed enable a device to perform operations comprising:

launching a Web browser including a browser interface via which a user is enabled to request access to a Web page, the Web page comprising HTML-based Web content having an original format including HTML code defining an original page layout, functionality, and design of corresponding content on the Web page;

retrieving, and translating at least a portion of the HTML-based Web content into scalable content that supports a scalable resolution-independent representation of the Web page that preserves the original page layout, functionality and design of the content defined by its original format when scaled and rendered; and

employing the scalable content to render the Web page on the Web browser using a first scale factor; and

enabling the Web page to be displayed at a different resolution by scaling the scalable content using a second scale factor to re-render the Web page on the Web browser,

wherein the original page layout, functionality, and design of the Web page content are preserved under both the first and second scale factors, and

wherein the Web browser enables a user of the device to browse billions of Web pages at multiple scale factors in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page at each scale factor.

312. The machine-readable medium of claim **311**, wherein the display is re-rendered in real-time.

313. The machine-readable medium of claim **311**, wherein the Web browser is configured to be installed on a hand-held device.

314. The machine-readable medium of claim **311**, wherein the Web browser is configured to be installed on at least one of a desktop computer, notebook computer or laptop computer.

315. The machine-readable medium of claim **311**, wherein execution of the instructions performs further operations comprising enabling a user to pan a view of the Web page in response to a corresponding user input.

316. The machine-readable medium of claim **315**, wherein execution of the instructions performs further operations comprising enabling the view of the Web page to be panned in real-time.

317. A hand-held wireless device, comprising:

a processor,

a wireless communications interface, to facilitate wireless communication with a network that supports access to the Internet;

a display; and

non-volatile memory, operatively coupled to the processor, in which software comprising a browser is stored, the browser comprising a plurality of instructions that when executed by the processor enable the device to perform operations including,

rendering a browser interface on the display via which a user is enabled to request access to a Web page including at least one image, at least one column, at least one hyperlink to an external reference and having a width and height;

retrieving the Web page via the wireless communications interface;

rendering the Web page on the display such that at least one of the width and height of the Web page is fully displayed; and

enabling the user to,

zoom and pan a view of the Web page;

activate a currently displayed hyperlink to an external reference while at a given zoom level and panned view, wherein in response to an activation of a hyperlink to an external reference, Web content associated with the external reference is retrieved and rendered on the display;

zoom in on an image of the Web page by tapping on the image via the display;

zoom in on a column of the Web page by tapping on the column via the display; and

zoom out to a previous view of the Web page.

318. The hand-held wireless device of claim **317**, wherein the Web page comprises HTML-based Web page content defining an original page layout, functionality, and design of the Web page content, and wherein the browser renders the Web page such that the original page layout, functionality, and design of the Web page are preserved at any selectable zoom level.

319. The hand-held wireless device of claim **318**, wherein the user is enabled to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web page content of each Web page.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO.	: 7,461,353 B2	Page 1 of 1
APPLICATION NO.	: 11/045757	
DATED	: December 2, 2008	
INVENTOR(S)	: Gary B. Rohrabaugh and Scott A. Sherman	

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In Claim 69,

at Col. 28, line 43, add --hand-held-- between “mobile” and “device”; and at Col. 28, line 46, delete “a e” between “Web” and “content” such that the claim recites,

“69. The mobile hand-held device of claim 36, wherein the device enables a user to browse, zoom, and pan billions of Web pages in a manner that preserves the original layout, functionality, and design of the HTML-based Web content of each Web page”.

In Claim 185,

at Col. 38, line 7, delete “hand-held” prior to “device” such that the subparagraph recites,

“re-rendering the Web page in response to associated user inputs to the device ...”.

In Claim 216,

Col. 40, line 5, replace “the” wireless communication means with --a-- wireless communications means, such that the subparagraph recites,

“retrieving the Web page via a wireless communication means, ...”.

Signed and Sealed this

Twenty-first Day of April, 2009

John Doll

JOHN DOLL

Acting Director of the United States Patent and Trademark Office

Exhibit 2

US007831926B2

(12) **United States Patent**
Rohrabaugh et al.

(10) **Patent No.: US 7,831,926 B2**
(45) **Date of Patent: *Nov. 9, 2010**

(54) **SCALABLE DISPLAY OF INTERNET CONTENT ON MOBILE DEVICES**

(75) Inventors: **Gary B. Rohrabaugh**, Bellingham, WA (US); **Scott A. Sherman**, Seattle, WA (US)

(73) Assignee: **SoftView LLC**, Bellingham, WA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 799 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **11/738,486**

(22) Filed: **Apr. 21, 2007**

(65) **Prior Publication Data**

US 2007/0288841 A1 Dec. 13, 2007

Related U.S. Application Data

(63) Continuation of application No. 09/878,097, filed on Jun. 8, 2001, now Pat. No. 7,210,099, which is a continuation-in-part of application No. 09/828,511, filed on Apr. 7, 2001, now abandoned.

(60) Provisional application No. 60/211,019, filed on Jun. 12, 2000, provisional application No. 60/217,345, filed on Jul. 11, 2000.

(51) **Int. Cl.**
G06F 17/00 (2006.01)

(52) **U.S. Cl.** **715/800**; 715/234; 715/243; 715/853

(58) **Field of Classification Search** 715/800, 715/238, 249, 204, 234
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,615,384	A	3/1997	Allard et al.
5,897,644	A	4/1999	Blumberg
5,952,994	A	9/1999	Ong et al.
5,956,025	A	9/1999	Goulden et al.
5,966,135	A	10/1999	Roy et al.
6,011,905	A	1/2000	Huttenlocher et al.

(Continued)

FOREIGN PATENT DOCUMENTS

GB	0009129.8		4/2000
GB	2344197	A	5/2000
JP	10-334082	A	12/1998

OTHER PUBLICATIONS

Scalable Vector Graphics (SVG) Specification W3C Working Draft Feb. 11, 1999 http://www.w3.org/TR/1999/WD-SVG-19990211/--Pub Feb. 11, 1999 by W3C pp. 1-7.

Fisher, B., G. Agelidis, J. Dill, P. Tan, G. Collaud and C. Jones. "CZWeb: Fish-Eye Views for Visualiziang the World-Wide Web", Proc. Seventh Int. Conf. on Human-Computer Interaction (HCI International '97), pp. 719-722, 1997.

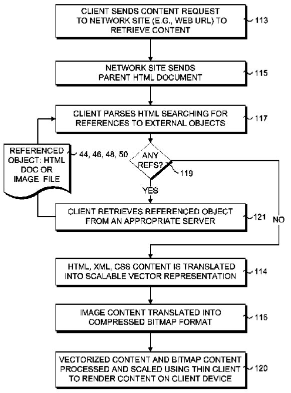
(Continued)

Primary Examiner—Doug Hutton
Assistant Examiner—Quoc A Tran
(74) *Attorney, Agent, or Firm*—Law Office of R. Alan Burnett

(57) **ABSTRACT**

Mobile devices enabled to support resolution-independent scalable display of Internet (Web) content to allow Web pages to be scaled (zoomed) and panned for better viewing on smaller screen sizes. The mobile devices employ software-based processing of original Web content, including HTML-based content, XML, cascade style sheets, etc. to generate scalable content. The scalable content and/or data derived therefrom are then employed to enable the Web content to be rapidly rendered, zoomed, and panned. Display lists may also be employed to provide further enhancements in rendering speed.

88 Claims, 22 Drawing Sheets


CLIENT SENDS CONTENT REQUEST TO NETWORK SITE (E.G., WEB URL) TO RETRIEVE CONTENT
113
NETWORK SITE SENDS PARENT HTML DOCUMENT
115
CLIENT PARSES HTML SEARCHING FOR REFERENCES TO EXTERNAL OBJECTS
117
REFERENCED OBJECT: HTML DOC OR IMAGE FILE
44, 46, 48, 50
ANY REFS?
119
YES
NO
CLIENT RETRIEVES REFERENCED OBJECT FROM AN APPROPRIATE SERVER
121
HTML, XML, CSS CONTENT IS TRANSLATED INTO SCALABLE VECTOR REPRESENTATION
114
IMAGE CONTENT TRANSLATED INTO COMPRESSED BITMAP FORMAT
116
VECTORIZED CONTENT AND BITMAP CONTENT PROCESSED AND SCALED USING THIN CLIENT TO RENDER CONTENT ON CLIENT DEVICE
120

U.S. PATENT DOCUMENTS

6,057,854	A	5/2000	Davis, Jr. et al.
6,076,166	A	6/2000	Moshfeghi et al.
6,211,856	B1	4/2001	Choi et al.
6,300,947	B1	10/2001	Kanevsky
6,337,693	B1	1/2002	Roy et al.
6,421,733	B1	7/2002	Tso et al.
6,449,639	B1	9/2002	Blumberg
6,456,305	B1	9/2002	Qureshi et al.
6,456,308	B1	9/2002	Agranet et al.
6,457,030	B1	9/2002	Adams et al.
6,466,203	B2	10/2002	Van Ee
6,546,397	B1	4/2003	Rempell
6,553,410	B2	4/2003	Kikinis
6,556,217	B1	4/2003	Mäkipää
6,615,212	B1	9/2003	Dutta et al.
6,674,445	B1	1/2004	Chithambaram et al.
6,704,024	B2	3/2004	Robotham et al.
6,781,600	B2	8/2004	Anwar
6,829,746	B1	12/2004	Schwerdtfeger et al.
6,857,102	B1	2/2005	Bickmore et al.
6,886,034	B2	4/2005	Blumberg
6,925,595	B1	8/2005	Whitledge et al.
6,978,418	B1	12/2005	Bain et al.
6,996,533	B2	2/2006	Ikeda et al.
7,023,572	B2	4/2006	Tuli
7,055,095	B1	5/2006	Anwar
7,072,984	B1	7/2006	Polonsky et al.
7,219,309	B2	5/2007	Kaasila et al.
7,308,649	B2	12/2007	Ehrich et al.
7,450,114	B2	11/2008	Anwar
2001/0015719	A1	8/2001	Van Ee et al.
2001/0047428	A1	11/2001	Hunter
2002/0023110	A1	2/2002	Fortin et al.
2002/0030699	A1	3/2002	Van Ee
2002/0112237	A1	8/2002	Ketts
2002/0158908	A1	10/2002	Vaajala et al.
2004/0049598	A1	3/2004	Tucker et al.
2005/0144256	A1	6/2005	Blumberg

OTHER PUBLICATIONS

Benjamin B. Bederson et al., Pad++: A Zoomable Graphical Sketchpad For Exploring Alternate Interface Physics, Sep. 19, 1995, pp. 1-30, http://www.cs.unm.edu/pad++.

Benjamin B. Bederson et al., A Zooming Web Browser, SPIE 1996, http://www.cs.umd.edu/hcil/jazz/learn/papers/spie-96-webbrowser.pdf, entirety (pages not numbered) .

Specification for Simple Vector Format (SVF) v1.1 Jan. 16, 1995, pp. 1-9.

Specification for Simple Vector Format (SFV) v2.0 Dec. 6, 2000, http://www.svf.org/spec.html, pp. 1-20.

SVF XML http://www.svf.org/svfxml.html, pp. 1-18.

Changes to SVF (Date unknown), pp. 1-2.

Scalable Vector Graphics (SVG) 1.0 Specification, W3C Candidate Recommendation Nov. 2, 2000, pp. 1-513.

Introduction to SVG, part of WD-SVG-19990211, http://www.w3.orgfTR/1999/ WD-SVG-19990211/intro.html#Document . . ., pp. 1-3.

W3C Scalable Vector Graphics (SVG)—History, http://www.w3.org/Graphics/SVG/History, pp. 1-18.

Janus Boye, SVG Brings Fast Vector Graphics to Web, Jul. 29, 1999 http://www.irt.org/articles/js176/, pp. 1-5.

Vector Markup Language (VML), World Wide Web Consortium Note May 13, 1998, Note-VML-19980513, http://www.w3.org/TR/1998/Note-VML_19980513, pp. 1-47.

Vector Markup Language, http://en.wikipedia.org/wiki/Vector_Markup_Language, pp. 1-2.

Precision Graphics Markup Language (PGML), World Wide Web Consortium Note Apr. 10, 1998, NOTE-PGML-19980410, http://www.w3.org/TR/1998/NOTE-PGML-19980410, pp. 1-31.

PCT International Search Report for International Application No. PCT/US01/40920, International Searching Authority, European Patent Office, Dec. 27, 2001, pp. 1-8.

PCT International Preliminary Exam Report for International Application No. PCT/US01/40920, International Preliminary Examining Authority, European Patent Office, Jun. 20, 2002, pp. 1-5.

Dennis, "Webraska Joins W3C for Next Generation Internet Graphics," Newsbytes, May 8, 2000, entirety (pages not numbered).

Fox et al. "Experience With Top Gun Wingman: A Proxy-Based Graphical Web Browser for the 3Com PalmPilot" Proceedings of Middleware '98, 1998, entirety (pages not numbered).

"Tadpole Technology: ILOG technology within Tadpole's Cartesia Redline system making repairs simpler," M2 Presswire, Jan. 1999, entirety (pages not numbered).

Thryft, "Wireless link ramped for 'always on' Internet," EETimes, Mar. 23, 1999, pp. 1-4.

Bharadvaj et al., "An Active Transcoding Proxy to Support Mobile Web Access," Proc. Of the IEEE Symposium on Reliable Distributed Systems, 1998, entirety (pages not numbered).

Björk, et al., "WEST: A Web Browser for Small Terminals," Proceedings of the 12 ACM symposium on User Interface Software and Technology (1999), entirety (pages not numbered).

Holmquist, "The Zoom Browser," Human IT Tridskrift för studier av IT ur ett humanvetenskaplight perspektive, Mar., 1998, pp. 1-18.

Chapple, "Rethinking the role of an embedded Internet client in digital set-top boxes," WebTV Workshop Submission Paper, Jun. 1998, http://www.w3.org/Architecture/1998/06/Workshop/paper11/, pp. 1-4.

Graham, "Mobile SVG at BitFlash Inc.", May 2001, http://www.w3.org/Talks/2002/1007-Di-Helsinki/bitflash/index.html, pp. 1-4.

Koncz, "Working with Autodesk Mapguide," Nov. 2000, http://www.directionsmag.com/article.php? article_id=77&trv=1, pp. 1-6.

Schaffer et al. "Navigating Hierarchically Cluster Networks through Fisheye and Full-Zoom Methods," Mar. 24, 1988—pp. 1-20.

Holmquist "Flip Zooming: Focus+Context Visualization of Linearly Ordered Discrete Visual Structures,"Published 2000, pp. 27-53.

Scalable Vector Graphics (SVG) Specification, W3C Working Draft Feb. 11, 1999 WD-SVG-19990211, http://www.w3.org/TR/1999/WD-SVG-19990211/ (HTML format—initial page).

Scalable Vector Graphics (SVG) Specification, W3C Working Draft Apr. 12, 1999 WD-SVG-19990412, http://www.w3.org/TR/1999/WD-SVG-19990412/ (HTML format—initial page).

Jones et al. "From the large screen to the small screen—retaining the designer's design for effective user interaction,"IEEE Colloquim on Issues for Networked Interpersonal Communicators, May 1997, pp. 3/1-3/4.

Pacchiano, "Adobe's New View," Computer Shopper, vol. 19, No. 4, Apr. 1999, p. 346.

Official USPTO Actions for U.S. Appl. No. 09/878,097. (Entirety—pages not numbered).

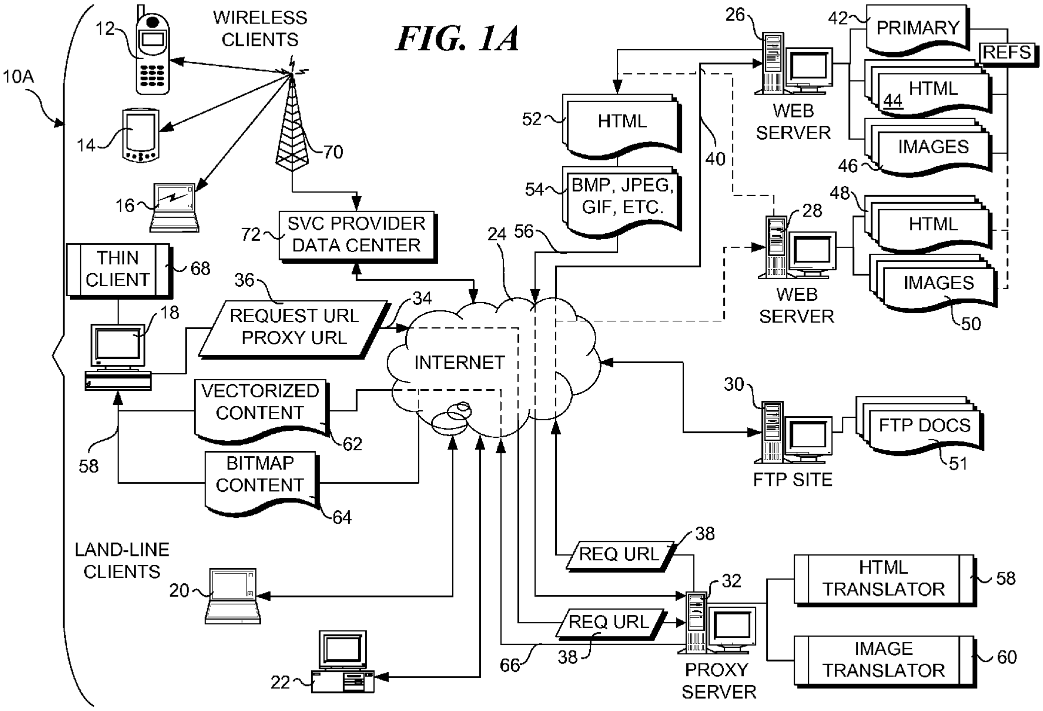
Official USPTO Actions for U.S. Appl. No. 11/045,649. (Entirety—pages not numbered).

Official USPTO Actions for U.S. Appl. No. 11/045,757. (Entirety—pages not numbered).

Official USPTO Actions for U.S. Appl. No. 11/738,932. (Entirety—pages not numbered).

Official USPTO Action for U.S. Appl. No. 11/868,124 Mailed Aug. 24, 2010 pp.1-15.

FIG. 1A


10A
12
WIRELESS CLIENTS
14
16
70
72
SVC PROVIDER DATA CENTER
THIN CLIENT
68
18
36
REQUEST URL PROXY URL
34
24
56
INTERNET
VECTORIZED CONTENT
62
58
BITMAP CONTENT
64
LAND-LINE CLIENTS
20
22
66
26
WEB SERVER
42
PRIMARY
REFS
HTML
44
IMAGES
46
52
HTML
54
BMP, JPEG, GIF, ETC.
40
28
WEB SERVER
48
HTML
IMAGES
50
30
FTP SITE
FTP DOCS
51
REQ URL
38
32
PROXY SERVER
REQ URL
38
HTML TRANSLATOR
58
IMAGE TRANSLATOR
60

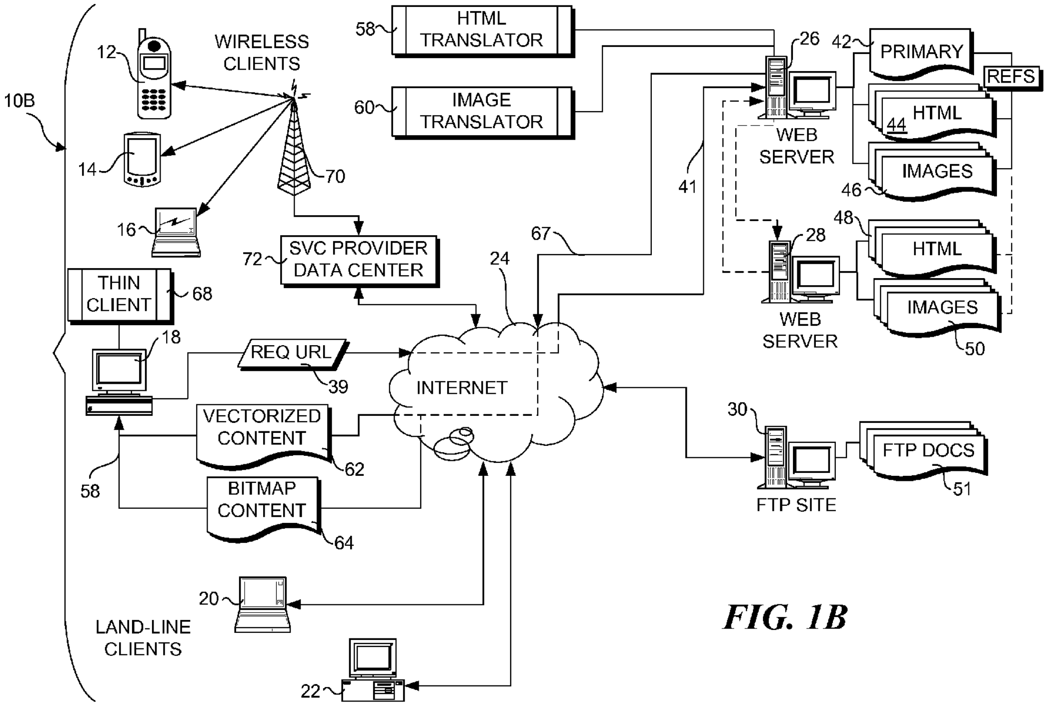

10B
12
WIRELESS CLIENTS
14
16
70
58
HTML TRANSLATOR
60
IMAGE TRANSLATOR
72
SVC PROVIDER DATA CENTER
THIN CLIENT
68
18
REQ URL
39
VECTORIZED CONTENT
62
BITMAP CONTENT
64
58
24
INTERNET
67
41
26
WEB SERVER
42
PRIMARY
REFS
HTML
44
IMAGES
46
48
28
WEB SERVER
HTML
IMAGES
50
30
FTP SITE
FTP DOCS
51
20
22
LAND-LINE CLIENTS

FIG. 1B

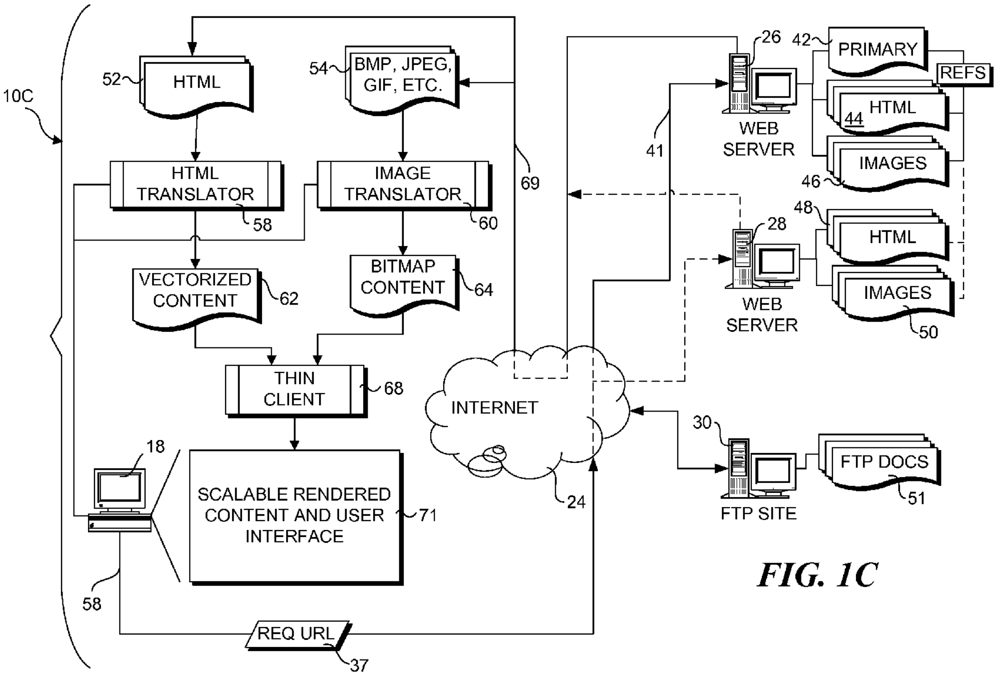

10C
52
HTML
54
BMP, JPEG, GIF, ETC.
HTML TRANSLATOR
58
IMAGE TRANSLATOR
60
69
VECTORIZED CONTENT
62
BITMAP CONTENT
64
THIN CLIENT
68
18
SCALABLE RENDERED CONTENT AND USER INTERFACE
71
58
REQ URL
37
INTERNET
24
41
26
WEB SERVER
42
PRIMARY
REFS
44
HTML
IMAGES
46
28
WEB SERVER
48
HTML
IMAGES
50
30
FTP SITE
FTP DOCS
51

FIG. 1C

FIG. 2A

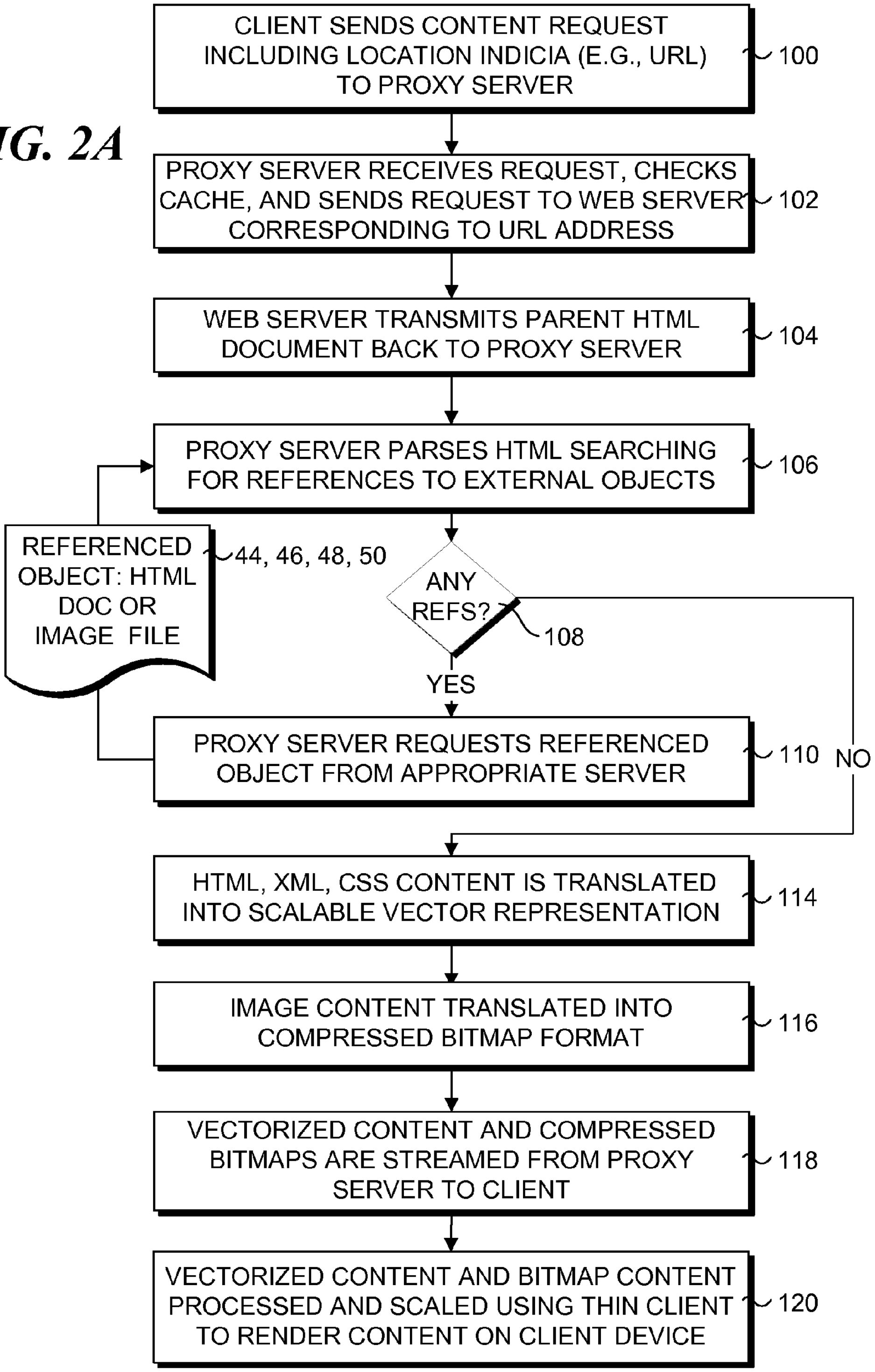

CLIENT SENDS CONTENT REQUEST INCLUDING LOCATION INDICIA (E.G., URL) TO PROXY SERVER
100
PROXY SERVER RECEIVES REQUEST, CHECKS CACHE, AND SENDS REQUEST TO WEB SERVER CORRESPONDING TO URL ADDRESS
102
WEB SERVER TRANSMITS PARENT HTML DOCUMENT BACK TO PROXY SERVER
104
PROXY SERVER PARSES HTML SEARCHING FOR REFERENCES TO EXTERNAL OBJECTS
106
REFERENCED OBJECT: HTML DOC OR IMAGE FILE
44, 46, 48, 50
ANY REFS?
108
YES
NO
PROXY SERVER REQUESTS REFERENCED OBJECT FROM APPROPRIATE SERVER
110
HTML, XML, CSS CONTENT IS TRANSLATED INTO SCALABLE VECTOR REPRESENTATION
114
IMAGE CONTENT TRANSLATED INTO COMPRESSED BITMAP FORMAT
116
VECTORIZED CONTENT AND COMPRESSED BITMAPS ARE STREAMED FROM PROXY SERVER TO CLIENT
118
VECTORIZED CONTENT AND BITMAP CONTENT PROCESSED AND SCALED USING THIN CLIENT TO RENDER CONTENT ON CLIENT DEVICE
120

FIG. 2B

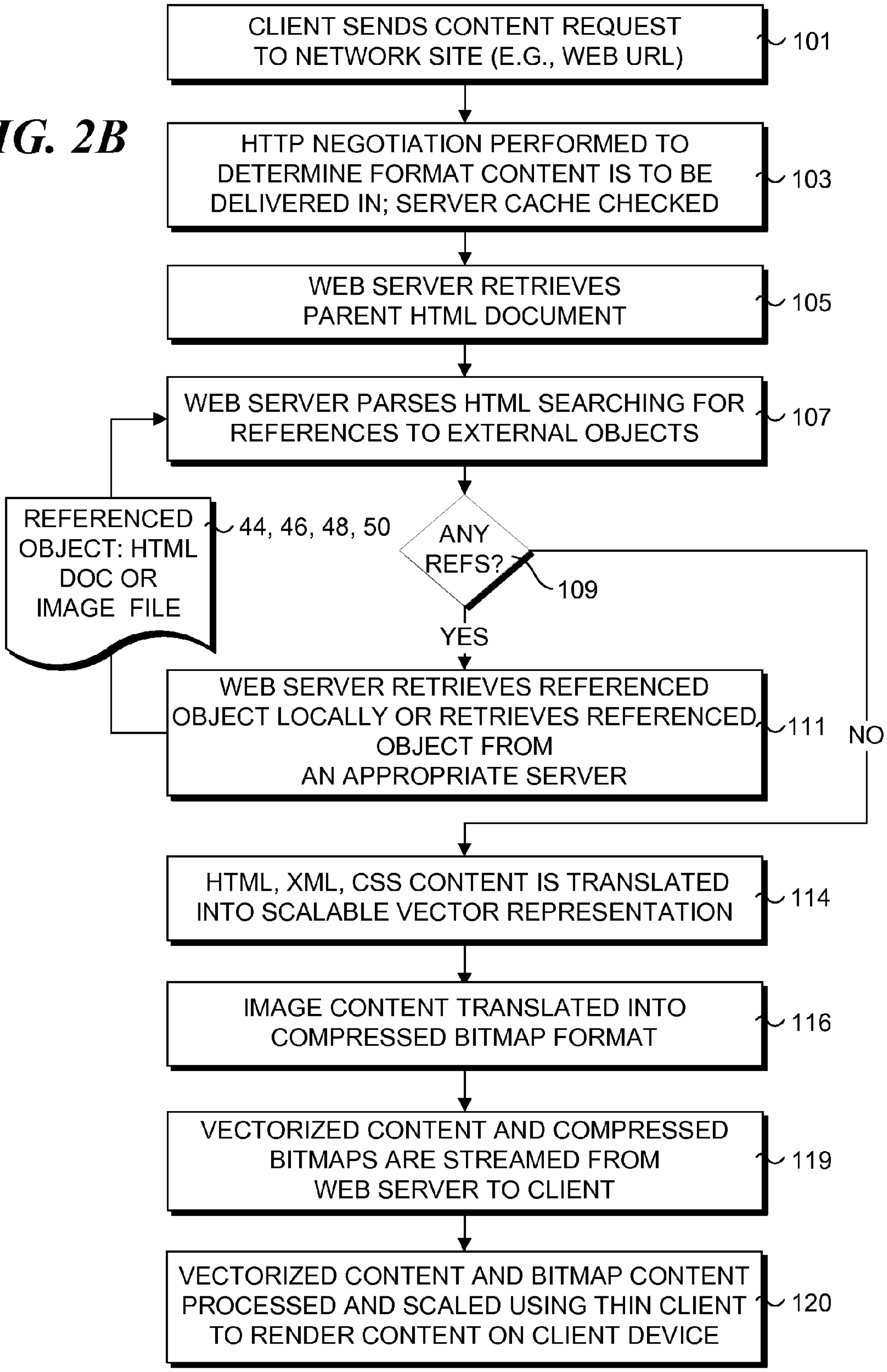
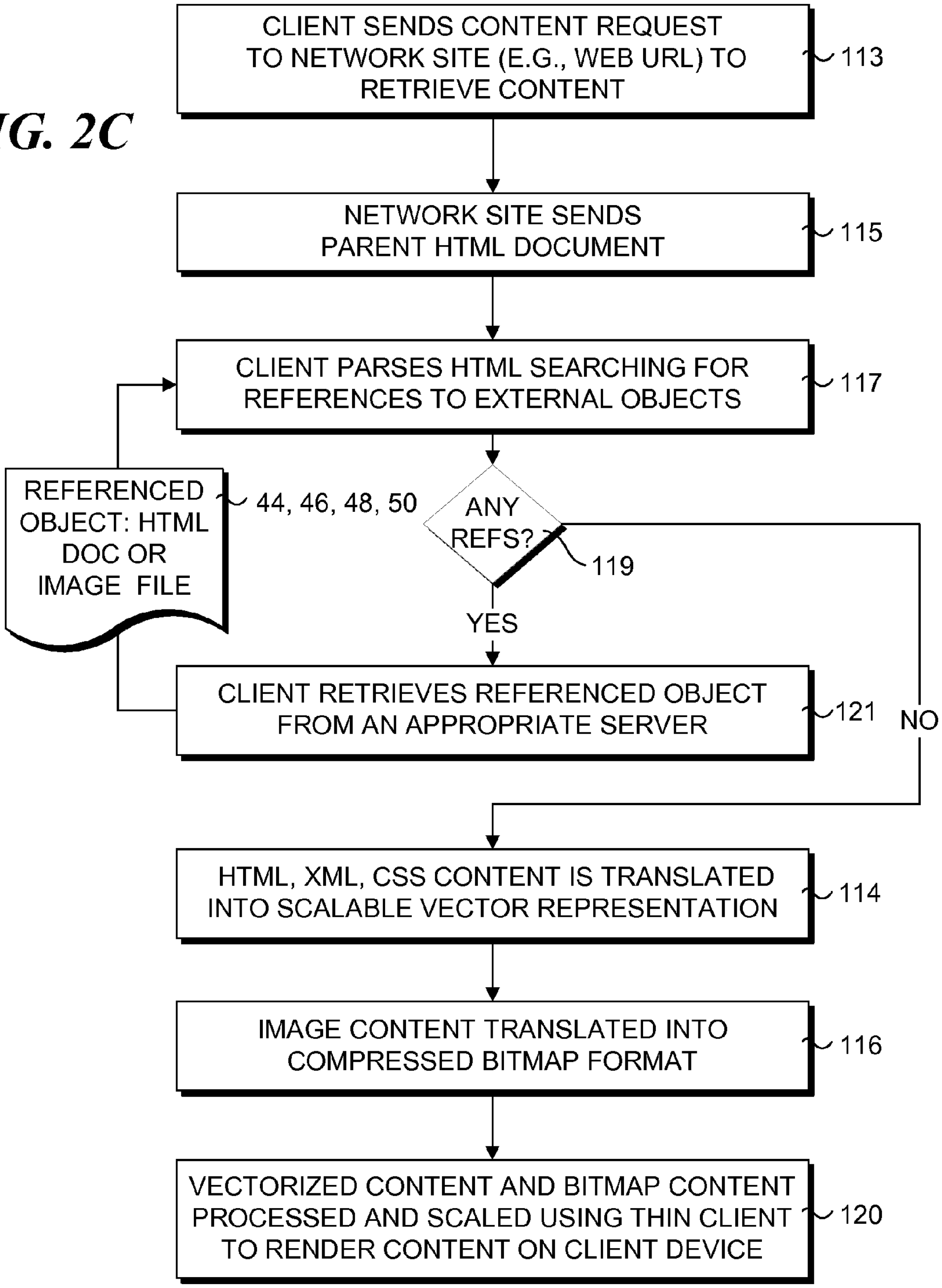

CLIENT SENDS CONTENT REQUEST TO NETWORK SITE (E.G., WEB URL)
101
HTTP NEGOTIATION PERFORMED TO DETERMINE FORMAT CONTENT IS TO BE DELIVERED IN; SERVER CACHE CHECKED
103
WEB SERVER RETRIEVES PARENT HTML DOCUMENT
105
WEB SERVER PARSES HTML SEARCHING FOR REFERENCES TO EXTERNAL OBJECTS
107
REFERENCED OBJECT: HTML DOC OR IMAGE FILE
44, 46, 48, 50
ANY REFS?
109
YES
NO
WEB SERVER RETRIEVES REFERENCED OBJECT LOCALLY OR RETRIEVES REFERENCED OBJECT FROM AN APPROPRIATE SERVER
111
HTML, XML, CSS CONTENT IS TRANSLATED INTO SCALABLE VECTOR REPRESENTATION
114
IMAGE CONTENT TRANSLATED INTO COMPRESSED BITMAP FORMAT
116
VECTORIZED CONTENT AND COMPRESSED BITMAPS ARE STREAMED FROM WEB SERVER TO CLIENT
119
VECTORIZED CONTENT AND BITMAP CONTENT PROCESSED AND SCALED USING THIN CLIENT TO RENDER CONTENT ON CLIENT DEVICE
120

FIG. 2C


CLIENT SENDS CONTENT REQUEST TO NETWORK SITE (E.G., WEB URL) TO RETRIEVE CONTENT
113
NETWORK SITE SENDS PARENT HTML DOCUMENT
115
CLIENT PARSES HTML SEARCHING FOR REFERENCES TO EXTERNAL OBJECTS
117
REFERENCED OBJECT: HTML DOC OR IMAGE FILE
44, 46, 48, 50
ANY REFS?
119
YES
NO
CLIENT RETRIEVES REFERENCED OBJECT FROM AN APPROPRIATE SERVER
121
HTML, XML, CSS CONTENT IS TRANSLATED INTO SCALABLE VECTOR REPRESENTATION
114
IMAGE CONTENT TRANSLATED INTO COMPRESSED BITMAP FORMAT
116
VECTORIZED CONTENT AND BITMAP CONTENT PROCESSED AND SCALED USING THIN CLIENT TO RENDER CONTENT ON CLIENT DEVICE
120

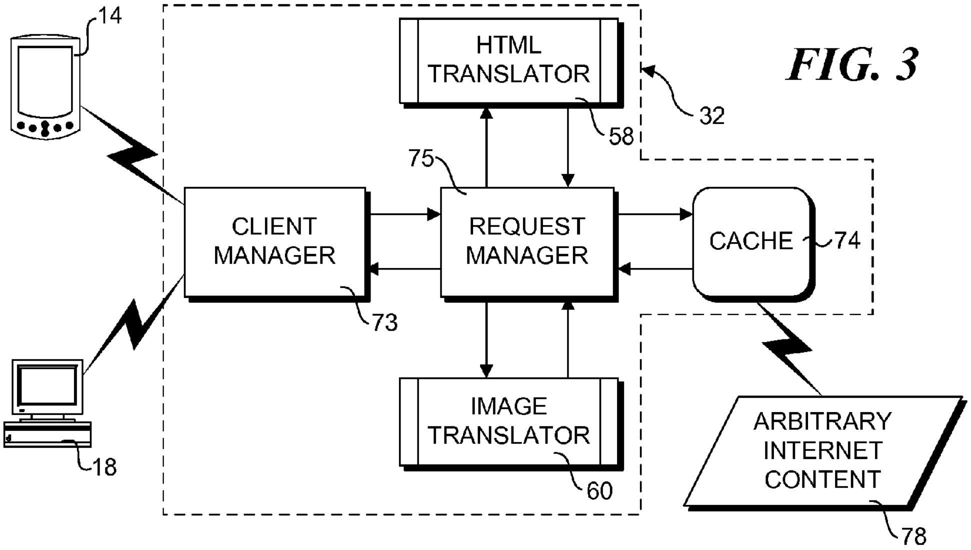

14
HTML TRANSLATOR
58
32
75
CLIENT MANAGER
73
REQUEST MANAGER
CACHE
74
IMAGE TRANSLATOR
60
18
ARBITRARY INTERNET CONTENT
78

FIG. 3

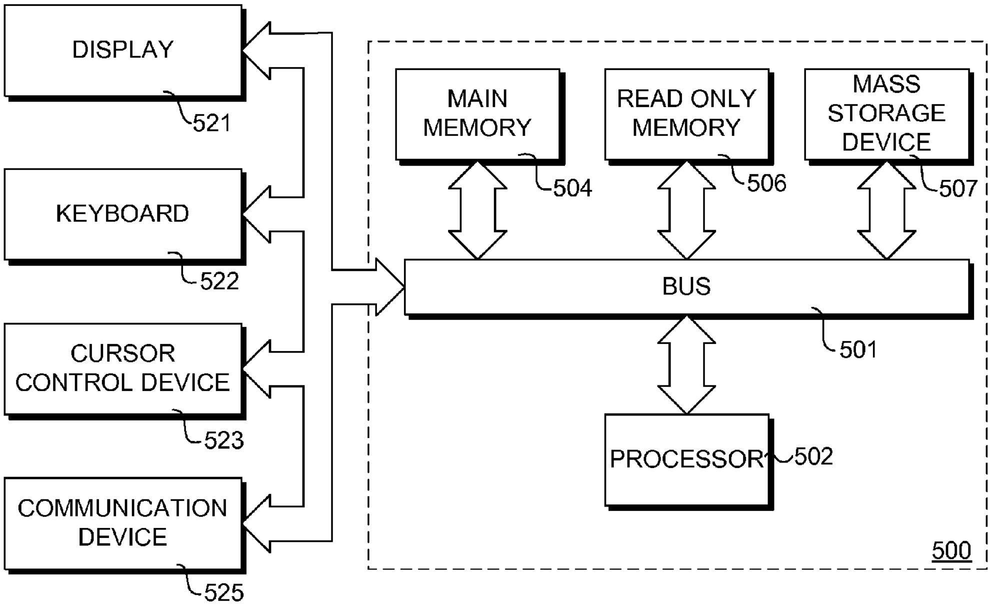
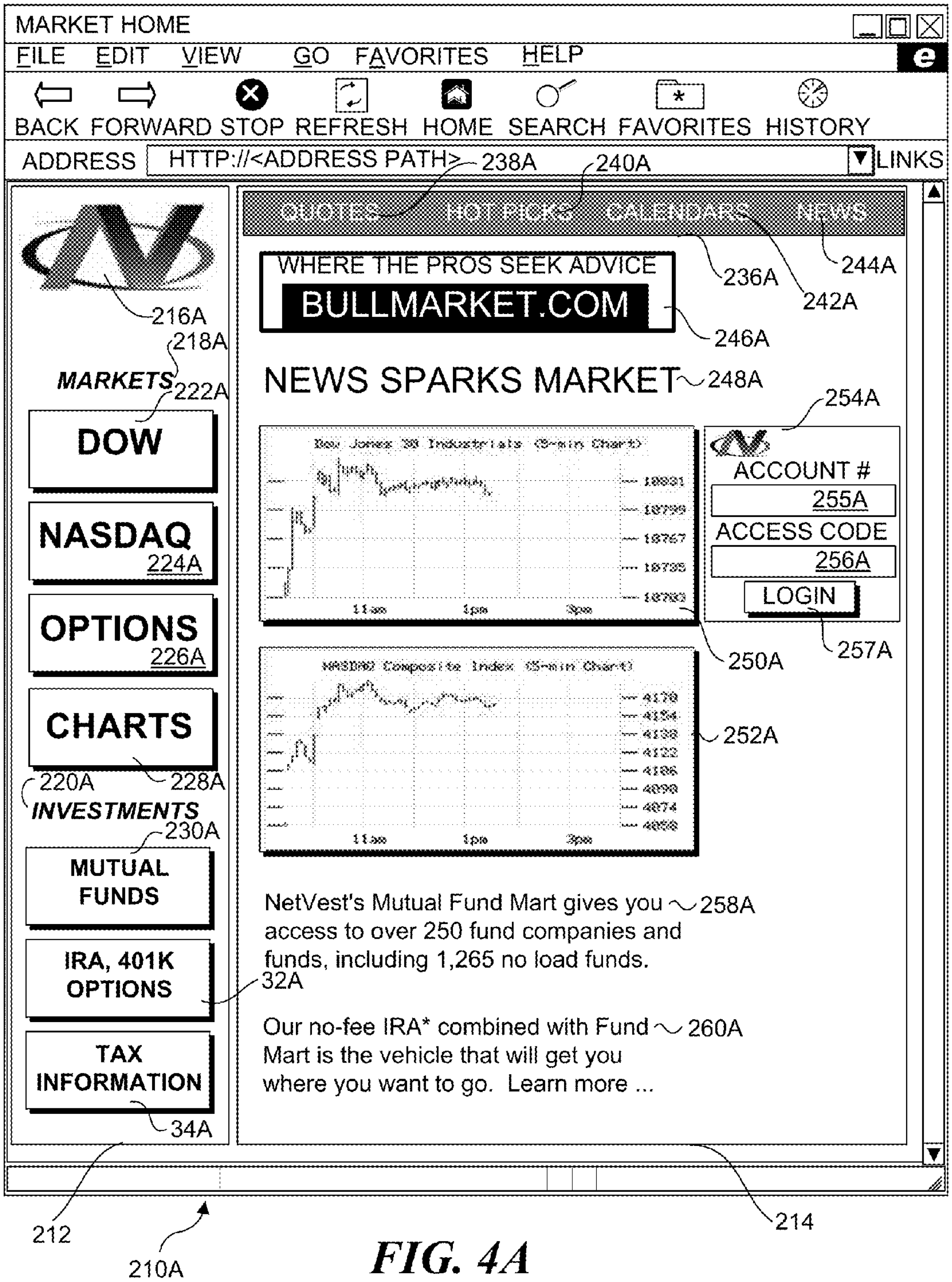

DISPLAY
521
KEYBOARD
522
CURSOR CONTROL DEVICE
523
COMMUNICATION DEVICE
525
MAIN MEMORY
504
READ ONLY MEMORY
506
MASS STORAGE DEVICE
507
BUS
501
PROCESSOR
502
500

FIG. 10



NEWS SPARKS MARKET

NetVest's Mutual Fund Mart gives you access to over 250 fund companies and funds, including 1,265 no load funds.

