IN THE 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 Nos. **10-389-LPS** 12-984-LPS 12-985-LPS 12-986-LPS 12-987-LPS 12-988-LPS 12-989-LPS 12-990-LPS 12-991-LPS

DEFENDANTS' OPENING CLAIM CONSTRUCTION BRIEF

MORRIS, NICHOLS, ARSHT & TUNNELL LLP
Jack B. Blumenfeld (#1014)
Rodger D. Smith II (#3778)
1201 North Market Street
P.O. Box 1347
Wilmington, DE 19899
(302) 658-9200
jblumenfeld@mnat.com
rsmith@mnat.com

Attorneys for Defendant AT&T Mobility LLC and Motorola Mobility LLC

POTTER ANDERSON & CORROON LLP
Richard L. Horwitz (#2246)
David E. Moore (#3983)
Hercules Plaza
6th Floor
1313 North Market Street
Wilmington, DE 19899
(302) 984-6000
rhorwitz@potteranderson.com
dmoore@potteranderson.com

Attorneys for Defendants Apple Inc. and AT&T Mobility LLC

EXHIBIT
Petitioner - Motorola
PX 1026

SHAW KELLER LLP
John W. Shaw (#3362)
David Fry (#5486)
800 Delaware Avenue
4th Floor
Wilmington, DE 19801
(302) 559-9623
jshaw@shawkeller.com
dfry@shawkeller.com

Attorneys for Defendants HTC Corp., HTC America Inc., and Exedea Inc.

RICHARDS, LAYTON & FINGER, P.A.
Frederick L. Cottrell, III (#2555)
Anne Shea Gaza (#4093)
920 North King Street
Wilmington, DE 19801
(302) 651-7700
cottrell@rlf.com
gaza@rlf.com

Attorneys for Defendants Huawei Technologies Co., Ltd. and Futurewei Technologies, Inc.

GREENBERG TRAURIG, LLP
Gregory Erich Stuhlman (#4765)
The Nemours Building
1007 North Orange Street
Suite 1200
Wilmington, DE 19801
(302) 661-7381
stuhlmang@gtlaw.com

Attorneys for Defendants LG Electronics, Inc., LG Electronics USA, Inc., and LG Electronics Mobilecomm USA Inc.

YOUNG, CONAWAY, STARGATT & TAYLOR LLP
Adam Wyatt Poff (#3990)
Monté Terrell Squire (#4764)
The Brandywine Building
1000 West Street
17th Floor
P.O. Box 391
Wilmington, DE 19899
(302) 571-6600
apoff@ycst.com
msquire@ycst.com

Attorneys for Defendants Samsung Electronics Co. Ltd., Samsung Electronics America, Inc., and Samsung Telecommunications America LLC

RICHARDS, LAYTON & FINGER, PA
Steven J. Fineman (#4025)
One Rodney Square
920 North King Street
Wilmington, DE 19801
(302) 651-7700
fineman@rlf.com

Attorneys for Defendant Dell Inc.

PHILLIPS, GOLDMAN & SPENCE, P.A.
John C. Phillips, Jr. (#110)
Megan C. Haney (#5016)
1200 North Broom Street
Wilmington, DE 19806
(302) 655-4200
jcp@pgslaw.com
mch@pgslaw.com

Attorneys for Defendants Kyocera Corp. and Kyocera Wireless Corp.

MORRIS, NICHOLS, ARSHT & TUNNELL LLP
Rodger D. Smith II (#3778)
1201 North Market Street
P.O. Box 1347
Wilmington, DE 19899
(302) 658-9200
rsmith@mnat.com

Attorneys for Defendant Sony Ericsson Mobile Communications (USA) Inc.

September 21, 2012

TABLE OF CONTENTS

TABLE OF AUTHORITIES

CASES **PAGE**

STATUTES

I. INTRODUCTION

U.S. Patent Nos. 7,461,353 and 7,831,926 (the "Patents") claim a certain system and method for allowing web pages to be rendered, zoomed, and panned in a web browser. Defendants' proposed constructions for the disputed terms and phrases of the Patents are consistent with the Patents' disclosures and well-supported by the intrinsic record.[1] In contrast, Plaintiff SoftView's proposed constructions are overbroad, and if adopted would expand the claims to cover unclaimed systems and methods of rendering, zooming, and panning that were already well-known in the field and that SoftView candidly distinguished in its application and during prosecution. Defendants respectfully request that the Court adopt their proposed constructions, as set out below.

II. APPLICABLE LAW

This Court is familiar with the standards for claim construction, as recited in recent opinions. *See, e.g.*, *St. Clair Intellectual Prop. Consultants, Inc. v. Acer, Inc.*, No. 1:09-cv-354-LPS, slip op. at 3-6 (D. Del. Aug. 7, 2012) (D.I. 747, copy attached); *Cadence Pharm., Inc. v. Paddock Labs., LLC*, No. 1:11-cv-733-LPS, slip op. at 1-4 (D. Del. Aug. 22, 2012) (D.I. 188, copy attached). Case law supporting specific proposed constructions is discussed as appropriate

---

[1] The '353 and the '926 patents descend from a common parent and share a nearly identical specification. In this brief, citations to the Patents' specification will be made to the '353 patent for convenience, but are also intended to reference the identical portion of the '926 patent specification. Defendants also rely on various portions of the prosecution and reexamination histories of the '353 and '926 patents and related applications, all of which constitute intrinsic evidence for claim construction. *See Ormco Corp. v. Align Tech., Inc.*, 498 F.3d 1307, 1314 (Fed. Cir. 2007) (prosecution history of other patents in family can inform construction of claims); *Advanced Cardiovascular Sys. v. Medtronic, Inc.*, 265 F.3d 1294, 1305 (Fed. Cir. 2001) (same); *Marine Polymer Techs., Inc. v. HemCon, Inc.*, 672 F.3d 1350, 1365 (Fed. Cir. 2012) (reexamination proceedings are part of intrinsic record and can affect claim interpretation); *St. Clair Intellectual Prop. Consultants, Inc. v. Canon Inc.*, 412 F. App'x 270, 275-76 (Fed. Cir. 2011) (examiner's remarks during reexamination "are relevant prosecution history when interpreting claims").

below.

III. BACKGROUND OF THE TECHNOLOGY AT ISSUE

1. The Patents disclose an alleged invention that translates HTML content into a "scalable vector representation" or "scalable content" for scaling, zooming, and panning web pages.

HyperText Markup Language ("HTML") is a standard, prior art format for defining text, links, and other objects in a web page. The asserted Patents, entitled "Scalable Display of Internet Content on Mobile Devices," share substantially the same disclosures and describe a system and method for translating web pages from a native file format—typically HTML—into a "scalable vector representation," also referred to as "vectorized content," or "scalable content." '353 patent at 1:42-50, 1:61-65, 2:21-34, 6:60-64, 7:38-44, 7:56-8:19, 8:57-61, 9:8-10, 12:12-14, Abstract. According to the Patents, it was familiar in the prior art Computer Aided Design ("CAD") field for graphics in a "vector" format to be magnified and moved around in real time. *Id.* at 4:67-5:17. Likewise, in accordance with the "present invention," the specification alleges that scale factors and offsets may be applied to the translated web content to simplify zooming and panning of the web page, or to more easily scale a page designed for a single, target resolution (typically, a desktop monitor) for display at different sizes or resolutions, such as a small Personal Digital Assistant ("PDA") or a large, "billboard"-style display. *Id.* at 2:4-26, 4:64-5:24, 5:3-24, 9:1-13, 17:42-45, 20:49-67, Figs. 7A-9B.