Our no-fee IRA* combined with Fund Mart is the vehicle that will get you where you want to go. Learn more ...

FIG. 4A

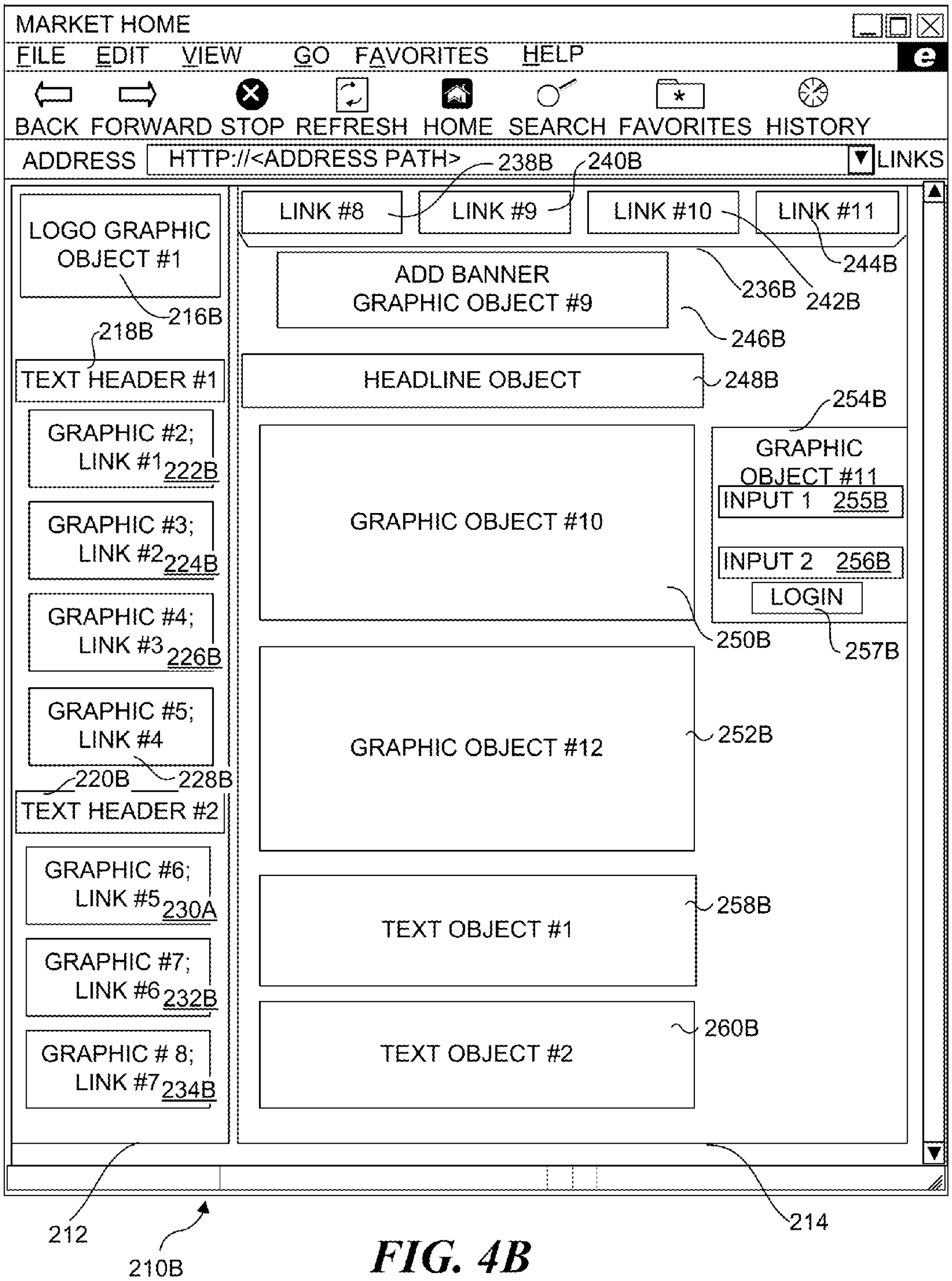

MARKET HOME
FILE EDIT VIEW GO FAVORITES HELP
BACK FORWARD STOP REFRESH HOME SEARCH FAVORITES HISTORY
ADDRESS HTTP://<ADDRESS PATH>
LINKS
238B
240B
LINK #8
LINK #9
LINK #10
LINK #11
244B
236B
242B
LOGO GRAPHIC OBJECT #1
216B
218B
ADD BANNER GRAPHIC OBJECT #9
246B
TEXT HEADER #1
HEADLINE OBJECT
248B
254B
GRAPHIC #2; LINK #1 222B
GRAPHIC OBJECT #11
INPUT 1 255B
INPUT 2 256B
LOGIN
257B
GRAPHIC #3; LINK #2 224B
GRAPHIC OBJECT #10
GRAPHIC #4; LINK #3 226B
250B
GRAPHIC #5; LINK #4
GRAPHIC OBJECT #12
252B
220B
228B
TEXT HEADER #2
GRAPHIC #6; LINK #5 230A
TEXT OBJECT #1
258B
GRAPHIC #7; LINK #6 232B
GRAPHIC # 8; LINK #7 234B
TEXT OBJECT #2
260B
212
210B
214

FIG. 4B

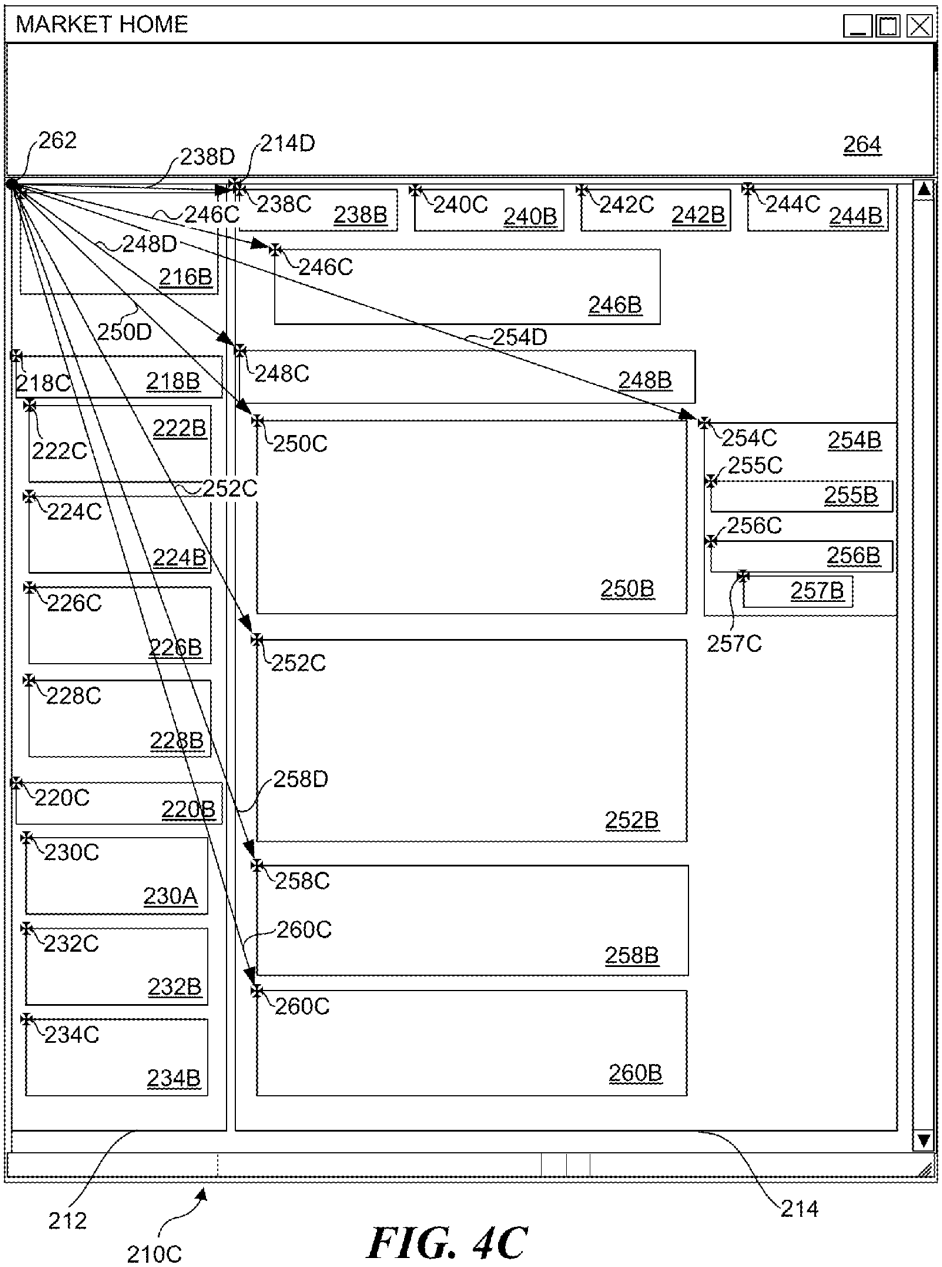

MARKET HOME
264
262
238D
214D
238C
238B
240C
240B
242C
242B
244C
244B
246C
248D
216B
246C
246B
250D
254D
248C
248B
218C
218B
222B
222C
254C
254B
255C
255B
252C
224C
256C
256B
224B
257B
250B
226C
257C
226B
252C
228C
228B
258D
220C
220B
252B
230C
230A
258C
260C
258B
232C
232B
234C
260C
234B
260B
212
214
210C

FIG. 4C

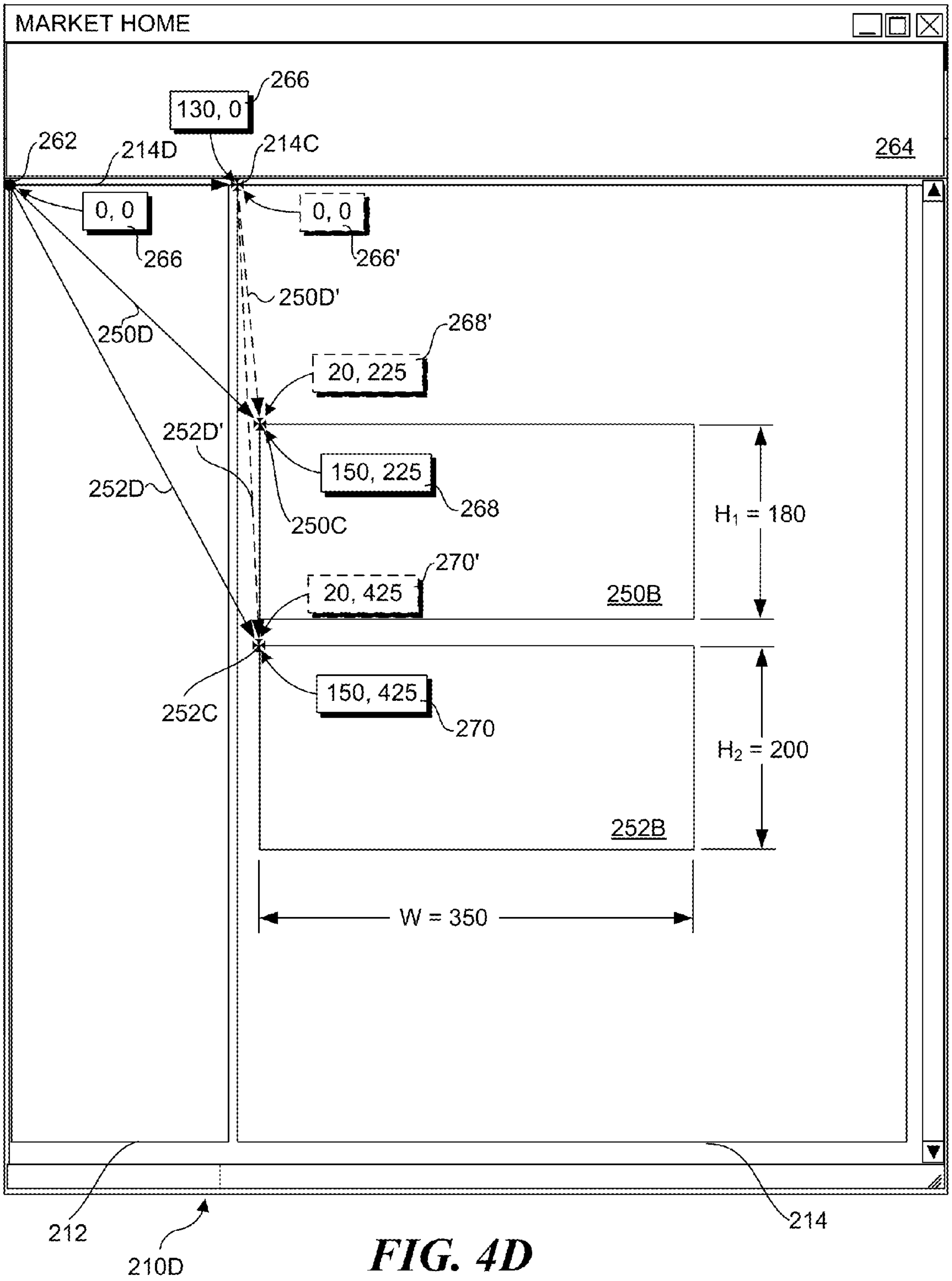
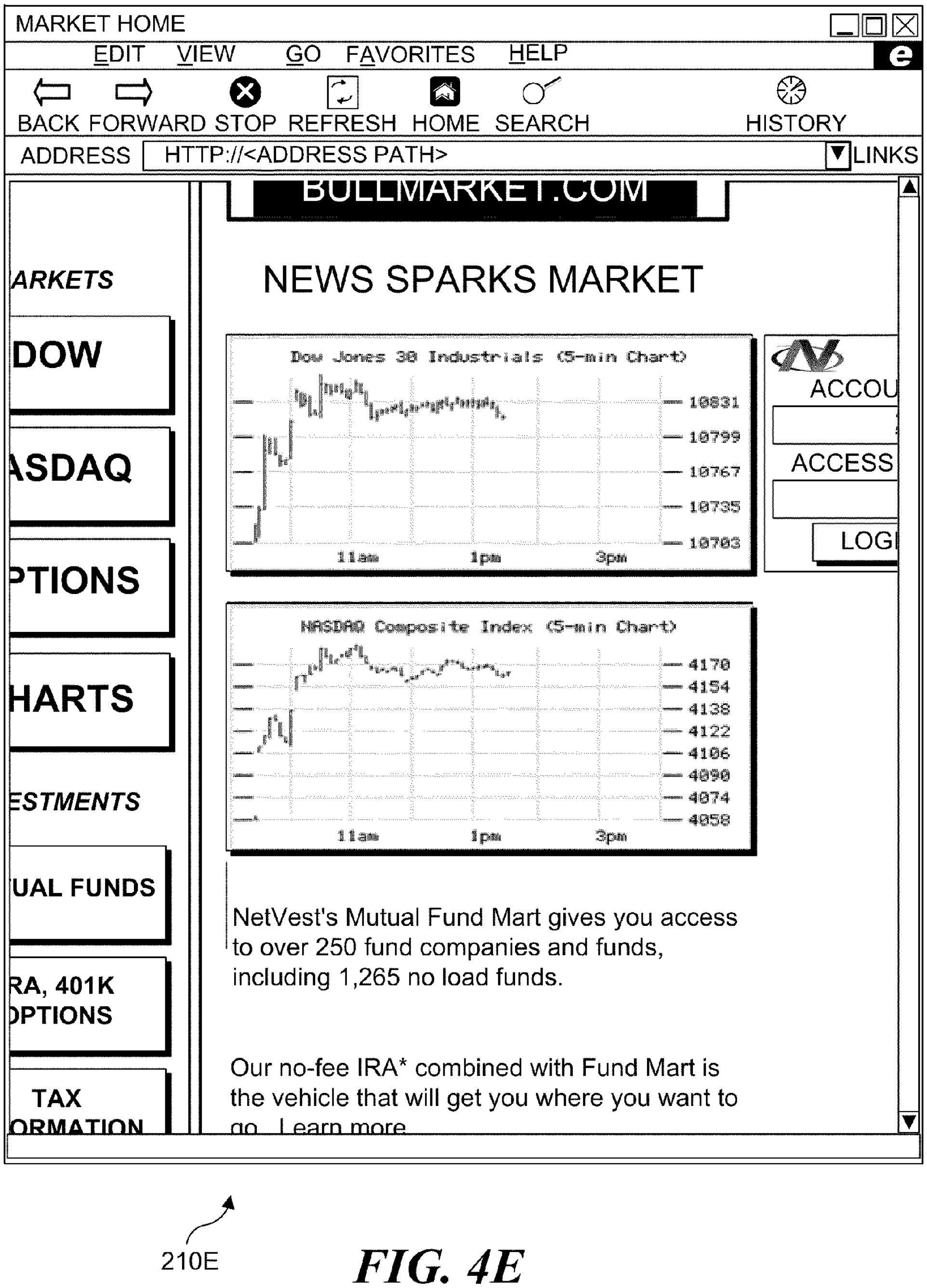

MARKET HOME
266
130, 0
262
214D
214C
264
0, 0
266
0, 0
266'
250D'
250D
268'
20, 225
252D'
150, 225
252D
268
250C
H1 = 180
270'
20, 425
250B
252C
150, 425
270
H2 = 200
252B
W = 350
212
214
210D

FIG. 4D



NEWS SPARKS MARKET

NetVest's Mutual Fund Mart gives you access to over 250 fund companies and funds, including 1,265 no load funds.

Our no-fee IRA* combined with Fund Mart is the vehicle that will get you where you want to go. Learn more

FIG. 4E

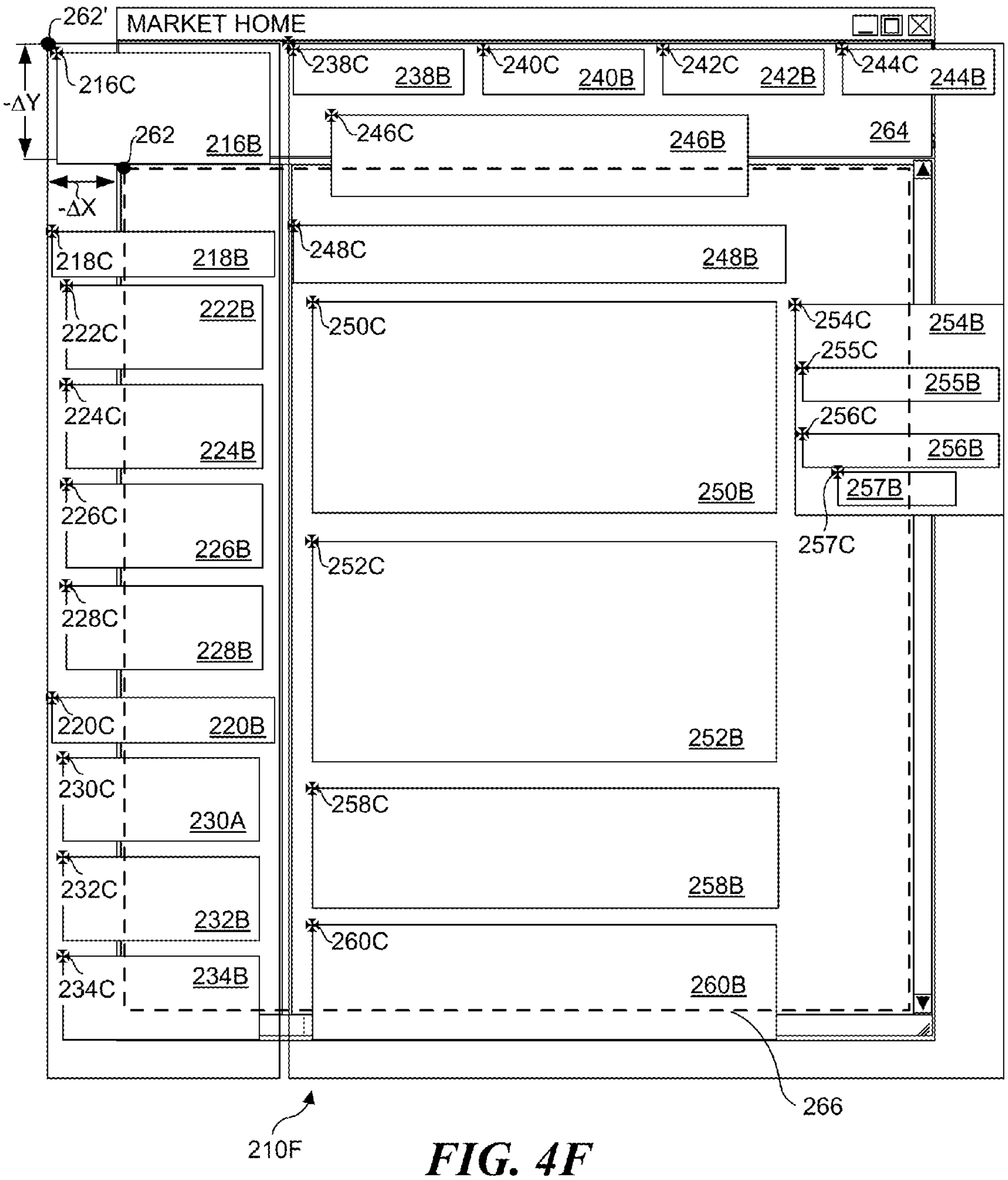

262'
MARKET HOME
216C
-ΔY
262
216B
-ΔX
238C
238B
240C
240B
242C
242B
244C
244B
246C
246B
264
248C
248B
218C
218B
222B
222C
250C
254C
254B
255C
255B
256C
256B
257B
257C
224C
224B
226C
226B
250B
252C
228C
228B
252B
220C
220B
230C
230A
258C
258B
232C
232B
260C
234C
234B
260B
266
210F

FIG. 4F

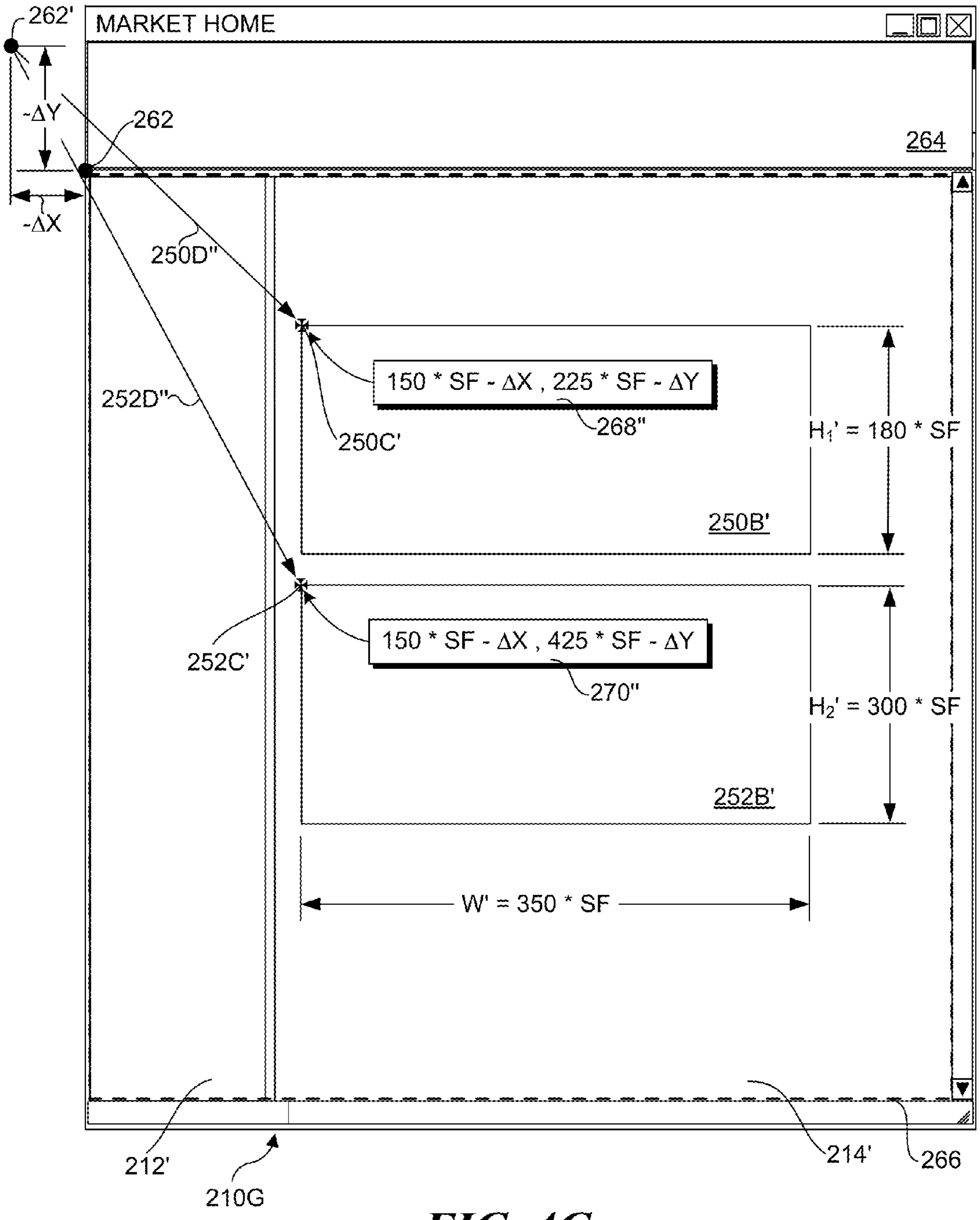

262'
MARKET HOME
-ΔY
262
264
-ΔX
250D"
150 * SF - ΔX , 225 * SF - ΔY
268"
250C'
252D"
H₁' = 180 * SF
250B'
150 * SF - ΔX , 425 * SF - ΔY
270"
252C'
H₂' = 300 * SF
252B'
W' = 350 * SF
212'
210G
214'
266

FIG. 4G

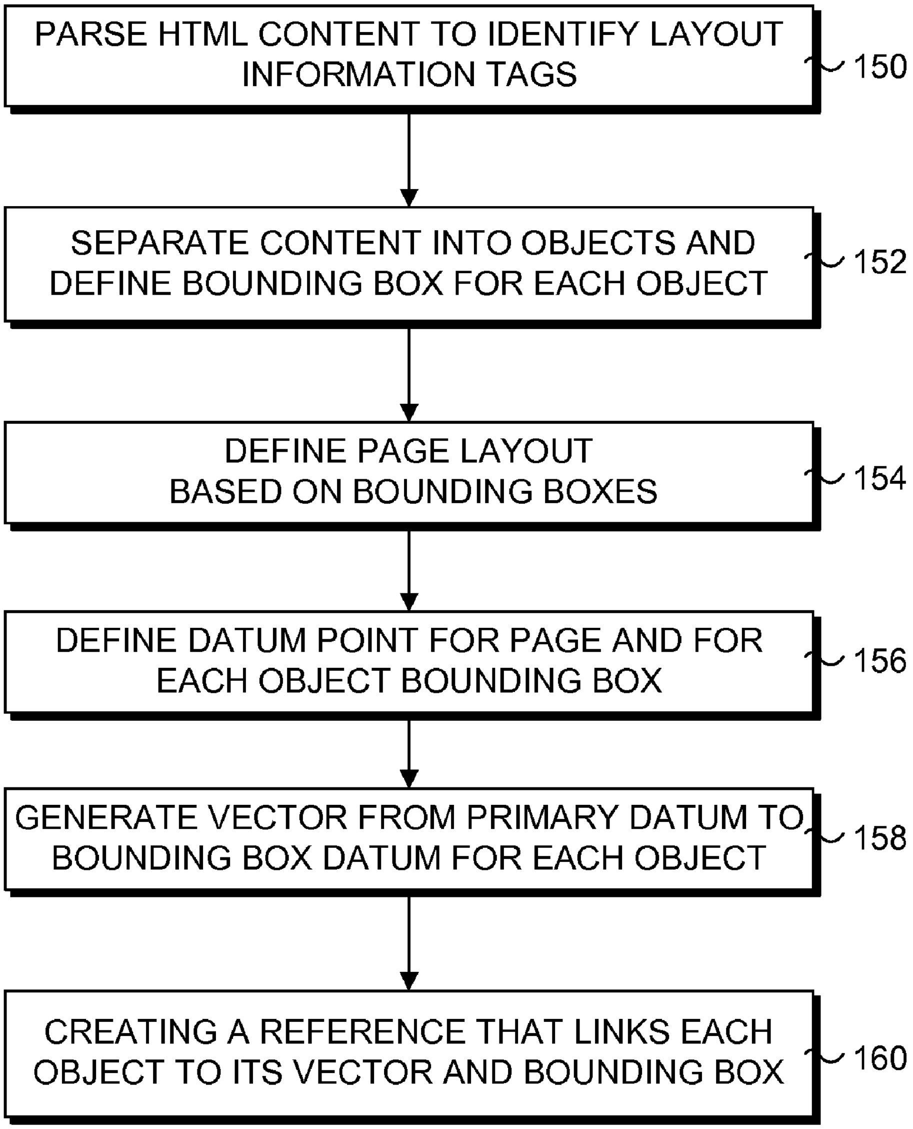

PARSE HTML CONTENT TO IDENTIFY LAYOUT INFORMATION TAGS
150
SEPARATE CONTENT INTO OBJECTS AND DEFINE BOUNDING BOX FOR EACH OBJECT
152
DEFINE PAGE LAYOUT BASED ON BOUNDING BOXES
154
DEFINE DATUM POINT FOR PAGE AND FOR EACH OBJECT BOUNDING BOX
156
GENERATE VECTOR FROM PRIMARY DATUM TO BOUNDING BOX DATUM FOR EACH OBJECT
158
CREATING A REFERENCE THAT LINKS EACH OBJECT TO ITS VECTOR AND BOUNDING BOX
160

FIG. 5

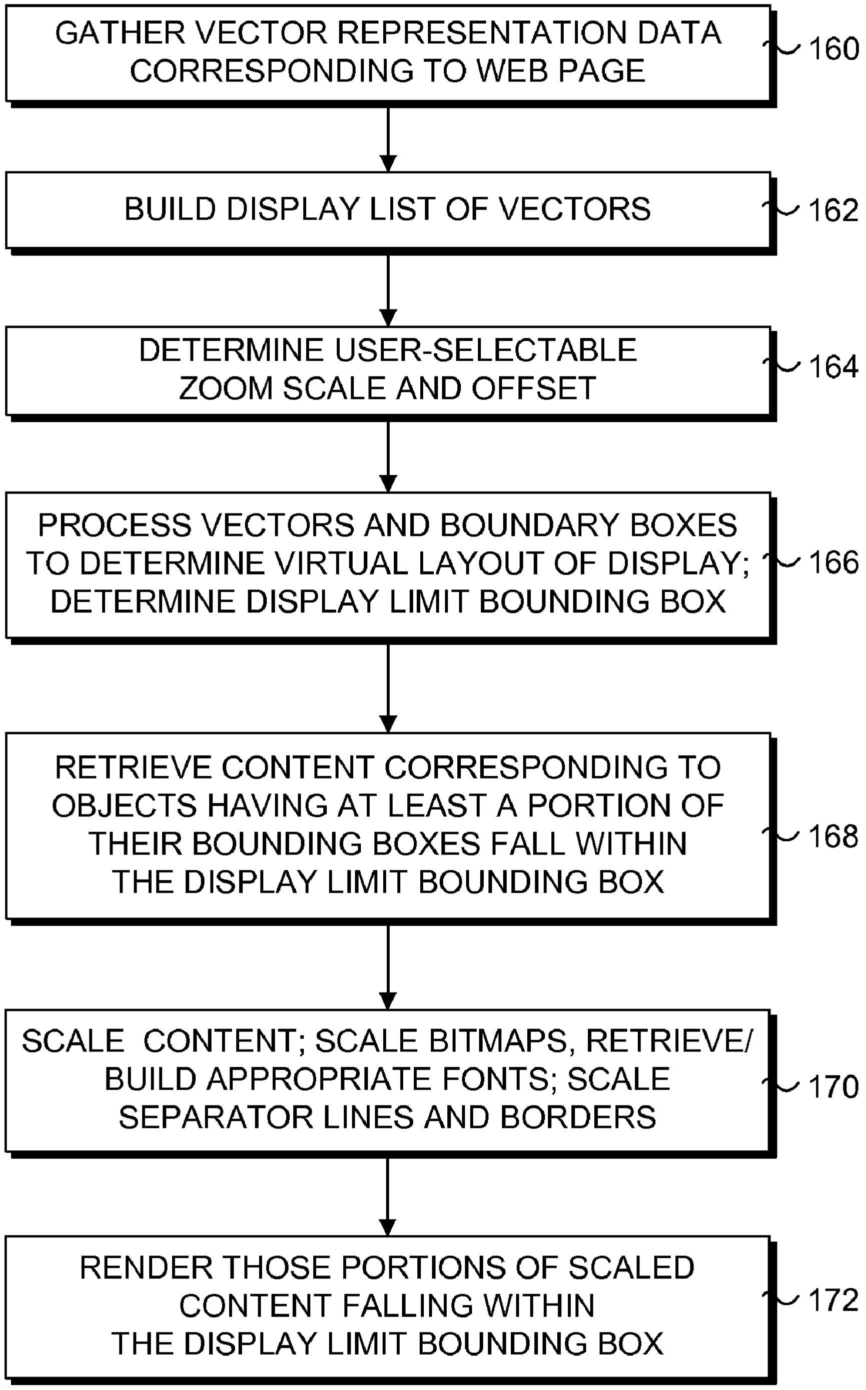

GATHER VECTOR REPRESENTATION DATA CORRESPONDING TO WEB PAGE
160
BUILD DISPLAY LIST OF VECTORS
162
DETERMINE USER-SELECTABLE ZOOM SCALE AND OFFSET
164
PROCESS VECTORS AND BOUNDARY BOXES TO DETERMINE VIRTUAL LAYOUT OF DISPLAY; DETERMINE DISPLAY LIMIT BOUNDING BOX
166
RETRIEVE CONTENT CORRESPONDING TO OBJECTS HAVING AT LEAST A PORTION OF THEIR BOUNDING BOXES FALL WITHIN THE DISPLAY LIMIT BOUNDING BOX
168
SCALE CONTENT; SCALE BITMAPS, RETRIEVE/ BUILD APPROPRIATE FONTS; SCALE SEPARATOR LINES AND BORDERS
170
RENDER THOSE PORTIONS OF SCALED CONTENT FALLING WITHIN THE DISPLAY LIMIT BOUNDING BOX
172

FIG. 6

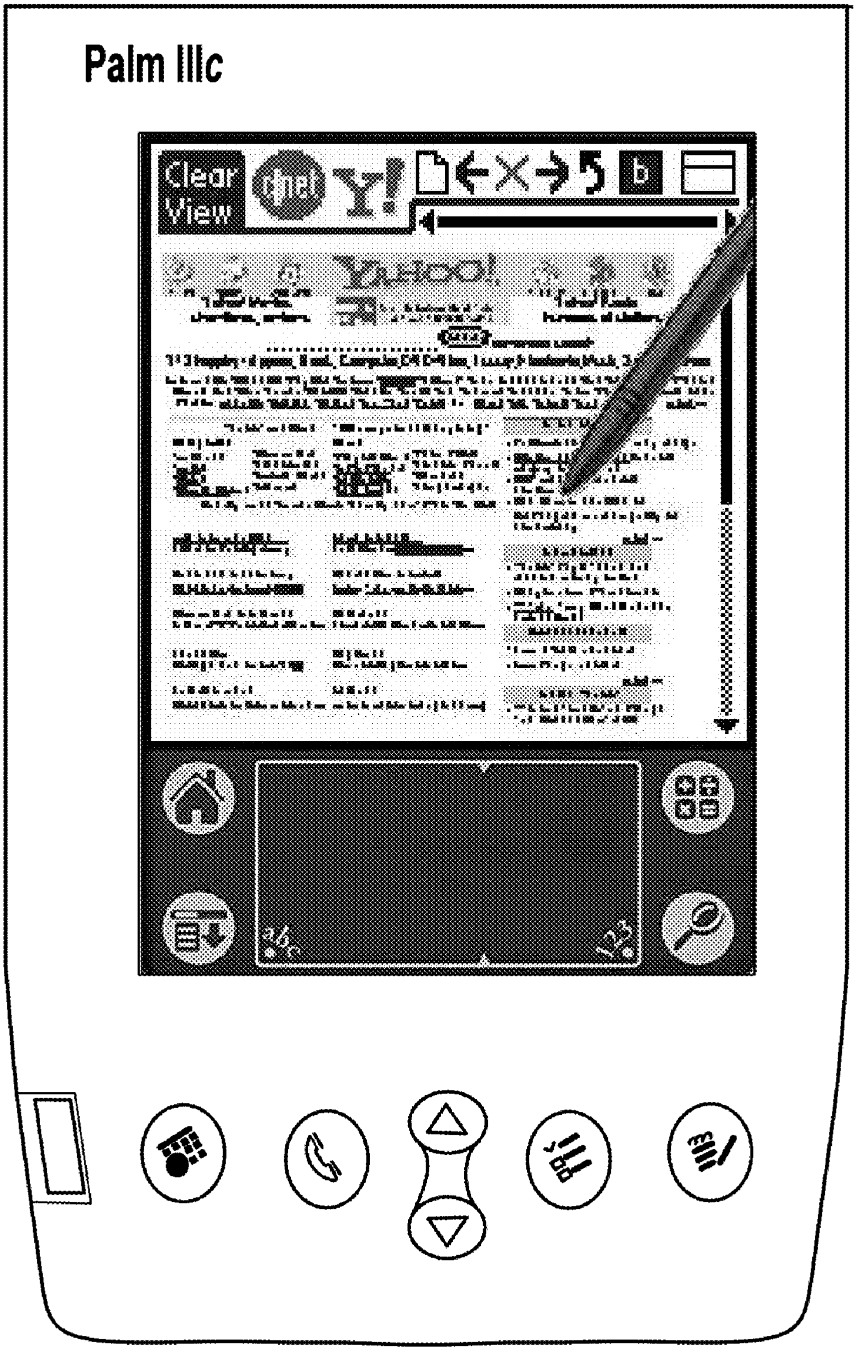
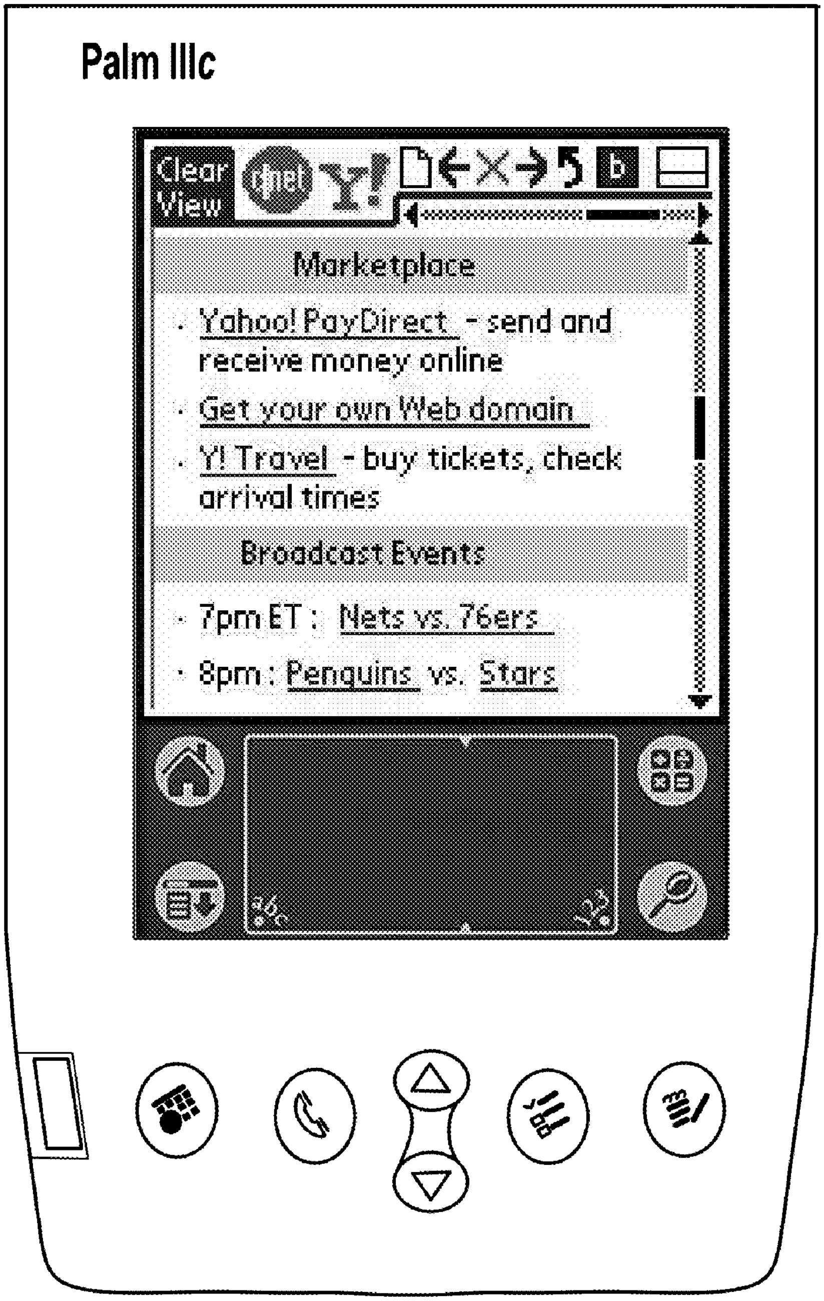


FIG. 7A



Marketplace

- Yahoo! PayDirect - send and receive money online
- Get your own Web domain
- Y! Travel - buy tickets, check arrival times

Broadcast Events

- 7pm ET : Nets vs. 76ers
- 8pm : Penguins vs. Stars

FIG. 7B

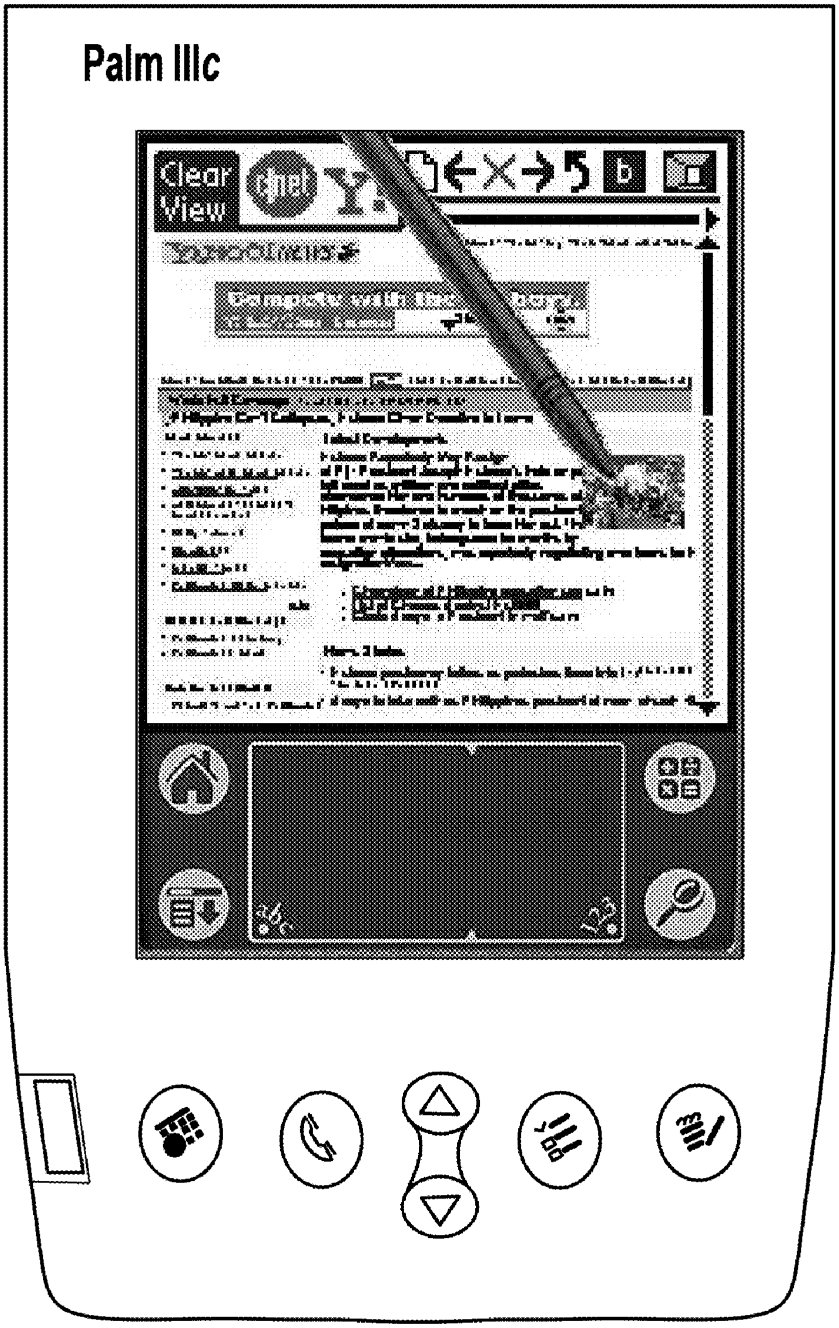


FIG. 8A

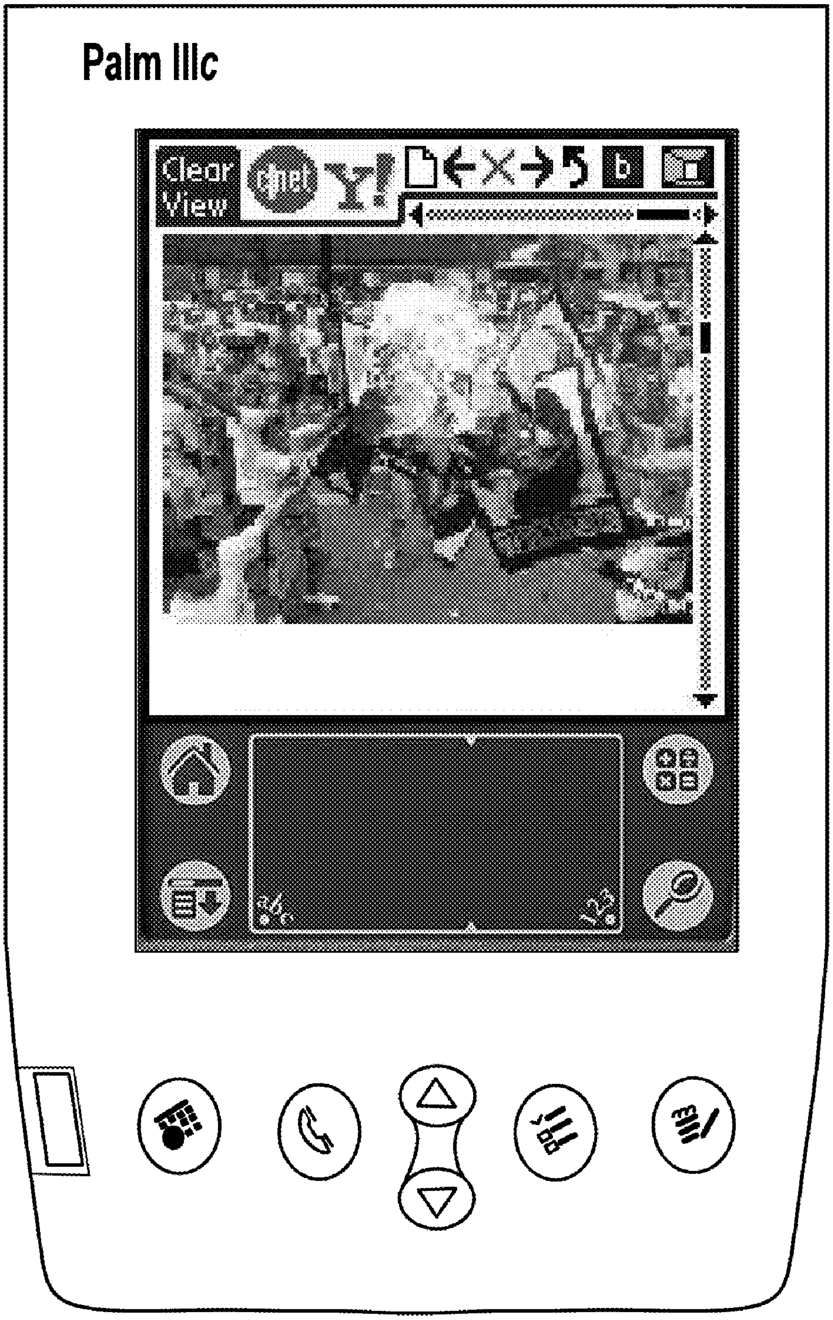


FIG. 8B

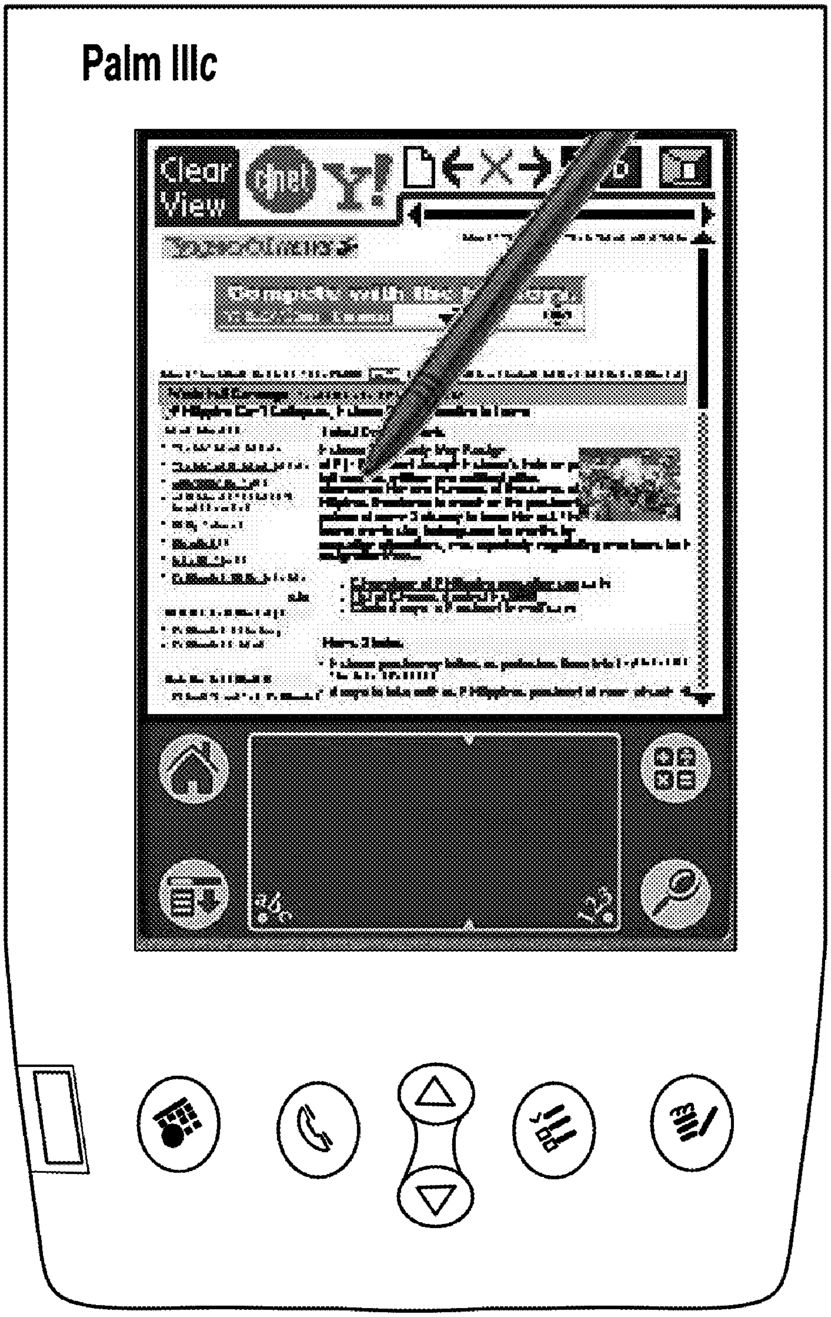
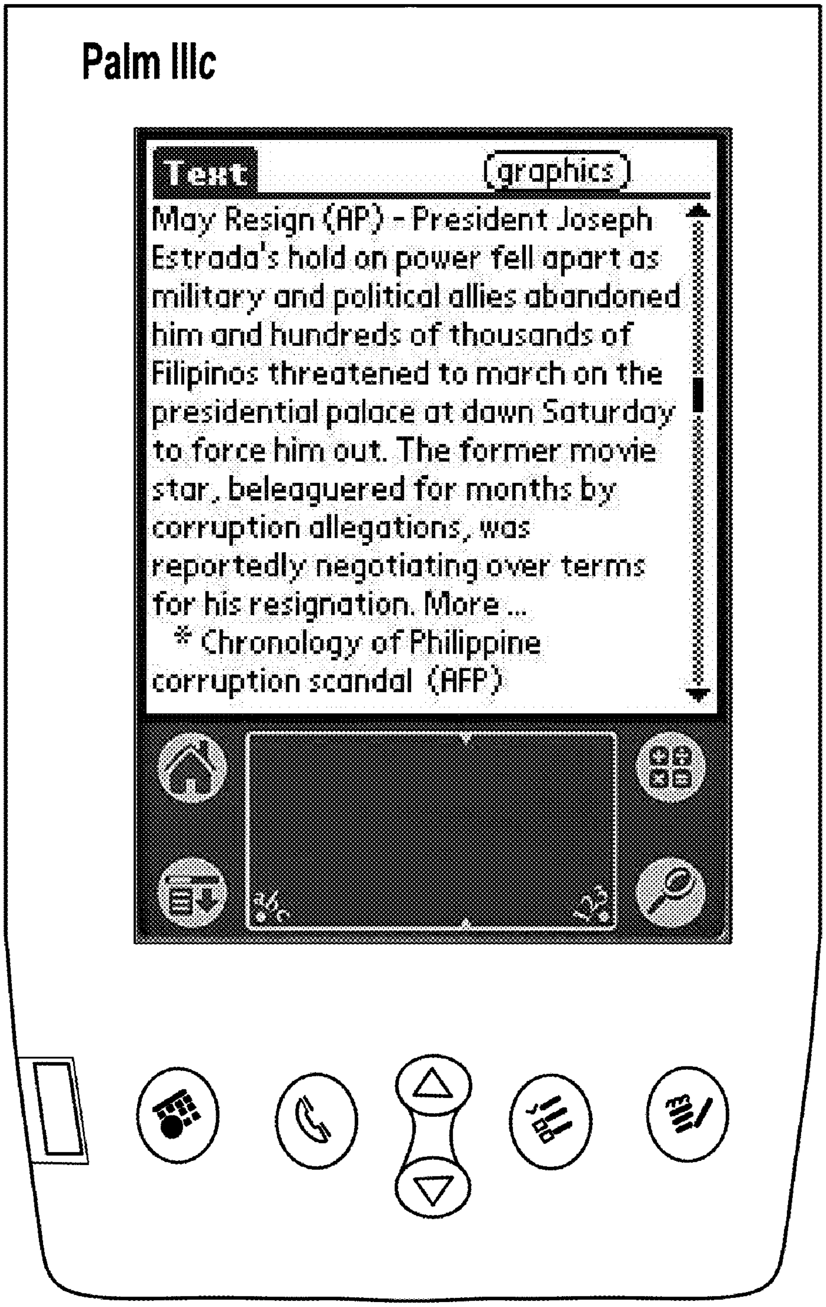


FIG. 9A



May Resign (AP) - President Joseph Estrada's hold on power fell apart as military and political allies abandoned him and hundreds of thousands of Filipinos threatened to march on the presidential palace at dawn Saturday to force him out. The former movie star, beleaguered for months by corruption allegations, was reportedly negotiating over terms for his resignation. More ...

* Chronology of Philippine corruption scandal (AFP)

FIG. 9B

SCALABLE DISPLAY OF INTERNET CONTENT ON MOBILE DEVICES

RELATED APPLICATIONS

This application is a Continuation of U.S. Non-provisional application Ser. No. 09/878,097, filed Jun. 8, 2001, (issued as U.S. Pat. No. 7,210,099) entitled "RESOLUTION INDEPENDENT VECTOR DISPLAY OF INTERNET CONTENT," which is a Continuation-in-Part of U.S. Non-provisional application Ser. No. 09/825,511, filed Apr. 7, 2001, (Abandoned) entitled "RESOLUTION INDEPENDENT VECTOR DISPLAY OF INTERNET CONTENT," the benefit of the filing dates of which is claimed under 35 U.S.C. §120. U.S. Non-provisional application Ser. No. 09/878,097 further claims the benefit of the filing dates of U.S. Provisional Application No. 60/211,019, filed Jun. 12, 2000, entitled "METHOD AND SYSTEM FOR RESOLUTION INDEPENDENT DISPLAY OF HTML AND XML CONTENT" and U.S. Provisional Application No. 60/217,345, filed Jul. 11, 2000, entitled "METHOD AND SYSTEM FOR SELECTION, RETRIEVAL, AND CONVERSION OF COMPUTER CONTENT TO VECTOR FORMAT FOR RESOLUTION INDEPENDENT DISPLAY," under 35 U.S.C. §119(e). The disclosure of each of the foregoing applications is incorporated by reference in its entirety herein for all purposes.

This application also contains subject matter related to Divisionals (of Ser. No. 09/878,097) U.S. Non-provisional application Ser. Nos. 11/045,649 (issued as U.S. Pat. No. 7,584,423) entitled METHOD, PROXY AND SYSTEM TO SUPPORT FULL-PAGE WEB BROWSING ON HANDHELD DEVICES, and 11/045,757 (issued as U.S. Pat. No. 7,461,353) entitled SCALABLE DISPLAY OF INTERNET CONTENT ON MOBILE DEVICES, both filed Jan. 28, 2005. This application also contains subject matter related to U.S. Non-provisional application Ser. Nos. 11/735,477 and 11/735,482, both filed on Apr. 15, 2007, 11/738,932 filed on Apr. 23, 2007, 11/868,124 filed on Oct. 5, 2007, and 12/326,092 filed on Dec. 1, 2008.

COPYRIGHT NOTICE

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates generally to viewing of Internet content on mobile devices, and more particularly concerns to novel processing of Internet and World Wide Web content to scalable forms for resolution-independent rendering and zoom- and pan-enabling the display of content on mobile devices.

2. Description of the Related Art

Text only Internet information browsers began as a project at the CERN, European Organization for Nuclear Research, facility in Geneva Switzerland. From its inception the intent was to provide a mesh or web of access to data with a common user interface. Browsers moved from the academic environment when NCSA, the National Center for Supercomputing Applications at the University of Illinois in Urbana-Champaign developed Mosaic, an Internet information browser and World Wide Web client.

Internet content is stored in multiple file formats. These formats include HTML (Hyper Text Markup Language) and XML (eXtended Markup Language) as well as graphic file format GIF (Graphics Interchange Format) and JPEG (Joint Photographic Experts Group). These four file formats constitute the majority of Internet content. Font size and resizing display area for content can alter the size of the display of Internet content in existing browsers. The majority of Internet content displays as a flat single resolution with no browser support for zoom.

Much of the Internet content has been designed for display on desktop computers with a single target resolution. Even though HTML has the ability to adapt to changes in screen resolution, major Internet content providers have chosen to create their Web pages using fixed resolution structures, such as tables. This gives them the ability to control the look and feel of their Web sites. This fixed resolution approach has evolved to the point that the fixed resolution layout of Web pages has become the most common method to brand or uniquely identify Web sites. While this fixed resolution approach is good for site branding and product differentiation it does present a daunting technical problem for display of Internet content (designed for desktop computers) on small screen, low resolution, or different aspect ratio devices, such as cell phones and hand held computers.

BRIEF SUMMARY OF THE INVENTION

In accordance with aspects of the invention, mobile devices enabled to support resolution-independent scalable display of Internet (Web) content to allow Web pages to be scaled (zoomed) and panned for better viewing on smaller screen sizes are disclosed. The mobile devices employ novel processing of original Web content, including HTML-based content, XML, cascade style sheets, etc. to generate scalable content. The scalable content and/or data derived therefrom are then employed to enable the Web content to be rapidly rendered, zoomed, and panned. Display lists may also be employed to provide further enhancements in rendering speed.

According to further aspects, the mobile devices employ touch-sensitive display screens that enable users to provide various inputs to control display of content within Web pages. Exemplary user inputs include tap-based inputs to selectively zoom in on columns, images, and paragraphs. Users can also define a window to zoom in on via the touch-sensitive display.

Other features of the present invention will be apparent from the accompanying drawings and from the detailed description that follows.

BRIEF DESCRIPTION OF THE DRAWINGS

The appended claims set forth the features of the invention with particularity. The invention, together with its advantages, may be best understood from the following detailed description taken in conjunction with the accompanying drawings of which:

FIG. 1A is a block schematic diagram illustrating a first exemplary system infrastructure in accordance with the present invention in which content translation services are performed by a third-party proxy service that translates content requested from a client that is retrieved from one or more network resources into a scalable vector representation and delivers the translated content to the client;

FIG. 1B is a block schematic diagram illustrating a second exemplary system infrastructure in which the translation of content is performed at a content provider's web site and delivered directly to the requesting client;

FIG. 1C is a block schematic diagram illustrating a third exemplary system infrastructure in which content received from one or more network sources is translated into a scalable vector representation at the client;

FIG. 2A is a flowchart illustrating how data is retrieved, processed and transferred in accordance with the system infrastructure of FIG. 1A;

FIG. 2B is a flowchart illustrating how data is retrieved, processed and transferred in accordance with the system infrastructure of FIG. 1B;

FIG. 2C is a flowchart illustrating how data is retrieved, processed and transferred in accordance with the system infrastructure of FIG. 1C;

FIG. 3 is a block schematic diagram illustrating an exemplary architecture corresponding to the proxy server of FIG. 1A;

FIG. 4A is a representation of an exemplary web page has displayed on a conventional browser;

FIG. 4B is a schematic diagram illustrates various objects that are generated based on the HTML code of the web page of FIG. 4A;

FIG. 4C is a schematic diagram illustrating a set of vectors and bounding boxes corresponding to the objects generated in FIG. 4B;

FIG. 4D is a schematic diagram illustrating how various vectors and bounding boxes may be defined in accordance with the invention;

FIG. 4E is a representation of the web page of FIG. 4A after it has been offset and scaled in accordance with the invention;

FIG. 4F is a schematic diagram illustrating new datum points and bounding boxes corresponding to the scaled and offset web page;

FIG. 4G is a schematic diagram illustrating new vectors and bounding box parameters for a pair of objects in the scaled and offset web page;

FIG. 5 is a flowchart illustrating the logic used by the invention when translating content into a scalable vector representation of that content;

FIG. 6 is a flowchart illustrating client-side operations that are performed to create a rendered display page based on the translated content the client receives and user-input;

FIGS. 7A and 7B are representations of a nominal and a zoomed in column view of an exemplary web page as they might appear on a Palm device;

FIGS. 8A and 8B are representation of nominal and zoomed in view of an exemplary graphic image as they might appear on the Palm device;

FIGS. 9A and 9B are representations of a nominal and zoomed in view of a text portion of a web page as they might appear on the Palm device; and

FIG. 10 illustrates an exemplary computer system that may be used for implementing various aspects of embodiments of the invention.

DETAILED DESCRIPTION OF THE INVENTION

Apparatus and methods are described for creating resolution independent vector display of Internet content to allow it to be scaled (zoomed) larger and smaller for better viewing or to fit any resolution or screen size. In addition, infrastructure and methods are provided for delivering such content to clients.

In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, to one skilled in the art that the present invention may be practiced without some of these specific details. In other instances, well-known structures and devices are shown in block diagram form.

The present invention includes various operations, which will be described below. The operations of the present invention may be performed by hardware components or may be embodied in machine-executable instructions, which may be used to cause a general-purpose or special-purpose processor or logic circuits programmed with the instructions to perform the operations. Alternatively, the operations may be performed by a combination of hardware and software.

The present invention may be provided as a computer program product that may include one or more machine-readable mediums having stored thereon instructions, which may be used to program a computer (or other electronic devices) to perform a process according to the present invention. The machine-readable medium may include, but is not limited to, floppy diskettes, optical disks, CD-ROMs, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, magnetic or optical cards, flash memory, or other type of media/machine-readable medium suitable for storing electronic instructions. Moreover, the present invention may also be downloaded as a computer program product, wherein the program may be transferred from a remote computer (e.g., a server) to a requesting computer (e.g., a client) by way of data signals embodied in a carrier wave or other propagation medium via a communication link (e.g., a modem or network connection). Accordingly, herein, a carrier wave shall be regarded as comprising a machine-readable medium.

Client Overview

According to one embodiment, an ultra-thin client-side viewer provides the graphics, linking, caching, and function handling capabilities necessary for extending the web to almost any platform. It is designed as a lightweight browser (micro-browser) running directly on device operating systems. In alternative embodiments, the client-side viewer may be deployed as a standard browser plug-in, or Java applet for extending browser functionality. In one embodiment, the client-side viewer attains its small size and efficiency by taking advantage of the power of SVF (Simple Vector Format) to describe almost any current web content. SVF files can be handled with a tiny fraction of the client code required by normal web browsers because current browsers must interpret a large and growing number of file types and their idiosyncrasies. SVF was originally designed to handle a superset of the most commonly used file formats in the complex world of CAD. It can accommodate not only new graphical functions, but the storage and transfer of almost any foreseeable new functional capability. SVF has been under consideration by the W3C (World Wide Web Consortium) for adoption as a standard for vector content on the World Wide Web.

By working tightly with a server-side content translator, web content and functionality can be passed seamlessly to the end user platform without any degradation in the look or feel of the output. In addition, because the resulting file graphics are handled as vectors, the end user can control real time changes in the size of text and graphics as well as what portion of the file is viewable in the display. This "zoom and pan" capability, familiar to CAD and other vector content software users, adds dramatically to the usability of non-standard display sizes. For very small displays, real time zooming and panning allows the user to see graphics and text at sizes that make them easily readable, and then "back up" to view an entire page for context or pan in any direction for navigation. Because the client-side viewer manipulates vectors, there is no loss in quality as the display is zoomed. The graphics rendering engine within the client is so efficient that file manipulation happens in a fraction of a second. There is no perceptible wait for the user as the file is resized, or the window is repositioned. Content created for one display resolution now can be sized, real time, for any other display without degradation. Besides making small displays eminently usable, this technology extends web content into some surprising new arenas. For example, it enables normal desktop displays to be effective for individuals with visual impairment, or for content designed for 640×480 standard PC monitors to be shown without degradation on web billboards now appearing in cities like Seattle and San Francisco.

With a client of such extraordinary power packed in a tiny footprint, end user device manufacturers can free up valuable memory space for pre-fetching, caching and pre-loading content, dramatically improving performance for use in low bandwidth and portable applications. In the example of a wireless handheld device where expensive flash memory must be used instead of more cost effective bulk storage technology, the difference between consuming 10's of megabytes of flash memory with a standard browser versus running the client-side viewer described herein is dramatic.

Those "saved" megabytes of memory are now available for impressive interfaces, caching of often used content, and pre-fetching of intelligently selected linked files or pre-loading of content for targeted applications. For example, in a mapping application, the map tiles surrounding the viewed map could be downloaded and stored while the user was working with the initial tile, enabling an experience remarkably free from the current frustrations of waiting for a new map to be transferred for even the smallest change in magnification or coverage. If the user knows ahead of time what city they will visit on a business trip, maps and additional travel information in great detail could also be pre-loaded using a high bandwidth connection at home or in the office before heading out to shop or conduct business in the city. Additionally, SVF is a more efficient way to store web content. Resulting content files are reduced in size by anywhere from 20 to 80 percent over their source. SVF is also very compressible. With target file size reduction in the range of 90%, SVF files can take up as little as 1/10th the space of the web files in current use. This means that pre-translated content can be moved up to 10 times the rate of current web pages, and as much as 10 times as many pages, maps, stock charts, etc. can be stored for instant retrieval on the hand held platform as can be handled with current web technology.

When used on content created natively in SVF, additional capability can be extended to the client-side viewer.

Graphing the performance of stocks over time is only one use of SVF's ability to handle streams of data. Handling the output from financial systems, transactional systems, ERP packages, and CRM systems becomes easier and more flexible. Of course, systems integrators don't have to use these powerful capabilities to start with. If the target system provides web interfaces, these can be viewed, as designed, with no additional software to write, and no changes to the design or layout of the interface.

Server Overview

Enabling the client-side viewer to be so small and powerful is the server-side content translator. The server-side content translator rapidly translates Web content to SVF, compresses and encrypts the SVF results if desired, and transfers the vector formatted results to the client-side viewer. Alternatively, SVF files can be cached or stored in a file system for fetching and transfer at a later time. Pre-translated or cached content transfers are significantly faster as no conversion overhead is incurred, and file sizes are reduced using the more efficient SVF. Combine that with standard compression algorithms selectable for use with the client-side viewer for additional performance improvements.

During the translation process, and in the process of serving cached, pre-translated, or native SVF content, output files are "streamed" to the client-side viewer. Although this does not decrease the total time for file transfer, it can significantly improve the effective system performance for the end user. Content can be selectively streamed, with text and links coming through first, followed by graphic images and other content, for example. Should the user be accessing a link, rather than having interest in the entire file served, links can be selected early in the transfer and the next file download started immediately. In addition to streaming, the server-side content converter may also layer the content by type. This means that text can be put in one layer, links in another, GIF images in another, Javascript in another and so on. Layers can be turned on or off depending upon client capabilities, making files for less capable clients, or for users interested in a reduced functionality, higher transfer performance mode to be handled automatically.

All operational modes may be controlled through an administrative interface or accessible through a straightforward API (Application Program Interface). Furthermore, the system works with existing firewalls and within standard security protocols. In more secure modes, the server-side content converter and the client-side viewer may operate using Public/Private key authentication and encryption.

Exemplary System Infrastructures

In the following paragraphs, a description of three exemplary system infrastructures is provided. Schematic illustrations of these system infrastructures are shown in FIGS. **1**A, **1**B, and **1**C. It is noted that like-numbered components in these Figures perform substantially the same function. Therefore, any discussion of the functions of a component with reference to one or more of the infrastructures generally may apply to the other infrastructures as well, unless specifically noted otherwise.

A first of exemplary system infrastructure **10**A for implementing the invention is shown in FIG. **1**A. Infrastructure **10**A enables various clients, including wireless devices such as a cellular phone **12**, a wireless-enabled PDA **14**, and a wireless-enabled laptop computer **16**, as well as landline computers **18**, **20**, and **22**, to request content that is accessible via a network such as the Internet **24** to be retrieved from selected network resources, including web servers **26** and **28** and an FTP site **30**, wherein the content is translated into a scalable vector representation (e.g., SVF, also referred to herein as "vectorized content") through use of a proxy server **32** and sent to the requesting client. Upon being received by the client, the vectorized content is processed and rendered using a thin client to enable a user to view the content on the client device.

With reference to the flowchart of FIG. **2**A, the foregoing process is initiated by a client in a block **100**, wherein the client submits a request to proxy server **32** to retrieve and convert selected content. As depicted by a transfer path **34**, this comprises sending data **36**, which includes content network location indicia from which the content can be retrieved and proxy server network location information by which the content request may be delivered to over Internet **24** to proxy server **32**. Typically, it will be desired to retrieve a particular web page. Accordingly, the content network location indicia will comprise a URL (uniform resource locator) for the web page. Similarly, the proxy server network location information may also comprise a URL corresponding to a network access point for the proxy server. Optionally, the location information may comprise a network IP address for one or both of the content location and the proxy server location. If the content is to be retrieved from an Internet resource, the request will typically be sent using the HyperText Transfer Protocol (HTTP) over the TCP/IP transport.

Next, in a block **102**, the request is received by the proxy server and the proxy server checks its cache to see if it already has the request content in its cache. If it does, it sends this cached content back to the client. If it does not have the requested content cached, the proxy server sends out a request to retrieve the content from the network resource. For illustrative purposes, it will be assumed for the present example that the desired content comprises a web page that is stored on web server **26**. Typically, when the requested content comprises a web page, the content may be retrieved using conventional web content retrieval techniques, such as that employed by various modern browser clients, including Netscape Navigator and Internet Explorer. This generally comprises providing routing information, such as the URL for the web page (URL **38**) to routing services provided by Internet **24**, which routes the request to an appropriate network resource (e.g., web server **26**), as depicted by a transfer path **40**.

Typically, the URL will correspond to a web page whose content is stored by the web server in an HTML (HyperText Markup Language) document comprising HTML code and embedded text content, in addition to other optional content languages, that may contain references to other objects (e.g., HTML documents and graphic image files) stored locally to the server or stored on a remote server. For example, the HTML content corresponding to a single-frame web page is often stored in a single file, while multiple-frame web pages may comprise content that is stored in a single file or in multiple files. These files may be stored locally on the web server (e.g., on one of the server's hard disks), or on a local storage device connected to the web server via a local area network (LAN), such as a network attached storage (NAS) filer. Optionally, some of the web page's content may comprise one or more documents that are stored at remote locations that may be accessed via a WAN (wide area network) or the Internet.

HTML is a standardized language that describes the layout of content on a web page, and attributes of that content. This layout and attribute information is defined by sets of tags contained in HTML code corresponding to the page. The tags define various HTML layout and display information, including tables, paragraph boundaries, graphic image positions and bounding box sizes, typeface styles, sizes, and colors, borders, and other presentation attributes. A portion or all of a web page's text content may be contained in the parent HTML document corresponding to the URL. In addition to basic HTML, web page documents may contain XML (eXtensable markup language) code, as well as scripting language code, such as javascript. However, for simplicity, any documents containing web page content other than only graphic content that are discussed herein will be referred to as HTML documents.

In addition to HTML and other markup and scripting language content, it is very common for web pages to include graphical content. In general, graphical content is usually stored in an image file or files that are external from the parent HTML document for the web page. For example, the parent HTML document may contain one or more embedded image tags that reference the location where those images are stored. As before, the graphic images may be stored locally, or may be stored on remote servers that are accessed by the web server via a WAN, or the Internet. These files will typically comprise data stored in one of several well-known graphic formats, including bitmap files (BMP), GIF (Graphics Interchange Format) files, and JPEG (Joint Photographic Experts Group) files.

In response to receiving the request for content, web server **26** begins sending a parent HTML document **42** back to proxy server **32** in a block **104**. In a block **106**, the HTML content of the parent HTML document is parsed to search for references to external objects such as HTML frames and graphics. In a decision block **108**, a determination is made to whether any references are found. For each reference to an external object that is found, proxy server **32** requests to have the object retrieved from an appropriate network resource (e.g., a web server) in a block **110**, and data corresponding to the object is transmitted back to the proxy server, as depicted by locally accessible HTML documents **44** and graphic images **46**, as well as remotely accessible HTML documents **48** and graphic images **50**, which may be accessed via web server **28**. If the external object is a graphic image, there is no further processing of the object at this point. If the object is an HTML document, the functions provided by blocks **106** and **108** are repeated. Generally, this set of processing functions is repeated iteratively until all of the external objects are retrieved. However, as described below, there will be some instances in which certain objects will be retrieved at a later point in time. In addition to content stored on web servers that are accessed using HTTP, content may also be retrieved from various network sites using the File Transfer Protocol (FTP), such as FTP documents **51**, which are accessed via FTP server **30**.

In general, HTML documents and graphic files will be sent as packetized data streams using HTTP over one or more TCP/IP network connections, wherein the data streams will usually be asynchronous. Retrieval of HTML documents and graphic files corresponding to the embedded references will usually require additional transfer time. Furthermore, graphic content oftentimes comprises significantly larger file sizes than HTML content, leading to significant transfer times in some instances. For simplicity, the transfer of the various HTML documents and graphic files for the content request are depicted by HTML documents **52** and graphic documents **54**, which are transferred over a transfer path **56**.

When the HTML documents and graphic content are received by proxy server **32**, a scalable vector representation of the web page is generated in a block **114** by an HTML translator **58**. In brief, HTML translator **58** translates HTML, XML, and cascaded style sheet (CSS) layout content into a scalable vector representation, such as SVF. Details of the HTML translation process are contained below. In addition, the graphic images are converted into a compressed bitmap format in a block **116** by a graphics translator **60**. The vectorized content **62** and compressed bitmaps **64** are then streamed back to the client (i.e., computer **18**) in a block **118**, as depicted by a transfer path **66**. In one embodiment, the content portions are sent in separate streams using multiple connections. In another embodiment, the content portions are sent via a multiplexed stream using a single connection. As the vectorized content and compressed bitmap data are received by the client device, they are processed by a thin client **68** running on the client device, whereby a representation of the original web page content may be rendered on the client device's display screen at various user-selectable scaled resolutions and pan offsets in a block **120**, thereby enabling a user to more clearly see an overview or details in the web page. Further details of the client side processing are provided below.

As discussed above, wireless clients may also access the vectorized network (e.g., web site) content provided via proxy server **24**. The majority of this process is identical to that described above for land-line clients (e.g., computers **18**, **20**, and **22**), except for provisions required for sending data to and receiving data from wireless devices. In general, most wireless devices will access the Internet via a wireless service provider (i.e., a wireless telecommunications carrier) that is particular to that wireless device. Accordingly, a portion of the transmission path to and from proxy server **24** will comprise infrastructure provided by that service provider and/or shared with other service providers. For simplicity, this infrastructure is shown as a cellular tower **70** and a service provider data center **72**, although it will be understood by those skilled in the art that the connection path may comprise additional infrastructure components, including appropriate gateways and routers, that enable wireless devices to access proxy server **24**.

In some implementations, there will be no special formatting/protocol services that need to be performed by proxy service **24**—from the viewpoint of the proxy service, it will be immaterial whether the client is a land-based or wireless client; the special handling provisions for wireless devices will be handled entirely by the service providers infrastructure transparently at both ends of the communications path. In other instances, it may be desired or necessary to reformat the data content delivered to the wireless device at the proxy service. This will generally be dependent on the particular wireless protocol used, and what services are provided by the service provider for the wireless client.

Currently, in the United States, wireless clients generally access Internet **24** by using the Wireless Application Protocol (WAP). In Japan, the most popular access means is NTT DoCoMo's i-Mode wireless protocol. In addition to these wireless standards, new standards are anticipated to be in force in the near future, including NTT DoCoMo's FOMA (Freedom of Mobile Multimedia Access), which is transported over W-CDMA (Wideband Code Division Multiple Access), and CDMA-2000. For the purposes of the invention herein, it will be understood that those skilled in the mobile telecommunications arts will be knowledgeable about any particular format and/or transport protocol requirements that pertain to the particular protocol that is to be used.

A second exemplary system infrastructure **10**B for implementing the invention is shown in FIG. **1**B. As will be readily recognized, much of infrastructure **10**B is similar to infrastructure **10**A; however, rather than have a separate proxy server perform the proxy functions (retrieve and translate content), these functions are performed on machines operated by the web site in infrastructure **10**B.

The logic implemented by the invention when providing content to a client using infrastructure **10**B is illustrated in the flowchart of FIG. **2**B, wherein the process begins in a block **101** in which the client sends a content request **39** directly to the network site (e.g., web server **26**), as depicted by a transfer path **41**. In a block **103**, HTTP negotiations are performed to determine the format the content is to be delivered in. For example, the request may contain indicia identifying the type of content requested, such as an SVF MIME type (e.g., image/vnd.svf). This is to inform the web server that the request is for specially-formatted content rather than conventional content. The server first checks to see if it already has cached the requested content. If it has, it sends the content to the requesting client; otherwise, it retrieves the parent HTML document in a block **107**. It then performs processing steps in blocks **107**, **109**, and **111** to retrieve content referenced through embedded tags in a manner substantially similar to that discussed above with reference to respective blocks **106**, **108**, and **110**. The primary difference in this instance is that the web server does not receive requests from or send documents to a proxy server—rather, the content is retrieved and processed at the web server, wherein the retrieved content may be stored local to the web server or retrieved from a remote server in a manner similar to that described above.

As before, the retrieved HTML documents are translated into scalable vector representations by HTML translator **58** in a block **114**, while the graphic images are translated into a compressed bitmap format by image translator **60** in a block **116**, as depicted by vectorized content **62** and bitmap content **64**. The vectorized content and bitmap content are then streamed from the web server to the client in a block **119**, as depicted by a transfer path **67**. Upon arriving at the client, the vectorized content and bitmap content are processed, scaled, and rendered on the client in a block **120**.

A third exemplary system infrastructure **10**C for implementing the invention is shown in FIG. **1**C. In this configuration, the proxy functions are performed at the client. As shown by a block **113** in FIG. **2**C, the process for providing vectorized content to a client in accordance with infrastructure **10**C begins in a block **113**, in which the client sends a content request **37** to a network site, such as web server **26**, via Internet **24**. In response, the network site retrieves the parent HTML document and sends it to the requesting client in a block **115**. In a manner similar to that discussed above with reference to blocks **106**, **108**, and **110** of FIG. **1**A, the client first parses the parent HTML document searching for embedded references to external objects and retrieves these objects, whereupon the embedded reference search is performed on the newly retrieved document until all of the content corresponding to the original content request has been retrieved. This content is depicted by HTML documents **52** and image files **54**, which are sent from the network site to the client via a transfer path **69**. At this point, the client performs translations on the HTML content and the graphic image content that are substantially similar to that performed by the proxy server in FIG. **1**A or at the web site in FIG. **1**B, as provided by blocks **114** and **116**. The vectorized and image content is then processed and scaled by thin client **68** in a block **120**, as depicted by device output **71**.

Attention now is focused on the functionality provided by proxy server **24** in system infrastructure **10**A of FIG. **1**. Fundamentally, the proxy server functions as a proxy. It accepts requests for content from client devices as full URLs using standard HTTP mechanisms carried over a multiplexed TCP connection. Standard HTTP content negotiations features specify the formats in which content is to be delivered (SVF, bitmap, and possibly others, which can be handed off to cooperating client-side display software). As described in further details below, in some embodiments the proxy server appears for the client as a normal proxy (that is, the client knows it is retrieving content via the proxy), while in other embodiments the proxy is transparent to the client.

The proxy server responds to client content requests by delivering content in one of the requested formats, by retrieving the content in an appropriate format from its cache, or from an upstream content source (again using standard HTTP content negotiation features), or by translating upstream content from a supported original format to SVF or the client bitmap format.

Requests from the server installation to its cache and from the cache to upstream content sources are made in HTTP carried over TCP using simple straightforward Web content requests. For example, requests from clients to the proxy server comprise HTTP proxy requests (e.g., "GET http://www/xyz.com/some_page.html HTTP/1.0 . . . ") carried over TCP or over a lightweight multiplexing protocol over TCP. The multiplexing protocol allows the server to push image thumbnails to the client before the SVF stream is available, as well as offering a channel for control and status information, more simultaneous channels than the client operating system may support, and a mechanism for prioritizing information flow from server to client under loose client control. In addition to HTTP requests, the proxy server architecture supports other user-level protocols, such as FTP and Gopher.

Details of some of the primary components of the proxy server architecture are shown in FIG. **3**. Internally, the proxy server comprises a suite of coordinated processes connecting to upstream content through an HTTP cache **74**. In one embodiment all functions except caching are performed in a single process, wherein multiple threads are used to effect asynchronous I/O. Separate processes communicated via persistent multiplexed connections carried over the most efficient reliable transport available (e.g., Unix sockets over single processor and symmetric multiprocessor (SMP) computers; TCP sockets between separate computers). All processes are capable of servicing multiple requests simultaneously. No process maintains client state outside the context of a single request, so all components can be repeated and load balanced across multiple CPU's of an SMP computer or across separate computers on a LAN.

The various content translators used by the proxy server accept (via HTTP PUT) or request (driven by HTTP proxy GET/POST) content in supported, but client-unsupported, formats; and return (via HTTP PUT or GET/POST response) one or more representations of that content in a client-supported format. In the embodiments illustrated in FIGS. **1**A-C, two translators are used: HTML translator **58** and image translator **60**. Future content types may be accommodated by new translators, by extending existing translators to cover the new content types, or by extending the client's capabilities. Standard HTTP content negotiation mechanisms are used to inform the proxy server of the client's capabilities and expectations on each request.

Managers at the proxy server coordinate the operations of other components. Two managers are presently defined; a client manager **73** that handles client proxy requests, and a request manager **75** that handles unproxied HTTP requests from other services. The managers accept requests, attempt to service them from HTTP cache **74**, and drive HTML translator **58** and image translator **60** when content does not match the clients' requirements. Managers also handle translator requests for inline content (e.g., image dimensions for page layout), and push translated content into HTTP cache **74**. Additionally, the client manager coordinates delivery of primary and inlined content, and provides process and status information to the clients.

As discussed above, HTML translator **58** creates a scalable vector representation of the original HTML content of a requested web page. In order to better explain how translation of HTML content is performed, one embodiment of a translation process is described below as applied to an exemplary web page. In addition, details of conventional web page client and server-side processing are provided so as to clarify how web content is laid out during a pre-rendering process on the client.

FIG. **4** shows a representation of a web page **210** served from an exemplary stock brokerage Internet web site as it would appear when rendered on a modern Internet browser, such as Microsoft's Internet Explorer or Netscape's Navigator. Web page **210** is exemplary of many web pages that implement frames, and includes two adjacent frames **212** and **214**. A logo graphic object **216**A is displayed at the top of frame **212**, which additionally includes a "MARKETS" text header **218**A, an "INVESTMENTS" text header **220**A, and a plurality of links with overlaying graphic objects, including a "DOW" link **222**A, a "NASDAQ" link **224**A, an "OPTIONS" link **226**A, a "CHARTS" link **228**A, a "MUTUAL FUNDS" link **230**A, a "IRA, 401K OPTIONS" link **232**A, and a "TAX INFORMATION" link **234**.

A horizontal group of links **236** is disposed at the top of frame **214**, and includes a "QUOTES" link **238**A, a "HOT PICKS" link **240**A, a "CALENDARS" link **242**A, and a "NEWS" link **244**A. An advertisement banner **246**A is displayed just below the horizontal group of links and just above a "NEWS SPARKS MARKET" headline **248**A. Frame **214** also includes a pair of graphic image objects, including a DOW chart **250**A and a NASDAQ chart **252**A. A set of user input objects is disposed adjacent to DOW chart **250**A within a graphic object **254**A, including an "ACCOUNT #" input box **255**A, an "ACCESS CODE" input box **256**A, and a "LOGIN" button **257**A. In addition to the foregoing objects, frame **214** also includes text objects **258**A and **260**A.

An HTML listing corresponding to web page **210** is presented below as LISTING 1. Note that LISTING 1 sometimes refers to object descriptions and link paths rather than the text or path location of actual objects for simplicity, and that other elements commonly found in HTML pages, such as META entries, are omitted for clarity.

LISTING 1

```
<html>
<head><title>"MARKET HOME"</title></head>

<body bgcolor="#FFFFFF" link="0033CC" vlink="0033CC">

<frameset cols="25%,75% frameborder=0 border=0>
<frame>
<align=left><align=top>
<img src="/directory path/logo.gif" align = left border="0" height="80" width="100">
<br><br>
<t3>TEXT HEADER #1 align=left</t3><br>

<table width="90%" border=0 cellspacing=10 cellpadding=0 bgcolor="#000000"
align=center>
     <tr>
```

-continued

LISTING 1

```
                <a href=”URL or path for LINK #5” <img src=”/directory
                path/GRAPHIC#2” height=”50” width =”150></a>
    <tr>
                <a href=”URL or path for LINK #6” <img src=”/directory
                path/GRAPHIC#3” height=”50” width =”150></a>
    <tr>
                <a href=”URL or path for LINK #7” <img src=”/directory
                path/GRAPHIC#4” height=”50” width =”150></a>
    <tr>
                <a href=”URL or path for LINK #8” <img src=”/directory
                path/GRAPHIC#5” height=”50” width =”150></a>
</table>
    <br>
                <t3>TEXT HEADER #1 align=left</t3>
    <br>
<table width=”90%” border=0 cellspacing=10 cellpadding=0 bgcolor="#000000"
  align=center>
    <tr>
                <a href=”URL or path for LINK #9” <img src=”/directory
                path/GRAPHIC#6” height=”50” width =”150></a>
    <tr>
                <a href=”URL or path for LINK #10” <img src=”/directory
                path/GRAPHIC#7” height=”50” width =”150></a>
    <tr>
                <a href=”URL or path for LINK #11” <img src=”/directory
                path/GRAPHIC#8” height=”50” width =”150></a>

</table>
</frame>

<frame>

<table>
    <tr>
                <table width="100%" border=0 cellspacing=15 cellpadding=0
                  bgcolor="#000000" align=center>
          <tr>
                <td><a href=”URL or path for link#1”> alt=”QUOTES”</a>
                <td><a href=”URL or path for link#2”> alt=”HOT PICKS”</a>
                <td><a href=”URL or path for link#3> alt=”CALENDERS”</a>
                <td><a href=”URL or path for link#4>alt=”NEWS”</a>
</table><br>
      <br>
                <img src=”URL for GRAPHIC #9” align=center
                  border=”0” height=”80” width=”325”>
    <br><t1>HEADLINE TEXT>/t1>
    <table>
            <Colgroup span=”2”>
                <Col width = “400” align=“center”>
                <Col width = “200” align=”center”>
                <tr><td>
                    <img src=”/directory path/GRAPHIC #10” align = center
                    border=“0” height=”180” width=”350”>
                <td>
  /* INPUT FOR ACCOUNT NUMBER AND ACCESS CODE */
    <SCRIPT LANGUAGE =”Javascript”>
  <!---
                [Javascript variable declarations]
                [Javascript functions to enable login] ---!>
  </SCRIPT>
        <table>
              <td>
                <img src=”/directory path/GRAPHIC #11” align = center>
                <table width=”150” height=”25”>
                <td>
                <font size=-2 face="arial,helvetica,verdana">Account #</font>
                <tr><input type=text name="USERID" maxlength=9 size=20>
                <tr><font size=-2 face="arial, helvetica">Access Code:</font>
                <tr><input type=password name="PASSWORD" maxlength=10 size=20
                    onKeyDown="SuppressEnterBell(event)"
                    onKeyPress="SuppressEnterBell(event)"
                    onKeyUp="SubmitOnEnter(event)">
                <br>&nbsp;
                <br><input type="button" value="Login"
                OnClick="ProcessForm( )">&nbsp;&nbsp;<input type="reset">
```

-continued

LISTING 1

```
 91.                        <br>&nbsp;
 92.                        </td>
 93.                        </table>
 94.              </table>
 95.     <tr>
 96.                        <img src="/directory path/GRAPHIC #12" border="0
 97.                          height="200" width="350">
 98.    <tr>
 99.                              <p>TEXT FOR TEXT OBJECT #1</p><br>
100.                              <p>TEXT FOR TEXT OBJECT #2</p>
101.              </table>
102.            </frame>
103.  </frameset>
104.  </html>
```

Web page documents comprise HTML code that is parsed, interpreted, and rendered by a browser. An HTML document comprises a plurality of HTML "markup" elements (tags) with corresponding attributes, that are used to describe the layout and formatting of various objects, including plain text and graphic objects, embedded between tag pairs. Exemplary elements include text tags (e.g., <b></b> for bolding text), links (e.g., <a href="URL"></a>), formatting (e.g., <p></p> for creating a new paragraph, graphical (e.g., <img src="name">), wherein "name" defines an absolute or relative location at where an image is stored, tables (e.g., <table></table>) creates a table, and forms (e.g., <form></form> creates all forms).