2. Prior art web browsers laid out web pages in a "pre-rendering" process.

According to the Patents, translating web content from HTML into a scalable vector representation or scalable content includes a "pre-rendering" process that was performed by prior art web browsers. *Id.* at 17:31-34. In the prior art, and still today, the layout of a HTML web page is not typically defined by designating specific coordinate locations for objects on a web page; instead, HTML usually defines layout by spatial relationships between objects (*e.g.*, requiring text to be placed below an image). *Id.* at 16:55-58. In a prior art "layout" process, browsers would retrieve, parse (*i.e.*, separate and identify the constituent parts of the page), and process the HTML to define a page layout based on the location of a "bounding box" for each object on the page. *Id.* at 15:43-16:38, 17:16-30, Fig. 5 (blocks 150-154). In prior art browsers, such as those using the "Mozilla" rendering engine, a data structure called a "render tree" would store, for each object on the page, the X,Y location of the object relative to a previously-defined object, called a "container." *Id.* at 17:16-41; Ex. A, U.S. App. No. 11/868,124, Applicant Remarks at 26-30 (Nov. 24, 2010). The Patents refer to this process as "pre-rendering." *Id.* at 15:43-17:41, Fig. 5 (blocks 150-154). Such prior art browsers would calculate the X,Y location of an object relative to the top-left corner of the page by "walking the render tree," or adding together the stored X,Y coordinates in the render tree for the object, its container, its container's container, and so forth. Ex. A at 27, 29-30; '353 patent at 17:53-56.

3. In the "present invention" of the Patents, a scalable vector representation is generated from the pre-rendering layout information.

The Patents assert "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." '353 patent at 12:12-14, 17:42-45. The Patents state that the translated content may be in a file format known as "Simple Vector Format" or "SVF." *Id.* at 4:50-53, 6:5-11, 6:61-67. The SVF specification dates back to 1995 and is cited prior art (as were other zooming web browsers). *Id.* at p. 2; Ex. B, U.S. App. No. 11/045,757, Info. Disclosure Statement at FH_DEF000168, 211, 220 (Jan. 28, 2005). A prior art SVF file could define displayable elements (*i.e.*, graphics) such as points, lines, text, and web hyperlinks in "vector" form, and these vector based graphical elements could subsequently be scaled, zoomed, and panned with a prior art SVF viewing program. *Id.* at FH_DEF000211.

Generating a scalable vector representation from the pre-rendered layout information begins by defining a datum point for the entire page and additional datum points for each object on the page. *See* ’353 patent at 17:42-18:32, Fig. 5 (blocks 156-160). The page datum for the entire page, or “primary datum,” may be at any point on the page, so long as that point is used consistently in calculating the coordinates of objects on the page. *Id.* at 17:47-56, Fig. 4C (item 262), claim 5. Likewise, the “object datum” may be at any point on an object (*e.g.*, the top-left corner of the bounding box for the object), so long as that location is used consistently across all objects. *Id.* at 17:57-64, Fig. 4C (items ending in “C”).

After datum points are defined, a “vector” for each object is generated from the page datum to each object datum. *Id.* at 17:65-67, Fig. 5 (block 158), Fig. 4C (items ending in “D”). If the page datum is chosen to be at coordinate 0,0, the vector for an object may simply be stored as the X,Y value of that object’s datum point. *Id.* at 17:67-18:8, Fig. 4D. The scalable vector representation is completed by creating a reference that associates an object’s content and attributes to its vector. *Id.* at 18:17-26, Fig. 5 (block 160).

4. The scalable vector representation is used to scale, zoom, and pan.

The scalable vector representation can then be used to scale the web page for displays of various sizes and resolutions and to zoom and pan the page at various user-selectable scaled resolutions and pan offsets. *Id.* at 5:3-24, 9:4-13, 18:47-19:3. According to the Patents, a page can be scaled simply by manipulating the vectors and resizing the bounding boxes for each object to be displayed, and then scaling the content. *Id.* at 19:32-56, 20:18-32, Fig. 6. More specifically, for each object to be zoomed or panned, the vector is offset and has a scale factor applied to it, and the bounding box is scaled by the same scale factor, as shown in Figure 4G of the Patents. *Id.* at 19:57-20:17. Thus, for example, zooming the web page in Figure 4A into the broken rectangle in Figure 4F results in the page displayed in Figure 4E. *Id.* at 3:31-46.

IV. AGREED CLAIM TERMS

As noted in the parties' Joint Claim Construction Chart, the parties do not dispute the constructions of "object[s]" and "bounding box." (D.I. 467 at 2.) Accordingly, Defendants respectfully request that the Court adopt the parties' agreed constructions for these terms.

V. DISPUTED CLAIM TERMS

A listing of the asserted claims in which the disputed claim terms appear can be found in the parties' Joint Claim Construction Chart.[2] (D.I. 467 at 3-12.)

1. Scalable content.

Defendants' Construction	SoftView's Construction
data in a format generated after pre-rendering that provides the layout, functionality, and design of the web page at multiple user-selectable scaled resolutions "pre-rendering": the process of blocks 150-154 of Figure 5 (*see* '353 patent at 15:43-17:41)	content capable of being rendered at multiple zoom levels

Defendants' construction of "scalable content" consists of two primary parts. First, scalable content is "data in a format generated after pre-rendering." Second, scalable content "provides the layout, functionality, and design of the web page at multiple user-selectable scaled resolutions." In contrast to SoftView's overly broad construction, both parts of Defendants' construction are firmly grounded in, and required by, the intrinsic record, including the patentees' own attempts to distinguish their alleged invention from the prior art.

a. Scalable content is data in a format generated after pre-rendering.

The specification dictates that "scalable content" be data in a format generated *after* the

[2] On September 17, 2012, SoftView informed Defendants that it had changed its proposed claim constructions for "scalable content," "vector-based content," and "scalable vector-based content" from those contained in the parties' Joint Claim Construction Chart (D.I. 467). SoftView's proposed constructions contained in Defendants' Opening Claim Construction Brief reflect these changed constructions. Defendants reserve the right to cite additional intrinsic and extrinsic evidence responsive to these changes.

“pre-rendering” process. As discussed, pre-rendering results in layout data, which is later translated into “scalable content” or a “scalable vector representation.” *See* Part III.3, *supra*. Specifically, in describing the “present invention,” the Patents expressly define the key divergence from the prior art as translating the layout information *after* pre-rendering:

> As will be recognized by those skilled in the art, the functions performed in blocks 150, 152, and 154 [of Figure 5] are *commonly performed by conventional browsers* *during* *a pre-rendering process*. . . .
>
> *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.

’353 patent at 17:31-45 (emphasis added). From the outset, the Patents emphasize that the “invention” employs “novel processing of original Web content, including HTML-based content, XML, cascade style sheets, etc. to generate scalable content.” *Id.* at 2:23-29.