As of Netscape Navigator 3.0 (and other later browsers), web pages could include frames. When using frames, the display page is divided into multiple framed areas. Framing enables a single display page to include source code from several HTML documents (one for each frame) or optionally, enables a single document to include more complicated grouping of contents whereby different content groups are contained in separate frames. Frames are commonly found on the web pages at sites that display a great deal of text and graphical content, such as MSN.com, ESPN.com, and USAToday.com.

With reference to the flowchart of FIG. **5**, the process for translating the HTML content into a scalable vector representation proceeds as follows. The process is initiated when the proxy server receives the HTML corresponding to the parent document (and frame documents, if appropriate), whereupon a pre-rendering parsing of the HTML is performed to determine where to place the various objects on the display page in a block **150**. For example, elements such as tables, column definitions, graphic images, paragraphs and line breaks are identified. If frames are included, each frame is examined in the sequential order it appears in the HTML document, or the order in which the HTML documents corresponding to the frames in a frameset are downloaded to the browser. During further processing, the actual objects are rendered in their respective positions. Some of these objects are rendered almost immediately, such as plain text, while other objects, such as graphic objects, must first be retrieved prior to being fully-rendered. With respect to tables, there are some instances in which all of the objects corresponding to the cells in the table must be retrieved prior to rendering any of the table, while a well-designed table can be rendered incrementally. For example, by using Column grouping, the format of the corresponding table can be quickly determined by the browser. In some instances, one or more bitmaps may actually need to be fetched before the page layout can be determined.

Next, in a block **152**, the content is separated into objects based on logical groupings of content portions and a page layout is built using bounding boxes that are produced for each object. As the primary HTML document is parsed, logical groupings of content will emerge. For instance, text content contained within paragraph tags <p> </p> forms a logical grouping of text content. In essence, a logical grouping means the content should appear together as a logical group, such as within a substantially rectangular outline, in the rendered page. Other logical groupings include frames, table content, row content, single line entries such as headlines and headers, and user-interface objects, as well as graphic layout objects, such as separator bars, and graphic images. In addition to logically grouping content into objects, a "bounding box" is defined for each object. In general, the bounding box defines an outlined shape within which the content (text or graphic image) will appear. In most instances, the bounding box will be substantially rectangular in shape. However, bounding boxes comprising more complex shapes may also be produced.

In further detail, the following explains how objects corresponding to graphic images are produced. In HTML, objects comprising graphic content are identified by an <img src="/local directory path/graphic image file" (for a local graphic image) or "URL" (for a remote graphic image)> or <object> or other tags. In the foregoing tag, local graphic images are typically stored on the same server as the web page, or another computer that is local to the site's server, and generally are located through a local directory path (absolute or relative to the location of the present page) that points to the graphic image file. Remote images are those images that are stored on servers at sites that are remote to the web server. For example, with reference to LISTING 1, when the parser encounters line 9, the browser identifies that data comprising a graphic image corresponding to logo graphic object 1 will be arriving (or may have already been received), and the displayed image is to have a height of 80 pixels and a width of 100 pixels. The location of each object on a display page will be dependent on previous HTML layout elements, such as tables, paragraphs, line breaks, and other graphic objects. The size and location of the other graphic objects (i.e., graphic objects #2-12) on the page are determined in a similar manner. The HTML code for these objects are shown in lines 16, 19, 22, 25, 34, 37, 40, 59, 67, 78 and 96, respectively. As identified in the HTML code, data corresponding to graphic objects #9 (advertisement banner **46**A) is forwarded to the browser from an external site (as indicated by the URL to GRAPHIC #9), while graphic objects 1-8 and 10-12 are sent from the web site the parent HTML document is sent from.

In a similar manner, the foregoing technique is applied to the HTML code in the primary document to identify other types of objects as well. In addition to parsing the primary HTML document, similar processing is performed on referenced documents, such as documents that include frame content that is defined and stored separate from the primary HTML document.

A representation of the results of the functions performed in block **152** are shown in FIG. **4**B. In the Figure, objects corresponding to the original content of FIG. **4**A are shown with an appended "B" that is added to each object's root reference number, wherein the root reference number for an object is that same as the logically grouped content in FIG. **4**A that it corresponds to, e.g., an object **248**B is generated for "NEWS SPARKS MARKET" headline **248**A, etc.

Next, in a block **154**, the page layout is defined based on the bounding boxes. In actuality, generation of the page layout information is performed in conjunction with defining the boundary boxes for the objects, wherein the location of a given object is based on the location of other related (e.g., if within a table) or non-related objects corresponding to HTML content that have been previously parsed. For example, the location of a given paragraph will depend on the other content for the page that are listed prior to the definition for the paragraph in the primary HTML document or referenced document, if applicable. As the HTML content of the primary and any referenced HTML documents are parsed, the page layout is generated based on the various HTML tags and the content embedded between tag pairs and/or referenced by a tag pair statement (e.g., graphic images).

As will be recognized by those skilled in the art, the functions performed in blocks **150**, **152**, and **154** are commonly performed by conventional browsers during a pre-rendering process. In some browsers, these functions are performed by the Mozilla rendering engine, which comprises open source software that is readily available for use by developers. At present, the software for the Mozilla rendering engine may be accessed via the Internet at www.mozilla.org. Accordingly, in one embodiment, the present invention uses core functionality provided by the Mozilla rendering engine source code to perform the functions of block **150**, **152**, and **154**.

At this point, the present invention deviates substantially from the prior art by using the various object layout data generated during the pre-rendering process to generate a scalable vector representation of the original page content. First, in a block **156**, a datum point is defined for the page and the bounding box for each object. For example, as shown in FIG. **4**C, a rendered page datum **262** is defined to be coincident with the upper left hand corner of the display frame of the rendered page for the web page. Generally, any point on the page may be used as the page datum—the only requirement is that the page datum that is selected is used consistently throughout the process. The use of the upper left hand corner of the display frame is advantageous since the location of the first object encountered in the HTML code for a page is located relative to this corner.

In general, the datum points for each object may also be located any place on the object, as long as the object datum points are used in a predictable manner. For example, as depicted in FIG. **4**C, various datum points for corresponding objects are defined to be coincident with the upper left hand corner of the bounding box for that object, wherein the object's datum point shares the root reference number of the object with an appended "C."

Once the page's datum point and an object's datum point are known, a vector between these points is generated for each object in a block **158**. With reference to FIG. **4**D, in one embodiment, wherein the page datum point corresponds to the upper left and corner of the display frame and is assigned an XY value 266 of 0,0, the vector for a given object may be stored as the XY value of the datum point of that object relative to 0,0, such as a value of 150, 225 (ref. num. **268**) for a vector **250**D pointing to an object datum **250**C, and a value of 150, 425 (ref. num. **270**) for a vector **252**D pointing to an object datum **252**C. In another embodiment, each vector may be stored as XY data relative to a 0,0 datum point corresponding to the upper left hand corner of the frame the object belongs to. For example, a vector **250**D' from a frame datum **214**D to object datum **250**C is stored as 20, 200 (ref. num. **268'**), while a vector **252**D from frame datum **214**D to object datum **252**C is stored as 20, 425. In this embodiment, offset information for each frame relative to a known datum will also be stored, as depicted by a vector **214**D.

The scalable vector representation is completed in a block **160**, wherein a reference is created for each object that includes or links an object's content and attributes, such as object type (e.g., text, image), object typeface, and boundary box parameters, to the object's vector. For example, object **250**B is a graphic image having a vector **250**D and a bounding box that is 180 pixels high and 350 pixels wide, while object **252**B is a graphic image having a vector **252**D and a bounding box that includes a height of 200 pixels and a width of 350 pixels. This enables client-side operations to be performed that only initially consider the vectors, wherein if it is determined that a vector's endpoint (and/or the bounding box corresponding to the object the vector points to) would appear off of a display, there is no need to retrieve the content and attribute data linked to the vector. This concept is explained in further detail in the following section.

It is noted that a portion of the display content produced on a client device will never contain any rendered content, as this portion is reserved for the browser's user interface. In WINDOWS™ environments, this portion will include the browser's window frame, as well as the pulldown and icon menus provided in the browser's user interface, which are depicted by a box **264** in the Figures herein.

Client-Side Software and Processing

As discussed above, the present invention supports a wide variety of clients, including land-based clients and wireless clients. Each client requires some client-side software that enables the scalable vector content data provided to it to be rendered at a user-selectable scale factor and offset on the client's display, such as a monitor or built-in LCD screen.

By enabling original content from a web site to be displayed in such a resolution-independent manner, users will be able to view content in a manner that did not previously exist, greatly enhancing the user experience. For example, in some implementations the client may be a personal computer (PC). Using a least-common denominator approach, many web pages are designed for a smaller resolution (for example 640×480 pixels, a minimum resolution commonly supported by nearly all PC's, including legacy PC's) than the resolution provided by the video output capabilities available with many of today's PC's, such as 1024×768 pixels, 1280×1024 pixels, and even 1600×1200 pixels. As a result, when these web pages are displayed on a high-resolution display, they occupy only a portion of the display, making portions of the pages, especially those portions containing small text, difficult to read. By enabling users to selectively magnify the entire page, these design flaws are easily overcome. Alternatively, the client may be a small device, such as a hand held computer or a cell phone, which has a smaller display resolution than common Web pages are designed for. As explained below, through use of the invention's scalable vector representation and client-side processing, users are enabled to view the entire content of billions of existing Web pages using hand-held devices in a simple and reasonable way.

In one embodiment, the client software may be a plug-in to a Web browser, such as Netscape Navigator or Microsoft Internet Explorer. Such a plug-in might have the browser download the data and display it in a sub-window of the browser. Alternatively, the client software may be a Java applet running in a browser. As another option, the client software may be a stand-alone program that interfaces with the proxy server or proxy software directly. The client software may bypass the proxy when requesting information that won't be translated to vectors, such as bitmaps.

With reference to FIG. **6**, client-side processing proceeds in the following manner. In a block **160**, the vector representation data (i.e., vectorized HTML content and compressed bitmap content) for the web page is gathered at the client. Typically, this data will be stored in a cache at the client as it is being received, and the client simply retrieved the data from the cache. In a block **162**, a display list of vectors is built. This process is well known in the CAD arts, and is enabling rapid zooming of vector-based objects. In a block **164**, user selectable scale and offset (pan) values are determined. Based on various user interactions with the user-interface of the client, the user is enabled to control the zoom (size) and offset of the rendered page. For example, suppose the user provides zoom and offset inputs to produce a rendered page **210**E, as shown in FIG. **4**E. In this rendered page, the original origin is now off of the screen (the page image is shifted upward and toward the left—see FIG. **4**F), and the view has been scaled approximately 1.3 times.

Next, in a block **166**, the vectors and boundary boxes are processed based on the scale and offset, and a bounding box defining the limits of the display content is determined. The results of this step are shown in FIG. **4**F, while FIG. **4**G shows specific details on how the vectors and bounding boxes corresponding to image objects **250**B and **252**B (now **250**B' and **252**B', respectively) are processed. Logically, there are generally two ways to scale and offset the rendered content. In one embodiment, vectors and bounding boxes are mapped to a virtual display area in memory that has much greater resolution (e.g., 100,000×100,000 pixels) than any real display, and a virtual display limit bounding box is scaled and moved around over the virtual display area. Accordingly, during subsequent processing described below, objects falling within the display bounding box are rendered by reducing the scaling of those objects in the virtual display to how the objects will appear on the client device display relative to the virtual display bounding box. In the alternate, a fixed reference frame corresponding to the display resolution of the client device screen is maintained, wherein all vectors and bounding boxes are scaled and offset relative to the fixed reference frame. Each scheme has its advantages and disadvantages. One advantage of the second method is that the display bounding box is always maintained to have a size that matches the resolution of the content display area on the client device.

As shown in FIG. **4**G, respective offsets in X and Y, (−ΔX and −ΔY in the Figure) are applied to the starting point of each of the vectors. The vectors are then scaled by a scale factor "SF." The results of the new vectors are depicted by vectors **250**D" and **252**D". This produces a new datum for each object's bounding box that is relative to rendered page datum **262**, which remains fixed. As discussed above, only a portion of the display screen will actually be used to display content (as defined by a display limit bounding box **266** in this embodiment), while other portions of the screen, including box **264**, will comprise a generally fixed-size user interface. Accordingly, rendered page datum **262** is not located at the upper left hand corner of the display area, although it possibly could be located at this point when either the current user interface is inactive (i.e., the display portion of the user interface is temporary disabled) or the user interface is contained in other portions of the display.

This foregoing process establishes a starting point (the new datum) for where the content in each object's bounding box will be rendered. At this point, each object's bounding box is then drawn from its new datum using the scaling factor. For example, in the original web page **210**D (FIG. **4**D), bounding box **250**B had an X-axes datum of 150 pixels, a Y-axis datum of 225 pixels, and a height and width of 180×350 pixels. In contrast, after being offset and scaled, bounding box **250**B' has an X-axis datum of 150*SF−ΔX, a Y-axis datum of 225*SF−ΔY, and a height and width of 180*SF×350*SF.

Returning to the flowchart of FIG. **6**, once the vectors and bounding boxes are offset and scaled, content corresponding to objects having at least a portion of their bounding boxes falling within the display limit bounding box is retrieved from the client device's display list in a block **168**. For examples, as shown in FIG. **4**F, content corresponding to all of the objects except for those falling entirely outside of display limit bounding box **266** (objects **216**, **238**, **240**, **242** and **244**) is retrieved from the display list. That content is then scaled in a block **170**. For image content, this comprises decompressing and scaling the compressed bitmaps corresponding to those images. For text content, this comprises scaling the font (i.e., typeface) that the text content portions of the web page are written in the parent HTML document and any referenced documents. There are various techniques for typeface scaling that may be implemented here, depending on the available resources provided by the operating system of the client device. For example, for WINDOWS™ operating systems, many TRUETYPE™ fonts are available, which use a common scalable definition for each font, enabling those fonts to be scaled to just about any size. In other cases, such as current PDA (e.g., Palm Pilots) operating systems, there is no existing feature that supports scaling fonts. As a result, bitmapped fonts of different font sizes and styles may be used. In addition to scaling image and text content, other types of content, such as separator lines and borders may also be scaled by block **170**.

The process is completed in a block **172**, wherein those portions of the scaled content falling within the display limit bounding box are rendered on the client device's display.

As discussed above, it is foreseen that the invention will be used with client devices having small, low resolution displays, such as PDAs and pocket PCs. Examples of various views of an exemplary web pages obtained from the YAHOO™ web site are shown in FIGS. **7**A-B, **8**A-B and **9**A-B. For instance, FIG. **7**A represents how the YAHOO™ home page might appear on a Palm IIIc color PDA.

In addition to directly scaling and offsetting content, the client user-interface software for PDA's provides additional functionality. For instance, a user may select to view a column (results represented in FIG. **7**B by tapping that column with a stylus, a shown in FIG. **7**A. Similarly, the user may select to zoom in on an image by tapping the image with the stylus, as shown in FIGS. **8**A and **8**B, or select to view a paragraph in an article by tapping on the paragraph, as shown in FIGS. **9**A and **9**B. It is noted that in some instances, the display of the paragraph may be reformatted to fit the characteristics of the display, rather than following the original format in the zoom-out view.

It is further noted that that different scaling factors can be applied to the X and Y axis so as to change the aspect ratio of the display. For example, a Web page may be designed to be displayed on a computer having a resolution of 800×600 pixels, or a 4X to 3Y aspect ratio. In this case, the display corresponds to a "landscape" layout, wherein there are more pixels along the X axis than along the Y axis. Conversely, many handheld devices display images having a "portrait" layout, wherein there are more pixels along the Y axis than the X axis. By enabling different scaling factors to be applied to the X and Y axes, the present invention enables the aspect ratio of a rendered display image to be adjusted to better fit the aspect ratio of the client device.

An Exemplary Computer Architecture

An exemplary machine in the form of a computer system **500** in which features of the present invention may be implemented will now be described with reference to FIG. **10**. Computer system **500** may represent a workstation, host, server, print server, or printer controller. Computer system **500** comprises a bus or other communication means **501** for communicating information, and a processing means such as processor **502** coupled with bus **501** for processing information. Computer system **500** further comprises a random access memory (RAM) or other dynamic storage device **504** (referred to as main memory), coupled to bus **501** for storing information and instructions to be executed by processor **502**. Main memory **504** also may be used for storing temporary variables or other intermediate information during execution of instructions by processor **502**. Computer system **500** also comprises a read only memory (ROM) and/or other static storage device **506** coupled to bus **501** for storing static information and instructions for processor **502**.

A data storage device **507** such as a magnetic disk or optical disc and its corresponding drive may also be coupled to bus **501** for storing information and instructions. Computer system **500** can also be coupled via bus **501** to a display device **521**, such as a cathode ray tube (CRT) or Liquid Crystal Display (LCD), for displaying information to an end user. Typically, an alphanumeric input device **522**, including alphanumeric and other keys, may be coupled to bus **501** for communicating information and/or command selections to processor **502**. Another type of user input device is cursor control **523**, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor **502** and for controlling cursor movement on display **521**.

A communication device **525** is also coupled to bus **501**. Depending upon the particular presentation environment implementation, the communication device **525** may include a modem, a network interface card, or other well-known interface devices, such as those used for coupling to Ethernet, token ring, or other types of physical attachment for purposes of providing a communication link to support a local or wide area network, for example. In any event, in this manner, the computer system **500** may be coupled to a number of clients and/or servers via a conventional network infrastructure, such as a company's Intranet and/or the Internet, for example.

Importantly, the present invention is not limited to having all of the routines located on the same computer system. Rather, individual objects, program elements, or portions thereof may be spread over a distributed network of computer systems. Additionally, it is appreciated that a lesser or more equipped computer system than the example described above may be desirable for certain implementations. Therefore, the configuration of computer system **500** will vary from implementation to implementation depending upon numerous factors, such as price constraints, performance requirements, and/or other circumstances. For example, according to one embodiment of the present invention, a cell phone or a hand held computer may comprise only a processor or a micro controller and a memory, such as a micro code ROM or RAM, for storing static or dynamically loaded instructions and/or data.

In the foregoing specification, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

What is claimed is:

1. A mobile device, comprising:

a processor,

a wireless communications device operatively coupled to the processor, to facilitate communication with a network via which Web content may be accessed;

a touch-sensitive display;

a memory, operatively coupled to the processor; and

storage means, operatively coupled to the processor, in which a plurality of instructions are stored that when executed by the processor enable the mobile phone to perform operations including,

enabling a user to request access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

retrieving HTML-based Web content associated with the Web page;

translating the HTML-based Web content to produce scalable vector-based page layout information;

employing the scalable vector-based page layout information and/or data derived therefrom to,

render a view of at least a portion of the Web page on the touch-sensitive display using a first scale factor; and

re-render the Web page in response to associated user inputs to enable a user to iteratively zoom in and out views of the Web page on the display while preserving the original page layout, functionality, and design of the content on the Web page defined by the HTML-based Web content,

wherein preservation of the functionality defined by the HTML-based content includes preservation of hyperlink functionality.

2. The mobile device of claim **1**, wherein the device comprises a mobile phone.

3. The mobile device of claim **1**, wherein the device comprises one of a hand-held device or a palm-held device.

4. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input made via the touch-sensitive display.

5. The mobile device of claim **1**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

6. The mobile device of claim **1**, wherein the Web content includes at least one hyperlink, and wherein execution of the instructions performs further operations comprising:

enabling the user to select the hyperlink via the touch-sensitive display; and, in response thereto, retrieving and translating HMTL-based Web content associated with the hyperlink to produce additional scalable vector-based page layout information; and

employing the additional scalable vector-based page layout information and/or data derived therefrom to render the Web content associated with the hyperlink on the touch-sensitive display.

7. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising enabling the Web content to be displayed at different resolutions by scaling the scalable vector-based page layout information to resize a view of the Web page on the display in response to associated user inputs.

8. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising returning the display of the Web content to a previous view in response to a corresponding user input made via the touch-sensitive display.

9. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising enabling a user to pan a display of the Web content in response to a corresponding user input made via the touch-sensitive display.

10. The mobile device of claim **9**, wherein execution of the instructions performs further operations comprising enabling the display of the Web content to be panned in real-time.

11. The mobile device of claim **1**, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein the scalable vector-based page layout information and/or data derived therefrom is scaled to render a display having a different aspect ratio.

12. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising enabling a user to view a column of the Web content at a higher resolution than a current resolution by tapping on the column via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is displayed to fit across the touch-sensitive display.

13. The mobile device of claim **1**, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to view an image at a higher resolution than a current resolution by tapping on the image via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that the image is displayed to fit across a width of a display area of the touch-sensitive display.

14. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising enabling a user to view a paragraph of the Web content at a higher resolution than a current resolution by tapping on the paragraph via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is displayed to fit across a width of a display area of the touch-sensitive display.

15. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising:

generating a display list derived, at least in part, via use of the vector-based page layout information; and

employing the display list to re-render the display of the Web page.

16. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising:

parsing HTML-based code corresponding to the received Web content to logically group content into objects, the objects including a plurality of display objects;

defining a primary datum corresponding to a page layout; and,

for each display object,

defining an object datum corresponding to a layout location datum for the object's associated display content;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to its corresponding vector.

17. The mobile device of claim **16**, wherein execution of the instructions performs further operations comprising:

mapping the object vectors to a virtual display area in memory.

18. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising:

parsing the HTML-based content to logically group content into objects;

generating page layout information including a bounding box for each object, the bounding box defining width and height dimensions for the object; and

storing information that links each object with its corresponding page layout information;

wherein the page layout information further includes information from which a page layout location of each of the bounding boxes can be determined.

19. The mobile device of claim **1**, wherein the scalable vector-based content includes scalable text content, and wherein execution of the instructions performs further operations comprising scaling a scalable font to render the scalable text content.

20. The mobile device of claim **1**, wherein at least a portion of the instructions comprise Java-based instructions configured to be executed on a Java virtual machine.

21. The mobile device of claim **1**, wherein translating the HTML-based Web content to produce scalable vector-based page layout information comprises:

processing the HTML-based Web content with a rendering engine to generate page layout information corresponding to the original page layout as interpreted by the rendering engine; and

employing the page layout information to produce scalable vector-based page layout information.

22. The mobile device of claim **21**, wherein the page layout information defines a layout location for a plurality of objects, including text objects, graphic layout objects, and/or image objects included on the Web page, and wherein producing vector-based page layout information comprises:

defining a primary datum corresponding to a page layout; and,

for each object,

defining an object datum corresponding to the layout location for the object on the page layout;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to its corresponding vector.

23. The mobile device of claim **22**, wherein execution of the instructions performs further operations comprising effecting a zoom operation combined with a pan operation by,

for each of the plurality of display objects to be included in a panned view of the Web page to be rendered on the display,

scaling page layout information associated with the display object using a scale factor corresponding to a zoom level associated with the zoom operation to determine a scaled datum;

determining an offset corresponding to the pan operation and combining the scaled datum with the offset to produce a scaled and offset datum that defines a location where the display object is to be rendered on the panned view of the Web page;

scaling content associated with the display object using the scale factor; and

rendering the scaled content at the location defined by the scaled and offset datum to render the display object on the panned view of the Web page.

24. The mobile device of claim **23**, wherein rendering scaled content associated with a text object comprises:

retrieving presentation attributes for the text object, the presentation attributes including a font typeface, size and color;

employing a scalable font associated with the font typeface to render text associated with the text object in a color associated with the color attribute, wherein the text is rendered relative to a location associated with the scaled and offset datum for the text object, and wherein the scale applied to the scalable font is a function of the scale factor and the font size.

25. The mobile device of claim **21**, wherein the original format of the Web page defines a width for the Web page, as interpreted by the rendering engine, and wherein execution of the instructions performs further operations comprising:

determining an applicable scale factor to fit the width of the Web page across a display area of the touch-sensitive display; and

employing the scale factor that is determined as the first scale factor.

26. The mobile device of claim **1**, wherein zooming operations are effected by applying a mathematical transformation to a plurality of points in a two-dimensional coordinate system comprising X and Y axes, including points comprising datum points having corresponding vectors included in the scalable vector-based page layout information defining page layout locations of corresponding text and image objects mapped to the two-dimensional coordinate system, wherein the mathematical transformation comprises,

$$X'=X*\mathrm{SF};$$

$$Y'=Y*\mathrm{SF};$$

wherein X, Y is the location of a point prior to transformation, X', Y' is the location of the point after transformation, and SF is the scale factor.

27. The mobile device of claim **26**, wherein the mathematical transformation is applied to points in a first coordinate system comprising a virtual coordinate system associated with a virtual display area onto which page layout information is mapped to a second coordinate system comprising a device coordinate system corresponding to a pixel resolution of the display of the mobile device, wherein points are mapped from the first coordinate system to the second coordinate system using the mathematical transformation.

28. The mobile device of claim **1**, wherein execution of the instructions performs further operations comprising maintaining at least one instance of the page layout information in a manner that is independent of the zoom levels used to view the web page on the display.

29. The mobile device of claim **1**, wherein the HTML-based Web content includes cascading style sheet content defining layout and presentation attributes for the Web page.

30. A mobile phone, comprising:

a processor,

wireless communications means operatively coupled to the processor, to facilitate communication with a mobile service provider network via which Web content may be accessed;

a touch-sensitive display;

a memory, operatively coupled to the processor; and

storage means, operatively coupled to the processor, in which a plurality of instructions are stored that when executed by the processor enable the mobile phone to perform operations including,

rendering a browser interface via which a user is enabled to request to access to a Web page having an original format comprising HTML-based content defining an original page layout, functionality, and design of content on the Web page;

retrieving HTML-based content associated with the Web page;

translating at least a portion of the HTML-based content from its original format to produce translated content including scalable vector-based content that supports a scalable resolution-independent representation of the HTML-based content that preserves an original page layout, functionality and design of the at least a portion of the HTML-based content when scaled and rendered; and

employing the scalable vector-based content to render a view of at least a portion of the Web page on the display using a first scale factor,

wherein preservation of the functionality defined by the HTML-based content includes preservation of hyperlink functionality.

31. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input made via the touch-sensitive display.

32. The mobile phone of claim **31**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

33. The mobile phone of claim **30**, wherein the Web content includes at least one hyperlink, and wherein execution of the instructions performs further operations comprising:

enabling the user to select the hyperlink via the touch-sensitive display; and, in response thereto,

retrieving and translating the Web content associated with the hyperlink to produce additional scalable vector-based content; and

employing the additional scalable vector-based content to render the Web content associated with the hyperlink on the touch-sensitive display.

34. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising:

parsing and processing markup language code associated with the Web page to determine the original page layout of display content within the Web page, wherein the original page layout defines a layout location for a plurality of objects, including text objects, graphic layout objects, and/or graphic image objects included in the Web page;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to the layout location for the object;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to its corresponding vector.

35. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising enabling the Web content to be displayed at different resolutions by scaling the scalable vector-based content to re-render the display in response to associated user inputs.

36. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising returning the display of the Web content to a previous view in response to a corresponding user input made via the touch-sensitive display.

37. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising enabling a user to pan a display of the Web content in response to a corresponding user input made via the touch-sensitive display.

38. The mobile phone of claim **37**, wherein execution of the instructions performs further operations comprising enabling the display of the Web content to be panned in real-time.

39. The mobile phone of claim **30**, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein the scalable vector-based content is scaled when rendered so as to produce a display having a different aspect ratio.

40. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising enabling a user to view a column of the Web content at a higher resolution than a current resolution by tapping on the column via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is displayed to fit across the touch-sensitive display.

41. The mobile phone of claim **30**, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to view an image at a higher resolution than a current resolution by tapping on the image via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that the image is displayed to fit across at least one of a width and height of a display area of the touch-sensitive display.

42. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising enabling a user to view a paragraph of the Web content at a higher resolution than a current resolution by tapping on the paragraph via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is displayed across at least one of a width and height of a display area of the touch-sensitive display.

43. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising:

generating a display list associated with the scalable vector-based content; and

employing the display list to re-render the display at different scale factors to enable rapid zooming of the Web page.

44. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising:

parsing and processing markup language code corresponding to the received Web content to determine page layout information corresponding to a page layout of the content on the Web page;

logically grouping selected content into objects;

defining a primary datum corresponding to the page layout; and,

for each object,

defining an object datum corresponding to a layout location datum for the object's associated display content;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to its corresponding vector.

45. The mobile phone of claim **44**, wherein execution of the instructions performs further operations comprising:

mapping the object vectors to a virtual display area in memory.

46. The mobile phone of claim **45**, wherein execution of the instructions performs further operations comprising:

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and pan corresponding to a rendered view of the Web content desired by a user;

determining a virtual display limit bounding box for the virtual display area associated with the first scale factor and offset;

identifying objects having at least a portion of their content falling within the virtual display limit bounding box; and,

for each of such objects,

retrieving content associated with that object; and

applying an appropriate scale factor and offset to the content to render the view of the Web content.

47. The mobile phone of claim **30**, wherein execution of the instructions performs further operations comprising:

parsing markup language code corresponding to the received Web content to logically group selected content into objects;

generating page layout information including a bounding box for each object, the bounding box defining width and height dimensions for the object; and

storing information that links each object with its corresponding page layout information;

wherein the page layout information further includes information from which a page layout location of each of the bounding boxes can be determined.

48. The mobile phone of claim **30**, wherein the scalable vector-based content includes scalable text content, and wherein execution of the instructions performs further operations comprising scaling a scalable font to render the scalable text content.

49. The mobile phone of claim **30**, wherein the original format of the Web page defines a width for the Web page, and wherein execution of the instructions performs further operations comprising:

determining an applicable scale factor to fit the width of the Web page across a display area of the touch-sensitive display; and

employing the scale factor that is determined as the first scale factor.

50. The mobile phone of claim **30**, wherein at least a portion of the instructions comprise Java-based instructions configured to be executed on a Java virtual machine.

51. The mobile device of claim **30**, wherein the original format of the Web page comprises HTML-based Web content and the vector-based scalable content comprises scalable vector-based page layout information, and wherein execution of the instructions performs further operations comprising:

processing the HTML-based Web content with a rendering engine to generate page layout information corresponding to the original page layout as interpreted by the rendering engine;

employing the page layout information to generate the scalable vector-based page layout information.

52. A mobile device, comprising:

a processor,

wireless communications means, to facilitate wireless communication with a network via which Web content may be accessed;

a touch-sensitive display;

flash memory, operatively coupled to the processor, in which a plurality of instructions are stored that when executed by the processor enable the mobile device to perform operations including,

rendering a browser interface via which a user is enabled to request access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

retrieving and processing the HTML-based Web content to produce scalable content; and

employing the scalable content and/or data derived therefrom to,

render a view of the Web page on the touch-sensitive display; and

re-render the Web page in response to associated user inputs to enable the user to iteratively zoom in and out views of the Web page while preserving an original page layout, functionality, and design defined by the HTML-based Web content as interpreted by a rendering engine,

wherein preservation of the functionality defined by the HTML-based Web content includes preservation of hyperlink functionality.

53. The mobile device of claim **52**, wherein the device comprises a mobile phone.

54. The mobile device of claim **52**, wherein the device comprises one of a Personal Digital Assistant (PDA) or pocket PC.

55. The mobile device of claim **52**, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input made via the touch-sensitive display.

56. The mobile device of claim **55**, wherein the user interface input enables the user to define a window of a current view of the Web page on which to zoom in on.

57. The mobile device of claim **52**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

58. The mobile device of claim **52**, wherein the Web page includes at least one hyperlink, and wherein execution of the instructions performs further operations comprising:

enabling the user to select the hyperlink via the touch-sensitive display; and, in response thereto,

retrieving and processing HMTL-based Web content associated with the hyperlink to produce additional scalable content; and

employing the additional scalable content and/or data derived therefrom to render the Web content associated with the hyperlink on the touch-sensitive display.

59. The mobile device of claim **52**, wherein at least a portion of the scalable content comprises scalable vector-based content.

60. The mobile device of claim **52**, wherein execution of the instructions performs further operations comprising returning the display of the Web page to a previous view in response to a corresponding user input made via the touch-sensitive display.

61. The mobile device of claim **52**, wherein execution of the instructions performs further operations comprising enabling a user to pan a display of the Web content in response to a corresponding user input made via the touch-sensitive display.

62. The mobile device of claim **61**, wherein execution of the instructions performs further operations comprising enabling the display of the Web content to be panned in real-time.

63. The mobile device of claim **52**, wherein the page layout of the Web page is defined to have an original aspect ratio, and wherein the scalable content and/or data derived therefrom is scaled to render a display having a different aspect ratio.

64. The mobile device of claim **52**, wherein execution of the instructions performs further operations comprising enabling a user to view a column of the Web content at a higher resolution than a current resolution by tapping on the column via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that content corresponding to the selected column is displayed across the touch-sensitive display.

65. The mobile device of claim **52**, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to view an image at a higher resolution than a current resolution by tapping on the image via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that the image is displayed to fit a width of a display area of the touch-sensitive display.

66. The mobile device of claim **52**, wherein execution of the instructions performs further operations comprising enabling a user to view a paragraph of the Web content at a higher resolution than a current resolution by tapping on the paragraph via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that content corresponding to the selected paragraph is displayed across at least one of a width and height of a display area of the touch-sensitive display.

67. The mobile device of claim **66**, wherein the content of the paragraph is reformatted to fit characteristics of the display area when the display is re-rendered.

68. The mobile device of claim **52**, wherein the Web page includes text, layout attributes, and images, and wherein execution of the instructions performs further operations comprising:

receiving content corresponding to the text and layout attributes via a first connection; and

receiving content corresponding to at least one image via a second connection.

69. The mobile device of claim **68**, further comprising dynamic memory having at least a portion employed for rendering purposes, wherein execution of the instructions performs further operations comprising:

mapping the object vectors and associated bounding boxes to a virtual display area in the dynamic memory.

70. The mobile device of claim **69**, wherein execution of the instructions performs further operations comprising:

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and pan corresponding to a rendered display of the Web page desired by a user;

determining a virtual display limit bounding box for the virtual display associated with the first scale factor and offset;

identifying object bounding boxes having at least a portion falling within the virtual display limit bounding box; and,

for each of such object bounding boxes,

retrieving content associated with that object bounding box; and

applying an appropriate scale factor to the content to render the display.

71. The mobile device of claim **52**, further comprising dynamic memory having at least a portion employed for rendering purposes, wherein execution of the instructions performs further operations comprising:

building a display list via use of the scalable content and rendering the display list on a virtual display area in the dynamic memory; and

scaling the display list to re-render the display of the Web page.

72. The mobile device of claim **52**, wherein execution of the instructions performs further operations comprising:

parsing HTML-based code corresponding to the received Web content to identify content on the Web page;

logically grouping selected content into objects;

defining a primary datum corresponding to the original page layout; and,

for each object,

defining an object datum corresponding to a layout location datum for the object's associated display content;

generating a vector from the primary datum to the object datum for the object; and

creating a reference that links the object to its corresponding vector.

73. The mobile device of claim **72**, wherein execution of the instructions performs further operations comprising:

generating a bounding box for each object, the bounding box representing a portion of a rendered display page occupied by the object's associated group of content.

74. The mobile device of claim **52**, wherein the scalable content includes scalable text content, and wherein execution of the instructions performs further operations comprising scaling a scalable font to render the scalable text content.

75. The mobile device of claim **52**, wherein the original format of the Web page defines a width for the Web page, and wherein execution of the instructions performs further operations comprising:

determining an applicable scale factor to fit the width of the Web page across a display area of the touch-sensitive display; and

employing the scale factor to render the display area.

76. The mobile device of claim **52**, wherein at least a portion of the instructions comprise Java-based instructions configured to be executed on a Java virtual machine.

77. The mobile device of claim **52**, wherein a portion of the HTML-based Web content comprises XML-based content.

78. The mobile device of claim **52**, wherein a portion of the HTML-based Web content comprises cascading style sheet data.

79. A mobile device, comprising:

processing means;

wireless communications means, to facilitate wireless communication with a network via which Web content may be accessed;

touch-sensitive display means, to facilitate user input and display rendered content;

programmed circuit means; and

storage means, in which a plurality of instructions are stored,

wherein, upon execution of the instructions by at least one of the processing means and programmed circuit means, the mobile device is enabled to perform operations, including,

rendering a browser interface via which a user is enabled to request to access to a Web page comprising HTML-based Web content defining an original page layout, functionality, and design of content on the Web page;

retrieving and processing the HTML-based Web content to produce scalable content; and

employing the scalable content and/or data derived therefrom to,

render a view of the Web page on the touch-sensitive display; and

re-render the Web page in response to associated user inputs made via the touch-sensitive display means to enable the user to iteratively zoom in and out views of the Web page while preserving an original page layout, functionality, and design defined by the HTML-based Web content as interpreted by a rendering engine,

wherein preservation of the functionality defined by the HTML-based Web content includes preservation of hyperlink functionality.

80. The mobile device of claim **79**, wherein the processing means includes a general-purpose processor.

81. The mobile device of claim **79**, wherein at least a portion of the programmed circuit means is embodied as a special-purpose processor.

82. The mobile device of claim **79**, wherein execution of the instructions performs further operations comprising enabling the user to zoom in on a user-selectable portion of a display of the Web page in response to a user interface input made via the touch-sensitive display.

83. The mobile device of claim **82**, wherein the user interface input enables the user to define a window of a current view of the Web page on which to zoom in on.

84. The mobile device of claim **79**, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

85. The mobile device of claim **79**, wherein execution of the instructions performs further operations comprising enabling a user to pan a display of the Web content in response to a corresponding user input made via the touch-sensitive display.

86. The mobile device of claim **85**, wherein execution of the instructions performs further operations comprising enabling the display of the Web content to be panned in real-time.

87. The mobile device of claim **79**, wherein the Web content includes at least one image, and wherein execution of the instructions performs further operations comprising enabling a user to view an image at a higher resolution than a current resolution by tapping on the image via the touch-sensitive display, wherein in response thereto, the display is re-rendered such that the image is displayed to fit across a width of a display area of the touch-sensitive display.

88. The mobile device of claim **79**, further comprising dynamic memory having at least a portion employed for rendering purposes, wherein execution of the instructions performs further operations comprising:

building a display list of scalable content via use of the scalable content and rendering the display list on a virtual display area in the dynamic memory; and

scaling the scalable content in the display list to re-render the display of the Web page.

* * * * *

Exhibit 3

accessed by an unsupported browser.

Overall, the percentage of Web pages that are either browser/device-specific or contain invalid HTML (of significance) is rather small (when considered among the literally billions of estimated Web pages); however, as discussed above, such pages do exist. As a result, one of skill in the art would not expect any browser implementation to be able to render all Web pages as designed and/or coded.

Under the claimed inventions of claims 383-390 and 393, users are enabled to view, zoom, and pan the HTML-based Web page content of substantially any Web page in a manner that preserves the original layout and attributes of the Web page content. This capability will generally be dependent on the rendering engine compatibility with the Web page definition (as defined by the Web page's corresponding HTML). That is, this will generally depend on whether the rendering engine can render the original HTML-based Web page content appropriately at its original resolution or as originally defined by the Web page's HTML.[2]. Under the principles and teachings of the present disclosure, Web pages that can be rendered (by the applicable rendering engine) at the original resolution or as originally defined by the Web page's corresponding HTML are enabled to be viewed, zoomed, and panned in a manner that preserves the original layout and attributes of the Web page (as rendered at the original resolution).

[2] That is the resolution the Web page was designed to be rendered at in accordance with the original HTML-based content defining the original layout and attributes of the Web page content. The original resolution for a predefined width corresponds to the width and height of a Web page (as to be rendered) in pixels. For example, many or today's Web pages are designed for a screen with a width resolution of 1024 pixels (or greater), while earlier Web pages were designed for screen widths of 800 (SVGA) or 640 (VGA) pixels. Meanwhile, some Web pages are coded (via corresponding HTML) to be centered or otherwise fit within the width of a current browser window. In this case, the width of the Web page is not predefined, but rather will be a function of the width of the current browser window. As a result, the rendering engine determines the applicable layout of the Web page in view of the current browser window width (which it obtains from the browser or operating system).

specification. In response to this argument, Examiner Tran stated that he did not examine this patent (US 5,956,025), and what another Examiner did was not material to examination of the present application.

In view of the foregoing, it became clear that the use of "substantially" in a claim in the present application was going to render the claim indefinite due to the broad interpretation of the term by Examiner Tran. Accordingly, the Applicants have elected to remove the word "substantially" from the pending claims. However, it is noted that the intended scope of the corresponding claims (*i.e.,* as originally intended by the Applicants) has not changed due to the removal of the word "substantially," as the Applicants never intended the term to have the breadth accorded by Examiner Tran. Accordingly, no *Festo*[1] estoppel shall apply, as no equivalence has been surrendered, as argued more specifically below.

It is well established that statements in the file history may be used to interpret the scope of the claim elements. See, *e.g., Warner-Jenkinson Co. v. Hilton Davis Chem. Co.*, 117 S. Ct. 1040, 41 USPQ2d 1865 (1997); *Markman v. Westview Instruments*, 52 F.3d 967, 34 USPQ2d 1321 (Fed. Cir. 1995), aff'd 116 S. Ct. 1384, 38 USPQ2d 1461 (1996); *Vitronics Corp. v. Conceptronic Inc.,* 90 F.3d 1576, 39 USPQ2d 1573 (Fed. Cir. 1996). Moreover, file histories of more recently issued patents and pending applications which have been published are available to the public via PAIR. Accordingly, applicants respectfully assert that the scope of the terminology and claim elements discusses below clearly renders each claim element to be definite, as such discussion is publically made available to those skilled in the art, as well as the public in general.

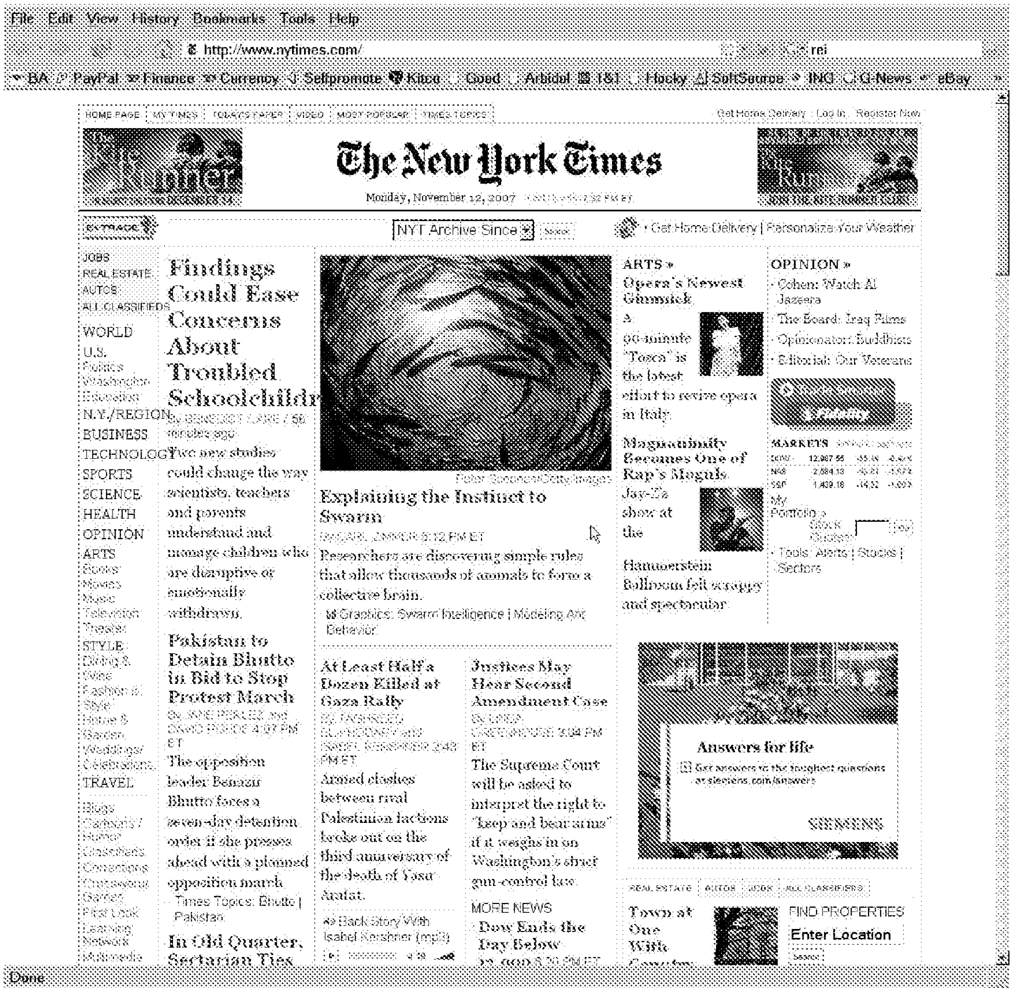
Scope of the terminology "the Web page is rendered to fit across the display"

Each of claims 71, 143, 180, 244, 271, and 337 contain claim elements

[1] *Festo Corp. v. Shoketsu Kinzoku Kogyokabushiki Co.* 535 U.S. 722 (2002) 234 F.3d 558.

including the language "the Web page is rendered to fit across the display," replacing the prior language "... fit substantially across the display." A discussion of the intended scope of this terminology was presented in the December 9, 2007 response to the Office Action of October 23, 2007; an augmented argument (to account for the removal of the word substantially) is presented below.