As identified above, the Patents describe the prior art “pre-rendering” process in Figure 5 as: (1) parsing the HTML (Fig. 5, block 150), (2) defining a “bounding box” for each object (Fig. 5, block 152), and (3) defining the web page layout based on the bounding boxes (Fig. 5, block 154)—and explain that prior art, including Mozilla, performed pre-rendering steps (1)-(3). *See id.* at 15:48-50, 16:19-21, 16:32-33, 17:16-22. Indeed, the patentees admit they used prior art Mozilla to perform these steps in an embodiment of the alleged invention. *Id.* at 17:31-41. Because “the invention” is described as something that happens *after* the process disclosed in the prior art, “scalable content” necessarily refers to data that is created after pre-rendering, which the Patents expressly define as the prior art process of blocks 150-154 of Figure 5.[3] Finally, for

[3] *See TiVo, Inc. v. EchoStar Commc’ns Corp.*, 516 F.3d 1290, 1300 (Fed. Cir. 2008) (“When a patent . . . describes the features of the ‘present invention’ as a whole, this description limits the scope of the invention.” (citation omitted)); *see also Kinik Co. v. Int’l Trade Comm’n*, 362 F.3d 1359, 1365 (Fed. Cir. 2004) (“Claims cannot be construed as encompassing the prior art that was distinguished in the specification . . . .”); *SciMed Life Sys., Inc. v. Advanced Cardiovascular Sys., Inc.*, 242 F.3d 1337, 1341 (Fed. Cir. 2001) (“Where the specification makes clear that the

the reasons set forth in Part V.3, *infra*, scalable content is data in a format.

b. Scalable content must provide the layout, functionality, and design of the web page at multiple user-selectable scaled resolutions.

Scalable content must also provide the layout, functionality, and design of the web page. As an initial matter, it is the original HTML that defines these aspects of the page. *E.g.*, ’353 patent at 7:56-63; *id.* at claim 1 (“HTML-based Web content having an original format *defining . . . an original page layout, functionality, and design*”). However, the alleged invention uses only the *translated* content, not the original HTML, to render, zoom, and pan the page. *See* Part III.3-III.4, *supra*. In allowing the claims to issue, the Patent Office described the allowable subject matter as translating HTML content from its original format, which defines the “layout, functionality, and design of the web page,” into scalable content, which preserves that “layout functionality, and design,” as follows:

> [T]he claimed invention [of the ’353 patent] 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*” . . . . Additionally, the claimed invention *preserves the functionality* of the original HTML web page **after** it has been translated . . . .

Ex. C, U.S. App. No. 11/045,757, Notice of Allowability at 3-4 (Aug. 8, 2008) (emphasis changed).

> [T]he claimed invention [of the ’926 Patent] *takes HTML-based Web content* in its original *page layout* of the web page and *translates* the HTML-based Web content into *“scalable content”* . . . for enabling Web pages in their original HTML-based content form to be accessed via mobile devices, viewed at various zoom levels by zooming in and out views of the Web pages and interacted with via the mobile devices in a manner that *preserves the original page layout*,

---

invention does not include a particular feature, that feature is deemed to be outside the reach of the claims of the patent, even though the language of the claims, read without reference to the specification, might be considered broad enough to encompass the feature in question.”); *O.I. Corp. v. Tekmar Co.*, 115 F.3d 1576, 1581 (Fed. Cir. 1997) (where written description distinguished invention over prior art “smooth-walled” passages, one of skill in the art would conclude that the term “passage” did not encompass a smooth-walled structure).

> *functionality (including preservation of hyperlink functionality), and design of the Web page content (as defined by its HTML-based content)*[.]

Ex. D, U.S. App. No. 11/738,486, Notice of Allowability at 28-29 (June 9, 2010) (emphasis changed).

These statements by the Examiners are consistent with all three preferred embodiments disclosed by the patentees, in which HTML-based web content is translated into "vectorized content," and only the vectorized content (along with graphics not originally part of the HTML), *not* the original HTML content, is sent to a "thin client," such as a cellular phone or PDA, for display. *See* '353 patent at 6:6-9, 6:42-10:55, Figs. 1A-1C at items 52-68. Thus, the scalable content *must* provide the layout, functionality, and design of the web page. Otherwise, as made clear throughout the intrinsic record, the client described in the three disclosed embodiments would not be able to render, scale, zoom, or pan the page on its display.[4]

Finally, according to the specification, the translated content provides the layout, functionality, and design of the web page at multiple user-selectable resolutions. *Id.* at 9:7-10 ("[A] representation of the original web page content may be rendered on the client device's display screen at *various user-selectable scaled resolutions* . . . ." (emphasis added)); *id.* at 2:29-31 ("The *scalable content* and/or data derived therefrom are then employed to enable the Web content to be *rapidly rendered, zoomed, and panned*." (emphasis added)).[5]

---

[4] *See, e.g.*, '353 patent at 4:64-67 ("By working tightly with a server-side content translator, web content *and functionality* can be passed seamlessly to the end user platform . . . ." (emphasis added)); 18:17-21 (generation of "scalable vector representation" is complete only after "a reference is created for each object that includes or links an object's . . . *attributes* . . . to the object's vector" (emphasis added)); Ex. E, U.S. App. No. 11/045,757, Supp. Amendment at 77-78, 90-91 (May 20, 2008) (discussing claim amendments that replaced "layout and attributes" with "layout, functionality, and design," and noting that "attributes" includes web page's functionality (*e.g.*, hyperlink functionality) and design).

[5] *See also* '353 patent at 18:44-45 ("scalable vector content . . . rendered at a user-selectable scale factor"); *id.* at 19:22-23 ("user selectable scale and offset (pan) values"); *id.* at 2:44-45

c. SoftView's overbroad construction is not supported by the intrinsic evidence.

SoftView's construction of "scalable content" is overly broad and contradicts the Patents' disclosures by encompassing content that is generated before or during the "pre-rendering" process, so long as that content is "capable of being rendered at multiple zoom levels." But, as discussed, the patentees expressly disclaimed the "pre-rendering" process as prior art, and further argued that scalable content is what is created *after* "pre-rendering," thereby distinguishing the "present invention."

2. Scalable / scaling / scaled.

Defendants' Constructions	SoftView's Construction
"scaling" / "scaled": These terms have a plain and ordinary meaning and do not need to be construed. However, if the Court chooses to construe these terms, the following constructions should be used: "scaling": setting to a user-selected resolution "scaled": set to a user-selected resolution "scalable": defined in the context of "scalable content" and "scalable vector-based content"	capable of being rendered at multiple zoom levels / rendering at multiple zoom levels / rendered at multiple zoom levels

The terms "scalable," "scaling," and "scaled" do not require construction. First, these terms are readily understandable, and SoftView's proposed constructions do not add meaning or clarify any dispute as to claim scope.[6] Second, in virtually every instance in the asserted claims, the word "scalable" is recited in the context of "scalable content" or other analogous terms that mean more than merely "content capable of being rendered at multiple zoom levels." *See* Part

("Users can also define a window to zoom in on . . . ."); *id.* at 19:25-26 ("[T]he user is enabled to control the zoom (size) . . . of the rendered page.").

[6] *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1314 (Fed. Cir. 2005) ("In some cases, the ordinary meaning of claim language as understood by a person of skill in the art may be readily apparent even to lay judges, and claim construction . . . involves little more than the application of the widely accepted meaning of commonly understood words.").

V.1, *supra*. Construing "scalable" outside the context of "scalable content" would only lead to confusion. SoftView should not be allowed to rely on a construction of "scalable" in the abstract to generalize or otherwise vary the particular meaning of the "scalable content" lexicography in the Patents.[7]

3. Translating.

Defendants' Construction	SoftView's Construction
converting the format of	This term has a plain and ordinary meaning and does not need to be construed. However, if the Court chooses to construe this term, the following construction should be used: converting

The parties do not dispute that "translating" requires "converting." But SoftView's proposed construction is incomplete and overly broad, because, according to the intrinsic evidence, the claimed "translation" is from the "original *format*" of web content, such as HTML, to a "scalable content" format, such as SVF (Simple Vector *Format*). *See id.* at 10:61-11:9 ("The proxy server . . . deliver[s] content in one of the requested *formats* . . . by *translating* . . . content *from a supported original format to SVF or the client bitmap format*." (emphasis added)); *id.* at 8:60-62 ("HTML translator 58 translates HTML, XML, and cascaded style sheet (CSS) layout content into a scalable vector representation, such as SVF. . . . [G]raphic images are converted into a compressed bitmap format . . . .").[8] Indeed, the Patent Office explained that

[7] If the court finds a construction is necessary, Defendants' proposed constructions for "scaling" and "scaled" should be used because the phrase "user-select[ed] resolution" comes directly from the patent specification. *See* '353 patent at 9:8-10, 18:44-45, 19:22-23.

[8] *See also, e.g.*, '353 patent at 10:21-24 ("HTML documents are *translated* into scalable vector representations by HTML translator . . . while the graphic images are *translated* into a compressed *bitmap format* by image translator . . . ." (emphasis added)); *id.* at 10:2-5 ("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)." (emphasis added)); *id.* at 11:3-9 ("The proxy server responds to client content requests by delivering content in one of the requested *formats* . . . by translating upstream content from a supported original *format to SVF or*

"the claimed invention takes **HTML**-based Web content **in its original format** . . . and ***translates*** the HTML-based Web content into '**scalable content**' . . . ." Ex. C at 3-4 (emphasis in original). And, as with HTML, a client would require translated content to be in some format to be recognizable and usable. '353 patent at 8:56-59, 10:2-6, 11:3-9, 18:41-19:31, Figs. 1A-1C (items 52, 58, 62, 68). In the Patents, therefore, "translating" means "converting the format of."

4. Processing [the] HTML-based Web content to produce scalable content.

Defendants' Construction	SoftView's Construction
converting [the] Web content from its HTML format to a scalable content format	processing [the] HTML-based Web content to produce content capable of being zoomed in or out

"Processing" HTML-based web content "to produce" scalable content requires generating scalable content from the web content's original HTML format. In describing the "aspects of *the invention*," the Patents state that the invention "employ[s] novel *processing* of original Web content, including HTML-based content, XML, cascade style sheets, etc. to *generate scalable content*." '353 patent at 2:26-29 (emphasis added); *see TiVo*, 516 F.3d at 1300 ("When a patent . . . describes the features of the 'present invention' as a whole, this description limits the scope of the invention." (citation omitted)). Moreover, "scalable content" must be "in a format." *See* Part V.3, *supra*. Thus, "processing . . . to produce scalable content" involves converting web page content from its original HTML format into a scalable content format.

5. Format.

Defendants' Construction	SoftView's Construction
structure of data in a file (*e.g.*, .htm, .css, .svf, .bmp, .jpg, .gif)	This term has a plain and ordinary meaning and does not need to be construed. However, if the Court chooses to construe this term, the following construction should be used: a particular way that information is encoded

The Patents' claims use "format" to mean a "file format," *i.e.*, the structure of data in a

---

the client bitmap format." (emphasis added)); *id*. at 4:50-54; *id*. at 11:41-46.

file. As discussed in Part III, *supra*, the alleged invention involves processing original internet content to create scalable content, which is contained in file formats such as SVF and compressed bitmap. The patentees consistently describe both the original web content and the generated scalable content in terms of content in a file format. For example, with respect to web content the Patents state:

> 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.

’353 patent at 1:61-66 (emphasis added); *see also id.* at 10:2-8. With respect to scalable content, the Patents state that “HTML translator 58 *translates HTML, XML, and cascaded style sheet (CSS) layout content into a scalable vector representation, such as SVF*. . . . [G]raphic images are converted into a compressed *bitmap format* . . . .” *Id.* at 8:60-65 (emphasis added); *see also id*. at 4:50-54.

Defendants’ proposed construction reflects the meaning of “format” as set forth in the Patents as well as the examples of specific “formats” for web content and scalable content that are discussed throughout the specification.[9] These examples will assist the jury in understanding the Patents because they make clear that certain terms used throughout the Patents are types of formats. Plaintiff’s argument that construction is not necessary should be rejected because it ignores the fact that, as shown above, the Patents use “format” in a specific sense—to mean “the

---

[9] *See* ’353 patent at 1:61-66, 4:50-59 (“[C]urrent browsers must interpret a large and growing number of file types . . . . *SVF was originally designed to handle a superset of the most commonly used file formats* . . . .” (emphasis added)); *id.* at 8:5-18 (web page graphical content “is usually stored in an image file or files . . . . *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.” (emphasis added)); *id.* at 8:60-65, 10:2-8, 10:21-24, 10:61-11:9, 11:41-46.

structure of data in a file (*e.g.*, .htm, .css, .svf, .bmp, .jpg, .gif)."

6. Vector-based content / scalable vector-based content.

Defendants' Construction	SoftView's Constructions
scalable content that includes, for each object, a directional data structure storing X & Y values from a single point for the page or frame to the object	"vector-based content": content that includes one or more vectors "scalable vector-based content": content that (1) is capable of being rendered at multiple zoom levels and (2) includes one or more vectors

Defendants' construction of "scalable vector-based content" reflects that the "scalable content" of this claim limitation is embodied in a "vector" implementation, as the term "vector" is defined by the Patents and in reexamination. The Patents assert that a vector may be stored as the X & Y value of the datum point of an object, relative to the page datum. *See* '353 patent at 17:65-18:8, Fig. 4D. Likewise, in the pending *inter partes* reexamination of the '353 patent, the examiner determined that "vector" as used in the Patents means "[a] directional data structure stores X & Y values from known datum (primary datum) to object bounding box (object datum)[. A] vector between these points is generated for each object (17:65-67)." Ex. F, U.S. Reexam. App. No. 95/000,634, Order Granting Request for *Inter Partes* Reexamination at 15 (Aug. 8, 2011); *see St. Clair*, 412 F. App'x at 276 ("Because an examiner in reexamination can be considered one of ordinary skill in the art, his construction of the asserted claims carries significant weight."). Defendants' proposed construction thus captures these definitions of "vector" as applied to the "scalable content" of this claim limitation.

Because the Patents use the terms consistently, Defendants propose that the construction of "vector-based content" should be the same as that for "scalable vector-based content." The term "vector-based content" appears in the claims of the Patents only as a form of "scalable content." *E.g.*, '353 patent claim 33 ("a portion of the scalable content comprises vector-based content"). The Patents never claim "vector-based content" that is not also "scalable content."

Indeed, the Patents expressly state that a "*scalable* vector representation" is "also referred to herein as '*vectorized* content,'" *id.* at 6:60-64 (emphasis added), and describe generating "vectorized content" for the alleged benefits of a "*scalable* vector representation," such as scaling, zooming, and panning web content. *See id.* at 2:23-26, 4:64-5:24, 6:53-67, 9:4-12. Thus, all "vector-based content" is necessarily "scalable vector-based content," and separate definitions for these terms are unnecessary.

SoftView's proposed constructions are inconsistent with the Patents' disclosures because by calling for simply "content that *includes one or more* vectors," SoftView's constructions would allow, in vector based content or scalable vector-based content, objects that are not associated with any vector, vectors that do not point to objects, and vectors having different origin points. This cannot be correct, because, as discussed, the Patents require a vector from a single point for the page or frame to every object on the page. *See id.* at 17:42-18:32; Part V.9.a, *infra*. Indeed, the scaling and zooming described in the Patents require that all objects on the page be associated with a vector. *See* '353 patent at 19:32-20:6.