Figs. 7A, 8A, and 9A show examples of Web pages rendered to fit across the display of the illustrated Palm IIIc touchscreen display. One of skill in the art would recognize that it may be desirable to provide a border of a few pixels or more around the edges of the rendered Web page for readability purposes and/or aesthetics. Additionally, depending on the scrolling scheme employed, a portion of the browser may be used for scroll bars or the like, such as shown in Figs. 7A, 8A, and 9A. Generally, depending on the underlying operating system (and possibly browser features), the width of the scroll bars may vary, no scroll bars may be displayed, or scroll bars may be overlaid over a portion of the browser's page rendering area, enabling the entire width of the display to be used for browser page rendering. Examples of operating systems and/or browser implementations with different scroll bar widths are shown below:



Findings Could Ease Concerns About Troubled Schoolchildr

By BENEDICT CAREY / 56 minutes ago

Two new studies could change the way scientists, teachers and parents understand and manage children who are disruptive or emotionally withdrawn.

Pakistan to Detain Bhutto in Bid to Stop Protest March

By JANE PERLEZ and DAVID ROHDE 4:07 PM ET

The opposition leader Benazir Bhutto faces a seven-day detention order if she presses ahead with a planned opposition march.

Times Topics: Bhutto | Pakistan

In Old Quarter, Sectarian Ties

Peter Scoones/Getty Images

Explaining the Instinct to Swarm

By CARL ZIMMER 5:12 PM ET

Researchers are discovering simple rules that allow thousands of animals to form a collective brain.

Graphics: Swarm Intelligence | Modeling Ant Behavior

At Least Half a Dozen Killed at Gaza Rally

By TAGHREED EL-KHODARY and ISABEL KERSHNER 2:43 PM ET

Armed clashes between rival Palestinian factions broke out on the third anniversary of the death of Yasir Arafat.

Back Story With Isabel Kershner (mp3)

Justices May Hear Second Amendment Case

By LINDA GREENHOUSE 3:04 PM ET

The Supreme Court will be asked to interpret the right to "keep and bear arms" if it weighs in on Washington's strict gun-control law.

MORE NEWS

- Dow Ends the Day Below 13,000 5:30 PM ET

ARTS »

Opera's Newest Gimmick

A 90-minute "Tosca" is the latest effort to revive opera in Italy.

Magnanimity Becomes One of Rap's Moguls

Jay-Z's show at the Hammerstein Ballroom felt scrappy and spectacular.

OPINION »

- Cohen: Watch Al Jazeera
- The Board: Iraq Films
- Opinionator: Buddhists
- Editorial: Our Veterans

MARKETS

DOW	12,987.55	[illegible]	[illegible]
NAS	2,584.13	[illegible]	[illegible]
S&P	1,439.18	[illegible]	[illegible]

My Portfolio »

Stock Quotes [ ] Go

- Tools: Alerts | Stocks | Sectors

Answers for life

Get answers to the toughest questions at siemens.com/answers

SIEMENS

REAL ESTATE | AUTOS | JOBS | ALL CLASSIFIEDS

Town at One With

FIND PROPERTIES

Enter Location

Search

NYT Web page as rendered on a Mozilla Firefox desktop browser running under the Microsoft Windows XP operating system

Sony Clié running Palm OS	Pocket PC running Windows CE OS

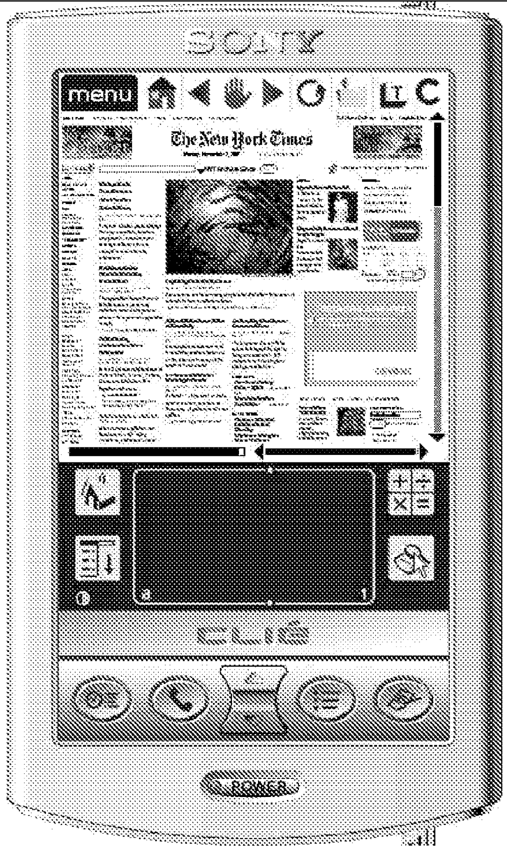
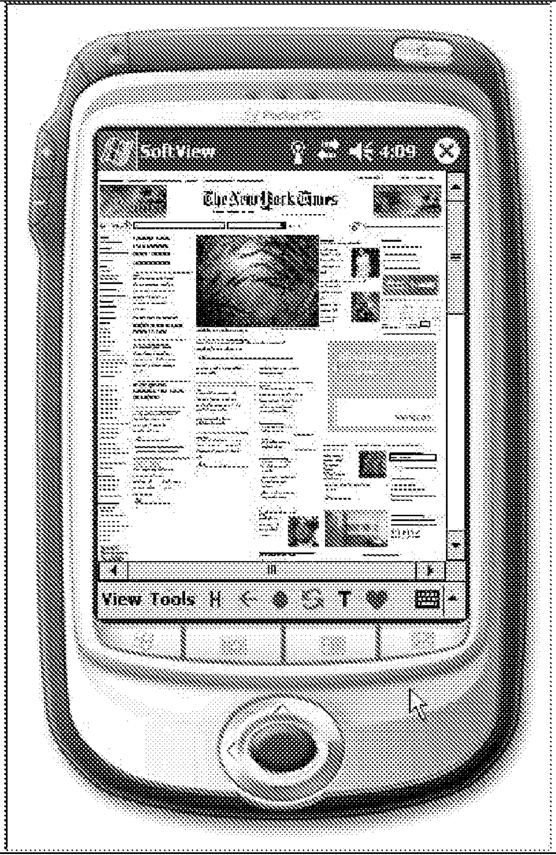

	

Image 1A	Image 1B

Image 1A shows a Web page rendered by a SoftView™ browser on a Sony Clié running a version of Palm OS, while Image 1B shows a Web page rendered by a SoftView™ browser on a Pocket PC running a version of Windows CE OS. (It is noted that the principle developers of the SoftView™ browsers were Gary Rohrabaugh and Scott Sherman, the inventors of the claimed inventions in the present application, and the SoftView™ browsers employ the resolution-independent Web page scaling, zooming, and scrolling (panning) techniques disclosed in the application.)

As illustrated in Image 1A, and similarly illustrated in Figs. 7A, 8A, and 9A of the present application (the Palm IIIc also ran on a version of Palm OS), the SoftView™ browser implementation running on a Palm OS employs a vertical and horizontal scroll bar with arrows at the ends that are wider than the bars themselves. Also, the completely filled horizontal scroll bars in each of Image 1A, Figs. 7A, 8A, and 9A, indicates that horizontal scrolling is not applicable, as the Web page view has been rendered to fit across the width of the browser display area.[2] The scroll bars used by Windows CE are somewhat different – they include separate arrow controls that are the same width as the scroll bars. In a manner similar to the Palm OS examples, the Web page in Image 1B is rendered to fit across the width of the browser display area. Of course, for operating systems/browsers that use overlaid scroll bars, the actual browser display area would be slightly larger. Thus, depending on the type of scroll bar implementation, the portion of the display available to render the Web page (*i.e.,* the browser display area) will vary a small amount. As noted above, border areas may also be desired for readability and/or aesthetics. Accordingly, the scope of the terminology "the Web page is rendered to fit across the display" is intended to cover each of the foregoing scroll bar schemes and/or border areas schemes and combinations thereof.

<u>Scope of the terminology "determining an applicable scale factor to display at least one of the width and height of the Web page to fit across a display area of the display"</u>

Each of claims 123, 238, and 330 recite the language, "determining an applicable scale factor to display at least one of the width and height of the Web page to fit across a browser display area of the display." The scope of this language is intended to cover a Web page being displayed such that at least one of the width and

[2] It is noted that one of ordinary skill in the art would recognize the browser display area is the portion of the rendered display reserved for rendering the Web page content. This typically includes the display area that is not occupied by browser menu items and/or icons, tool bars (as applicable) and scroll bars (as applicable).

height of the Web page occupies the browser display area (which will vary depending on the scroll bar scheme), with the optional use of small borders.

<u>Scope of the terminology "the content corresponding to the selected column is rendered to fit across the display"</u>

Each of claims 111, 190, 315, 365, 368, 371, 374, 377, and 380 recites the language, "the content corresponding to the selected column is rendered to fit across the display." The scope of this language is intended to cover the rendering of a column (in response to a zoom to column user input) such that the column is rendered to span the width of the applicable browser display area, with optional small borders (such as illustrated in FIG. 7B), in a manner similar to that discussed above when rendering a Web page.

<u>Scope of the terminology "the display is re-rendered such that the image is rendered to fit across the display"</u>

Each of claims 113, 136, 193, 318, 366, 369, 372, 375, 378, and 381 recites the language, "the display is re-rendered such that the image is rendered to fit across the display." The scope of this language is intended to cover the rendering of an image (in response to a zoom to image user input) such that the image is rendered to span the width of the applicable browser display area, with optional small borders (such as illustrated in FIG. 8B), in a manner similar to that discussed above when rendering a Web page to fit across the display. This does not imply that the claimed zoom to image operation will cause *all* images to be rendered to fit across the display, as the claim language clearly does not state this. One of skill in the art would recognize that when a selected image has a native resolution (*i.e.,* the 1:1 resolution for the image) that is less than the resolution of the applicable browser display area, it generally would be preferable to render the image at its 1:1 resolution, as rendering the image beyond this resolution (*e.g.,* "blowing" up the image) will generally result in a blurred image. For example, if an image has a native resolution of 150 x 150 pixels and the applicable browser display area is 300 pixels wide, it is preferable to display the image at 1:1 (150 pixels wide) rather than render the image to span the width of the display area. This is illustrated below:

USPTO Seal on http://www.uspto.gov/ at native 1:1 resolution (131 x 131 pixels)

UNITED STATES PATENT AND TRADEMARK OFFICE
DEPARTMENT OF COMMERCE

Same USPTO Seal blown up 200% (262 x 262 pixels)

On the flip side, when the native resolution of an image is greater than or equal to the applicable display area, it is advantageous to render the image to fit the applicable display area, as claimed. An a manner analogous to that described above, "re-rending an image to fit across the display" is intended to cover situations where the image is rendered to span the width of the applicable display area, with the optional use of small borders.

<u>Scope of the terminology "the content corresponding to the selected paragraph is</u>

rendered to fit across the display"

Each of claims 195, 367, 370, 373, 376, 379, and 382 recites the language, "the content corresponding to the selected paragraph is rendered to fit across the display." The scope of this language is intended to cover the rendering of paragraph content (in response to a zoom to paragraph user input) such that the content is rendered to span the width of the applicable browser display area, with optional small borders (such as illustrated in FIG. 9B), in a manner similar to that discussed above when rendering a Web page. Likewise, the terminology, "the content corresponding to the selected paragraph is rendered to fit across a display area of a display" recited in each of claims 114 and 320 is intended to have similar scope.

Scope of the terminology "in real-time"

The term "substantially" in "substantially in real-time" has been removed from each of claims 73, 79, 104, 109, 133, 135, 151, 156, 175, 179, 182, 188, 216, 221, 255, 257, 266, 270, 273, 279, 308, 313, 348, 350, 360, and 364. In general, "in real time" pertains to zooming and/or panning operations (as applicable) in each of these claims. The scope of the terminology "in real-time" is intended to pertain to the concept of real-time as perceived by humans when interacting with software, as opposed to the use of real-time to describe machine operations (*e.g.,* a real-time operating system), as argued in the December 9, 2007 response below (a portion of which is augmented to account for the removal of "substantially."

One of skill in the art would recognize the meaning of the terminology "real time" varies depending on the particular use context. For example, for an embedded real-time operating system or implementation, real-time might mean a timeframe in the millisecond or even microsecond range. In this context, the time context is machine time and real-time means instantaneous. In another use context, such as replying to e-mail, real-time is significantly longer. For example, many people refer to responding to e-mail in "real time" – this means the people respond

Exhibit 4

example, there are several web sites that include mouse "hover-over" menus under which a menu will pop-up or otherwise appear when a user places a mouse or pointer cursor over a corresponding portion of the screen. Since there is no mouse or equivalent on a typically mobile device, the browser's rendering/layout engine may be configured to ignore support of such functionality.

The claim language "preservation of the original page layout, functionality and design" is to be considered in the context of the browser implementation itself. That is, the preservation is relative to the interpretation of the page by the browser implementation itself, as opposed to preservation of the original layout, functionality and design based on some rigid consideration of a "perfect" interpretation of the page. As discussed above, a perfect interpretation does not realistically exist, as the HTML and related design specifications are too imprecise to begin with, and due to legacy considerations may remain so in the future.

Returning to the screen shots, Figs. 2a-d represent comparisons between the nytimes.com home page using Firefox 3.0.5 and the SoftView™ proxy server – browser client implementation. It is noted that the SoftView™ implementation (employed for the screenshots herein) uses a version of the Mozilla rendering engine from approximately 2002. As a result, there may be certain types of page design aspects that are not included in the rendered page, since corresponding aspects of the Cascading Style Sheets (CSS) specifications were not supported by this earlier version of the Mozilla rendering engine.[8] These include support for rendering some of the thin light gray column separator elements, which are defined in the common layout CSS document for the nytimes.com Web site.

[8] See http://en.wikipedia.org/wiki/Cascading_Style_Sheets for an excellent discussion of the Cascading Style Sheets, their history, and in particular the section concerning difficulty with adoption.

```
.auto-style5 {
	font-family: "Lucida Sans", "Lucida Sans Regular", "Lucida Grande", "Lucida Sans Unicode",
Geneva, Verdana, sans-serif;
	font-size: x-large;
	margin-left: 360px;
	margin-top: 30px;
}
```

Notably, a font size of “extra large” corresponding to this font family may be interpreted differently by different browsers. In the case of the Sony Clie examples shown herein, which employs a Palm operating system, a system font may typically be substituted for this font family. Since this web page was designed using Microsoft software, it is believed that the font definition in the HTML above is clearly understood by a Microsoft Internet Explorer browser, but may not be understood by other browsers. Moreover, the font size of “x-large” may be interpreted differently by different rendering engines.

The left and top margins are relative to the page “body” or “body block,” which is the top-level container block for this page’s content. In more complex web pages, there may be multiple container blocks, including nested container blocks. Under HTML, the margin offset information defines a margin offset for the object relative to each object’s container block.

The layout locations for the other objects are determined in a similar manner, with the results depicted in the figure below. The object datums are depicted as X, Y pixel coordinates corresponding to the upper left-hand corner of the bounding box for each object. The selection of the upper left-hand corner is exemplary and not limiting, as the lower left-hand corner may also be used. Moreover, a combination of upper and lower left-hand corners may be employed for object datums, as long as the use for particular object types is consistent. For example, since it is common to layout text content based on a baseline location, using a lower-left hand corner for the datum for a text object bounding box may be employed by some implementations.

schematically shows the page layout and corresponding content being scaled by 1/3, including graphically depicting the point vectors for the object datums being scaled. Since a vector in a two-dimensional coordinate system is defined by a length and direction (i.e., angle), when a vector is scaled the length is multiplied by the applicable scale factor while the direction remains the same. This result is achieved by multiplying each of the X and Y floating point coordinate values by the same scale factor (in this case simply dividing each floating point coordinate by 3). For example, when the original datum for the 300x300 Seattle skyline image is scaled by 1/3, its X, Y coordinates are transformed from 360.0f, 108,0f to 120.0f, 36.0f, as shown below. Also, since the coordinates corresponding to the corners of the bounding boxes are scaled, the width and height dimensions of the bounding boxes are likewise scaled by the same amount. For example, the size of the bounding box for the 300x300 Seattle skyline image is 100.0f by 100.0f after being scaled by 1/3.

Exhibit 5

Merriam-Webster's Collegiate® Dictionary

TENTH EDITION

Merriam-Webster, Incorporated
Springfield, Massachusetts, U.S.A.

Library of Congress Cataloging in Publication Data
Main entry under title:

Merriam-Webster's collegiate dictionary. — 10th ed.
p. cm.
Includes index.
ISBN 0-87779-708-0 (unindexed : alk. paper). — ISBN 0-87779-709-9 (indexed : alk. paper). — ISBN 0-87779-710-2 (deluxe indexed : alk. paper). — ISBN 0-87779-707-2 (laminated cover, unindexed).
1. English language—Dictionaries. I. Merriam-Webster, Inc.
PE1628.M36 1998
423—dc21 97-41846
CIP

Made in the United States of America

30313233RT:WC01

ab·la·tion \a-ˈblā-shən\ *n* (15c) : the process of ablating: as **a** : surgical removal **b** : loss of a part (as ice from a glacier or the outside of a nose cone) by melting or vaporization

^{1}**ab·la·tive** \ˈa-blə-tiv\ *adj* (15c) : of, relating to, or constituting a grammatical case expressing typically the relations of separation and source and also frequently such relations as cause or instrument — **ablative** *n*

^{2}**ab·la·tive** \a-ˈblā-tiv\ *adj* (ca. 1569) **1** : of or relating to ablation **2** : tending to ablate ⟨~ material on a nose cone⟩ — **ab·la·tive·ly** *adv*

ablative absolute \ˈa-blə-tiv-\ *n* (ca. 1828) : a construction in Latin in which a noun or pronoun and its adjunct both in the ablative case form together an adverbial phrase expressing generally the time, cause, or an attendant circumstance of an action

ab·laut \ˈä-blau̇t, ˈa-; ˈäp-ˌlau̇t\ *n* [G, fr. *ab* away from + *Laut* sound] (1849) : a systematic variation of vowels in the same root or affix or in related roots or affixes esp. in the Indo-European languages that is usu. paralleled by differences in use or meaning (as in *sing, sang, sung, song*)

ablaze \ə-ˈblāz\ *adj or adv* (1801) **1** : being on fire **2** : radiant with light or emotion ⟨his face all ~ with excitement —Bram Stoker⟩

able \ˈā-bəl\ *adj* **abler** \-b(ə-)lər\; **ablest** \-b(ə-)ləst\ [ME, fr. MF, fr. L *habilis* apt, fr. *habēre* to have — more at HABIT] (14c) **1 a** : having sufficient power, skill, or resources to accomplish an object **b** : susceptible to action or treatment **2** : marked by intelligence, knowledge, skill, or competence — **ably** \ˈā-b(ə-)lē\ *adv*

-able *also* **-ible** *adj suffix* [ME, fr. OF, fr. L *-abilis, -ibilis,* fr. *-a-, -i-,* verb stem vowels + *-bilis* capable or worthy of] **1** : capable of, fit for, or worthy of (being so acted upon or toward) — chiefly in adjectives derived from verbs ⟨break*able*⟩ ⟨collect*ible*⟩ **2** : tending, given, or liable to ⟨agree*able*⟩ ⟨perish*able*⟩ — **-ably** *also* **-ibly** *adv suffix*

able–bod·ied \ˌā-bəl-ˈbä-dēd\ *adj* (ca. 1622) : having a sound strong body

able–bodied seaman *n* (ca. 1909) : ABLE SEAMAN

able seaman *n* (1702) : an experienced deck-department seaman qualified to perform routine duties at sea

abloom \ə-ˈblüm\ *adj* (1855) : abounding with blooms : BLOOMING ⟨parks ~ with roses⟩

ab·lut·ed \ə-ˈblü-təd, a-\ *adj* [back-formation fr. *ablution*] (1650) : washed clean

ab·lu·tion \ə-ˈblü-shən, a-\ *n* [ME, fr. MF or LL; MF, fr. LL *ablution-, ablutio,* fr. L *abluere* to wash away, fr. *ab-* + *lavere* to wash — more at LYE] (1533) **1 a** : the washing of one's body or part of it (as in a religious rite) **b** *pl* : the act or action of bathing **2** *pl, Brit* : a building housing bathing and toilet facilities on a military base — **ab·lu·tion·ary** \-shə-ˌner-ē\ *adj*

ABM \ˌā-(ˌ)bē-ˈem\ *n, pl* **ABM's** *or* **ABMs** \-ˈemz\ (1963) : ANTIBALLISTIC MISSILE

Ab·na·ki \ab-ˈnä-kē\ *var of* ABENAKI

ab·ne·gate \ˈab-ni-ˌgāt\ *vt* **-gat·ed; -gat·ing** [back-formation fr. *abnegation*] (1623) **1** : DENY, RENOUNCE ⟨*abnegated* their God⟩ **2** : SURRENDER, RELINQUISH ⟨*abnegated* her powers⟩ — **ab·ne·ga·tor** \-ˌgā-tər\ *n*

ab·ne·ga·tion \ˌab-ni-ˈgā-shən\ *n* [LL *abnegation-, abnegatio,* fr. L *abnegare* to refute, fr. *ab-* + *negare* to deny — more at NEGATE] (14c) : DENIAL; *esp* : SELF-DENIAL

^{1}**ab·nor·mal** \(ˌ)ab-ˈnȯr-məl, əb-\ *adj* [alter. of F *anormal,* fr. ML *anormalis,* fr. L *a-* + LL *normalis* normal] (ca. 1836) : deviating from the normal or average : UNUSUAL, EXCEPTIONAL ⟨~ behavior⟩ — **ab·nor·mal·ly** \-mə-lē\ *adv*

^{2}**abnormal** *n* (1912) : an abnormal person

ab·nor·mal·i·ty \ˌab-nər-ˈma-lə-tē, -(ˌ)nȯr-\ *n, pl* **-ties** (1854) **1** : the quality or state of being abnormal **2** : something abnormal

abnormal psychology *n* (ca. 1903) : a branch of psychology concerned with mental and emotional disorders (as neuroses, psychoses, and mental deficiency) and with certain incompletely understood normal phenomena (as dreams and hypnosis)

abo \ˈa-(ˌ)bō\ *n, pl* **ab·os** (1908) *Austral, often disparaging* : ABORIGINE

^{1}**aboard** \ə-ˈbōrd, -ˈbȯrd\ *adv or adj* (14c) **1** : ALONGSIDE **2 a** : on, onto, or within a vehicle (as a car or ship) **b** : in or into a group, association, or organization ⟨her second promotion since coming ~⟩ **3** *baseball* : on base

^{2}**aboard** *prep* (15c) : ON, ONTO, WITHIN ⟨go ~ ship⟩ ⟨~ a plane⟩

ABO blood group \ˌā-(ˌ)bē-ˈō-\ *n* (1949) : any of the four blood groups A, B, AB, and O comprising the ABO system

abode \ə-ˈbōd\ *n* [ME *abod,* fr. *abiden* to abide] (13c) **1** *obs* : WAIT, DELAY **2** : a temporary stay : SOJOURN **3** : the place where one abides : HOME

aboil \ə-ˈbȯi(ə)l\ *adj or adv* (1810) **1** : being at the boiling point : BOILING **2** : intensely excited or stirred up ⟨the meeting was ~ with controversy⟩

abol·ish \ə-ˈbä-lish\ *vt* [ME *abolisshen,* fr. MF *aboliss-,* stem of *abolir,* fr. L *abolēre;* prob. akin to *adolescere* to grow up — more at ADULT] (15c) **1** : to end the observance or effect of : ANNUL **2** : DESTROY — **abol·ish·able** \-li-shə-bəl\ *adj* — **abol·ish·er** *n* — **abol·ish·ment** \-mənt\ *n*

ab·o·li·tion \ˌa-bə-ˈli-shᵊn\ *n* [MF, fr. L *abolition-, abolitio,* fr. *abolēre*] (1529) **1** : the act of abolishing : the state of being abolished **2** : the abolishing of slavery — **ab·o·li·tion·ary** \-ˈli-shᵊn-ˌer-ē\ *adj*

ab·o·li·tion·ism \-ˈli-shᵊn-ˌi-zəm\ *n* (1808) : principles or measures fostering abolition esp. of slavery — **ab·o·li·tion·ist** \-ist\ *n or adj*

ab·oma·sum \ˌa-bō-ˈmā-səm\ *n, pl* **-sa** \-sə\ [NL, fr. L *ab-* + *omasum* ox's tripe] (ca. 1706) : the fourth compartment of the ruminant stomach that follows the omasum and has a true digestive function — compare RUMEN, RETICULUM — **ab·oma·sal** \-səl\ *adj*

A–bomb \ˈā-ˌbäm\ *n* (1945) : ATOMIC BOMB 1

abom·i·na·ble \ə-ˈbäm-nə-bəl, -ˈbä-mə-\ *adj* (14c) **1** : worthy of or causing disgust or hatred : DETESTABLE ⟨the ~ treatment of the poor⟩ **2** : quite disagreeable or unpleasant ⟨~ weather⟩ — **abom·i·na·bly** \-blē\ *adv*

abominable snow·man \-ˈsnō-mən, -ˌman\ *n, often cap A&S* (1921) : a mysterious creature with human or apelike characteristics reported to exist in the high Himalayas — called also *yeti*

abom·i·nate \ə-ˈbä-mə-ˌnāt\ *vt* **-nat·ed; -nat·ing** [L *abominatus,* pp. of *abominari,* lit., to deprecate as an ill omen, fr. *ab-* + *omin-, omen* omen] (1644) : to hate or loathe intensely : ABHOR **syn** see HATE — **abom·i·na·tor** \-ˌnā-tər\ *n*

abom·i·na·tion \ə-ˌbä-mə-ˈnā-shən\ *n* (14c) **1** : something abominable **2** : extreme disgust and hatred : LOATHING

ab·oral \(ˌ)a-ˈbōr-əl, -ˈbȯr-\ *adj* (1857) : situated opposite to or away from the mouth ⟨a sea urchin's ~ surface⟩ — **ab·oral·ly** \-ə-lē\ *adv*

^{1}**ab·orig·i·nal** \ˌa-bə-ˈrij-nəl, -ˈri-jə-nᵊl\ *adj* (1667) **1** : being the first or earliest known of its kind present in a region ⟨~ forests⟩ ⟨~ rocks⟩ **2 a** : of or relating to aborigines **b** *often cap* : of or relating to the indigenous peoples of Australia **syn** see NATIVE — **ab·orig·i·nal·ly** *adv*

^{2}**aboriginal** *n* (1767) **1** : ABORIGINE 1 **2** *often cap* : ABORIGINE 2

ab·orig·i·ne \ˌa-bə-ˈrij-(ˌ)nē, -ˈri-jə-\ *n* [L *aborigines,* pl., fr. *ab origine* from the beginning] (1533) **1** : an aboriginal inhabitant esp. as contrasted with an invading or colonizing people **2** *often cap* : a member of any of the indigenous peoples of Australia

^{1}**aborn·ing** \ə-ˈbȯr-niŋ\ *adv* [^{1}*a-* + E dial. *borning* (birth)] (1916) : while being born or produced ⟨a resolution that died ~⟩

^{2}**aborning** *adj* (1943) : being born or produced ⟨the ~ fiasco⟩

^{1}**abort** \ə-ˈbȯrt\ *vb* [L *abortus,* pp. of *aboriri* to miscarry, fr. *ab-* + *oriri* to rise, be born — more at ORIENT] *vi* (1580) **1** : to bring forth premature or stillborn offspring **2** : to become checked in development so as to degenerate or remain rudimentary ~ *vt* **1 a** : to induce the abortion of or give birth to prematurely **b** : to terminate the pregnancy of before term **2 a** : to terminate prematurely : CANCEL ⟨~ a project⟩ ⟨~ a spaceflight⟩ **b** : to stop in the early stages ⟨~ a disease⟩ — **abort·er** *n*

^{2}**abort** *n* (1944) : the premature termination of a flight (as of an aircraft or spacecraft), a mission, or an action or procedure relating to a flight ⟨a launch ~⟩

abor·ti·fa·cient \ə-ˌbȯr-tə-ˈfā-shənt\ *n* (1873) : an agent (as a drug) that induces abortion — **abortifacient** *adj*

abor·tion \ə-ˈbȯr-shən\ *n* (1547) **1** : the termination of a pregnancy after, accompanied by, resulting in, or closely followed by the death of the embryo or fetus: as **a** : spontaneous expulsion of a human fetus during the first 12 weeks of gestation — compare MISCARRIAGE **b** : induced expulsion of a human fetus **c** : expulsion of a fetus by a domestic animal often due to infection at any time before completion of pregnancy — compare CONTAGIOUS ABORTION **2** : MONSTROSITY **3** : arrest of development (as of a part or process) resulting in imperfection; *also* : a result of such arrest

abor·tion·ist \-sh(ə-)nist\ *n* (1871) : one who induces abortions

abor·tive \ə-ˈbȯr-tiv\ *adj* (14c) **1** *obs* : prematurely born **2** : FRUITLESS, UNSUCCESSFUL **3** : imperfectly formed or developed **4** : tending to cut short — **abor·tive·ly** *adv* — **abor·tive·ness** *n*

ABO system \ˌā-(ˌ)bē-ˈō-\ *n* (1944) : the basic system of antigens of human blood behaving in heredity as an allelic unit to produce any of the ABO blood groups

abound \ə-ˈbau̇nd\ *vi* [ME, fr. MF *abonder,* fr. L *abundare,* fr. *ab-* + *unda* wave — more at WATER] (14c) **1** : to be present in large numbers or in great quantity : be prevalent **2** : to be copiously supplied ⟨life ~*ed* in mysteries —Norman Mailer⟩ ⟨institutions ~ with evidence of his success —*Johns Hopkins Mag.*⟩

^{1}**about** \ə-ˈbau̇t\ *adv* [ME, fr. OE *abūtan,* fr. ^{1}*a-* + *būtan* outside — more at BUT] (bef. 12c) **1 a** : reasonably close to ⟨~ a year ago⟩ **b** : ALMOST ⟨~ starved⟩ **c** : on the verge of — usu. used with *be* and a following infinitive ⟨is ~ to join the army⟩; used with a negative to express intention or determination ⟨not ~ to quit⟩ **2** : on all sides : AROUND **3 a** : in rotation **b** : around the outside **4** : HERE AND THERE **5** : in the vicinity : NEAR **6** : in succession : ALTERNATELY ⟨turn ~ is fair play⟩ **7** : in the opposite direction ⟨face ~⟩ ⟨the other way ~⟩

^{2}**about** *prep* (bef. 12c) **1** : in a circle around : on every side of : AROUND **2 a** : in the immediate neighborhood of : NEAR **b** : on or near the person of **c** : in the makeup of ⟨a mature wisdom ~ him⟩ **d** : at the command of ⟨has his wits ~ him⟩ **3** : engaged in ⟨act as if they know what they're ~ —T. S. Matthews⟩ **4 a** : with regard to : CONCERNING **b** : concerned with **c** : fundamentally concerned with or directed toward ⟨poker is ~ money —David Mamet⟩ **5** : over or in different parts of

^{3}**about** *adj* (1815) **1** : moving from place to place; *specif* : being out of bed **2** : AROUND 2

about–face \ə-ˈbau̇t-ˈfās\ *n* [fr. the imper. phrase *about face*] (1861) **1** : a 180° turn to the right from the position of attention **2** : a reversal of direction **3** : a reversal of attitude, behavior, or point of view — **about–face** *vi*

about–turn \-ˈtərn\ *n* (1893) *Brit* : ABOUT-FACE

^{1}**above** \ə-ˈbəv\ *adv* [ME, fr. OE *abufan,* fr. *a-* + *bufan* above, fr. *be-* + *ufan* above; akin to OE *ofer* over] (bef. 12c) **1 a** : in the sky : OVERHEAD **b** : in or to heaven **2 a** : in or to a higher place **b** : higher on the same page or on a preceding page **c** : UPSTAIRS **d** : above zero ⟨10 degrees ~⟩ **3** : in or to a higher rank or number ⟨30 and ~⟩ **4** *archaic* : in addition : BESIDES **5** : UPSTAGE

^{2}**above** *prep* (bef. 12c) **1** : in or to a higher place than : OVER **2 a** : superior to (as in rank, quality, or degree) **b** : out of reach of **c** : in preference to **d** : too proud or honorable to stoop to **3** : exceeding in number, quantity, or size : more than **4** : as distinct from and in addition to ⟨heard the whistle ~ the roar of the crowd⟩

^{3}**above** *n, pl* **above** (13c) **1 a** : something that is above **b** : a person whose name is written above **2 a** : a higher authority **b** : HEAVEN

usage Although still objected to by some, the use of *above* as a noun in sense 1a ⟨none of the *above*⟩ ⟨the *above* is Theseus's opinion —William Blake⟩ and as an adjective ⟨without the *above* reserve —O. W. Holmes †1935⟩ ⟨I was brought up on the *above* words —Viscount Montgomery⟩ has been long established as standard.

^{4}**above** *adj* (1604) : written or discussed higher on the same page or on a preceding page ***usage*** see ^{3}ABOVE

above all *adv* (14c) : before every other consideration : ESPECIALLY

^{1}**above·board** \ə-ˈbəv-ˌbōrd, -ˌbȯrd\ *adv* [fr. the difficulty of cheating at cards when the hands are above the table] (1594) : in a straightforward manner : OPENLY

Vau•dois \vō-ˈdwä, ˈvō-ˌ\ *n pl* [MF, fr. ML *Valdenses*] (1560) : WALDENSES

¹**vault** \ˈvȯlt\ *n* [ME *voute*, fr. MF, fr. (assumed) VL *volvita* turn, vault, fr. fem. of *volvitus*, alter. of L *volutus*, pp. of *volvere* to roll — more at VOLUBLE] (14c) **1 a** : an arched structure of masonry usu. forming a ceiling or roof **b** : something (as the sky) resembling a vault **c** : an arched or dome-shaped anatomical structure **2 a** : a space covered by an arched structure; *esp* : an underground passage or room **b** : an underground storage compartment **c** : a room or compartment for the safekeeping of valuables **3 a** : a burial chamber **b** : a prefabricated container usu. of metal or concrete into which a casket is placed at burial — **vaulty** \ˈvȯl-tē\ *adj*

²**vault** *vt* (14c) : to form or cover with or as if with a vault : ARCH

³**vault** *vb* [MF *volter*, fr. OIt *voltare*, fr. (assumed) VL *volvitare* to turn, leap, freq. of L *volvere*] *vi* (1538) **1** : to bound vigorously; *esp* : to execute a leap using the hands or a pole **2** : to do or achieve something as if by a leap ~ *vt* : to leap over; *esp* : to leap over by or as if by aid of the hands or a pole

⁴**vault** *n* (1576) : an act of vaulting : LEAP

vault•ed \ˈvȯl-təd\ *adj* (1533) **1** : built in the form of a vault : ARCHED **2** : covered with a vault

vault•er \-tər\ *n* (ca. 1552) : one that vaults; *esp* : an athlete who competes in the pole vault

¹**vault•ing** \-tiŋ\ *n* (1512) : vaulted construction

²**vaulting** *adj* (1593) **1** : reaching or stretching for the heights ⟨~ ambition⟩ ⟨a ~ imagination⟩ **2** : designed for use in vaulting or in gymnastic exercises ⟨a ~ block⟩ — **vault•ing•ly** \-tiŋ-lē\ *adv*

vaulting horse *n* (ca. 1875) **1** : a gymnastics apparatus used in vaulting that consists of a padded rectangular or cylindrical form supported in a horizontal position above the floor **2** : an event in which vaults are made over a vaulting horse

¹**vaunt** \ˈvȯnt, ˈvänt\ *vb* [ME, fr. MF *vanter*, fr. LL *vanitare*, freq. of (assumed) L *vanare*, fr. L *vanus* vain] *vi* (15c) : to make a vain display of one's own worth or attainments : BRAG ~ *vt* : to call attention to pridefully and often boastfully ⟨people who ~ their ingenuity⟩ ***syn*** see BOAST — **vaunt•er** *n* — **vaunt•ing•ly** \ˈvȯn-tiŋ-lē, ˈvän-\ *adv*

²**vaunt** *n* (15c) **1** : a vainglorious display of what one is or has or has done **2** : a bragging assertive statement

vaunt–cou•ri•er \vȯnt-ˈku̇r-ē-ər, vänt-, -ˈkər-ē-, -ˈkə-rē-\ *n* [MF *avant-courrier*, lit., advance courier] (1560) *archaic* : FORERUNNER

vaunt•ed \ˈvȯn-təd, ˈvän-\ *adj* (1579) : highly or widely praised or boasted about ⟨his own much ~ ferocity —Calvin Tomkins⟩

vaunt•ful \ˈvȯnt-fəl, ˈvänt-\ *adj* (1590) : VAINGLORIOUS, BOASTFUL

vaunty \ˈvȯn-tē, ˈvän-\ *adj* (1724) *Scot* : PROUD, BOASTFUL, VAIN

vav *var of* WAW

vav•a•sor *or* **vav•a•sour** \ˈva-və-ˌsȯr, -ˌsōr, -ˌsu̇r\ *n* [ME *vavasour*, fr. MF *vavassor*, prob. fr. ML *vassus vassorum* vassal of vassals] (14c) : a feudal tenant ranking directly below a baron

va•ward \ˈvau̇-ˌȯrd, -ˌwȯrd\ *n* [ME *vauntwarde*, *vaward*, fr. ONF *avantwarde*, fr. *avant* before (fr. LL *abante*) + *warde* guard, fr. *warder* to guard — more at ADVANCE, REWARD] (1597) *archaic* : the foremost part : FOREFRONT ⟨the ~ of our youth —Shak.⟩

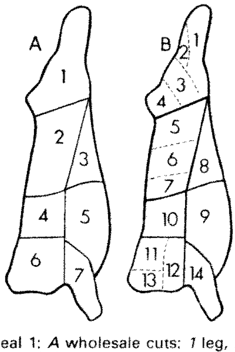
VCR \ˌvē-(ˌ)sē-ˈär\ *n* [*v*ideo*c*assette *r*ecorder] (1971) : a videotape recorder that uses videocassettes

V–day \ˈvē-ˌdā\ *n* [*v*ictory *day*] (1941) : a day of victory

've \v, əv\ *vb* [by contr.] (ca. 1613) : HAVE ⟨we*'ve* been there⟩

Ve•adar \ˈvā-ˌä-ˌdär, ˈvā-ə-\ *n* [Heb *wĕ-Adhār*, lit., and Adar (i.e., the second Adar)] (ca. 1864) : the intercalary month of the Jewish calendar following Adar in leap years — see MONTH table

¹**veal** \ˈvē(ə)l\ *n* [ME *veel*, fr. MF, fr. L *vitellus* small calf, dim. of *vitulus* calf — more at WETHER] (14c) **1** : the flesh of a young calf **2** : CALF; *esp* : VEALER



veal 1: *A* wholesale cuts: *1* leg, *2* loin, *3* flank, *4* rib, *5* breast, *6* shoulder, *7* shank; *B* retail cuts: *1* hind shank, *2* heel of round, *3* round, *4* rump roast, *5* sirloin steak, *6* loin chops, *7* kidney chops, *8* flank, *9* breast, *10* rib roast, *11* blade steak, *12* arm steak, *13* shoulder roast, *14* foreshank

²**veal** *vt* (1847) : to kill and dress (a calf) for veal

veal•er \ˈvē-lər\ *n* (ca. 1895) : a calf grown for or suitable for veal

vealy \ˈvē-lē\ *adj* (1769) **1** : resembling or suggesting veal or a calf **2** : IMMATURE

¹**vec•tor** \ˈvek-tər\ *n* [NL, fr. L, carrier, fr. *vehere* to carry — more at WAY] (1846) **1 a** : a quantity that has magnitude and direction and that is commonly represented by a directed line segment whose length represents the magnitude and whose orientation in space represents the direction; *broadly* : an element of a vector space **b** : a course or compass direction esp. of an airplane **2 a** : an organism (as an insect) that transmits a pathogen **b** : POLLINATOR a **3** : a sequence of genetic material (as a transposon or the genome of a bacteriophage) used to introduce specific genes into the genome of an organism — **vector** *adj* — **vec•to•ri•al** \vek-ˈtōr-ē-əl, -ˈtȯr-\ *adj* — **vec•to•ri•al•ly** \-ə-lē\ *adv*

²**vector** *vt* **vec•tored; vec•tor•ing** \-t(ə-)riŋ\ (1941) **1** : to guide (as an airplane, its pilot, or a missile) in flight by means of a radioed vector **2** : to change the direction of (the thrust of a jet engine) for steering

vector product *n* (1878) : a vector *c* whose length is the product of the lengths of two vectors *a* and *b* and the sine of their included angle, whose direction is perpendicular to their plane, and whose direction is that in which a right-handed screw with axis *c* will move along *c* when *a* is rotated into *b* — called also *cross product*

vector space *n* (1937) : a set representing a generalization of a system of vectors and consisting of elements which comprise a commutative group under addition, each of which is left unchanged under multiplication by the multiplicative identity of a field, and for which multiplication by the multiplicative operation of the field is commutative, closed, distributive such that both $c(A + B) = cA + cB$ and $(c + d)A = cA + dA$, and associative such that $(cd)A = c(dA)$ where A, B are vectors and c, d are elements of the field

vector sum *n* (ca. 1890) : the sum of a number of vectors that for the sum of two vectors is geometrically represented by the diagonal of a parallelogram whose sides represent the two vectors being added

Ve•da \ˈvā-də\ *n* [Skt, lit., knowledge; akin to Gk *eidenai* to know — more at WIT] (1734) : any of four canonical collections of hymns, prayers, and liturgical formulas that comprise the earliest Hindu sacred writings

ve•da•lia \vi-ˈdāl-yə\ *n* [NL, genus name] (1889) : an Australian ladybug (*Rodolia cardinalis*) introduced to many countries to control scale insects

Ve•dan•ta \vā-ˈdän-tə, və-, -ˈdan-\ *n* [Skt *Vedānta*, lit., end of the Veda, fr. *Veda* + *anta* end; akin to OE *ende* end] (1788) : an orthodox system of Hindu philosophy developing esp. in a qualified monism the speculations of the Upanishads on ultimate reality and the liberation of the soul — **Ve•dan•tism** \-ˈdän-ˌti-zəm, -ˈdan-\ *n* — **Ve•dan•tist** \-ˈdän-tist, -ˈdan-\ *n*

Ve•dan•tic \-ˈdän-tik, -ˈdan-\ *adj* (1882) **1** : of or relating to the Vedanta philosophy **2** : VEDIC

Ved•da *or* **Ved•dah** \ˈve-də\ *n* [Sinhalese *vedda* hunter] (1681) : a member of an aboriginal people of Sri Lanka

Ved•doid \ˈve-ˌdȯid\ *n* (1928) : a member of an ancient race of southern Asia characterized by wavy to curly hair, chocolate-brown skin color, and slender body build — **Veddoid** *adj*

ve•dette \vi-ˈdet\ *n* [F, fr. It *vedetta*, alter. of *veletta*, prob. fr. Sp *vela* watch, fr. *velar* to keep watch, fr. L *vigilare* to wake, watch, fr. *vigil* awake — more at VIGIL] (ca. 1702) : a mounted sentinel stationed in advance of pickets

Ve•dic \ˈvā-dik\ *adj* (1848) : of or relating to the Vedas, the language in which they are written, or Hindu history and culture between 1500 B.C. and 500 B.C.

vee \ˈvē\ *n* (ca. 1883) **1** : something shaped like the letter V **2** : the letter *v*

vee•jay \ˈvē-ˌjā\ *n* [*v*ideo *j*ockey] (1981) : an announcer of a program (as on television) that features music videos

vee•na *var of* VINA

veep \ˈvēp\ *n* [fr. *v. p.* (abbr. for *vice president*)] (1949) : VICE PRESIDENT

¹**veer** \ˈvir\ *vt* [ME *veren*, of LG or D origin; akin to MD *vieren* to slacken, MLG *vīren*] (15c) : to let out (as a rope)

²**veer** *vb* [MF *virer*, fr. OF, to throw with a twisting motion, perh. modif. of L *vibrare* to wave, propel suddenly — more at VIBRATE] *vi* (1582) **1** *of the wind* : to shift in a clockwise direction — compare BACK **2** : to change direction or course **3** : to wear ship ~ *vt* : to direct to a different course; *specif* : WEAR 7 ***syn*** see SWERVE — **veer•ing•ly** \-iŋ-lē\ *adv*

³**veer** *n* (1611) : a change in course or direction ⟨a ~ to the right⟩

vee•ry \ˈvir-ē\ *n, pl* **veeries** [prob. imit.] (1838) : a thrush (*Catharus fuscescens*) common in the eastern U.S.

veg \ˈvej\ *n, pl* **veg** (1918) *chiefly Brit* : VEGETABLE

Ve•ga \ˈvē-gə, ˈvā-\ *n* [NL, fr. Ar (*al-Nasr*) *al-Wāqiʿ*, lit., the falling (vulture)] : a star of the first magnitude that is the brightest in the constellation Lyra

veg•an \ˈvē-gən *also* ˈvā- *also* ˈve-jən *or* -ˌjan\ *n* [by contr. fr. *vegetarian*] (1944) : a strict vegetarian who consumes no animal food or dairy products; *also* : one who abstains from using animal products (as leather) — **veg•an•ism** \ˈvē-gə-ˌni-zəm, ˈvā-gə-, ˈve-jə-\ *n*

¹**veg•e•ta•ble** \ˈvej-tə-bəl, ˈve-jə-\ *adj* [ME, fr. ML *vegetabilis* vegetative, fr. *vegetare* to grow, fr. L, to animate, fr. *vegetus* lively, fr. *vegēre* to enliven — more at WAKE] (15c) **1 a** : of, relating to, constituting, or growing like plants **b** : consisting of plants : VEGETATIONAL **2** : made or obtained from plants or plant products **3** : resembling or suggesting a plant (as in inertness or passivity)

²**vegetable** *n* (1582) **1** : PLANT 1b **2** : a usu. herbaceous plant (as the cabbage, bean, or potato) grown for an edible part that is usu. eaten as part of a meal; *also* : such edible part **3** : a human being whose mental and physical functioning is severely impaired

vegetable ivory *n* (1842) **1** : the hard white opaque endosperm of the ivory nut that takes a high polish and is used as a substitute for ivory **2** : IVORY NUT

vegetable marrow *n* (1816) : any of various smooth-skinned elongated summer squashes with creamy white to deep green skins

vegetable oil *n* (1797) : an oil of plant origin; *esp* : a fatty oil from seeds or fruits

vegetable oyster *n* (ca. 1818) : SALSIFY

vegetable pear *n* (1887) : CHAYOTE

vegetable wax *n* (1815) : a wax of plant origin secreted commonly in thin flakes by the walls of epidermal cells

veg•e•ta•bly \ˈvej-tə-blē, ˈve-jə-\ *adv* (1651) : in the manner of or like a vegetable

veg•e•tal \ˈve-jə-tᵊl\ *adj* [ML *vegetare* to grow] (15c) **1** : VEGETABLE **2** : VEGETATIVE **3** : of or relating to the vegetal pole of an egg or to that part of an egg from which the endoderm normally develops ⟨~ blastomeres⟩

vegetal pole *n* (1896) : the point on the surface of an egg that is diametrically opposite to the animal pole and usu. marks the center of the protoplasm containing more yolk, dividing more slowly and into larger blastomeres than that about the animal pole, and giving rise to the hypoblast of the embryo

¹**veg•e•tar•i•an** \ˌve-jə-ˈter-ē-ən\ *n* [²*vege*table + *-arian*] (1839) **1** : one who believes in or practices vegetarianism **2** : HERBIVORE

²**vegetarian** *adj* (1849) **1** : of or relating to vegetarians **2** : consisting wholly of vegetables, fruits, and sometimes eggs or dairy products ⟨a ~ diet⟩

veg•e•tar•i•an•ism \-ē-ə-ˌni-zəm\ *n* (ca. 1851) : the theory or practice of living on a diet made up of vegetables, fruits, grains, nuts, and sometimes eggs or dairy products

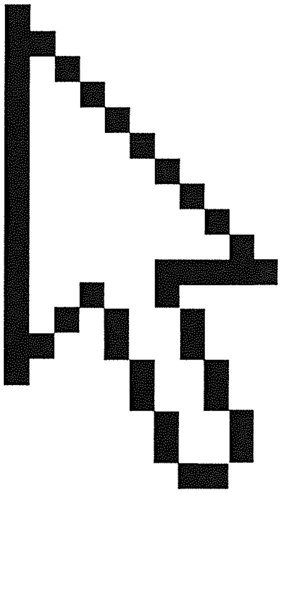
veg•e•tate \ˈve-jə-ˌtāt\ *vb* **-tat•ed; -tat•ing** [ML *vegetatus*, pp. of *vegetare* to grow] *vi* (1605) **1 a** : to grow in the manner of a plant; *also* : to grow exuberantly or with proliferation of fleshy or warty outgrowths **b** : to produce vegetation **2** : to lead a passive existence without exertion of body or mind ~ *vt* : to establish vegetation in or on

Exhibit 6

Microsoft

Microsoft Computer Dictionary

Fifth Edition



PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Library of Congress Control Number: 2002019714

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to *mspinput@microsoft.com.*

Active Desktop, Active Directory, ActiveMovie, ActiveStore, ActiveSync, ActiveX, Authenticode, BackOffice, BizTalk, ClearType, Direct3D, DirectAnimation, DirectDraw, DirectInput, DirectMusic, DirectPlay, DirectShow, DirectSound, DirectX, Entourage, FoxPro, FrontPage, Hotmail, IntelliEye, IntelliMouse, IntelliSense, JScript, MapPoint, Microsoft, Microsoft Press, Mobile Explorer, MS-DOS, MSN, Music Central, NetMeeting, Outlook, PhotoDraw, PowerPoint, SharePoint, UltimateTV, Visio, Visual Basic, Visual C++, Visual FoxPro, Visual InterDev, Visual J++, Visual SourceSafe, Visual Studio, Win32, Win32s, Windows, Windows Media, Windows NT, Xbox are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Acquisitions Editor: Alex Blanton
Project Editor: Sandra Haynes

Body Part No. X08-41929

lectromag-
uch as radio
ignificantly
formation
f glass or
dreds of
s are essen-
. *See also*

finite series
the two
3, 5, 8, 13,
the thir-
acci of Pisa.
peed binary
to groups in
numbers in
et of 34
nd another
of 13, the
is divided
ue until the
terms in the
Ratio, a
oportions of
y things,
rtions of
in Athens,

ewsgroup
rotocol orig-
om Jen-
ctor in its
e e-mail
vate compa-
nd individu-

rticular type
ECORD
Name,
rent-Salary,
are charac-
of data (for

example, alphabetic, numeric, or financial) that can be placed in them. The facility for creating these specifications usually is contained in the data definition language (DDL). In relational database management systems, fields are called *columns*. **2.** A space in an on-screen form where the user can enter a specific item of information.

field-effect transistor *n. See* FET.

field expansion *n. See* date expansion.