7. Vector.

Defendants' Construction	SoftView's Construction
for each object, a directional data structure that stores X & Y values from a single point for the page or frame to the object	A mathematical expression representing a length and a direction in a two-dimensional space. In an X, Y coordinate system, a vector is represented by a value X2, Y2 relative to an origin point, represented by X1, Y1.

Defendants' construction of "vector" is consistent with the Patents and definition of "vector" provided by the reexamination patent examiner, as discussed for "scalable vector-based content." *See* Part V.6, *supra*. Further, the Patents limit "the invention" to the use of vectors that are drawn from a single point for the page or frame. *See* Part V.9.a, *infra*.

By contrast, SoftView's construction is overly broad because it does not require the vector to be "stored." Under SoftView's construction, any use of an X,Y value in a coordinate system could constitute a vector because the X,Y point could "represent" a vector relative to an origin point. However, the Patents require that the vector be "stored." *See* '353 patent at 18:3-5 ("[T]he vector for a given object may be *stored* as the XY value of the datum point of that object . . . ." (emphasis added)); *id.* at 18:8-9 ("In another embodiment, each vector may be *stored* as XY data . . . ." (emphasis added)). Storing the vectors is necessary because the vectors are linked by a reference to an object, *see id.* at 18:17-21, the vectors may be considered separately to avoid unnecessary retrieval of content, *see id.* at 18:26-31, and the vectors may be manipulated to carry out repeated zooming and panning, *see, e.g.*, *id.* at 9:4-13, 19:57-60.

8. Object datum.

Defendants' Construction	SoftView's Construction
X,Y coordinate for an object that is defined across objects at a consistent location on the objects	reference point for an object

As set forth in Defendants' construction, the Patents consistently use the term "object datum" to mean a point that is both (1) defined by an X,Y coordinate, and (2) defined across objects at a consistent location on the objects. First, the Patents' description of "the present invention" requires the use of "object datum" points defined by X,Y coordinates.[10] *See id.* at 17:42-18:16. The specification confirms that an "object datum" is an X,Y coordinate by stating that vectors may be stored as "the XY value of the datum point of that object." *Id.* at 18:3-8. Likewise, Figure 4D depicts an "object datum" in terms of X,Y coordinates (*e.g.*, object datum 250C is defined at 150, 225). Second, the invention requires that "object datum" points be consistent, in that they can be "located any place *on* the object, as long as . . . used *in a*

[10] *See, e.g.*, *Verizon Servs. Corp. v. Vonage Holdings Corp.*, 503 F.3d 1295, 1308 (Fed. Cir. 2007) ("When a patent thus describes the features of the 'present invention' as a whole, this description limits the scope of the invention." (citation omitted)).

predictable manner." *Id.* at 17:57-64 (emphasis added). All "object datum" points depicted in the drawings are defined consistently at the same top-left corner location on the objects. *See id.* Figs. 4C-4D, 4F-4G ("object datum" points designated by an appended "C"). Without this consistency—which could be violated under SoftView's construction—the scaling process would be complicated in a manner not accounted for by the Patents' disclosure. If "object datum" points were not consistent, but rather, could be located at differing places on the objects, the position of the "object datum" on each object would have to be separately scaled relative to the rest of the object, which is nowhere included in the process described in the Patents. *See* Ex. J, Grimes Decl. ¶¶ 17-18.

9. Primary datum.

Defendants' Construction	SoftView's Construction
single point defined at a fixed X,Y coordinate on the full-size web page	an origin point defined at an X,Y coordinate

a. The "primary datum" is a single point.

In describing the *invention*, as opposed to a preferred embodiment, the specification explains that a single datum is defined as the origin point of the vectors. *See id.* at 17:42-18:16. Further, this datum is defined by reference to either the *page* (in one embodiment) or *frame* (in another embodiment) containing the object. *See id.* at 17:46 ("a datum point is defined for the page"); *id.* at 18:8-11 ("[E]ach 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."). Regardless of whether a particular embodiment uses a datum point for the page or the frame, the invention contemplates that there is a single origin point for each object that is "used consistently throughout the process." *Id.* at 17:52-53.

The intrinsic evidence uniformly confirms that the same single datum defined for the page or frame is the "primary datum" from which vectors are generated to the objects. *See, e.g.*, *id.* at 17:65-67, Fig. 5 (blocks 156-158). The specification never discloses or even suggests that vectors may be generated to an object from multiple "primary datum" locations for the page or frame. To the contrary, the starting point of the vector to each object is a single "primary datum" defined for either the page or the frame to which that object belongs.

Figures 4F and 4G make clear that the same singular "primary datum" is the starting point of the vectors even as the user pans and zooms around the page (the dotted line represents panning and zooming). The "primary datum" always remains at the same, single point on the full size web page (point 262' Figures 4F and 4G). *See* Ex. J ¶¶ 19-21.

b. The "primary datum" is fixed on the full-size web page.

The intrinsic evidence also requires that the "primary datum" is fixed on the full-size web page. The specification explains that a vector representation of the web page is generated from initial page layout information as *originally* determined for the page during pre-rendering. *See, e.g.*, *id.* at 17:42-45. That initial page layout information—including the "primary datum"—corresponds to the full-size web page. The web page has not yet been scaled or panned when the "primary datum" is defined and the initial vectors are generated. *See, e.g.*, *id.* Figs. 4C, 5, claim 58; '926 patent claim 72. A scale factor and offsets are applied to vectors to define new locations for objects on the scaled webpage only after vectors have been defined on the basis of coordinates on the full-size web page. *See* '353 patent at 19:57-64. This type of scaling scheme is possible only if the starting point (*i.e.* "primary datum") and endpoints of those initial vectors correspond to coordinates on the full-size web page.

The figures of the Patents further demonstrate that Defendants' proposed construction is correct. In particular, once the web page is scaled (Figs. 4F-4G, *supra*), the "primary datum" (262') appears off the screen, because it remains a fixed point on the full-size web page. Objects on the page are all scaled consistently, by reference to that single fixed point.

SoftView's construction does not require the "primary datum" to be at a fixed point on the full-size web page, and would permit the primary datum—the starting point for the object vectors—to move around the page as the user zoomed or panned. This type of scheme is not contemplated in the Patents and is inconsistent with the description of the invention. *See* Ex. J ¶¶ 19-21.

10. Layout location datum.

Defendants' Construction	SoftView's Construction
point defined at a fixed X,Y coordinate on the full-size web page that corresponds to an object	one or more points corresponding to the location of the object

As discussed above, the page layout information is defined by a coordinate system. *See* Parts V.8-V.9, *supra*. For the same reasons, "layout location datum"—like the other datum points—also has an X,Y value. Moreover, each asserted claim that contains the term "layout location datum" requires "defining an *object datum corresponding to a layout location datum* for the object's associated display content" and then "generating a vector *from the primary datum to the object datum* for the object."[11] By definition then, a "layout location datum" shares the same X,Y coordinate as the endpoint of an object's vector at the beginning of the process (*i.e.*, before the page is scaled). *See, e.g.*, *id.* at 17:67-18:16, claim 58, Figs. 4D, 4G.

Likewise, for the reasons discussed in Part V.9.b, *supra*, an object's "layout location datum" is defined as to the full-size web page, because it is located at the endpoint of the *initial* vector generated in accordance with the full-size page. *See id.* Therefore, although new "object datums" are defined (and given new values) whenever a page is scaled, the "layout location datum" for an object always remains fixed relative to the full-size web page. *See, e.g.*, *id.* at 19:60-64, Fig. 4G.

[11] The asserted claims requiring these elements are: (1) claims 58, 59 (by reference to 58), and 139 (by reference to 138) of the '353 patent; and (2) claim 72 of the '926 patent.

Thus, Defendants' proposed construction is consistent with the page layout and scaling scheme disclosed in the Patents as "the present invention." *Id.* at 17:42-45. By contrast, SoftView's proposed construction is divorced from the claimed invention and ambiguous, as it provides no guidance as to how the "one or more points" correspond to an object's location.

11. Enabling the user to zoom and pan a view of the Web page.

Defendants' Construction	SoftView's Construction
using the scalable content to allow the user to resize and move around the web page	enabling the user to zoom and move around the web page

This claim language requires the use of scalable content when displaying a device-specific view of a web page, as provided by Defendants' construction. From the abstract through the figures through the entire specification, the *only* invention the Patents disclose uses scalable content to resize and move around the web page:

- Abstract: "The *scalable content* . . . are then employed to *enable* the Web content to be rapidly rendered, zoomed, and panned." '353 patent, Abstract (emphasis added).
- Background of the Invention: "The invention relates generally to translation of . . . Web content to scalable vector representation." *Id.* at 1:43-45.
- Brief Summary of the Invention: "The *scalable content* and/or data derived therefrom *are then employed to enable* the Web content to be rapidly rendered, zoomed, and panned." *Id.* at 2:29-31 (emphasis added).
- Detailed Description of the Invention: "[T]he present invention supports a wide variety of clients . . . *[e]ach 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 . . . ." *Id.* at 18:41-45 (emphasis added). "[T]hrough use of the invention's *scalable vector representation* . . . users are *enabled* to view the entire content of billions of existing Web pages using handheld devices in a simple and reasonable way." *Id.* at 18:67-19:3 (emphasis added).

SoftView's proposal ignores the Patents' own summary of the purported invention and each and every disclosed embodiment, and would improperly expand claim 317 of the '353 patent (the only claim that recites this disputed term) to encompass something different and far broader than the purported invention. *See id.* at 6:64-67, 10:22-31, 10:50-56. The Patents' singular disclosure precludes the claims from covering a process that does *not* use scalable content.[12] Indeed, the Patents cite the use of scalable content to distinguish the alleged invention from the prior art: "[T]he present invention deviates substantially from the prior art by . . . generat[ing] a scalable vector representation of the original page content." *Id.* at 17:42-45; *see SciMed*, 242 F.3d at 1343 (holding that "claims should not be read so broadly so as to encompass the distinguished prior art"); *see also supra* n.3.

The Patents' consistent and repeated teaching that the invention's use of scalable content is what "enables" viewing web content on displays of varying sizes or resolutions, *see, e.g.*, '353 patent at 2:26-31, 18:67-19:3, dictates that the "enabling" claim term must mean using the scalable content to achieve the described web-viewing capabilities.

12. Original.

Defendants' Constructions	SoftView's Construction
"original": as viewed on a conventional desktop browser, such as Internet Explorer or Netscape Navigator, when the browser window is set to the full width resolution for which the page was designed "original width and height of the Web page": the width and height of the Web page as viewed on a conventional desktop browser, such as Internet Explorer or Netscape Navigator, when the browser window is set to the full width resolution for which the page was designed "original page layout, functionality, and design": the page layout, functionality, and design of the Web page as viewed on a conventional desktop browser, such as Internet Explorer or Netscape Navigator, when the browser window is set to the full width resolution for which the page was designed	This term has a plain and ordinary meaning and does not need to be construed.

[12] *See SciMed*, 242 F.3d at 1341 ("Where the specification makes clear that the invention does not include a particular feature, that feature is deemed to be outside the reach of the claims of the patent, even though the language of the claims, read without reference to the specification, might be considered broad enough to encompass the feature in question."); *see also Genzyme Corp. v. Transkaryotic Therapies, Inc.*, 346 F.3d 1094, 1099 (Fed. Cir. 2003) (disclosure in the summary of the invention is not a mere preferred embodiment).

"original page layout": the page layout of the Web page as viewed on a conventional desktop browser, such as Internet Explorer or Netscape Navigator, when the browser window is set to the full width resolution for which the page was designed	

Defendants' proposed constructions are not only correct but also necessary, because the "plain meanings" of these terms are in dispute. *See O2 Micro Int'l Ltd. v. Beyond Innovation Tech. Co.*, 521 F.3d 1351, 1361 (Fed. Cir. 2008) (term must be construed if it has more than one plain meaning or its plain meaning would not resolve the parties' dispute). The dispute arises from whether the "original" layout (and other attributes as claimed) refers to a web page as it (1) would be viewed on a conventional desktop browser at full width resolution, or (2) is first rendered on the particular browser specified in the claims. As explained below, the first interpretation, corresponding to Defendants' constructions, is correct. Yet SoftView asserted during prosecution that "original" refers to the second interpretation, *see* Ex. E at 92, as well as the first, *see* Ex. G, U.S. App. No. 11/045,757, Amendment and Resp. to Office Action at 83 (Dec. 9, 2007), thus raising a dispute as to the plain meaning of these terms.

Only Defendants' constructions are consistent with the claim language and intrinsic record. In particular, the claims expressly require that the "original" layout be defined by the HTML-based content *as requested by the browser*. *See, e.g.*, '353 patent claim 1. By necessity, a browser requests HTML content before it determines the layout of the page. *See id.* at 15:45-50 (browser obtains HTML before determining "where to place the various objects on the display page"); *id.* at 17:16-30. Therefore, the "original" layout of the HTML as requested by the browser cannot necessarily be the layout as it will appear in that browser (as SoftView asserted during prosecution), because such layout is not yet determined when the HTML is requested. Instead, "original" must refer to layout as viewed on a conventional desktop browser, which is how the HTML is defined at the time of request—*i.e.*, before it is interpreted by the browser in question. *See id.* at 2:4-5 (alleged benefit of claimed invention is to allow scaling of common web pages "designed for display on *desktop computers with a single target resolution*" for displays having different sizes and resolutions (emphasis added)); *id.* at 18:47-66; Ex. G at 83 (setting forth patentees' assertions that pending claims called for "Web page content in its *original form as stored on (a) Web server(s) and made available for download to conventional desktop browsers*" (some emphasis removed)).

Other aspects of Defendants' constructions have been included for clarity. For example, the specification refers to Internet Explorer and Netscape Navigator as common web browsers at the time of the alleged invention. '353 patent at 19:4-6. Also, "full width resolution" clarifies that the "original" layout is in reference to the entire web page, not just a portion of the page. *See* Ex. H, U.S. App. No. 11/045,757, Supp. Amendment at 66 n.2 (Jan. 12, 2008) ("[T]he rendering engine determines the applicable layout of the Web page in view of the current browser window width.").

13. Fit across.

Defendants' Construction	**SoftView's Construction**
"fit across": fully fill "fit across the display": fully fill the display "rendered to fit across the display": rendered to fully fill the display "displayed to fit across the touch-sensitive display": displayed to fully fill the touch-sensitive display "displayed to fit across at least one of a width and height of a display area of the touch-sensitive display": displayed to fully fill at least one of a width and height of a display area of the touch-sensitive display	This term has a plain and ordinary meaning and does not need to be construed.