Field Programmable Gate Array *n. See* FPGA.

field-programmable logic array *n.* An integrated circuit containing an array of logic circuits in which the connections between the individual circuits, and thus the logic functions of the array, can be programmed after manufacture, typically at the time of installation in the field. Programming can be performed only once, typically by passing high current through fusible links on the chip. *Acronym:* FPLA. *Also called:* PLA, programmable logic array.

field separator *n.* Any character that separates one field of data from another. *See also* delimiter, field (definition 1).

FIFO *n. See* first in, first out.

fifth-generation computer *n. See* computer.

fifth normal form *n. See* normal form (definition 1).

file *n.* A complete, named collection of information, such as a program, a set of data used by a program, or a user-created document. A file is the basic unit of storage that enables a computer to distinguish one set of information from another. A file is the "glue" that binds a conglomeration of instructions, numbers, words, or images into a coherent unit that a user can retrieve, change, delete, save, or send to an output device.

file allocation table *n.* A table or list maintained by some operating systems to manage disk space used for file storage. Files on a disk are stored, as space allows, in fixed-size groups of bytes (characters) rather than from beginning to end as contiguous strings of text or numbers. A single file can thus be scattered in pieces over many separate storage areas. A file allocation table maps available disk storage space so that it can mark flawed segments that should not be used and can find and link the pieces of a file. In MS-DOS, the file allocation table is commonly known as the FAT. *See also* FAT file system.

file attribute *n.* A restrictive label attached to a file that describes and regulates its use—for example, hidden, system, read-only, archive, and so forth. In MS-DOS, this information is stored as part of the file's directory entry.

file backup *n. See* backup.

file compression *n.* The process of reducing the size of a file for transmission or storage. *See also* data compression.

file control block *n.* A small block of memory temporarily assigned by a computer's operating system to hold information about an opened file. A file control block typically contains such information as the file's identification, its location on a disk, and a pointer that marks the user's current (or last) position in the file. *Acronym:* FCB.

file conversion *n.* The process of transforming the data in a file from one format to another without altering the data—for example, converting a file from a word processor's format to its ASCII equivalent. In some cases, information about the data, such as formatting, may be lost. Another, more detailed, type of file conversion involves changing character coding from one standard to another, as in converting EBCDIC characters (which are used primarily with mainframe computers) to ASCII characters. *See also* ASCII, EBCDIC.

file extension *n. See* extension (definition 1).

file extent *n. See* extent.

file format *n.* The structure of a file that defines the way it is stored and laid out on the screen or in print. The format can be fairly simple and common, as are files stored as "plain" ASCII text, or it can be quite complex and include various types of control instructions and codes used by programs, printers, and other devices. Examples include RTF (Rich Text Format), DCA (Document Content Architecture), PICT, DIF (Data Interchange Format), DXF (Data Exchange File), TIFF (Tagged Image File Format), and EPSF (Encapsulated PostScript Format).

file fragmentation *n.* **1.** The breaking apart of files as they are stored by the operating system into small, separate segments on disk. The condition is a natural consequence of enlarging files and saving them on a crowded disk that no longer contains contiguous blocks of free space large enough to hold them. File fragmentation is not an integrity problem, although it can eventually slow read and write access times if the disk is very full and storage is badly fragmented. Software products are available for redistributing (optimizing) file storage to reduce fragmentation. **2.** In a database, a situation in which records are not stored in their optimal access sequence because of accumulated additions and deletions of records. Most database protected mode, and automatically allocates space in RAM rather than requiring the user to reserve space for the cache. *See also* cache, driver, protected mode, RAM, VFAT.

vCalendar *n.* A specification defining the format for applications to exchange scheduling information. The vCalendar specification is based on existing industry standards, including international standards for representing dates and times, and permits the exchange of schedules and "to-do" lists of the sort users commonly enter into personal calendars. Like the companion vCard specification for electronic business cards, it was created by the versit consortium founded by Apple, AT&T, IBM, and Siemens. Handed off to the Internet Mail Consortium (IMC) in 1996, vCalendar is supported by numerous hardware and software vendors. *See also* vCard.

vCard *n.* A specification for creating an electronic business card (or personal-information card) and for the card itself. Designed to be exchanged through applications such as e-mail and teleconferencing, a vCard includes information such as name, address, telephone and fax number(s), and e-mail address. It can also include timezone, geographic location, and multimedia data such as photographs, company logos, and sound clips. Based on the ITU's X.500 directory services specification, vCard was developed by versit, a consortium whose principal members include Apple, AT&T, IBM, and Siemens. The specification is under the guidance of the Internet Mail consortium Version 3.0 of the vCard specification has been approved as a proposed standard by the IETF. A companion specification known as vCalendar supports electronic exchange of scheduling information. *See also* vCalendar, X series.

V-chip *n.* Electronic chip for installation in a television, VCR, cable box, or stand-alone device to provide adults with the ability to block programming they deem inappropriate. Intended to provide parents with a means of controlling the programming viewed by children, the V-chip allows adults to screen programs based on a rating level transmitted in the portion of the TV signal known as the vertical blanking interval (the same portion that carries closed captioning information). When programs exceed the chosen level, the V-chip signals the television, which then displays an "unauthorized to receive" message on a blank screen.

VCOMM *n.* The communications device driver in Windows 9x that provides the interface between Windows-based applications and drivers on one side, and port drivers and modems on the other. *See also* driver.

VCPI *n. See* Virtual Control Program Interface.

VCR-style mechanism *n.* **1.** A user interface for playing movie files that has controls similar to those on a videocassette recorder (VCR). **2.** A type of motorized docking mechanism in which a laptop or notebook computer is physically locked into place by the docking station. The advantage to a VCR-style mechanism is that it provides an electrically consistent, secure bus connection. *See also* docking mechanism, docking station, laptop, portable computer.

VDD *n.* Acronym for **v**irtual **d**isplay **d**evice driver. *See* virtual device driver.

VDL *n.* Acronym for **V**ienna **D**efinition **L**anguage. A metalanguage, containing both a syntactic and a semantic metalanguage, used to define other languages. *See also* metalanguage.

VDM *n. See* video display metafile.

VDSL *n.* Short for **v**ery-high-speed **d**igital **s**ubscriber **l**ine. The high-speed version of the xDSL (digital subscriber line) communication technologies, all of which operate over existing phone lines. VDSL can deliver up to 52 Mbps downstream, but it is effective only within about 4500 to 5000 feet of the central exchange. The data delivery rate is, in fact, related to the distance the signal must travel. To attain a rate of 52 Mbps, for example, the subscriber must be within 1000 feet of the exchange office. At a distance of 3000 feet, the data rate drops to about 26 Mbps; and at 5000 feet, the data rate drops to about 13 Mbps. *See also* central office, xDSL.

VDT *n.* Acronym for **v**ideo **d**isplay **t**erminal. A terminal that includes a CRT (cathode-ray tube) and keyboard. *See also* CRT.

VDU *n.* Acronym for **v**ideo **d**isplay **u**nit. A computer monitor. *See also* monitor.

vector *n.* **1.** In mathematics and physics, a variable that has both distance and direction. *Compare* scalar. **2.** In computer graphics, a line drawn in a certain direction from a starting point to an endpoint, both of whose locations are identified by the computer using *x-y*-coordinates on a grid. Vectors are used in the output of some graphics programs instead of groups of dots (on paper) or pixels (on screen). *See also* vector graphics. **3.** In data structures, a one-dimensional array—a set of items arranged in a single column or row. *See also* array, matrix.

vector display *n.* A CRT (cathode-ray tube), commonly used in oscilloscopes and DVST (direct view storage tube) displays, that allows the electron beam to be arbitrarily deflected, based on *x-y*-coordinate signals. For example, to draw a line on a vector display, the video adapter sends signals to the X and Y yokes to move the electron beam over the path of the line; there is no background composed of scan lines, so the line drawn on the screen is not constructed of pixels. *See also* CRT, yoke. *Compare* raster display.

vector font *n.* A font in which the characters are drawn using arrangements of line segments rather than arrangements of bits. *See also* font. *Compare* bitmapped font.

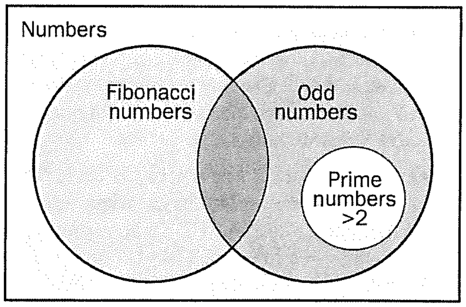
vector graphics *n.* Images generated from mathematical descriptions that determine the position, length, and direction in which lines are drawn. Objects are created as collections of lines rather than as patterns of individual dots or pixels. *Compare* raster graphics.

Vector Markup Language *n. See* VML.

vector table *n. See* dispatch table.

Velocity Engine *n.* A component of Apple's Macintosh G4 processor that processes data in 128-bit chunks. The Velocity Engine is capable of over one gigaflop of floating-point operations per second.

Venn diagram *n.* A type of diagram, used to express the result of operations on sets, in which a rectangle represents the universe and circles inside the rectangle represent sets of objects. Relationships between sets are indicated by the positions of the circles in relation to one another. The Venn diagram is named after John Venn (1834–1923), an English logician at Cambridge University. See the illustration.



Venn diagram.

verbose *adj.* Displaying messages as English text rather than as concise (but cryptic) codes.

verify *vb.* To confirm either that a result is correct or that a procedure or sequence of operations has been performed.

Veronica *n.* Acronym for very easy rodent-oriented Net-wide index to computerized archives. An Internet service developed at the University of Nevada that searches for Gopher archives by keywords. Users can enter Boolean operators, such as AND, OR, and XOR, to help narrow or expand their search. If any matching archives are found, they are listed on a new Gopher menu. *See also* Boolean operator, Gopher. *Compare* Archie, Jughead.

version *n.* A particular issue or release of a hardware product or software title.

version control *n.* The process of maintaining a database of all the source code and related files in a software development project to keep track of changes made during the project.

version number *n.* A number assigned by a software developer to identify a particular program at a particular stage, before and after public release. Successive public releases of a program are assigned increasingly higher numbers. Version numbers usually include decimal fractions. Major changes are generally marked by a change in the whole number, whereas for minor changes only the number after the decimal point increases.

verso *adj.* The publishing term for a left-hand page, which is always even-numbered. *Compare* recto.

vertex *n.* The highest point of a curve, the point where a curve ends, or the point where two line segments meet in a polygon or freeform.

vertical application *n.* A specialized application designed to meet the unique needs of a particular business or industry—for example, an application to keep track of billing, tips, and inventory in a restaurant.

vertical bandwidth *n.* The rate at which a display screen is refreshed entirely, expressed in hertz (Hz). The vertical bandwidth of display systems ranges from 45 Hz to over 100 Hz. *Also called:* vertical scan rate, vertical sync, V-sync.

vertical blanking interval *n.* The time required for the electron beam in a raster-scan display to perform a vertical retrace. *See also* blanking, vertical retrace.

vertical recording *n. See* perpendicular recording.

vertical redundancy check *n. See* VRC.

vertical retrace *n.* On raster-scan displays, the movement of the electron beam from the lower right corner back to the upper left corner of the screen after the beam has completed a full sweep of the screen. *See also* blanking, vertical blanking interval. *Compare* horizontal retrace.

Exhibit 7

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST-NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
95/000,634	05/20/2011	7,461,353	18157.0045.353	8853

29000 7590 08/08/2011
IRELL & MANELLA LLP
1800 AVENUE OF THE STARS
SUITE 900
LOS ANGELES, CA 90067

EXAMINER	
STEELMAN, MARY J	
ART UNIT	PAPER NUMBER
3992	
MAIL DATE	DELIVERY MODE
08/08/2011	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patents and Trademark Office
P.O.Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

DO NOT USE IN PALM PRINTER

THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS
NOVAK DRUCE & QUIGG, LLP
(NDQ REEXAMINATION GROUP)
1000 LOUISIANA ST., FIFTY-THIRD FLOOR
HOUSTON, TX 77002

Date:

Transmittal of Communication to Third Party Requester
Inter Partes Reexamination

REEXAMINATION CONTROL NO. : 95000634
PATENT NO. : 7461353
TECHNOLOGY CENTER :.3999
ART UNIT : 3992

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified Reexamination proceeding. 37 CFR 1.903.

Prior to the filing of a Notice of Appeal, each time the patent owner responds to this communication, the third party requester of the inter partes reexamination may once file written comments within a period of 30 days from the date of service of the patent owner's response. This 30-day time period is statutory (35 U.S.C. 314(b)(2)), and, as such, it cannot be extended. See also 37 CFR 1.947.

If an ex parte reexamination has been merged with the inter partes reexamination, no responsive submission by any ex parte third party requester is permitted.

All correspondence relating to this inter partes reexamination proceeding should be directed to the Central Reexamination Unit at the mail, FAX, or hand-carry addresses given at the end of the communication enclosed with this transmittal.

PTOL-2070(Rev.07-04)

ORDER GRANTING/DENYING REQUEST FOR* INTER PARTES *REEXAMINATION	**Control No.** 95/000,634	**Patent Under Reexamination** 7,461,353	
	Examiner MARY STEELMAN	**Art Unit** 3992	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address. --

The request for *inter partes* reexamination has been considered. Identification of the claims, the references relied on, and the rationale supporting the determination are attached.

Attachment(s): ☒ PTO-892 ☒ PTO/SB/08 □Other: _____

1. ☒ The request for *inter partes* reexamination is GRANTED.

   □ An Office action is attached with this order.

   ☒ An Office action will follow in due course.

2. □ The request for *inter partes* reexamination is DENIED.

This decision is not appealable. 35 U.S.C. 312(c). Requester may seek review of a denial by petition to the Director of the USPTO within ONE MONTH from the mailing date hereof. 37 CFR 1.927. EXTENSIONS OF TIME ONLY UNDER 37 CFR 1.183. In due course, a refund under 37 CFR 1.26(c) will be made to requester.

All correspondence relating to this *inter partes* reexamination proceeding should be directed to the **Central Reexamination Unit** at the mail, FAX, or hand-carry addresses given at the end of this Order.

Transmittal of Communication to Third Party Requester Inter Partes Reexamination

Control No.	Patent Under Reexamination
95/000,634	7,461,353
Examiner	**Art Unit**
MARY STEELMAN	3992

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address. --

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above-identified reexamination proceeding. 37 CFR 1.903.

Prior to the filing of a Notice of Appeal, each time the patent owner responds to this communication, the third party requester of the *inter partes* reexamination may once file written comments within a period of 30 days from the date of service of the patent owner's response. This 30-day time period is statutory (35 U.S.C. 314(b)(2)), and, as such, it cannot be extended. See also 37 CFR 1.947.

If an *ex parte* reexamination has been merged with the *inter partes* reexamination, no responsive submission by any *ex parte* third party requester is permitted.

All correspondence relating to this inter partes reexamination proceeding should be directed to the **Central Reexamination Unit** at the mail, FAX, or hand-carry addresses given at the end of the communication enclosed with this transmittal.

INTER PARTES REEXAMINATION

A substantial new question of patentability ("SNQ") affecting claims 1-319 of USPN 7,461,353 B2 to Rohrabaugh et al. is raised by the inter partes reexamination request filed 05/20/2011. The real party in interest making this request is Apple, Inc. An Office action on the merits will follow in due course.

Extensions of Time

Extensions of time under 37 CFR 1.136(a) will not be permitted in these proceedings because the provisions of 37 CFR 1.136 apply only to "an applicant" and not to parties in a reexamination proceeding. Additionally, 35 U.S.C. 314(c) requires that inter partes reexamination proceedings "will be conducted with special dispatch" (37 CFR 1.937). Patent Owner extensions of time in inter partes reexamination proceedings are provided for in 37 CFR 1.956. Extensions of time are not available for third party requester comments, because a comment period of 30 days from service of patent owner's response is set by statute. 35 U.S.C. 314(b)(3).

Notification of Other Proceedings

The patent owner is reminded of the continuing responsibility under 37 CFR 1.985(a) to apprise the Office of any litigation activity, or other concurrent proceeding, involving the patent under reexamination throughout the course of this reexamination proceeding. The third party requester is also reminded of the ability to similarly apprise the Office of any such activity or proceeding throughout the course of this reexamination proceeding. See MPEP §2686 and 2686.04.

References Cited in Request

Bederson, Benjamin B. and James D. Hollan, Pad++: A Zoomable Graphical Interface System, CHI '95 Mosaic of Creativity, May 1995 ("**Bederson-1**")

Bederson, Benjamin B. and George W. Furnas, Space-Scale Diagrams: Understanding Multiscale Interfaces, CHI '95 Proceedings, 1995 ("**Bederson-2**")

Bederson, Benjamin B., et al, A Zooming Web Browser, SPIE, Vol. 2667, 260-271, May 1996 ("**Bederson-3**")

Bederson, Ben and Jon Meyer, Implementing a Zooming User Interface: Experience Building Pad + +, Software-Practice and Experience, Vol. 28 (1), 1101-1135, August 1998 ("**Bederson-4**")

Bederson, Benjamin B., et al., Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics, Journal of Visual Languages and Computing, Vol. 7, 3-31, 1996 ("**Bederson-5**")

The five Bederson prior art citations above are combined and referred to as "**Pad++**." Each of the Pad++ references was published between 1995 and 1998 and each qualify as prior art under 35 U.S.C. §102(b) to the '353 Patent.

The Pad++ references (combination of five references) were not applied by the Examiner during the prosecution of the '353 Patent. Bederson-5 and Bederson-3 were noted on an IDS, but never applied in a rejection during prosecution of the '353 Patent. Bederson-1 was used in an obvious rejection of the parent application ('097). Bederson-5 and Bederson-3 were discussed during prosecution of the published parent application (09/878,097, which issued as USPN 7,210,099). To the extent that any teachings of Pad++ were previously considered, the combination of the 5 Bederson prior art presents the teachings in a new light.

Japanese Application Publication No, H10-326169 to Hara , et al. (**JP 169**) was published on December 8, 1998 and qualifies as a 102(b) prior art reference. JP 169 was not applied during the prosecution of the '353 Patent. An English translation is included. All citations map to the English translation provided.

Nokia Unveils World's First All-In-One Communicator for the Americas, Nokia Press Release, September 19, 1996 ("**Nokia**") Nokia qualifies as prior art to the '353 Patent under 35 U.S.C. §102(b). Nokia was not previously cited or considered by the Examiner in the prosecution of the '353 Patent. Nokia discloses a mobile phone enabled to access the Internet and to make voice calls.

3Com Announces the Palm VII Connected Organizer, the First Handheld Solution for Out-of-the-box Wireless Internet Access, Palm TM press release, December 2, 1998 ("**Palm**")

The Palm press release qualifies as prior art to the '353 Patent under 35 U.S.C. §102(b). Palm was not previously cited or considered by the Examiner in the prosecution of the '353 Patent. Palm discloses a personal digital assistant (PDA) devices including a touch screen that is enabled to access the internet and transmit electronic mail.

Robles, Emilo, and Jeni Johnston, Apple Outlines Plethora of Newton Wireless Communications Solutions at MessagePad 120 Launch, published January 30, 1995 ("**Newton**") Newton qualifies as prior art to the '353 Patent under 35 U.S.C. §102(b). Newton was not cited or considered by the Examiner in the prosecution of the '353 Patent. Newton discloses (pp. 2-3) a palm-held or hand-held PDA which can be configured to operate as a mobile phone.

Locatio Beginner's Guide was published July 30, 1999 (including English translation thereof) ("**Locatio**") All citations map to the English translation provided.

It is not clear to Examiner what is meant by the following statement, as there appears to be no explanation in section IV(E): (Request, p. 23, paragraph in middle of page) "The Locatio Beginner's Guide ("Locatio") was published on July 30, 1999, and is therefore prior art to the '353 Patent under 35 U.S.C. §102(a) are with respect to claims 2, 30-51, and 35 U.S.C. §102(b) art with respect to claim 53 (as will be explained in section IV(E), below)."

Regarding the suggestion of an SNQ raised over the combination of Pad++ and Locatio (Request, p. 23) or the combination of JP 169 and Locatio (Request, p. 26), Examiner notes that it may inferred that Requester intended to assert that Locatio qualifies as a 102(a) prior art reference with respect to claims 30, 32, 37, 81, 82, 107, 108, 144, 150, 205, 206, 245, 247, 253, 305, and 306 of the '353 Patent. Examiner believes that a typo in Request at p. 26 fails to note the JP 169 / Locatio proposed SNQ includes claim 37. Locatio was not cited or considered by the Examiner in the prosecution of the '353 Patent. Locatio discloses (English Translation, Chapter 3, p. 65) "PDAs that can be used as telephones .... If the old style was to have a cellphone or a PHS phone and a PDA as two separate devices, the Locatio (COM) style is to have all in one, which is easier." Locatio further recites (p. 48) that "[y]ou can connect to the Internet and browse webpages [with Locatio]" and (p. 61) "operations of Locatio are performed using the touch pen and the multimedia controller."

Bray, Tim, "XML Support in IE5," <http://www.xml.com/pub/a/1999/O3/ie5/first-x.html>, published March 18, 1999 ("**Bray**"). Bray qualifies as prior art to the '353 Patent under 35 U.S.C. §102(b). Bray was not cited or considered by the Examiner in the prosecution of the '353 Patent. Bray discloses (p. 1) that XML and/or Cascading Style Sheets (CSS) can be delivered over the Web. Bray further recites (p. 4) that Web browsers can retrieve and work with XML and/or CSS.

Fox, Armondo, et al., "Experience with Top Gun Wingman: A Proxy-Based Graphical Web Browser for the 3Com PalmPilot", published June 22, 1998 ("**Fox**"). Fox qualifies as prior art to the '353 Patent under 35 U.S.C. §102(b). Fox was not cited or considered by the Examiner in the prosecution of the '353 Patent. Fox (2.4) describes a Wingman browser (e.g., a proxy-based graphical Web browser) for a 3Com PalmPilot PDA that "allows users to zoom in on scaled-down images."

Prosecution History

USPN 7,461,353 B2 to Rohrabaugh et al. (Application no. 11/045,757; file date January 28, 2005; issue date December 2, 2008) is a division of Application No. 09/878,097 (file date June 8, 2001; now USPN 7,201,099) which is a continuation in part of Application No. 09/828,511 (file date April, 7, 2001, now abandoned).

As noted throughout the Request, Requester has recited such examples as "cited in an IDS during prosecution of the '486 Application for the '353 Patent" ('486 may be in reference to Application 11/738486 which issued as USPN 7,831,926, currently requested for reexamination as 95/000635**).** Examiner considers this to be a typo, as Application 11/045757 ('757) issued as USPN 7,461,353 ('353).

Prosecution History of Parent Application 09/878,097 (now USPN 7,210,099)

The family of Patent Applications and Patents issued related to the parent patent '099 is as follows:

09/878097, filed 06/08/2001, now USPN 7,210,099 and having 1 RCE-type filing therein
is a continuation in part of 09/828511, filed 04/07/2001, now abandoned
claims priority from Provisional Application 60/211019, filed 06/12/2000

claims priority from Provisional Application 60/217345, filed 07/11/2000.

Child Data of Application '097 / U.S. Patent '099

11/045649, filed on 01/28/2005 ,now USPN 7,584,423 and having 1 RCE-type filing therein, is a division of 09/878097, filed on 06/08/2001, now USPN 7,210,099 and having 1 RCE-type filing therein

11/045757, filed on 01/28/2005, now USPN 7,461,353 is a division of 09/878097, filed on 06/08/2001 ,now USPN 7,210,099 and having 1 RCE-type filing therein

11/735477, filed on 04/15/2007 is a continuation of 09/878097, filed on 06/08/2001, now USPN 7,210,099 and having 1 RCE-type filing therein

11/735477, filed on 04/15/2007 is a division of 09/878097, filed on 06/08/2001, now USPN 7,210,099 and having 1 RCE-type filing therein

11/735482, filed on 04/15/2007, now USPN 7,844,889 is a continuation of 09/878097, filed on 06/08/2001, now USPN 7,210,099 and having 1 RCE-type filing therein

11/735482, filed on 04/15/2007, now USPN 7,844,889 is a division of 09/878097, filed on 06/08/2001 ,now USPN 7,210,099 and having 1 RCE-type filing therein

11/738486, filed on 04/21/2007, now USPN 7,831,926 is a continuation of 09/878097, filed on 06/08/2001, now USPN 7,210,099 and having 1 RCE-type filing therein

11/738932, filed on 04/23/2007, now USPN 7,823,083 is a continuation of 09/878097, filed on 06/08/2001, now USPN 7,210,099 and having 1 RCE-type filing therein

12/941106, filed on 11/08/2010 is a division of 09/878097, filed on 06/08/2001, now USPN 7,210,099 and having 1 RCE-type filing therein.

Notably, **USPN 7,831,926** is also requested for inter partes reexamination (95/000635).

Regarding the prosecution of parent application, 09/878097, Examiner rejected (Non Final Office Action 01/12/2006, p. 4) claims as unpatentable over *Blumberg* in view of *Bederson.* [Referenced in the current reexam, 95/000634, as Bederson-1.]

(Request, p. 16), "Also as discussed, the Applicants, in a communication dated April 10, 2006 [Note this statement is in reference to parent application 09 / 878097, Applicant Remarks received 04/13/2010], attempted to overcome the Examiner's rejection of the pending claims over Blumberg in light of Bederson-1 by submitting additional references relating to Pad++ and characterizing these references and the Pad++ system generally."

Applicant's remarks (parent application, 90/878097, Remarks 04/13/2006, p. 8, 10) supported the independent claim language: "...receiving, at the client device, vector-formatted Web content comprising a machine-readable scalable vector representation of the Web content that provides a scalable resolution-independent display of the Web content that substantially retains an original page layout and attributes of the Web content corresponding to an appearance of the Web page when it is rendered at its predetermined resolution; and rendering the vector-formatted Web content on the client device such that it is displayed to have a different resolution than the predetermined resolution. (Emphasis added)"

Applicant argued (parent application, 90/878097, Remarks 04/13/2006, p. 10): With respect to claim 15, the neither [sic] *Blumberg* or *Bederson* teach or suggest the operations of "receiving, at the client device, vector-formatted Web content comprising a machine-readable scalable vector representation of the Web content that provides a scalable resolution-independent display of the Web content that substantially retains an original page layout and attributes of the Web content corresponding to an appearance of the Web page when it is rendered at its predetermined resolution," or "rendering the vector-formatted Web content on the client device such that it is displayed to have a different resolution than the predetermined resolution."

Applicant summarized (parent application, 90/878,097, Remarks 04/13/2006, pp. 13-16) the Pad++ invention citing to *Pad++: A Zoommable Graphical Sketchpad For Exploring Alternative Interface Physics,* by Benjamin B. Bederson et al. [Noted in Reexamination 95/000634 as Bederson-5.]:

> As described in the "2: Description" section of the Pad++ reference,
>
> **Pad++** ***is a general-purpose substrate for creating and interacting with structured information based on a zoomable interface.*** It adds scale as a first class parameter to all items, as well as various mechanisms for navigating through a multiscale space. It has several efficiency mechanisms which help maintain interactive frame-rates with large and complicated graphical scenes."
>
> "While Pad++ is not an application itself, it directly supports creation and manipulation of multiscale graphical objects, and navigation through spaces of these objects. ***It is implemented as a widget in Tcl/Tk*** [25] (described in a later section) which provides an ***interpreted scripting language for creating zoomable applications.*** The standard objects that Pad++ supports are colored text, graphics, images, portals, and hypertext markup language (HTML). Standard input widgets (buttons, sliders, etc.) are supplied as extensions." (Emphasis added)

Applicants comments on MSML (which Requester disagrees): (parent application, 90/878,097, Remarks 04/13/2006, pp. 15-16), "Details of how a Web browser based on Pad++ works are disclosed in A *Zooming Web Browser,* Benjamin B. Bederson et al. (attached with the

Supplemental IDS filed concurrently herewith). [Noted in Reexamination 95/000634 as Bederson-3.] It is apparent from this document, that the HTML documents (i.e., Web content) that Pad++ operates on are conventional HTML documents (e.g., documents available via typical Web sites) with an extension to HTML called *Multi-Scale Markup Language* (MSML), which allows Pad++ objects to be added to HTML documents so they can be made accessible to the Pad++-based browser. Importantly, this augmented Web content clearly does not comprise vector-formatted Web content, but rather objects written in the Tcl scripting language. Zooming functionality is enabled via the Pad++ browser through use of the Tk API, which is employed for generating user interfaces employing Pad++ objects defined in the Tcl scripts. This leads to two serious deficiencies overcome by embodiments of the present invention: 1) the Pad++ scheme requires more Web content data to be transferred rather than less (e.g., embodiments of the present invention may typically reduce the size of the Web content from 20-80+% (see paragraph [0039] of the present application); and 2) the size and complexity of the client is increased - the Pad++ client must include a Tcl interpreter and a Tk toolkit by definition of the operation of the Tcl/Tk scripting language. Both of these aspects of Pad++ clearly teach away from the present invention." "The multi-scale aspect of MSML does not imply scaling in the conventional sense. Rather, MSML enables different graphics (objects) to be associated with a common object, wherein a selected graphic from among the different graphics is displayed depending on the context of a current view. For example, Figures 5-7 and the accompanying text in the *A Zooming Web Browser* reference show that as a user selects to zoom in on an object (in this case, the New Mexico map), different graphic objects are used at each zoom level, rather than scaling a common graphic object. This supports semantic zooming, as described in the Pad++ reference.

As stated on page 2, 4th paragraph, "We use what we call portals to simulate lenses, and a notion we call *semantic zooming* to scale data in non-geometric ways." (Emphasis in original text.) As further stated in the first paragraph under the Semantic Zooming section header on page 8, "When zoomed out, instead of simply seeing a scaled down version of the object, it is potentially more effective to see *a different representation* of it." (Emphasis added.) (Underlining emphasis added)"

Prosecution History of 11/045,757 (USPN 7,461,353)

An application (Application No. 11/045, 757) for a patent was received 01/28/2005, that included 16 claims. '757 is a divisional of parent Application 09/878,097 (issued as USPN 7,210,099) and incorporates by reference Provisional Applications 60/217,345 & 60/211,019, and Application 09/828,511.

Applicant (03/31/2007) amended the Specification, original claims and added claims 17-39. IDS received 05/06/2007 included the Bederson-3 and Bederson-5 documents. Applicants (06/06/2007) replaced the Abstract, amended the Specification, cancelled claims 1-39 and added new claims 40-141. The amended Specification added the following paragraph, noting continuations of the '097 application:

> "The present application is also related to U.S. Application No. 11/735,477 filed on April 15, 2007, US Application No. 11/735,482 filed on April 15, 2007, U.S. Application Serial No. 11/738,486 filed on April 21, 2007, and U.S. Application No. 11/738,932 filed on April 23, 2007, each of which are continuations of U.S. Application No. 09/878,097, now U.S. Patent No. 7,210,099."

Applicants amended/added (07/19/2007 & 08/31/2007) claims and responded to a requirement for restriction election (08/15/2007) electing, without traverse, claims 71-150 and new claims 151-179. A "Terminal Disclaimer" for copending patent application number 09/878,097 was filed on 06/08/2008 and approved on 06/30/2008. In response to persuasive Applicant's arguments (05/20/2008), a Notice of Allowance was issued (08/08/2008) for claims 71-92; 94-242; 244-335; 337-389 and 391-393 (renumbered as claims 1-319).

Examiner recited (Notice of Allowance 08/08/2008, pp. 3-4):

> "Interpreting the claims in light of the specification, Examiner finds the claimed invention is patentably distinct from the prior art of record, *Chithambaram* et al. US006674445B1 -Provisional No.60/159,069 filed 10/12/1999, in view of Roy et al. US006642925B2 -Continuation of No.08/757,706 filed 10/30/1996, further in view of *Blumberg* US006886034B2 -Continuation of No.09/267,951 filed 03/11/1999, which set forth in the previous rejection mailed on 10/23/2007."
>
> "Under the broadest reasonable interpretation of the claimed limitation consistence with the Applicant's Specification, the prior art of record fail to teach all of the Applicant's claimed limitations. The claimed invention advantageously provides a finer level of detail when displaying HTML Web pages, designed for desktop computers, on a "small-screen" device, such as a cell phone and/or a PDA. In particular, the claimed invention takes HTML-based Web content in its original format (which defines the page layout, functionality and design of the web page) and *translates* the HTML-based Web content into "scalable content" that supports a scalable, resolution-independent representation of the HTML-based Web content. In other words, **the claimed invention *converts* an HTML web page into a "vector graphics" web page** and displays the web page on a PDA. **When viewing the "vector graphics" web page on the PDA, the user may zoom** in and *out* of the displayed web page, in order to increase/decrease the size of the web page components that are displayed on the PDA. Additionally, the claimed invention preserves the functionality of the original HTML web page after it has been ***translated* into a "vector graphics"** web page and displayed on the PDA. See Applicant's Remarks on Pages 91-94 of the Response dated 05/20/2008. See also independent claims 71, 99, 128, 143, 174, 180, 211,244, 265,271,303, 337 and 359." "The Examiner asserts that the claims overcome the prior art of record when the limitations are read in combination with the respective claimed limitations in their entirety."

As noted in Request (05/20/2011, pp. 14-15), "...when the Examiner allowed the claims of the '353 Patent, his reasons for allowance stated--consistent with both the scope of the disclosure and the Applicants' arguments distinguishing the cited Pad++ references in the published '097 application--that their broadest reasonable scope was confined to *"resolution-independent vector display* of Internet content (emphasis added)."

An Issue fee payment for Application ‘757 was received on 08/08/2008. With regards to the prosecution of 11,045,757 (USPN 7,461,353), Applicant did not dispute Examiner’s conclusion of the ‘757 Reasons for Allowance. Contrary to Third Party Requester’s presumption (See Request p. 15, footnotes 9 & 10), Applicants did not file Comments on Reasons for Allowance. There was no document entered July 9, 2010 related to the '353 patent (or the '757 application). Any Comments on the Reasons for Allowance would have been required to be submitted on or before the receipt of Issue Fee.

However, in subsequent related division / continuation applications, Applicant did dispute the scope of claim limitations given by Examiner in Reasons for Allowance, asserting that the plain language of the claims may suggest a broader interpretation claims than the Examiner appreciated when allowing them, arguing that claims lacking such explicit limitations *were not* limited to a *"resolution-independent"* or *"vector-based"* implementation (emphasis added).

Application 11/045,757 (USPN 7,461,353) issued A Certificate of Correction (requested 03/28/2009) on 04/21/2009, making edits to claims 69, 185, and 216.

Claim Terms and other Meaningful Citations

aspect ratio (21: 1-13) different scaling factors applied to the X and Y axis so as to change the aspect ratio of the display – used to adjust to better fit

vector (FIG. 4C) directional data structure stores X & Y values from known datum (primary datum) to object bounding box (object datum) vector between these points is generated for each object (17: 65-67) Evidence supporting the definition of the term 'vector' is provided by Examiner from the Microsoft Computer Dictionary, Fifth Edition.

datum point defined for page / "page datum" and bounding box for each object (17:45-64), relative to a consistent point of reference

scaling vector causes new datum (new starting point-19: 57-20:10) for each object's bounding box relative to rendered page datum (i.e., to use starting point datum of frame, use **offset** relative to page datum to scale vectors) "This produces a new datum for each object's bounding box that is relative to rendered page datum, which remains fixed." (20:7-9), "This foregoing process establishes a starting point (the new datum) for where the content in each object's bounding box will be rendered."

scaled / zoomed (2: 24-25), zoom size (19: 25) magnification

pan /offset (19: 24) i.e., "offset (pan) values..." offset information relative to a known datum will be stored (18: 14-16)

offsets in X and Y (delta X, delta Y) are applied to the starting point of each of the vectors. The vectors are then scaled by a scale factor SF This produces a new datum for each object's bounding box that is relative to rendered page datum 262 which remains fixed. (19:57-63; 18: 45 & FIG 4G)

bounding box (FIG. 4C; 16: 32-35) defined by HTML translator, defines an outlined shape within which the content (text or graphic image) will appear

generate "**scalable vector representation of the original page content"** (17: 44-45);
after **vectors and bounding boxes are offset and scaled**, then content falling within display limit bounding box is retrieved for rendering (20: 18-22) DO NOT retrieve content if vector endpoint would appear off of display (18: 30)

logical groupings (16: 23-32) – related content found within pairs of tags

scalable resolution independent representation - allows scaling/zooming and panned (2: 24-25) presumed by Examiner to also mean "**scalable content that is resolution independent**", where the term 'resolution' is in reference to the given display size and "**resolution-independent**' means the retrieved and translated web content may be rendered on a user's display of any size.

scalable vector representation, FIG. 5, "**vectorized content**" is generated (6: 61-67) HTML translator translates HTML, XML, and cascaded style sheet (CSS) layout content into a scalable vector representation, such as SVF "content is translated into a scalable vector representation (e.g., [for example] **SVF, also referred to herein as "vectorized content"**)..." (8: 58-67), generated by an HTML translator (12: 12-13) (18: 67) See inventive concept at 17: 44-45. Translation of HTML into **scalable vector representation (vectorized content)** at server by server side content translator (6: 6-7) <u>OR</u> (FIG. 1C) at client

"**vector representation data**" vectorized HTML content and compressed bitmap content (19: 15-16)

client can display/render using various **user-selectable scaled/zoomed resolutions and pan/offsets** (9: 9-11) (10: 21-32) (15: 43-18: 46)

display lists improve rendering speed (3: 35)

SVF (Simple Vector Format) can describe web content, "considered" by W3C as a standard for vector content (4: 43-63)

See attached definition of "**vector**." Microsoft Computer Dictionary, Fifth Edition (published 05/01/2002), p. 690, "In computer graphics, a line drawn in a certain direction from a starting point to and endpoint, both of whose locations are identified by the computer using x-y-coordinates on a grid. Vectors are used in the output of some graphics programs instead of groups of dots (on paper) or pixels (on screen)...In data structures, a one-dimensional array-a set of items arranged in a single column or row. See also array, matrix."

<u>Third Party Requester's Proposed Grounds of Rejections based on the Pad++ References</u>

Third Party Requester has proposed the following grounds of rejection:

A. Claims 1-29, 31, 33-36, 38-64, 67-80, 83-104, 109-143, 145-149, 151-180, 183-202, 205, 207-244, 246, 248-252, 254-280, 283-302, and 307-319 of the '353 Patent are rendered obvious by the Pad++ References in view of Common Knowledge of One of Ordinary Skill in the Art. (Request pp. 19 & 30)

B. Claims 30, 32, 37, 81, 82, 107, 108, 144, 150, 205, 206, 245, 247, 253, 305, and 306 of the '353 Patent are obvious in view of the Pad ++ References in view of Nokia and further in view of Palm. (Request pp. 21-22, 60) (Request, p. 21), “The Pad++ in view of Nokia and Palm disclose certain limitations of the claims of the '926 [Examiner presumes '926 is a typo and should be '353] Patent.