Because the parties disagree about whether "fit across" requires the displayed content to fill the entire display (or display area) and the term "fit across" does not specify how much of the display must be filled, construction is needed.

The claims as originally submitted recited the term "fit substantially across."[13] The patent examiner rejected those claims as indefinite because it was impossible to determine exactly what portion of a display would be considered "substantial." *See* Ex. E at 67. The patentees responded to that rejection by amending the claims to delete "substantially," thus relinquishing what might otherwise have been a more expansive meaning of the disputed term. *See Schindler Elevator Corp. v. Otis Elevator Co.*, 593 F.3d 1275, 1285 (Fed. Cir. 2010) ("[A]n amendment that clearly narrows the scope of a claim . . . constitutes a disclaimer of any claim interpretation that would effectively eliminate the limitation or that would otherwise recapture the claim's original scope."). Based on this intrinsic record, the term "fit across" cannot be left open to recapture the subject matter that the applicant disclaimed—*i.e.* content that "substantially" fills the display. Instead, the term should be construed, as Defendants have proposed, to mean fully filling the display.

14. Tapping.

Defendants' Construction	SoftView's Construction
making contact with the display using a stylus	This term has a plain and ordinary meaning and does not need to be construed.

"Tapping" should be construed to require the single disclosed embodiment of the invention: one that uses a stylus. *See Toro Co. v. White Consol. Indus., Inc.*, 199 F.3d 1295, 1301 (Fed. Cir. 1999). The embodiments described in the Patents' specification and figures disclose only an invention that requires use of a stylus. *See* '353 patent at 20:56-59 ("[A] user may select to view a column . . . by tapping that column *with a stylus* . . . . Similarly, the user may select to zoom in on an image by tapping the image *with the stylus*." (emphasis added)).

[13] The modifier "substantial" implies "'approximate,' rather than 'perfect.'" *See Playtex Prods., Inc. v. Procter & Gamble Co.*, 400 F.3d 901, 907 (Fed. Cir. 2005).

"Nowhere in the specification . . . is" contact with the display disclosed "without" a stylus used to make such contact. *Toro*, 199 F.3d at 1301. Accordingly, the claim term "tapping" should be construed to require tapping "with a stylus."

15. Preserve(s) / preserved / preserving / preservation.

Defendants' Construction	SoftView's Construction
These terms are indefinite and therefore cannot be construed.	This term has a plain and ordinary meaning and does not need to be construed.

These terms are indefinite because the intrinsic record leaves insolubly ambiguous the degree to which the claims require page layout be "preserved" when content is scaled. *See, e.g.*, '353 patent claim 1. On one hand, the patentees assert that "preservation" of layout need not be exact, but rather merely "substantial." *See id.* at 2:31-34 ("[T]he rendered displays provide *substantially the same or identical layout* as the original Web page . . . ."); *id.* at 20:64-67 ("[I]n 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."); Ex. G at 63 ("[T]he corresponding scaled page *may not result in an exact scaling* of the Web page content."); Ex. E at 5, 67-68 (assertion by SoftView that deletion of "substantially" from "substantially preserving" did not alter the claim scope). Yet, on the other hand, the patentees never define what degree of preservation constitutes "substantial." *See* Ex. I, U.S. App. No. 11/045,757, Supp. Info. Disclosure Statement Remarks at 17 (Jan. 22, 2008) (conclusory assertion by SoftView that, in the Opera 3.60 prior art reference, page layout was "clearly *not preserved* at many zoom levels" (emphasis added)); Ex. E at 92-93 (open-ended assertion that, "due to rendering limitations *such as* fixed size fonts, renderings of the same page when viewed at different zoom levels may result in small variations, as opposed to an exact scaled version of the same content (as if viewed by a magnifying glass)." (emphasis added)). Likewise, one of ordinary skill would not be able to ascertain, based on the intrinsic record, when scaling has sufficiently "preserved" layout, and when it has not. Ex. J ¶¶ 8-15. Thus, the intrinsic record sets forth no objective basis to determine exactly when layout is sufficiently "preserved," leaving the claims indefinite. *See Datamize, LLC v. Plumtree Software Inc.*, 417 F.3d 1342, 1351 (Fed. Cir. 2005) ("When a word of degree is used the district court must determine whether the patent's specification provides some standard for measuring that degree." (citation omitted)).

16. Machine-readable medium.

Defendants' Construction	SoftView's Construction
a medium for electronic instructions, such as a floppy disk, optical disk, CD-ROM, magneto-optical disk, ROM, RAM, EPROM, EEPROM, magnetic or optical card, flash memory, or carrier wave	The machine-readable medium may include, but is not limited to, floppy diskettes, 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.

The parties dispute whether the Patents include a carrier wave as a type of "machine-readable medium." Defendants' construction, consistent with the intrinsic evidence, proposes that "machine-readable medium" includes a "carrier wave." The Patents' specification expressly discloses that a carrier wave is a type of machine-readable medium that can transfer computer programs (instructions) from a remote computer:

> [T]he 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*.

'353 patent at 4:34-41 (emphasis added). *CCS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 1366 (Fed. Cir. 2002) (A patentee may act as his own lexicographer by "clearly set[ting] forth a definition of the disputed claim term in . . . the specification.").

17. Storage means.

Defendants' Construction	SoftView's Construction
Function: storing a plurality of instructions Corresponding structure: a floppy disk, optical disk, CD-ROM, magneto-optical disk, ROM, RAM, EPROM, EEPROM, magnetic or optical card, flash memory or carrier wave	Function: storing a plurality of instructions Corresponding structure: memory

The parties agree "storage means" is subject to Section 112 ¶ 6 and that the function is "storing a plurality of instructions." Pursuant to Section 112 ¶ 6, the scope of "storage means," is limited to the corresponding structure described in the specification for "storing a plurality of instructions" and equivalents. Both Patents disclose that the corresponding structure is the "machine-readable medium" (discussed in Part V.16, *supra*), stating that "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 . . . ." *Id.* at 4:24-27 (emphasis added). Accordingly, "storage means" should be construed to have the same scope as "machine-readable medium," discussed above.

18. Processing means.

Defendants' Construction	SoftView's Construction
Function #1: 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 Structure corresponding to Function #1: Indefinite, as the specification does not disclose any algorithm for rendering a browser interface. Function #2: retrieving the Web page via the wireless communication means Structure corresponding to Function #2: Indefinite, as specification does not disclose any structure corresponding to "wireless communication means."	Function: processing Corresponding structure: a processor, microcontroller, or logic circuitry