C. Claims 30, 32, 37, 81, 82, 107, 108, 144, 150, 205, 206, 245, 247, 253, 305, and 306 of the ‘353 Patent are obvious in view of the Pad++ References in view of Newton. (Request pp. 22-23, 61)

D. Claims 30, 32, 37, 81, 82, 107, 108, 144, 150, 205, 206, 245, 247, 253, 305, 306 of the '353 Patent are obvious in view of the Pad ++ References in view of Locatio. (Request pp. 23-24, 62) (Request, p. 23), "Pad++ in view of Locatio discloses certain limitations of the claims of the '926 Patent." [Examiner presumes typo here (Request, p. 23). Sentence should reference the '353 Patent].

E. Claims 65, 66, 105, 106, 181, 182, 203, 204, 281, 282, 303, and 304 of the '353 Patent are obvious in view of the Pad++ References in view of Bray. (Request pp. 24-25, 63) [Noting the inconsistencies at Request page iv and p. 24 & 63, Examiner assumes that claims 65 and 66 are intended to be included in this proposed rejection.]

Proposed SNQs

SNQ #1 – Requester proposes an SNQ as to claims 1-319 of USPN 7,461,353 B2 to Rohrabaugh et al., raised by the combination of Pad++ references (and in combination with secondary references as noted above).

Support for SNQ based on the Pad++ references:

Requester addressed four key points to support the use of the five combined Pad++ references:

(i) (Request 09/20/2011, pp. 15-16), "In particular, under the Applicants' asserted position that many claims of the '353 Patent should not be limited in scope to a "**resolution-independent**," "**vector**"-based implementation, the cited Pad++ references cannot be distinguished from the claims of the '353 Patent on even the bases [sic] the Applicants themselves set forth during prosecution of the published '097 application. Thus, the cited Pad++ references alone present a substantial new question of patentability, at least as to the '353 Patent claims lacking explicit "resolution-independent" and/or "vector" limitations, specifically claims 1-4, 6-22, 28-32, 34-42, 44-57, 62-83, 85-121, 123-136, 142-146, 148-155, 157-173, 178-182, 184-219, 221-237, 243-247, 249-258, 260-273, 278-282, and 284-319."

[Examiner disagrees with Requester. The list of claim numbers lacking "resolution-independent" and / or "vector" limitations are claims 36-42, 44-57, 62-83, 85-111,149-155, 157-173, 178-182, 184-208, 252-258, 260-273, 278-282, 284-310, and 317-319, i.e., claim 1 recites "resolution-independent representation".]

(ii) (Request 05/20/2011, pp. 16-18), Prior Applicant comments (in the prosecution of parent application '097, 04/13/2006) on the Pad++ arguments were mischaracterized. Requester asserts that Pad++ is able to present Web pages written in standard HTML and able to conventionally zoom Web pages.

(iii) (Request, pp. 17-18), The use of MSML-based objects to embed special Pad++ content within Web pages, is unrelated to the presentation, zooming, or panning of Web pages written in standard HTML.

(iv) (Request, pp. 18-19), Arguments presented related to MSML and “semantic zooming” ("Semantic zooming" refers to the ability to change the appearance of a displayed object (rather than simply scaling the object) when the user zooms in or out far enough. Bederson-3 at 262.) are irrelevant. "...the Pad++ browser supports conventional, or "geometric," scaling of Web pages, in addition to semantic zooming.” (Bederson-3 at 262.)

(Request, pp. 19-20), “The Pad++ references are published technical articles pertaining to the Pad++ product that provides "a graphical interface system based on zooming, as an alternative to traditional window and icon-based approaches." Bederson-1, Abstract. Pad++ provides a Web browser that is designed to operate on a wide range of platforms, including Personal Digital Assistants ("PDAs"). Pad++ allows a user to zoom in and out of Web pages in real time while preserving such Web pages' original layout, design and functionality, *e.g.*, Bederson-3 at pp. 5, 11-12. The Pad++ references further describe: Translation of standard HTML into Pad++ objects that comprise "scalable" and "vector"-based content and/or page layout information (see e.g., Bederson-2 at 3-4); Vector-based layout of the elements of an HTML-based Web page comprising resolution-independent coordinates relative to a fixed datum point (see e.g., Bederson-4 at 1129-1132); Zooming and panning of Web pages in real time by multiplying these coordinates by a scale factor and applying an offset (see e.g., Bederson-5 at 7 and Bederson-4 at 1129- 1132); and preserving the layout, design and functionality of the Web pages when scaled, zoomed and panned (see e.g., Bederson-4 at 1128-1129).”

Requester asserts (Request 05/20/2011, pp. 17 & 20) that the Pad++ references teach a zooming Web browser in which the original page layout, functionality, and design defined by the

HTML-based Web content [are preserved]..., wherein preservation of the functionality defined by the HTML-based Web content includes preservation of hyperlink functionality.

Examiner agrees that the Pad++ References **do raise an SNQ** as to claims 1-4, 6-22, 28-32, 34-42, 44-57, 62-83, 85-121, 123-136, 142-146, 148-155, 157-173, 178-182, 184-219, 221-237, 243-247, 249-258, 260-273, 278-282 and 284-319 of the '353 patent. The references provide new technical teachings (features related to zooming and panning of web content, fitting Web content to display screens at various resolutions, not limited to vector embodiments) of limitations not present during the original examination that would have been important to a reasonable examiner in considering the patentability of the claims. Given the above noted teachings, along with the mapping of the references to the claims as provided in the body of the Request, sufficient evidence is provided such than an SNQ is raised by the proposal. The teachings of the combination of Pad++ References are not cumulative to any written discussion on the record of the teachings of the prior art, were not previously considered nor addressed during a prior examination, and the same question was not the subject of a final holding of invalidity in the Federal Courts.

The combinations of Pad++ references **do not raise an SNQ** related to claims 5, 23-27, 33, 43, 58-61, 84, 122, 137-141, 147, 156, 174-177, 183, 220, 238-242, 248, 259, 274-277, and 283 of the '353 patent (all claims with "vector" limitations).

Pad++ teaches (p. 1129) a "view" that specifies the visible portion of a surface with <u>a point</u> and a magnification (scale or zoom). "The <u>point</u> specifies the portion of the surface that will appear at the <u>center of the view</u>." Pad++ does disclose (p. 1132) the view and object coordinates (inferred relative to a primary datum of (0,0)). Pad++ does disclose coordinate transformations, noting the “each view has an offset and magnification, which are stored as a triplet (xview, yview, zoom). Every object has a position and a scale (xoffset, yoffset, scale). To apply a coordinate transformation from object to screen coordinates, we start by applying the transform for the current view..." The transform algorithm at p. 1132, zoom * (xoffset-xview), relies on an (X,Y) value relating the object (object datum) and an (X,Y) value representing the center of a panned view. While the xoffset may be relative to an inferred primary datum (0,0), there is no disclosed vector generated from a primary datum to an object datum. The references do not provide new technical teachings of “vector” related limitations that would have been important to a reasonable examiner in considering the patentability of the claims.

The combinations of Pad++ References do not raise an SNQ based on the lack of obvious teaching of the following “vector” related limitations:

(similar limitations found in claims 5, 23, 24, 58, 122, 138, 174, 220, 239 and 274) “generating a vector from the primary datum to the object datum for the object; and creating a reference that links the object to the vector that is generated.”

(similar limitations found in claims 23, 137, and 238) “generating a vector-based display list associated with the scalable content...”

(similar limitations found in claims 26, 60, 140, 176, 241, and 276) “mapping the object vectors and associated bounding boxes...”

(similar limitations found in claims 33, 43, 84, 147, 156, 183, 248, 259, and 283) "vector based content..." or "scalable vector based content..."

Claims dependent upon the above noted limitations will not be reexamined.

Re. Ground #1:

Requester's proposed combinations of prior art based on Pad++ References **do not raise an SNQ** as related to claims 5, 23-27, 33, 43, 58-61, 84, 122, 137-141, 147, 156, 174-177, 183, 220, 238-242, 248, 259, 274-277, and 283 of the '353 patent. The **Order is denied** for reexamination of claims 5, 23-27, 33, 43, 58-61, 84, 122, 137-141, 147, 156, 174-177, 183, 220, 238-242, 248, 259, 274-277, and 283 of the '353 patent, based on the Pad++ references.

Requester's proposed combinations of prior art based on Pad++ **do raise an SNQ** (features related to zooming and panning of web content, fitting Web content to display screens at various resolutions, not limited to "vector" embodiments) as related to claims 1-4, 6-22, 28-32, 34-42, 44-57, 62-83, 85-121, 123-136, 142-146, 148-155, 157-173, 178-182, 184-219, 221-237, 243-247, 249-258, 260-273, 278-282 and 284-319 of the '353 patent. An **Order is granted** on the reexamination of claims 1-4, 6-22, 28-32, 34-42, 44-57, 62-83, 85-121, 123-136, 142-146, 148-155, 157-173, 178-182, 184-219, 221-237, 243-247, 249-258, 260-273, 278-282 and 284-319 of the '353 patent, with respect to the Pad++ reference combinations.

Third Party Requester's Proposed Grounds of Rejections based on the JP 169 Reference

Third Party Requester has proposed the following grounds of rejection:

F. Claims 1-10, 15-21, 23-28, 31, 33-36, 38-47, 51-55, 57-63, 67-80, 83-87, 92-103, 109-127, 131-135, 137-142, 145-149, 151-160, 165-171, 173-179, 183-185, 190-201, 205, 207-225, 230-236, 238-243, 246, 248-252, 254-263, 267-271, 273-279, 283-285, 290-302, and 307-316 of the '353 Patent are rendered obvious by the JP 169 reference in view of common knowledge of one of ordinary skill in the art. (Request, p. 65)

G. Claims 30, 32, 37, 81, 82, 107, 08, 144, 150, 205, 206, 245,247, 253, 305, 306 of the '353 Patent are rendered obvious by the JP 169 reference in view of Locatio. (Request, p. 88)

H. Claims 105, 106, 181, 182, 203,204, 281,282, 303, and 304 of the '353 Patent are rendered obvious by the JP 169 reference in view of Bray. (Request, p. 89)

I. Claims 11-14, 22, 29, 48-50, 56, 64, 104, 128-130, 136, 143, 161-164, 172, 180, 186-189, 202, 226-229, 237, 244, 264-266, 272, 280, 286-289, 302 and 317-319 of the '353 Patent are rendered obvious by the JP 169 reference in view of knowledge of one of ordinary skill in the art and in further view of the Pad++ References. (Request, p. 90)

J. Claims 11-14, 48-50, 128-130, 161-164, 186-189, 226-229, 264-266, 286- 289, and 317-319 of the '353 Patent are rendered obvious by the JP 169 reference in view of knowledge of one of ordinary skill in the art and in further view of the Fox. (Request, p. 98)

SNQ #2 – Requester has proposed an SNQ as to claims 1-319 of USPN 7,461,353 B2 to Rohrabaugh et al., raised by the obvious teachings of JP 169 (and in combination with secondary references as noted above).

Support for SNQ

Ground #2: JP 169 discloses that "[i]n WWW browsers, the machine on the information provider side is not matched to the machine on the information receiver side .... the display size of the displayed data is changed in accordance with the resolution of the display screen... if, for example, the image data is low-resolution, it is still possible to display with a size that matches the display screen." JP 169 at ¶¶ [0099]-[0100]. JP 169 discloses that WWW content can include clickable data (hyperlinks) associated with a linked destination. JP 169 further discloses that "the clickable control table 36 is stored in the RAM 12, as illustrated in FIG. 9 (c). This clickable control table 36 stores the [S]HAPE of the clickable region, the coordinate values thereof, and information indicating the link destination." JP 169 at ¶ [0066]. Therefore, JP 169 discloses a zooming (pan and zoom) Web browser in which the "original page layout, functionality, and design defined by the HTML-based Web content [are preserved]..., wherein preservation of the functionality defined by the HTML-based Web content includes preservation of hyperlink functionality.

Examiner agrees that JP 169 **raises an SNQ** to claims 1-4, 6-22, 28-32, 34-42, 44-57, 62-83, 85-121, 123-136, 142-146, 148-155, 157-173, 178-182, 184-219, 221-237, 243-247, 249-258, 260-273, 278-282 and 284-319 of the '353 patent. JP 169 provides obvious new technical teachings of limitations not present during the original examination that would have been important to a reasonable examiner in considering the patentability of the claims (features related to zooming and panning of web content, fitting Web content to display screens at various resolutions, not limited to "vector" embodiments). Given the above noted teachings, along with the mapping of the references to the claims as provided in the body of the Request, sufficient evidence is provided such than an SNQ is raised (on the claims noted above) by the proposal. The teachings of JP 169 are not cumulative to any written discussion on the record of the teachings of the prior art, were not previously considered nor addressed during a prior examination, and the same question was not the subject of a final holding of invalidity in the Federal Courts.

An **SNQ is not raised** related to claims 5, 23-27, 33, 43, 58-61, 84, 122, 137-141, 147, 156, 174-177, 183, 220, 238-242, 248, 259, 274-277, and 283 of the '353 patent ("vector" related claims).

JP 169 discloses zooming features (magnification) and broadly discloses panning ([0092], "corrects the coordinate values within the clickable control table 36 based on the magnification stored in the magnification parameter table...a shift is produced in the coordinate locations for the clickable data..."). Although new coordinate values (new (X,Y) values) are suggested, in no way does JP 169 define a "primary datum", an "object datum", and generating a vector from the primary datum to the object datum for the object; creating a reference that links the object to the vector that is generated. While it could be inferred that (0,0) reads on a primary datum, and a given pair of values (X,Y) read on an object datum, there is no teaching of generating a vector. Requester's proposed citation (Request, p. 72) references magnification (a new height and new width) to fit the resolution of the display screen.

JP 169 does not teach the following "vector" related limitations:

(similar limitations found in claims 5, 23, 24, 58, 122, 138, 174, 220, 239 and 274) "generating a vector from the primary datum to the object datum for the object; and creating a reference that links the object to the vector that is generated."

(similar limitations found in claims 23, 137, and 238) "generating a vector-based display list associated with the scalable content..."

(similar limitations found in claims 26, 60, 140, 176, 241, and 276) "mapping the object vectors and associated bounding boxes..."

(similar limitations found in claims 33, 43, 84, 147, 156, 183, 248, 259, and 283) "vector based content..." or "scalable vector based content..."

Claims dependent upon the above noted limitations will not be reexamined.

Re. Ground #2:

Requester's proposed combinations of prior art based on JP 169 **do not raise an SNQ** related to claims 5, 23-27, 33, 43, 58-61, 84, 122, 137-141, 147, 156, 174-177, 183, 220, 238-242, 248, 259, 274-277, and 283 of the '353 patent. **An Order** to reexamine claims 5, 23-27, 33, 43, 58-61, 84, 122, 137-141, 147, 156, 174-177, 183, 220, 238-242, 248, 259, 274-277, and 283 of the '353 patent based on the JP 169 reference **is denied**.

Requester's proposed combinations of prior art based on JP 169 **do raise an SNQ related to** (features related to zooming and panning of web content, fitting Web content to display screens at various resolutions, not limited to "vector" embodiments) claims 1-4, 6-22, 28-32, 34-42, 44-57, 62-83, 85-121, 123-136, 142-146, 148-155, 157-173, 178-182, 184-219, 221-237, 243-247, 249-258, 260-273, 278-282 and 284-319 of the '353 patent. An **Order is granted** on the reexamination of claims 1-4, 6-22, 28-32, 34-42, 44-57, 62-83, 85-121, 123-136, 142-146, 148-155, 157-173, 178-182, 184-219, 221-237, 243-247, 249-258, 260-273, 278-282 and 284-319 of the '353 patent with respect to JP 169.

Conclusion

Service of Papers

Any paper filed with the USPTO, i.e., any submission made, by either the Patent Owner or the Third Party Requester must be served on every other party in the reexamination proceeding, including any other third party requester that is part of the proceeding due to merger of the reexamination proceedings. As proof of service, the party submitting the paper to the Office must attach a Certificate of Service to the paper, which sets forth the name and address of the party served and the method of service. Papers filed without the required Certificate of Service may be denied consideration. 37 CFR 1.903; MPEP 2666.06.

Amendment in Inter Partes Reexamination Proceedings

Any proposed amendment to the specification and/or claims in this reexamination proceeding must comply with 37 CFR 1.530(d)-(j), must be formally presented pursuant to 37

CFR 1.52(a) and (b), and must contain any fees required by 37 CFR 1.20(c). Amendments in an inter partes reexamination proceeding are made in the same manner that amendments in an ex parte reexamination are made. MPEP 2666.01. See MPEP 2250 for guidance as to the manner of making amendments in a reexamination proceeding.

Requirements of responses, written comments, and briefs in inter partes reexamination are found in 37 CFR 1.943.

Submissions

In order to ensure full consideration of any amendments, affidavits or declarations, or other documents as evidence or patentability, such documents must be submitted in response to this Office action. Submissions after the next Office action, which is intended to be an Action Closing Prosecution (ACP), will be governed by 37 CFR 1.116(b) and (d), which will be strictly enforced.

All correspondence relating to this inter partes reexamination proceeding should be directed:

By EFS: Registered users may submit via the electronic filing system EFS-Web, at https://sportal.uspto.gov/authenticate/authenticateuserlocalepf.html.

By Mail to: Mail Stop Inter Partes Reexam
Attn: Central Reexamination Unit
Commissioner for Patents
United States Patent & Trademark Office

P.O. Box 1450
Alexandria, VA 22313-1450

By FAX to: (571) 273-9900
Central Reexamination Unit

By Hand: Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Any inquiry concerning this communication or earlier communications from the examiner, or as to the status of this proceeding, should be directed to the Central Reexamination Unit at telephone number (571) 272-7705.

/Mary Steelman/
Primary Examiner, Reexamination Specialist
Central Reexamination Unit 3992
(571) 272-3704

Conferees:

Notice of References Cited	Application/Control No. 95/000,634	Applicant(s)/Patent Under Reexamination 7,461,353	
	Examiner MARY STEELMAN	Art Unit 3992	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-			
	B	US-			
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

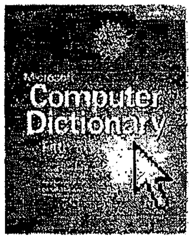
*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	Microsoft Computer Dictionary, Fifth Edition, published 05/01/2002, definition of "vector", p. 690, retrieved from URL <http://academic.safaribooksonline.com/book/communications/0735614954
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Safari Books Online

[Start a New Search] Entire Site SEARCH ADVANCED SEARCH

Sign Out & Clear Session • Personal Sign In



Microsoft® Computer Dictionary, Fifth Edition

By: Microsoft Press
Publisher: Microsoft Press
Pub. Date: May 1, 2002
Print ISBN-13: 978-0-7356-1495-6
Print ISBN-10: 0-7356-1495-4
Pages in Print Edition: 656
Amazon.com® Reviews
Subscriber Rating: [0 Ratings]
Popular Tags: None Yet

ADD NOTE / TAGS

START READING

BUY FROM PUBLISHER

Safari Books Online Subscribers Get Up to 35% OFF

V-chip *n.* Electronic chip for installation in a television, VCR, cable box, or stand-alone device to provide adults with the ability to block programming they deem inappropriate. Intended to provide parents with a means of controlling the programming viewed by children, the V-chip allows adults to screen programs based on a rating level transmitted in the portion of the TV signal known as the vertical blanking interval (the same portion that carries closed captioning information). When programs exceed the chosen level, the V-chip signals the television, which then displays an "unauthorized to receive" message on a blank screen.

VCOMM *n.* The communications device driver in Windows 9x that provides the interface between Windowsbased applications and drivers on one side, and port drivers and modems on the other. *See also* driver.

VCPI *n.* *See* Virtual Control Program Interface.

VCR-style mechanism *n.* **1.** A user interface for playing movie files that has controls similar to those on a videocassette recorder (VCR). **2.** A type of motorized docking mechanism in which a laptop or notebook computer is physically locked into place by the docking station. The advantage to a VCR-style mechanism is that it provides an electrically consistent, secure bus connection. *See also* docking mechanism, docking station, laptop, portable computer.

VDD *n.* Acronym for virtual display device driver. *See* virtual device driver.

VDL *n.* Acronym for Vienna Definition Language. A metalanguage, containing both a syntactic and a semantic metalanguage, used to define other languages. *See also* metalanguage.

VDM *n.* *See* video display metafile.

VDSL *n.* Short for very-high-speed digital subscriber line. The high-speed version of the xDSL (digital subscriber line) communication technologies, all of which operate over existing phone lines. VDSL can deliver up to 52 Mbps downstream, but it is effective only within about 4500 to 5000 feet of the central exchange. The data delivery rate is, in fact, related to the distance the signal must travel. To attain a rate of 52 Mbps, for example, the subscriber must be within 1000 feet of the exchange office. At a distance of 3000 feet, the data rate drops to about 26 Mbps; and at 5000 feet, the data rate drops to about 13 Mbps. *See also* central office, xDSL.

VDT *n.* Acronym for video display terminal. A terminal that includes a CRT (cathode-ray tube) and keyboard. *See also* CRT.

VDU *n.* Acronym for video display unit. A computer monitor. *See also* monitor.

vector *n.* **1.** In mathematics and physics, a variable that has both distance and direction. *Compare* scalar. **2.** In computer graphics, a line drawn in a certain direction from a starting point to an endpoint, both of whose locations are identified by the computer using *x-y-* coordinates on a grid. Vectors are used in the output of some graphics programs instead of groups of dots (on paper) or pixels (on screen). *See also* vector graphics. **3.** In data structures, a one-dimensional array—a set of items arranged in a single column or row. *See also* array, matrix.

vector display *n.* A CRT (cathode-ray tube), commonly used in oscilloscopes and DVST (direct view storagetube) displays, that allows the electron beam to be arbitrarily deflected, based on *x-y-*coordinate signals. For example, to draw a line on a vector display, the video adapter sends signals to the X and Y yokes to move the electron beam over the path of the line; there is no background composed of scan lines, so the line drawn on the screen is not constructed of pixels. *See also* CRT, yoke. *Compare* raster display.

vector font *n.* A font in which the characters are drawn using arrangements of line segments rather than arrangements of bits. *See also* font. *Compare* bitmapped font.

vector graphics *n.* Images generated from mathematical descriptions that determine the position, length, and direction in which lines are drawn. Objects are created as collections of lines rather than as patterns of individual dots or pixels. *Compare* raster graphics.

Doc code: IDS
Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (01-10)
Approved for use through 07/31/2012. OMB 0651-0031
U.S. 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 contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT **(Not for submission under 37 CFR 1.99)**		
	Application Number	11045757
	Filing Date	2005-01-28
	First Named Inventor	Gary B. Rohrabaugh
	Art Unit	
	Examiner Name	
	Attorney Docket Number	18157.0045.353

U.S.PATENTS

Examiner Initial*	Cite No	Patent Number	Kind Code1	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Patent citation information please click the Add button.

U.S.PATENT APPLICATION PUBLICATIONS

Examiner Initial*	Cite No	Publication Number	Kind Code1	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Published Application citation information please click the Add button.

FOREIGN PATENT DOCUMENTS

Examiner Initial*	Cite No	Foreign Document Number3	Country Code2 i	Kind Code4	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T5
/M.S./	1	10-326169	H		1998-12-08	HARA, MASUO		☒

If you wish to add additional Foreign Patent Document citation information please click the Add button

NON-PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T5

95000634 - GAU: 3992

INFORMATION DISCLOSURE STATEMENT BY APPLICANT **(Not for submission under 37 CFR 1.99)**	Application Number		11045757
	Filing Date		2005-01-28
	First Named Inventor	Gary B. Rohrabaugh	
	Art Unit		
	Examiner Name		
	Attorney Docket Number		18157.0045.353

/M.S./	1	Bederson, Benjamin B. and James D. Hollan, Pad++: A Zoomable Graphical Interface System, CHI '95 Mosaic of Creativity, May 1995	□
	2	Bederson, Benjamin B. and George W. Furnas, Space-Scale Diagrams: Understanding Multiscale Interfaces, CHI '95 Proceedings, 1995	□
	3	Bederson, Benjamin B., et al, A Zooming Web Browser, SPIE, Vol. 2667, 260-271, May 1996	□
	4	Bederson, Ben and Jon Meyer, Implementing a Zooming User Interface: Experience Building Pad ++, Software-Practice and Experience, Vol. 28(1), 1101-1135, August 1998	□
	5	Bederson, Benjamin B., et al., Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics, Journal of Visual Languages and Computing, Vol. 7, 3-31, 1996	□
	6	Nokia Unveils World's First All-In-One Communicator for the Americas, Nokia Press Release, September 19, 1996	□
	7	3Com Announces the Palm VII Connected Organizer, the First Handheld Solution for Out-of-the-box Wireless Internet Access, Palm™ press release, December 2, 1998	□
	8	Robles, Emilo, and Jeni Johnston, Apple Outlines Plethora of Newton Wireless Communications Solutions at MessagePad 120 Launch, published January 30, 1995	□
	9	Locatio Beginner's Guide published July 30, 1999 (including English translation thereof)	☒
	10	Bray, Tim, "XML Support in IE5," http://www.xml.com/pub/a/1999/03/ie5/first-x.html, published March 18, 1999	□
/M.S./	11	Fox, Armondo, et al., Experience with Top Gun Wingman: A Proxy-Based Graphical Web Browser for the 3Com PalmPilot, published June 22, 1998	□

EFS Web 2.1.17

INFORMATION DISCLOSURE STATEMENT BY APPLICANT **(Not for submission under 37 CFR 1.99)**	Application Number	11045757
	Filing Date	2005-01-28
	First Named Inventor	Gary B. Rohrabaugh
	Art Unit	
	Examiner Name	
	Attorney Docket Number	18157.0045.353

	12		□

If you wish to add additional non-patent literature document citation information please click the Add button

EXAMINER SIGNATURE

Examiner Signature	/Mary Steelman/	Date Considered	08/05/2011

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

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

Exhibit 8

[COUNSEL LISTED ON SIGNATURE PAGES]

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN JOSE DIVISION

APPLE INC., a California corporation, Plaintiff, v. SAMSUNG ELECTRONICS CO., LTD., a Korean corporation; SAMSUNG ELECTRONICS AMERICA, INC., a New York corporation; and SAMSUNG TELECOMMUNICATIONS AMERICA, LLC, a Delaware limited liability company, Defendants.	Case No. 11-cv-01846-LHK **APPLE'S STATEMENT IDENTIFYING THE CLAIMS IT WILL ASSERT AT TRIAL**

Pursuant to the Court's instructions at the May 2, 2012 Case Management Conference and the minute order received today, Apple submits this statement identifying the claims it will assert at trial.

I. APPLE'S POSITION

Samsung's infringing conduct imposes massive, continuing harm on Apple. This Court has already found that Samsung is likely infringing valid design and utility patents (Dkt. No. 452 at 27, 55-56), and that Samsung's sales of the accused phones and tablets are causing Apple irrecoverably to lose market share and customers (*Id*. at 31-32, 48-49.) In order to preserve its July 30 trial date, Apple is prepared to limit its offensive intellectual property claims to four (4) utility patent claims, and design rights in its iPhone and iPad — as further set out herein. These specific claims were chosen because they present unitary themes that should be easy for the jury to understand and evaluate.

1. The Case Is Ready for Trial

The parties have concluded extensive discovery on the full range of issues raised by their respective claims. The resulting evidence confirms the merit of Apple's claims. This evidence came to light even though Samsung has repeatedly impeded Apple's discovery efforts.[1]

While the parties have been readying the case for trial Samsung has vaulted into first place in worldwide sales of smartphones, with massive sales of its copycat products. (http://www.washingtonpost.com/business/industries/samsung-electronics-reports-record-profit-on-strong-smartphone-sales/2012/04/26/gIQAraz0jT_story.html.) Samsung's infringement of Apple's intellectual property has already resulted in damages that reach billions of dollars. At trial Apple will seek to recover those losses and to obtain adequate injunctive relief to prevent further losses. It is critical to Apple to start trial on July 30, to put an end to Samsung's continuing infringement.

2. Apple Proposes to Narrow its Case on Apple-Asserted Patents

To preserve the July 30 trial date, Apple is willing to narrow the case on its patents for

[1] Samsung has already been sanctioned for disobeying three separate court orders. Apple's motion concerning spoliation of evidence is pending.

jury trial to four utility patent claims and a small set of design-related claims:

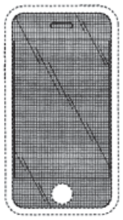
- Infringement of U.S. Patent No. 7,469,381 (rubber banding)—claim 19
- Infringement of U.S. Patent No. 7,844,915 (scroll v. gesture)—claim 8
- Infringement of U.S. Patent No. 7,663,607 (multipoint touchscreen)—claim 8
- Infringement of U.S. Patent No. 7,864,163 (tap to zoom and navigate)—claim 50
- Imitation of the iPhone design, as protected by:
    - U.S. Patent Nos. D618,677, D593,087, D617,334, and D604,305 (iPhone body-style & icon layout design patents [Apple will drop one more of these designs after related motions have been decided])
    - The iPhone trade dress (based on the trade dress Registration No. 3,470,983, the unregistered combination iPhone trade dress, and the unregistered iPhone 3G trade dress [to further narrow the case, Apple is prepared to go to a jury trial on only dilution of the iPhone trade dress])
- Imitation of the iPad design, as protected by:
    - U.S. Patent No. D504,889 (tablet body-style design patent)
    - The iPad trade dress (based on unregistered iPad/iPad 2 trade dress)

This proposal represents a significant reduction in the scope of Apple's jury trial case from that set forth in the May 1 Joint Case Management Statement, which was itself a dramatic narrowing of Apple's original case. The Court should allow Apple to proceed to a jury trial on July 30 under this proposal, for several reasons.

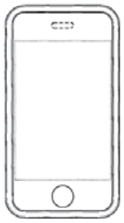
First, the scope of Apple's streamlined case is well within the parameters established by decisions limiting the number of claims that may be asserted at trial. *See, e.g., In re Katz Interactive Call Processing Patent Litig.*, 639 F.3d 1303, 1301-12 (Fed. Cir. 2011) (approving limit of 64 claims); *Gen-Probe Inc. v. Becton Dickinson and Co.*, 2012 U.S. Dist. LEXIS 21744, at *9 (S.D. Cal. Feb. 22, 2012) (limiting number of claims at trial to 30); *Oasis Research, LLC v. Adrive, LLC*, 2011 U.S. Dist. LEXIS 153466, at *10 (E.D. Tex. Sept. 13, 2011) (limiting number of claims at trial to 31).

Second, the remaining patents in Apple's jury case have been chosen so that they can be presented quickly and clearly to a jury. The four remaining utility patents all relate to Apple's unique multi-touch functions. Three of them relate to simple gestures that jurors will readily understand from seeing an iPhone in operation, and the fourth reads only on Samsung's two tablet products. Moreover, these utility patents collectively read on just four versions of Samsung's operating system software. Thus, even though a larger number of phones infringes those patents, only four "accused software versions" need be presented to and analyzed by the jury, because they are run on all the accused products. In addition, the Court has construed key terms in two of the utility patents (the '381 and '915 Patents), obviating any need to address those claim construction issues during trial. (Dkt. No. 849 at 17-23, 38-42.)

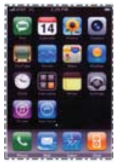
The design-related claims—whether design patents or trade dress—are non-technical, visual claims to the body-style and graphical user interface of the phones and tablets:



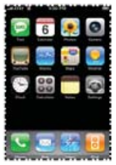
D'677



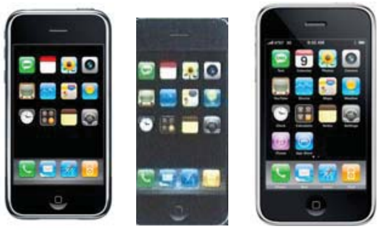
D'087



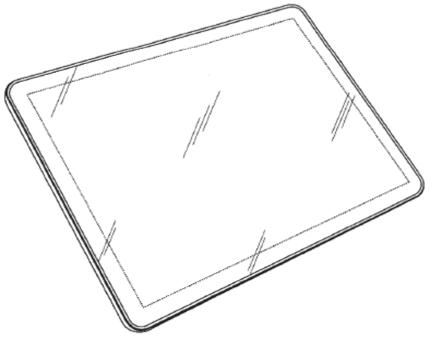
D'334



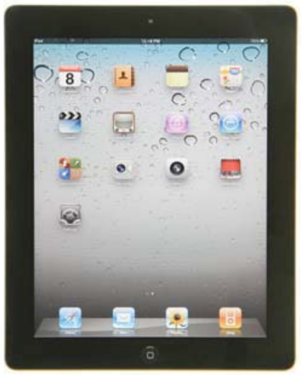
D'305[2]



iPhone Trade Dress



D'889 Patent



iPad/iPad 2 Trade Dress

[2] Apple will drop one of these designs from its jury trial case after related motions have been decided.

As this Court has observed, a trier of fact can "determine almost instinctively" whether an asserted design creates the same visual impression as the accused product. (Dkt. No. 452 at 18.) With narrowly focused utility patents and an instinctive design and trade dress case, Apple will be able expeditiously to present the case on its patents in a manner readily understandable by the jurors.

Third, the reasons why this Court granted Apple's motion to expedite the trial continue to hold true. With each passing day, Apple loses customers and revenue as a result of Samsung's infringement. In light of the market conditions in which Apple operates, any substantial delay in the trial date vitiates its case.

3. Options for Remaining Claims

Under today's proposal, Apple is willing to drop from its jury case all claims arising from three utility patents, two design patent claims, all infringement claims based on the iPhone trade dress (and any reliance on iPhone trade dress Registration No. 3,457,218), and six trademark claims. That proposal builds on Apple's previous proposal to drop from its jury case all claims arising from an additional utility patent, an additional design patent, one registered iPhone trade dress asset, two unregistered iPhone trade dress assets, and two registered trademarks. (Dkt. No. 893 at 1-3.)

These "dropped" claims remain ripe for trial and some should be tried. In particular, the utility patents Apple proposes to drop from its jury trial case (U.S. Patent No. 6,493,002 (status bar), U.S. Patent No. 7,920,129 (touchscreen shielding), U.S. Patent No. 7,812,828 (ellipse-fitting algorithms to interpret touches), and U.S. Patent No. 7,853,891 (timed window)) and Apple's icon-related trademarks are discrete assets that do not duplicate the assets remaining in Apple's jury trial case. Discovery on these claims is essentially complete. Accordingly, Apple requests that the Court consider a motion to bifurcate those claims and set them for a bench trial on the earliest possible date following a July 30, 2012 jury trial. If the Court is willing to entertain such a motion, Apple will waive its claims for damages as to those claims and seek solely injunctive relief.

In the alternative, if the Court is not willing to entertain a motion to bifurcate for Court trial, Apple requests that the Court dismiss without prejudice all claims based on patents, trade dress, and trademarks that Apple proposes to drop from its jury case.

4. Samsung Should be Required to Make a Correspondingly Substantial Reduction in the Scope of its Claims

Pursuant to the Court’s direction, lead counsel met in person on Thursday afternoon to meet and confer over reduction proposals. On Saturday morning, May 5, 2012, Apple sent Samsung an initial version of its proposed reductions. Despite two requests, Samsung had not provided a substantive response as of 5:30 pm on Monday, May 6, 2012, the date this report was due. Consequently, Apple has been required to file this proposal without the benefit of knowing Samsung’s position.

In light of Apple’s concessions to resolve the Court’s concerns regarding the breadth of material to be covered at trial, Apple respectfully requests that Samsung be held to the same standard if *its* case is to be tried to the jury beginning on July 30. Specifically, Samsung should be limited to not more than four utility patent claims, the same number Apple will be presenting to the jury. Obviously, Apple is not willing and should not be required to waive any right to a jury trial on claims and defenses that arise from Samsung’s continued assertion of patents that

Samsung contends are essential to practice the UMTS telecommunication standard, including Apple's Twenty-Fifth through Twenty-Ninth Counterclaims in Reply.

Dated: May 7, 2012

HAROLD J. MCELHINNY (CA SBN 66781)
hmcelhinny@mofo.com
MICHAEL A. JACOBS (CA SBN 111664)
mjacobs@mofo.com
JENNIFER LEE TAYLOR (CA SBN 161368)
jtaylor@mofo.com
ALISON M. TUCHER (CA SBN 171363)
atucher@mofo.com
RICHARD S.J. HUNG (CA SBN 197425)
rhung@mofo.com
JASON R. BARTLETT (CA SBN 214530)
jasonbartlett@mofo.com
MORRISON & FOERSTER LLP
425 Market Street
San Francisco, California 94105-2482
Telephone: (415) 268-7000
Facsimile: (415) 268-7522

WILLIAM F. LEE
william.lee@wilmerhale.com
WILMER CUTLER PICKERING
HALE AND DORR LLP
60 State Street
Boston, MA 02109
Telephone: (617) 526-6000
Facsimile: (617) 526-5000

MARK D. SELWYN (SBN 244180)
mark.selwyn@wilmerhale.com
WILMER CUTLER PICKERING
HALE AND DORR LLP
950 Page Mill Road
Palo Alto, California 94304
Telephone: (650) 858-6000
Facsimile: (650) 858-6100

By: *~/s/ Harold J. McElhinny~*

MORRISON & FOERSTER LLP
Attorneys for Plaintiff and Counterclaim-Defendant
APPLE INC.

Exhibit 9

iPad

Vision

Hearing

Physical & Motor Skills

Accessibility Solutions for iPhone and iPad.

You can find a variety of hardware and software products (sold separately) that adapt iPhone and iPad for specific needs. Learn more ▸

Vision

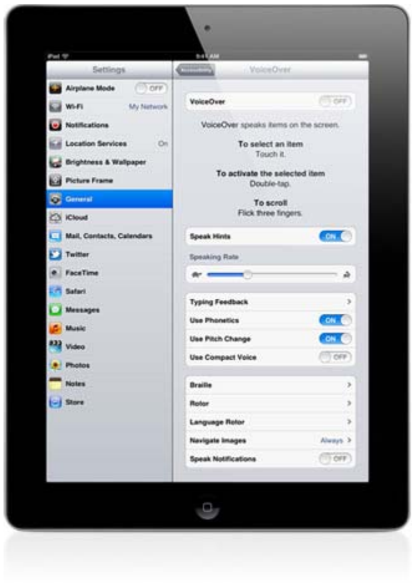
iPad includes an amazing screen reader along with other innovative accessibility features that make it easier to use for those who are blind or have impaired vision.

VoiceOver

The same VoiceOver screen reader available on iPhone comes standard on iPad. It's the world's first gesture-based screen reader, and it allows you to enjoy the fun and simplicity of iPad even if you can't see the screen.

With VoiceOver, you use simple gestures to physically interact with items on the screen. Instead of memorizing key commands or repeatedly pressing arrow keys to find what you're looking for, just touch the screen to hear an item's description, then gesture with a double-tap, drag, or flick to control iPad.

Because VoiceOver on iPad allows you to interact directly with objects, you can understand their location and context. When you touch the upper-left corner of the screen, you hear what's in the upper-left corner of a web page. And as you drag your finger around the screen, you learn what's nearby, providing an unprecedented sense of relationship and context.



VoiceOver

VoiceOver OFF

VoiceOver speaks items on the screen.

To select an item
Touch it.

To activate the selected item
Double-tap.

To scroll
Flick three fingers.

Speak Hints ON

Speaking Rate

Typing Feedback ›
Use Phonetics ON
Use Pitch Change ON
Use Compact Voice OFF

Braille ›
Rotor ›
Language Rotor ›
Navigate Images Always ›
Speak Notifications OFF

VoiceOver on iPad also gives you information about your device — including battery level, network signal level, and time of day. It even lets you know when the display changes to landscape or portrait orientation and when the screen is locked or unlocked.

Adjustable speaking rate

The speaking rate in VoiceOver is adjustable so you can set it to a speed that best suits you. VoiceOver uses distinctive sound effects to alert you when an application opens, when the screen is updated, when a message dialog appears, and more. And when VoiceOver is talking, the volume of background sounds and music is automatically lowered, "ducking" under the voice, so you can clearly hear what VoiceOver is telling you.

"App Store. Two new items. Double-tap to open."
"Mail. Seven new items. Double-tap to open."

It speaks your language

VoiceOver includes built-in voices that speak 36 languages:

- Arabic
- Chinese (Cantonese)
- Chinese (China)
- Chinese (Taiwan)
- Czech
- Danish
- Dutch
- English (Australia)
- English (Britain)
- English (Irish)
- English (South African)
- English (United States)
- Finnish
- Flemish (Belgian Dutch)
- French (Canadian)
- French (France)
- German
- Greek
- Hindi
- Hungarian
- Indonesian

- Italian
- Japanese
- Korean
- Norwegian
- Polish
- Portuguese (Brazil)
- Portuguese (Portugal)
- Romanian
- Russian
- Slovak
- Spanish (Mexico)
- Spanish (Spain)
- Swedish
- Thai
- Turkish

Getting started
VoiceOver is built right into iPad so there's nothing extra to purchase or install. All you need is the latest version of iTunes and a Mac or PC. You can activate your iPad and enable VoiceOver without sighted assistance using Setup Assistant. Sighted users can also enable VoiceOver directly on iPad using the Accessibility menu in Settings.

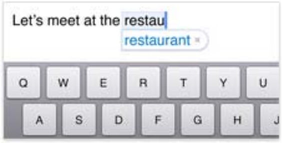
How it works
With VoiceOver enabled, you'll use a different but simple set of gestures to control iPad. For example, instead of tapping to activate a button, tap the button to hear a description of it, double-tap to activate it, and swipe up or down to adjust a slider.

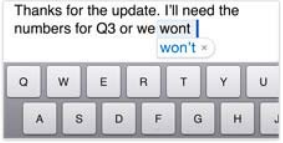
When an item on the screen is selected, a black rectangle called the VoiceOver Cursor appears around it. The VoiceOver Cursor is displayed for the benefit of sighted users with whom you may be sharing your iPad. When you prefer privacy, you can activate a screen curtain to disable the imaging on your display.

In addition to touching and dragging around the screen, you can also flick left and right to move the VoiceOver Cursor to the next or previous item on the screen — no matter how big or small it is. By flicking, you can make precise choices about what you hear even if it's difficult to place your finger on the item.

Entering text
When you're typing text, such as an email message or a note, VoiceOver echoes each character on the keyboard as you touch it, then again to confirm your selection. You can also enable Touch Typing, which automatically enters the last character you hear when you lift your finger. You can even set VoiceOver to speak each completed word instead of — or in addition to — individual characters as you type them. Move the insertion point cursor left or right by flicking up or down within text. With VoiceOver, you can edit a word just as easily and precisely as you can type it in the first place.

To help you type more quickly and accurately, iPad offers word prediction and spelling corrections. With Speak Auto-text enabled, you'll hear a sound effect and the suggested word spoken automatically. Keep typing to ignore the word or press the Space key to have iPad type it for you.


Let's meet at the restau


Thanks for the update. I'll need the numbers for Q3 or we wont

The rotor
VoiceOver features an innovative virtual control called a rotor. Turn the rotor on by rotating two fingers on the screen as if you were turning an actual dial. This gesture changes the way VoiceOver moves through a document based on a setting you choose. For example, a flick up or down might move the cursor through text word by word. But when the character setting is selected, the same gesture will move the cursor through the text character by character — perfect when you're proofreading or editing text.

You can also use the rotor to navigate web pages. When you're on a web page, the rotor contains the names of common items, such as headers, links, form elements, images, and more. You select a setting, then flick up and down to move to the previous or next occurrence of that item on the page, skipping over items in between.

Applications
VoiceOver works with all of the built-in applications that come on iPad, including Safari, Mail, App Store, iTunes, Music, Calendar, and Notes. And with the element

labeler, you can create custom labels for buttons so it's easier to surf the web, email your friends, manage your calendar, download new apps, read books, and more. Apple is also working with iPad software developers to make even more applications VoiceOver compatible. Learn more ▸

iBooks
With the iBooks app (available as a free download), you can download, organize, and read ebooks on your iPad. iBooks is fully compatible with VoiceOver, so you can have books read aloud in any of 33 languages. And you can tailor iBooks to suit the way you read. Read in either portrait or landscape orientation. Choose larger font sizes or different fonts. It also works with the white-on-black text setting. When you want to add new books, visit the iBookstore directly from your iPad and take advantage of VoiceOver to browse the store.

Wireless braille displays
iPad includes built-in support for refreshable braille displays that use Bluetooth wireless technology. You can use them to read VoiceOver output in contracted and non-contracted braille. In addition, braille displays with input keys and other controls can be used to control iPad when VoiceOver is turned on. Learn more about supported braille displays ▸

Zoom

While many iPad applications let you zoom in and out specific elements such as images in Mail or web page columns in Safari, Zoom lets you magnify the entire screen of any application you're using to help you see what's on the display. Zoom can be enabled on iPad using iTunes when you're setting up iPad for yourself or someone else, or later, using the Accessibility menu in the Settings application.

Zoom works everywhere — including the Home, Lock, and Spotlight screens — even in applications that you purchase from the App Store.

Here's how it works. Double-tap with three fingers to instantly zoom in and out 200 percent. Or double-tap and drag three fingers to dynamically adjust the screen's magnification between 100 percent and 500 percent. Even when zoomed in, you can continue using all the iPad gestures you're familiar with — flick, pinch, tap — to run your favorite applications.

White on Black

If you prefer higher contrast, you can change the display on your iPad to white on black. This reverse-video effect works in all applications and on the Home, Lock, and Spotlight screens, and it can be used with Zoom and VoiceOver.

Speak Selection

Want to select text from websites, email, messages, and more? Speak Selection lets you highlight text in any application by double-tapping it. Even if you don't have VoiceOver enabled, Speak Selection will read you the highlighted text and give you formatting options like cut, copy, and paste. Turn on Speak Selection in Settings.

Tactile Buttons

iPad includes a few, easily discernible physical buttons: the Sleep/Wake button, located on the top edge; the Side switch and volume control buttons, located on the upper-right edge; and the Home button, centered below the display.

Large Text

You can increase the font size to see up to 56-point text in alerts, Calendar, Contacts, Mail, Messages, and Notes.

Headset Compatibility

iPad works with a variety of headsets, including Apple earphones and in-ear headphones that have a high-performance microphone capsule built into the cord. Control music playback or record your voice in compatible applications when you click the microphone capsule on your headset.

Audible Alerts

iPad lets you activate audio alerts for incoming and outgoing mail and calendar event requests. iPad also offers an audio option for confirming keyboard actions.

Accessible iPad User Guide

The iPad User Guide has been designed with accessibility in mind. Read the iPad User Guide in HTML format using a web browser with your favorite screen reader on a Mac, PC, and iPad. Or listen to the iPad User Guide in ePub format using VoiceOver in the iBooks app on iPad (iBooks and the user guide can be downloaded at no charge from the App Store and iBookstore, respectively). You can also read the iPad User Guide in tagged PDF format using Preview in OS X and Adobe Acrobat in Windows.

Hearing ›

Exhibit 10

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN JOSE DIVISION

APPLE INC., A CALIFORNIA CORPORATION,	)	C-11-01846 LHK
PLAINTIFF,	)	SAN JOSE, CALIFORNIA
VS.	)	AUGUST 3, 2012
	)	VOLUME
SAMSUNG ELECTRONICS CO., LTD., A KOREAN BUSINESS ENTITY; SAMSUNG ELECTRONICS AMERICA, INC., A NEW YORK CORPORATION; SAMSUNG TELECOMMUNICATIONS AMERICA, LLC, A DELAWARE LIMITED LIABILITY COMPANY,	)	PAGES 556-930
DEFENDANTS.	)	

TRANSCRIPT OF PROCEEDINGS
BEFORE THE HONORABLE LUCY H. KOH
UNITED STATES DISTRICT JUDGE

APPEARANCES ON NEXT PAGE

OFFICIAL COURT REPORTER: LEE-ANNE SHORTRIDGE, CSR, CRR
CERTIFICATE NUMBER 9595

APPEARANCES:

FOR PLAINTIFF
APPLE:

MORRISON & FOERSTER
BY: HAROLD J. MCELHINNY
MICHAEL A. JACOBS
RACHEL KREVANS
425 MARKET STREET
SAN FRANCISCO, CALIFORNIA 94105

FOR COUNTERCLAIMANT
APPLE:

WILMER, CUTLER, PICKERING,
HALE AND DORR
BY: WILLIAM F. LEE
60 STATE STREET
BOSTON, MASSACHUSETTS 02109

BY: MARK D. SELWYN
950 PAGE MILL ROAD
PALO ALTO, CALIFORNIA 94304

FOR THE DEFENDANT:

QUINN, EMANUEL, URQUHART,
OLIVER & HEDGES
BY: CHARLES K. VERHOEVEN
50 CALIFORNIA STREET, 22ND FLOOR
SAN FRANCISCO, CALIFORNIA 94111

BY: VICTORIA F. MAROULIS
KEVIN P.B. JOHNSON
555 TWIN DOLPHIN DRIVE
SUITE 560
REDWOOD SHORES, CALIFORNIA 94065

BY: MICHAEL T. ZELLER
WILLIAM C. PRICE
JOHN B. QUINN
865 SOUTH FIGUEROA STREET
10TH FLOOR
LOS ANGELES, CALIFORNIA 90017

INDEX OF WITNESSES

PLAINTIFF'S

PHILIP SCHILLER

SCOTT FORSTALL

JUSTIN DENISON

(WHEREUPON, PLAINTIFF'S EXHIBIT NUMBER 1046, HAVING BEEN PREVIOUSLY MARKED FOR IDENTIFICATION, WAS ADMITTED INTO EVIDENCE.)

BY MR. MCELHINNY:

Q SIR, CAN YOU TELL US, AT A GENERAL LEVEL, WHAT YOUR INVENTION DEALT WITH?

A SO THIS INCLUDES A LOT OF THINGS. LET ME DESCRIBE ONE EXAMPLE THAT'S COVERED BY THIS INVENTION.

WHEN YOU'RE BROWSING THE WEB IN A WEB BROWSER, LIKE SAFARI, THERE ARE A LOT OF DIFFERENT STORIES AND SO, AGAIN, IMAGINE THE NEW YORK TIMES HOME PAGE. YOU GO THERE AND THERE'S COLUMNS OF STORIES, THERE'S HORIZONTAL STORIES, THERE MIGHT BE AN IMAGE OF A DIFFERENT SIZE, MAYBE A MOVIE, ALL THESE DIFFERENT PIECES OF CONTENT ON THE WEB PAGE.

AND THE ISSUE IS -- WELL, WHAT THIS TRIES TO SOLVE IS TO MAKE A REALLY EASY WAY FOR YOU TO NAVIGATE BETWEEN THOSE DIFFERENT STORIES, THOSE DIFFERENT PIECES OF CONTENT ON THE WEB PAGE.

Q AND HOW -- CAN YOU TELL US HOW IT DOES THAT?

A YES. SO IN THAT EXAMPLE, YOU CAN JUST -- WHAT WE IMPLEMENTED WAS YOU CAN DOUBLE TAP ON ONE OF THE STORIES, AND JUST BY DOUBLE TAPPING ON THE STORY, THE IPHONE ITSELF WILL FIGURE OUT WHICH STORY YOU MEAN AND THEN ZOOM IT UP SO IT SETS THE FONT SIZE RIGHT AND POSITIONS IT AS BEST IT CAN FOR YOU TO READ THAT STORY.

AND THEN YOU CAN TAP, DOUBLE TAP ON ANOTHER STORY NEXT TO IT, AND IT'LL AUTOMATICALLY CHANGE THE SCALING AND MOVE THAT PART OF THE STORY SO YOU CAN READ THAT ONE REALLY WELL ALSO.

SO IT ALLOWS YOU TO REALLY EASILY NAVIGATE THROUGH A WEBSITE JUST BY TAPPING AROUND, OR DOUBLE TAPPING AROUND.

Q CAN WE PLEASE HAVE PLAINTIFF'S DEMONSTRATIVE EXHIBIT 25?

CAN YOU -- CAN YOU TELL US WHAT THIS IS SHOWING, PLEASE?

A SO I THINK THIS MIGHT BE THE NEW YORK TIMES WEBSITE, I CAN'T SEE THE TOP OF IT HERE, BUT IT LOOKS LIKE A WEBSITE WITH A NUMBER OF NEWS STORIES.

CAN WE PLAY IT?

(WHEREUPON, A VIDEOTAPE WAS PLAYED IN OPEN COURT OFF THE RECORD.)

THE WITNESS: SO YOU SEE THERE, HE DOUBLE TAPPED, IT FIGURED OUT THE STORY THE PERSON WANTED TO READ, ZOOMED IN TO SHOW THAT STORY, SO THE STORY LOOKS GREAT.

NOW IF HE DOUBLE TAPS SOMEWHERE ELSE, IT AUTOMATICALLY ZOOMS AND MOVES THAT OTHER STORY OR INFORMATION -- THIS IS SOME STOCK PRICES -- INTO VIEW.

SO IT MAKES IT REALLY SIMPLE FOR YOU TO MOVE AROUND, NAVIGATE AROUND THE WEBSITE JUST BY DOUBLE TAPPING ON WHAT YOU WANT TO SEE.

BY MR. MCELHINNY:

Q HOW DID YOU COME UP WITH THIS INVENTION?

A I REMEMBER AS WE BUILT THE IPHONE, I SPENT A LOT OF TIME USING THE EARLY PROTOTYPES MYSELF. AND I WOULD USE THEM TO SEND ALL MY E-MAIL, TO BROWSE THE WEB, BASICALLY ANYTHING I COULD DO ON THE PROTOTYPE I WOULD DO ON THE PROTOTYPE INSTEAD OF ON A COMPUTER.

AND SO I SPENT A LOT OF TIME BROWSING THE WEB ON THE EARLY PROTOTYPE. IT WAS REALLY COOL, BECAUSE YOU COULD PINCH IN TO STORIES, YOU COULD USE YOUR FINGER TO SCROLL THROUGH THE STORY, MOVE TO ANOTHER STORY AND YOU COULD PINCH OUT.

BUT I FOUND I SPENT A LOT OF TIME CAREFULLY PINCHING A STORY TO BE JUST RIGHT, SO IT WOULD FIT JUST RIGHT WITH THE RIGHT FONT SIZE, AND THEN SCROLLING IT, AND THEN PINCHING THE NEXT STORY TO BE JUST RIGHT, SITUATED EXACTLY WHERE I WANTED

IT ON THE PHONE.

AND I THOUGHT, I'M HOLDING THIS INCREDIBLY POWERFUL DEVICE IN MY HAND, WHY CAN'T IT FIGURE OUT WHAT I KEEP ON DOING OVER AND OVER AGAIN AND JUST DO IT FOR ME?

AND SO I CHALLENGED THE TEAM TO ENABLE YOU TO JUST DOUBLE TAP ON A STORY AND THEN HAVE IT DO THE ZOOM UP AND CENTER IT, SUBSTANTIALLY IN THIS CASE, CENTER IT FOR ME TO READ THAT STORY.

AND THE TEAM AT FIRST THOUGHT THIS IS GOING TO BE SUPER HARD, MAYBE NOT POSSIBLE, AND THEY WENT BACK AND WORKED REALLY HARD AND MADE IT POSSIBLE.

Q   IS THERE ANY RELATIONSHIP BETWEEN THE FEATURES THAT WE'RE SHOWING AND THE ACTUAL STRUCTURE OF THE WEB PAGE ITSELF?

A   YES. THE WEB PAGE, BY ITS NATURE, IS -- IT'S AN ELECTRONIC DOCUMENT, THE WEB PAGE IS, AND WEB PAGES ARE GENERALLY MADE WITH THIS LANGUAGE CALLED HTML, AND HTML DEFINES PIECES OF STRUCTURE.

AND SO WHAT WE IMPLEMENTED WAS WHEN THE USER DOUBLE TAPS ON AN AREA, IT FIGURES OUT THE PIECE OF STRUCTURE THAT IS DEFINED BY THE HTML AND DEFINED BY THE APPEARANCE ON THE SCREEN, DETERMINES THE SORT OF BOX OF STRUCTURE YOU CARE ABOUT, AND

THAT'S THE ONE THAT IT ZOOMS UP FOR YOU.

Q PHYSICALLY, HOW IS THIS FEATURE IMPLEMENTED? HOW DO YOU PUT IT INTO THE PHONE SYSTEM? IS IT A SOFTWARE STRUCTURE?

A IT IS -- THE WHOLE THING IS A PIECE OF SOFTWARE. IT'S BUILT RIGHT INTO THE OS, AND BUILT INTO A NUMBER OF PLACES IN THE OS.

SO YOU CAN USE IT, YOU KNOW, IN THIS EXAMPLE IN A BROWSER. YOU CAN USE IT IN MAIL, SAY YOU RECEIVE DOCUMENTS.

YOU CAN DO IT, I THINK, TO PREVIEW OTHER SORTS OF DOCUMENTS, LIKE PDF'S.

SO IT'S SORT OF BUILT THROUGHOUT THE OS TO NAVIGATE THROUGH STRUCTURED DOCUMENTS.

Q ARE THERE ANY CHALLENGES TO CODING OR PUTTING THIS FEATURE INTO SOFTWARE?

A YEAH, THERE WERE.

UNDERSTANDING THAT STRUCTURE AND, IN FACT, THE STRUCTURE THAT THE USER CARES ABOUT, THAT'S THE CHALLENGE. THAT'S A LOT OF THE CHALLENGE.

YOU CAN IMAGINE A STORY ON THE WEBSITE WHERE THE FIRST LETTER OF THE FIRST WORD OF A STORY IS A BIG CAPITAL LETTER, A GIANT LETTER, AND THE REST OF THE STORY IS SMALLER.

WHEN YOU DOUBLE TAP AROUND THAT FIRST LETTER, THE USER PROBABLY DOESN'T MEAN TO ZOOM INTO ONE LETTER. THEY PROBABLY MEAN TO ZOOM INTO THE COLUMN OF THAT STORY.

SO THERE WAS A LOT OF WORK TO FIGURE OUT WHAT WAS THE CORRECT STRUCTURE AND THE CORRECT BOX THAT THE USER CARED ABOUT AND ZOOM UP THAT CORRECT ONE.

Q IS IT THE GOAL HERE TO MOVE EACH THING THAT THE PERSON WANTS INTO THE EXACT CENTER OF THE SCREEN?

MR. JOHNSON: OBJECTION. LEADING.

THE COURT: SUSTAINED.

BY MR. MCELHINNY:

Q CAN YOU TELL US WHETHER OR NOT IT IS THE GOAL OF THIS TO MOVE THE BOX DIRECTLY INTO THE CENTER OF THE STRUCTURE?

A SO THE GOAL IS TO -- WE CALL IT SUBSTANTIALLY CENTER IT, BUT THE GOAL IS TO MOVE IT TO THE BEST VIEWING PLACE FOR YOU.

IF YOU HAVE A PICTURE WHICH IS SURROUNDED BY A LOT OF OTHER TEXT OR PICTURES AND THAT PICTURE IS IN THE CENTER, WHEN YOU DOUBLE TAP ON THAT PICTURE, IT'LL LIKELY END UP EXACTLY CENTERED ON THE PHONE.

BUT IF YOU HAVE A, A COLUMN OF INFORMATION, LET'S SAY IT'S A COLUMN OF FIRST NAMES AND THEY'RE VERY SHORT NAMES AND IT'S ON THE VERY LEFT-HAND SIDE OF A WEB PAGE, IF YOU DOUBLE TAP ON THAT COLUMN, IT WOULD BE FOOLISH TO CENTER THAT COLUMN IN THE MIDDLE OF THE PHONE BECAUSE YOU LEAVE ALL THIS EMPTY SPACE BEYOND IT TO THE LEFT. YOU'RE JUST WASTING SPACE.

YOU CAN STILL PERFECTLY READ THOSE NAMES LINED UP ON THE EDGE OF THE PHONE AND SEE MORE OF THE WEB PAGE TO THE RIGHT.

SO WE TALKED ABOUT IT AS BEING SUBSTANTIALLY CENTERED, MEANING CENTER IT WHERE IT MAKES SENSE, BUT DON'T GO BEYOND THE EDGE OF A DOCUMENT BECAUSE THERE'S NO REASON TO DO THAT.

Q   SIR, AS THE DEVELOPER OF ALL OF THE IOS SYSTEM, DO YOU CONSIDER THIS A SIGNIFICANT FEATURE IN THE IPHONE?

A   ABSOLUTELY.

Q   AND WHAT IS THE SIGNIFICANCE OF IT?

A   I REMEMBER WHAT IT WAS LIKE TO USE BEFORE WE HAD THIS, WHILE WE WERE DEVELOPING IT, AND AFTER WE IMPLEMENTED THE FEATURE, AND IT DRAMATICALLY CHANGED HOW I USED THE WEB. FOR ME PERSONALLY, IT ENABLED ME TO BROWSE THE WEB MUCH MORE QUICKLY, MUCH MORE FLUIDLY, AND I WOULD THINK TO GO AND BROWSE CERTAIN WEBSITES ON MY PHONE WHERE, BEFORE, I MIGHT HAVE THOUGHT I HAVE TO GO TO A COMPUTER FOR THAT.

AND WE KNOW FROM OUR USERS THAT BROWSING THE WEB IS ONE OF THE MORE POPULAR THINGS THEY DO ON OUR IPHONES, AND ESPECIALLY ON OUR IPADS.

AND SO THIS, I THINK, ENABLES YOU TO HAVE A, A DRAMATICALLY BETTER EXPERIENCE BROWSING THE WEB ON OUR DEVICES.

Q SIR, WAS THE FEATURE THAT'S COVERED BY THIS PATENT, YOUR INVENTION, WAS IT EVER THE SUBJECT OF A SPECIFIC AD THAT APPLE DID FOR ITS PRODUCTS?

A YES. WE MADE A TELEVISION AD SPECIFICALLY TO HIGHLIGHT THIS.

MR. MCELHINNY: YOUR HONOR, AT THIS POINT I WOULD LIKE TO SHOW THAT AD, WHICH IS PART OF PLAINTIFF'S EXHIBIT 12, WHICH IS IN EVIDENCE.

THE COURT: ALL RIGHT. GO AHEAD, PLEASE.

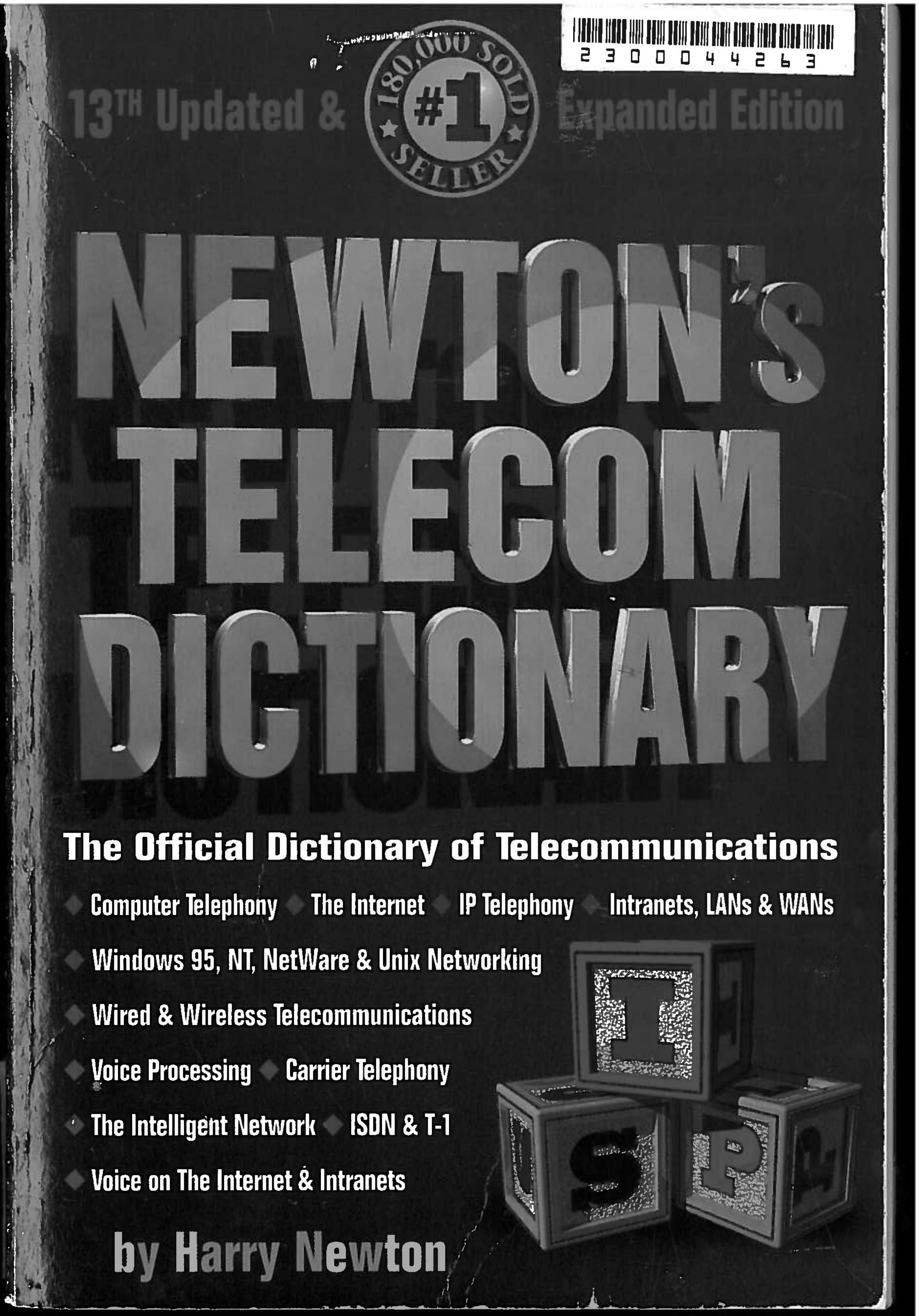
(WHEREUPON, A VIDEOTAPE WAS PLAYED IN OPEN COURT OFF THE RECORD.)

MR. MCELHINNY: THANK YOU, MR. FORSTALL.

I HAVE NO FURTHER QUESTIONS.

THE COURT: ALL RIGHT. THE TIME IS NOW 1:40.

Exhibit 11



13TH Updated & Expanded Edition

NEWTON'S TELECOM DICTIONARY

The Official Dictionary of Telecommunications

- Computer Telephony
- The Internet
- IP Telephony
- Intranets, LANs & WANs
- Windows 95, NT, NetWare & Unix Networking
- Wired & Wireless Telecommunications
- Voice Processing
- Carrier Telephony
- The Intelligent Network
- ISDN & T-1
- Voice on The Internet & Intranets

by Harry Newton

NEWTON's TELECOM DICTIONARY

Published in the United States by
Telecom Books and Flatiron Publishing,
divisions of Miller Freeman, Inc.

12 West 21 Street
New York, NY 10010
212-691-8215 Fax 212-691-1191
1-800-999-0345
1-800-LIBRARY
email: Harry_Newton@FlatironPublishing.com
www.telecombooks.com

ISBN # 1-57820-018-0

January, 1998

Manufactured in the United States of America

Thirteenth Expanded and Updated Edition
Cover Design by Saul Roldan
Printed at Command Web, Secaucus, New Jersey
www.commandweb.com

t of the traffic offered to a group of
izes a server on a LAN.
ich provides communications cir-
o "private" and "common." A pri-
service. A "common" carrier can't.
ndustry — your local phone com-
, etc. — are common carriers.
ated. Private carriers are not.
continuous frequency capable of
rmation. It is the modifications or
ier's basic frequency that become
lodifications are made via ampli-
he process of modifying a carrier
on. A carrier is modulated and
tracted at the other end) according
the wideband (i.e. multi-frequen-
"carriers." T1, which typically has
uits, is known as a carrier system.
of frequencies that can be modu-
n a specific transmission system.

distance phone company provides
n switching office and a customer's
local phone company. Bypass is
r or the long distance company
e to get service faster. Sometimes
imply can't deliver fast enough.
her level circuit (DS-1, DS-3,
) that has been designed to carry
DS-1).
ie Charge Also called Access
long distance phone companies
ies to complete their long distance
illy are much more than the local
ts normal business and residential
phone calls of the same distance.
designed to help the local phone
they do. They also encourage
g term, be self-defeating.
try Electronic components which
rier signal and thus determine if a
ppen. Used in modems.
A proposal for modifying the
ism for Gigabit Ethernet. Under a
evice in the network transmits, the
ger time before another device can
s an Ethernet frame travel a longer
ises the potential network diameter.
e frequency of a carrier wave. The
ed wave capable of being modulat-
ond (information-carrying) signal.
e carrier frequency is also referred
"

n Codes CIC. Three digit num-
stomers to reach the services of
ough equal access arrangements.
ice is reached by dialing "1" plus
rty number. Secondary IX carriers
ng 10 plus the CIC assigned to the
rs are used to dial around the car-
calling telephone (e.g. 10-XXX or
e CIC).
nted carrier remaining after carrier
ed carrier transmission system.

Carrier Liaison Committee CLC. A committee formed to help industry participants work together to resolve the issues of implementing 800 Portability. CLC is sponsored by the Exchange Carriers Standards Association (ECSA) and is comprised of the LECs (local exchange carriers), long distance carriers and users of 800 service.

Carrier Loss In T-1, carrier loss means too many zeros. A carrier loss in T-1 is said to occur when 32 consecutive zeros appear on the network. Carrier is said to return when the next 1 is detected.

Carrier Noise Level The noise level resulting from undesired variations of a carrier in the absence of any intended modulation.

Carrier Power (of a Radio Transmitter) The average power supplied to the antenna transmission line by a transmitter during one radio frequency cycle taken under the condition of no modulation. Does not apply to pulse modulation or frequency-shift keying.

Carrier Provided Loop MCI is responsible for ordering coordinating, maintaining, and billing the local loop.

Carrier Select Keys Buttons at the bottom of a payphone used to choose a long distance carrier.

Carrier Selection As a result of Judge Greene's Modified Final Judgment which lead to the breakup of the Bell System, most local phone companies must offer their customers (business and home) the opportunity to select which long distance company they would like to be use on a "primary" basis. That means when you dial 1+ (one plus) you get that carrier. To use any other long distance company you have to dial more digits, e.g. 1-0288 (for AT&T). See NANP.

Carrier Sense In a local area network, a PC or workstation uses its network card to detect if another station is transmitting. See CSMA.

Carrier Sense Multiple Access CSMA. In local area networking, CSMA is a way of getting onto the LAN. Before starting to transmit, personal computers on the LAN "listen" to make sure no other PC is transmitting. Once the PC figures out that no other PC is transmitting, it sends a packet and then frees the line for other PCs to transmit. With CSMA, though stations do not transmit until the medium is clear, collisions still occur. Two alternative versions (CSMA/CA and CSMA/CD) attempt to reduce both the number of collisions and the severity of their impact. See CSMA/CA and CSMA/CD.

Carrier Sense Multiple Access/Collision Avoidance CSMA/CA. A protocol that requires the PC to sense if another PC is transmitting. If not, it begins transmitting. Under CSMA/CA, a data station that intends to transmit sends a jam signal; after waiting a sufficient time for all stations to pick up the jam signal, it sends a transmission frame; if while transmitting, it detects another station's jam signal, it stops transmitting for a designated time and then tries again.

Carrier Sense Multiple Access/Collision Detection A network control scheme. It is a contention access control scheme. It "listens" for conflicting traffic to avoid data collisions. The Ethernet LAN uses CSMA/CD, then waits a small amount of time and then tries again. See CSMA/CD and ETHERNET.

Carrier Shift 1. A method of keying a radio carrier for transmitting binary data or teletypewriter signals, which consists of shifting the carrier frequency in one direction for a marking signal and in the opposite direction for a spacing signal.

2. In amplitude modulation, a condition resulting from imperfect modulation whereby the positive and negative excursions of the envelope pattern are unequal, thus effecting a change in the power associated with the carrier. There can be positive or negative carrier shift.

Carrier Signal A continuous waveform (usually electrical) whose properties are capable of being modulated or impressed with a second information-carrying signal. The carrier itself conveys no information until altered in some fashion, such as having its amplitude changed (amplitude modulation), its frequency changed (frequency modulation) or its phase changed (phase modulation). These changes convey the information.

Carrier Synchronization In a radio receiver, the generation of a reference carrier with a phase closely matching that of a received signal.

Carrier System A system where several different signals can be combined onto one carrier by changing some feature of the signals transmitting them (modulation) and then converting the signals back to their original form (demodulation). Many information channels can be carried by one broadband carrier system. Common types of carrier systems are frequency division, in which each information channel occupies an assigned portion of the frequency spectrum; and time division, in which each information channel uses the transmission medium for periodic assigned time intervals.

Carrier Terminal The modulation, demodulation and multiplex equipment used to combine and separate individual channels at the ends of a transmission system.

Carrier To Noise Ratio CNR. In radio receivers, the ratio, expressed in decibels, of the level of the carrier to that of the noise in the receiver bandwidth before any nonlinear process such as amplitude limiting and detection takes place.

Carrier Wave The radio frequency wave generated at a transmitting station for the purpose of carrying the modulated or audio frequency wave.

Carrierband Same as single-channel broadband. See also CARRIER BAND.

CARS Cable Television Relay Service Station. A fixed or mobile station used for the transmission of television and related audio signals, signals of standard and FM broadcast stations, signals of instructional television fixed stations, and cablecasting from the point of reception to a terminal point from which the signals are distributed to the public.

Carterfone A device for connecting a two-way mobile radio system to the telephone network invented by Thomas Carter. It was electrically connected to the base station of the mobile radio system. Its electrical parts were encased in bakelite. When someone on the radio wanted to speak on a "landline" (the phone system), the base station operator would dial the number on a separate phone then place the telephone handset on the Carterfone device. The handset was acoustically, not electrically, connected to the phone system. No more than 4,000 Carterfones were ever installed, yet the Bell System thought they were the most dangerous device ever invented. Tom Carter died in Gun Barrel, TX where he lived, in the early part of 1991. He died not a rich man. See CARTERFONE DECISION.

Carterfone Decision In the summer of 1968 the FCC said that the Carterfone and other customer phone devices could be connected to the nation's phone network — if they were "privately beneficial, but not publicly harmful." The Carterfone decision was a landmark. It allowed the connection of non-telephone company equipment to the public telephone network. This decision marked the beginning of the telephone interconnect business as we know it today. The Carterfone decision made a lot of lawyers rich before all the

Exhibit 12

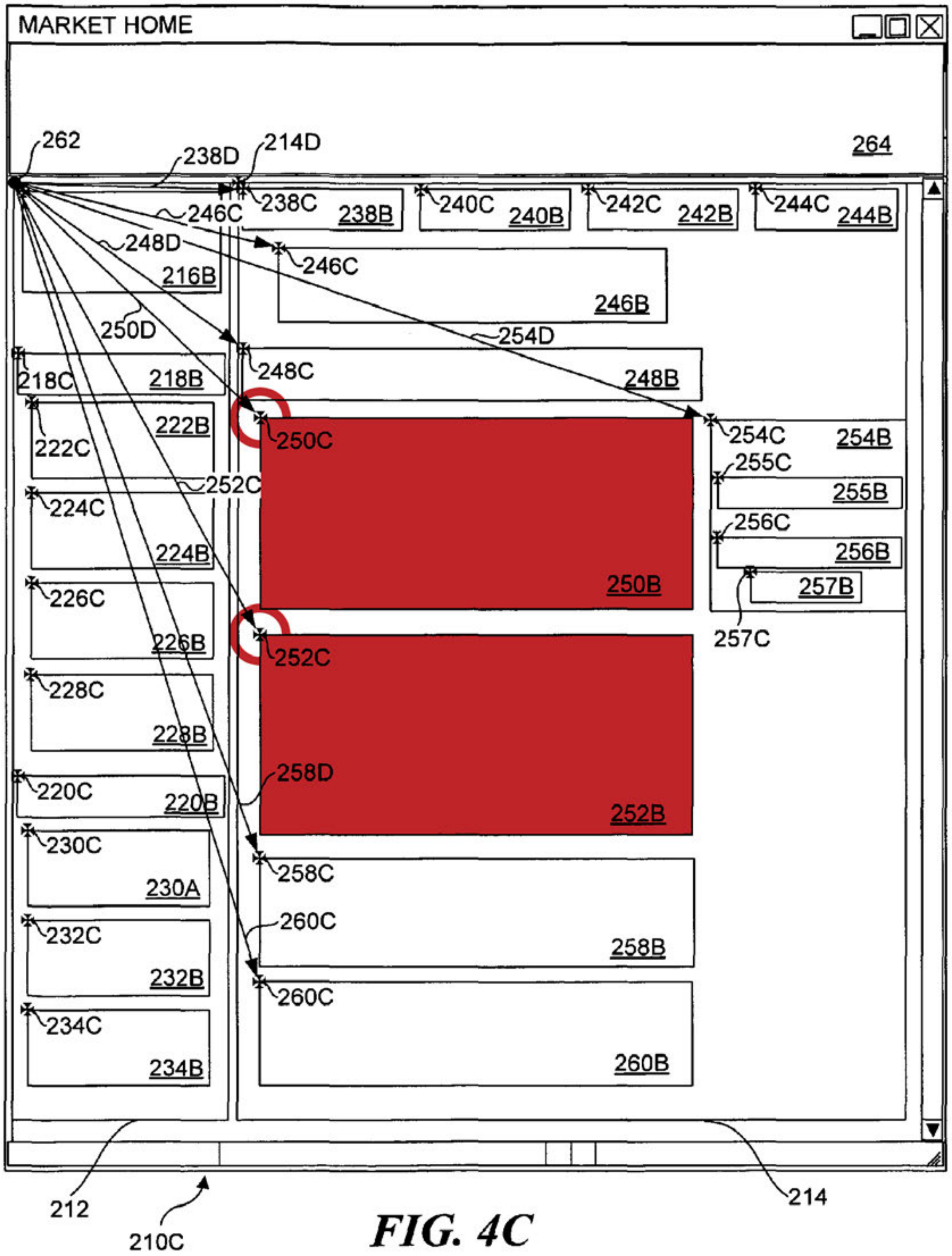

MARKET HOME
262
238D
214D
264
246C
238C
238B
240C
240B
242C
242B
244C
244B
248D
216B
246C
246B
250D
254D
218C
218B
248C
248B
222C
222B
250C
254C
254B
255C
255B
252C
224C
224B
256C
256B
257B
226C
226B
250B
257C
252C
228C
228B
258D
220C
220B
252B
230C
230A
258C
260C
258B
232C
232B
260C
234C
234B
260B
212
210C
214

FIG. 4C

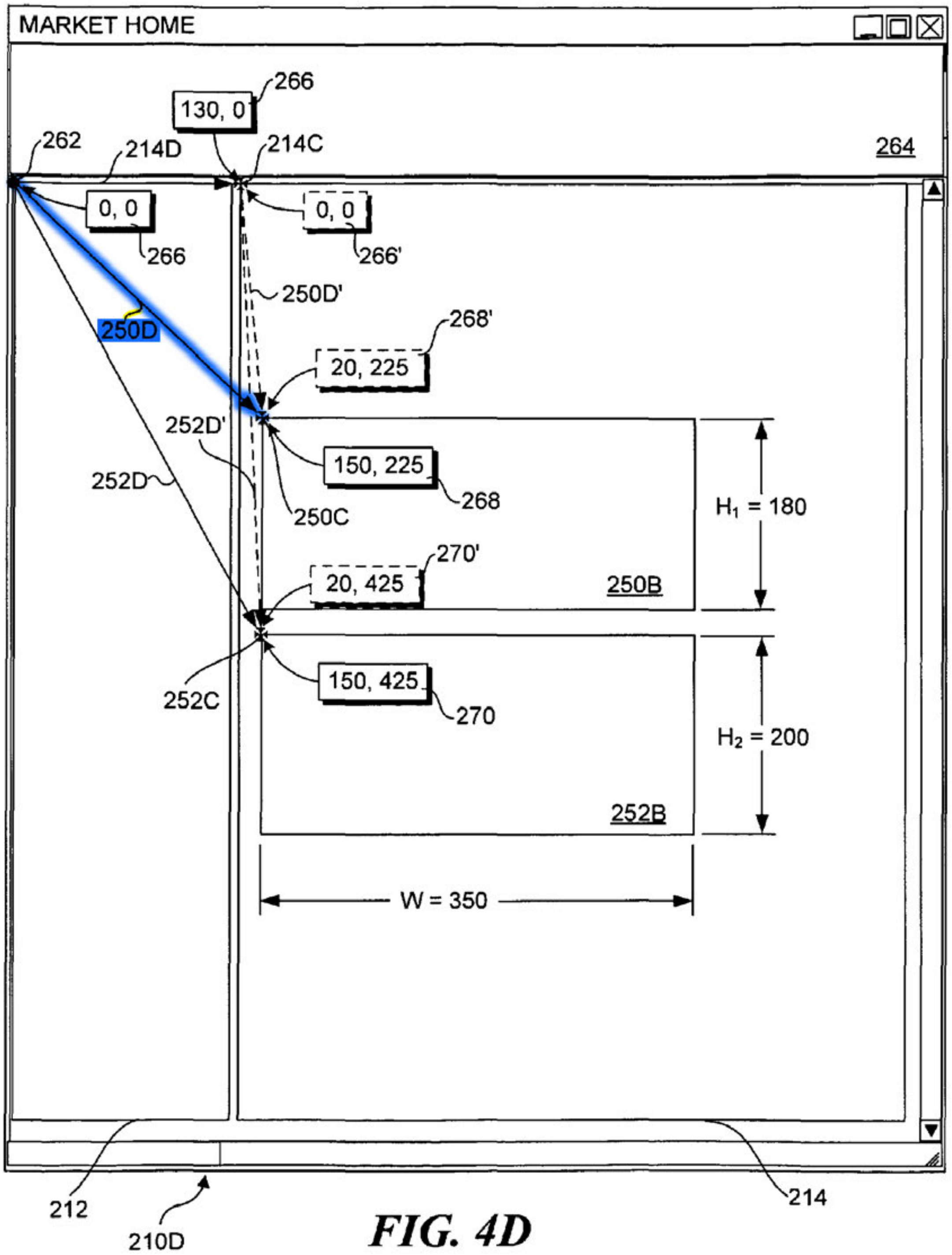

MARKET HOME
266
130, 0
262
214D
214C
264
0, 0
266
0, 0
266'
250D'
250D
268'
20, 225
252D'
150, 225
268
252D
250C
H₁ = 180
270'
20, 425
250B
252C
150, 425
270
H₂ = 200
252B
W = 350
212
210D
214

FIG. 4D

Exhibit 13

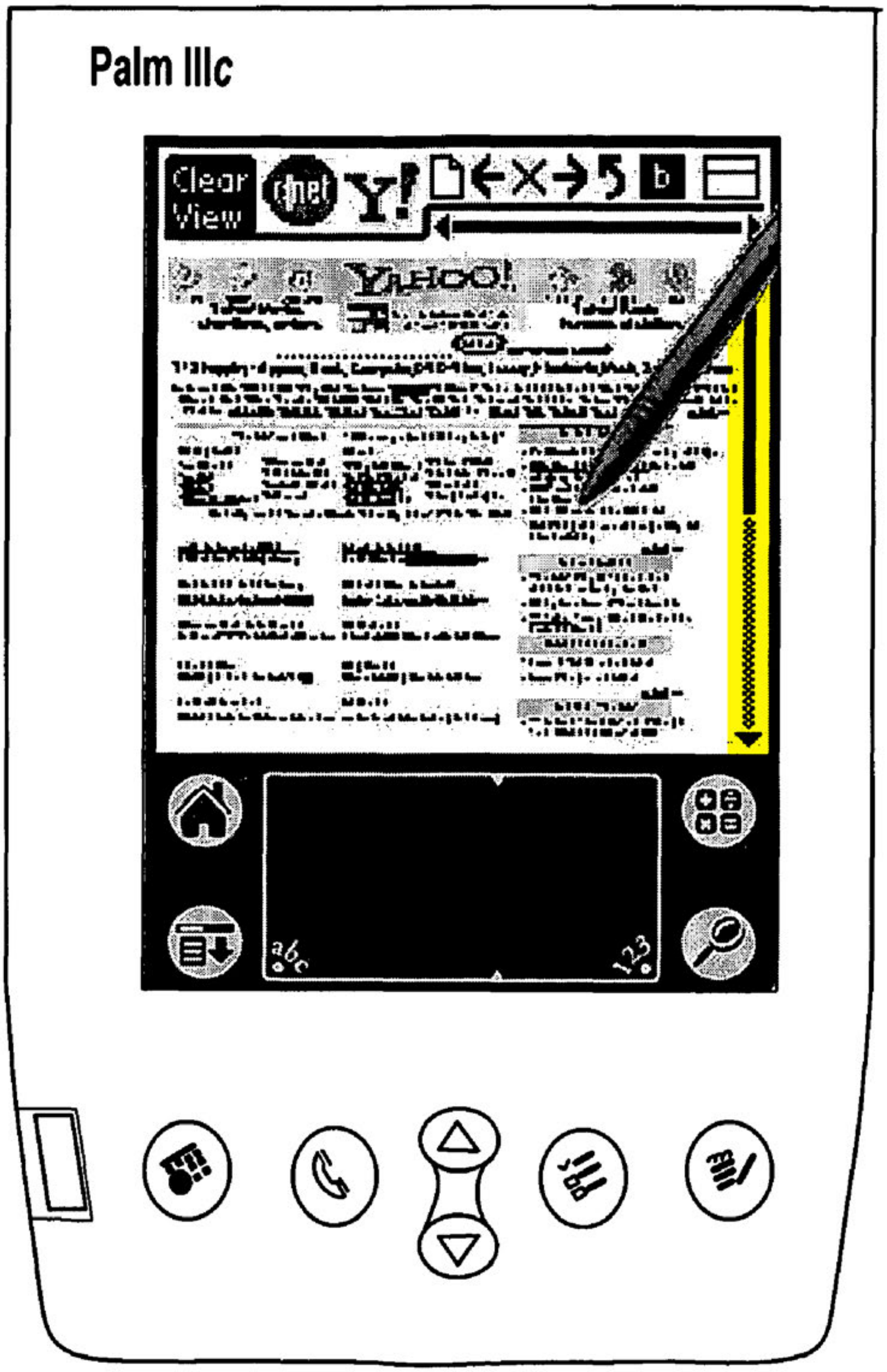


FIG. 7A

Exhibit 14

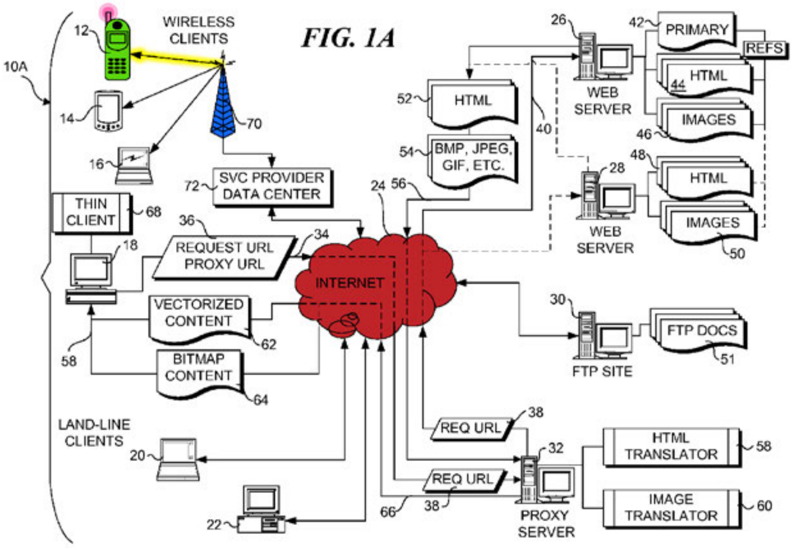


FIG. 1A

10A
12
WIRELESS CLIENTS
14
16
70
72
SVC PROVIDER DATA CENTER
THIN CLIENT
68
18
36
REQUEST URL PROXY URL
34
VECTORIZED CONTENT
62
58
BITMAP CONTENT
64
LAND-LINE CLIENTS
20
22
INTERNET
24
56
52
HTML
54
BMP, JPEG, GIF, ETC.
40
26
WEB SERVER
42
PRIMARY
REFS
HTML
44
IMAGES
46
48
28
WEB SERVER
HTML
IMAGES
50
30
FTP SITE
FTP DOCS
51
REQ URL
38
32
PROXY SERVER
REQ URL
38
66
HTML TRANSLATOR
58
IMAGE TRANSLATOR
60

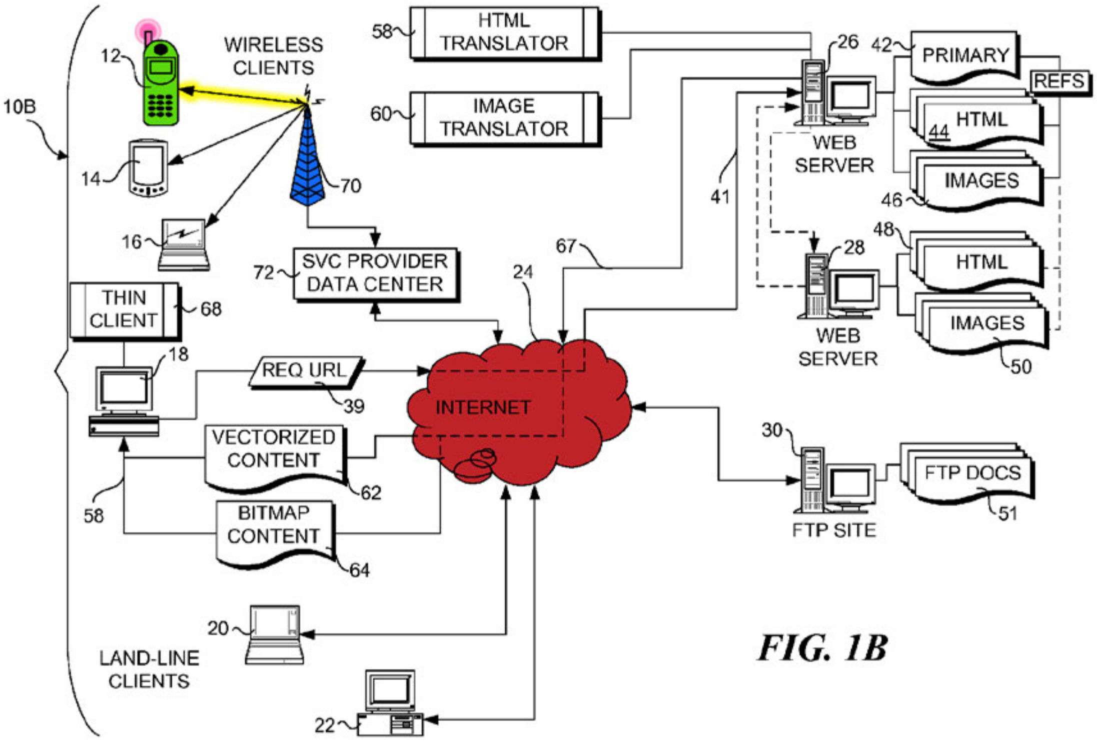

10B
12
WIRELESS CLIENTS
14
16
70
58
HTML TRANSLATOR
60
IMAGE TRANSLATOR
72
SVC PROVIDER DATA CENTER
THIN CLIENT
68
18
REQ URL
39
VECTORIZED CONTENT
62
58
BITMAP CONTENT
64
24
INTERNET
67
41
26
WEB SERVER
42
PRIMARY
REFS
44
HTML
46
IMAGES
28
WEB SERVER
48
HTML
IMAGES
50
30
FTP SITE
FTP DOCS
51
20
22
LAND-LINE CLIENTS

FIG. 1B