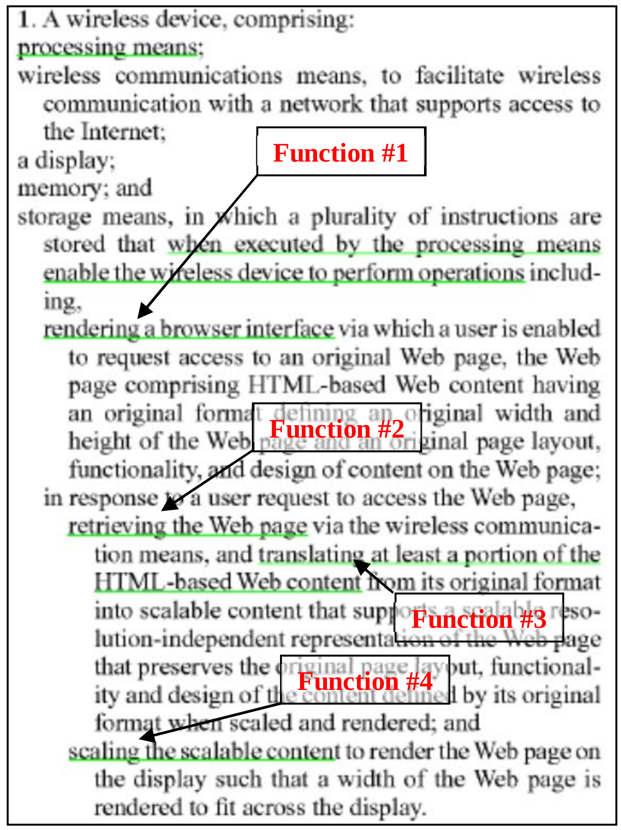
Function #3: translating at least a portion of the HTML-based Web content from its original format into scalable content that supports a scalable resolution independent representation of the Web page that preserves the original page layout, functionality and design of the content defined by its original format when scaled and rendered Structure corresponding to Function #3: The algorithm depicted at boxes 150-160 of Fig. 5 (also depicted at box 114 of Fig. 2C), and ’353 patent at 3:50-52, 10:31-55, 15:43-18:39. Function #4: 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 Structure corresponding to Function #4: The algorithm depicted at boxes 160-172 of Fig. 6, and corresponding text at ’353 patent at 3:53-55, 19:14-20:47, wherein the scale factor is chosen such that the width of the Web page fits across the display.	

The parties agree that this term is governed by Section 112 ¶ 6, but disagree about both the functions and the corresponding structure. SoftView’s proposed construction, that the function of the processing means is “processing” and the corresponding structure is a processor, is inconsistent with the law, which requires disclosure of structure beyond a general purpose processor. In the context of computer-implemented terms, it is well settled that the corresponding structure disclosed in the specification must be a particular algorithm, not simply a general purpose processor.[14]

[14] *Harris Corp. v. Ericsson Inc.*, 417 F.3d 1241, 1253 (Fed. Cir. 2005) (“A computer-implemented means-plus-function term is limited to the corresponding structure disclosed in the specification and equivalents thereof, and the corresponding structure is the algorithm.”); *Aristocrat Techs. Austl. Pty Ltd. v. Int’l Game Tech.*, 521 F.3d 1328, 1333 (Fed. Cir. 2008) (“[S]imply disclosing a computer as the structure designated to perform a particular function does not limit the scope of the claim to ‘the corresponding structure, material, or acts’ that perform the function, as required by section 112 paragraph 6.”). A narrow exception to this rule, not applicable here, is recognized “in the rare circumstances where any general purpose computer without any special programming can perform the function [in which case] an algorithm need not be disclosed.” *Ergo Licensing, LLC v. CareFusion 303, Inc.*, 673 F.3d 1361, 1365 (Fed. Cir. 2012).

The function and structure on which SoftView relies are deficient both for lack of any algorithm and because SoftView would not require that the processor actually perform the claimed functions. The language of claim 1 reveals that the processing means must be capable of several functions. The specification must thus disclose the algorithms used to perform these claimed functions, or the corresponding claim limitations are invalid for failing to satisfy the requirements of Section 112 ¶ 6.[15]

Claim 1 recites, *inter alia*, processing means and storage means for storing instructions that are executed by processing means to perform certain operations—rendering, retrieving, translating, and scaling—that are the functions associated with the processing means. The stored instructions are "executed by the processing means [to] enable the wireless device to perform operations . . . ." '353 patent claim 1.



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.

As noted above, when the function of a means-plus-function term is performed by a general-purpose processor, the corresponding structure is an algorithm. *Aristocrat*, 521 F.3d at 1333. Where multiple functions are claimed, the specification must disclose an algorithm for performing each of the claimed functions. *Noah Sys., Inc. v. Intuit Inc.*, 675 F.3d 1302, 1319

[15] *See, e.g.*, *Aristocrat*, 521 F.3d at 1334-35; *Dealertrack, Inc. v. Huber*, 674 F.3d 1315, 1330 (Fed. Cir. 2012) (claim indefinite because the "specification discloses no algorithm pursuant to which the 'central processing means' could perform the claimed function of 'tracking'").

(Fed. Cir. 2012) ("We cannot allow disclosure as to one function to fill the gaps in a specification as to a different, albeit related, function."). The claim language at issue identifies four functions ("rendering," "retrieving," "translating," and "scaling") of the processing means, but the specification discloses algorithms for only "translating" and "scaling." Because there is no algorithm disclosed for "rendering" or "retrieving," "processing means" is indefinite for failing to comply with Section 112 ¶ 6. *Aristocrat*, 521 F.3d at 1333.

19. Wireless communication[s] means.

Defendants' Construction	SoftView's Construction
Functions (the appropriate function depends upon the specific claim):	Functions:
facilitating wireless communication with a network that supports access to the Internet ('353 patent claim 33 by reference to claim 1)	facilitate wireless communication with a network that supports access to the Internet
facilitating communication with a mobile service provider network via which Web content may be accessed ('926 patent claim 30 and claims 31, 40, 41, and 43 by reference to claim 30)	facilitate communication with a mobile service provider network via which Web content may be accessed
facilitating wireless communication with a network via which Web content may be accessed ('926 patent claims 55, 59, 72, and 75 by reference to claim 52)	facilitate wireless communication with a network via which Web content may be accessed
Corresponding structure: Indefinite, as the specification does not disclose any structures corresponding to this limitation.	Corresponding structure: an antenna

The parties agree that this term is a means-plus-function term within the meaning of Section 112 ¶ 6 and agree on the recited functions (which depend upon the specific claim). The parties dispute the corresponding structure.

SoftView identifies "an antenna" as the corresponding structure. However, the word "antenna" appears nowhere in the Patents' specification. Nor can an antenna alone perform the claimed functions. An antenna is merely a simple conductor (*i.e.* a wire) that forms an output for a radio signal generated by a radio transmitter or an input to direct a signal to a radio receiver.

To enable wireless functionality, an antenna must be accompanied, at a minimum, by a transmitter, receiver, and wireless protocol processing circuitry. There is no disclosure of any such structures in the specification clearly linked or associated with the claimed functions. That one of skill in the art may be able to identify structure for performing to this limitation does not relieve the patentee from the requirements of Section 112. *Ergo*, 673 F.3d at 1364 ("[A] patentee is only entitled to 'corresponding structure . . . *described in the specification* and equivalents thereof,' not any device capable of performing the function."). An antenna alone cannot facilitate the complex communication functions of the claims, and the patentee cannot rely on the skilled artisan to fill in the necessary structure. *Med. Instrumentation and Diagnostics Corp. v. Elekta AB*, 344 F.3d 1205, 1211 (Fed. Cir. 2003) ("If the specification is not clear as to the structure that the patentee intends to correspond to the claimed function, then the patentee has not paid that price but is rather attempting to claim in functional terms . . . ."). Because the only disclosed structure corresponding with the claimed functionality associated with the "wireless communications means" claim elements is an "antenna," and an antenna cannot possibly accomplish those functions, "wireless communications means" is indefinite. *See Ergo*, 673 F.3d at 1363-64 (holding claim invalid where disclosed structure could not perform the claimed function).

VI. CONCLUSION

For the foregoing reasons, Defendants respectfully request that the Court adopt their proposed constructions for the disputed terms and phrases of the '353 and '926 patents.

MORRIS, NICHOLS, ARSHT & TUNNELL LLP

OF COUNSEL:

William H. Boice
KILPATRICK, TOWNSEND & STOCKTON LLP
1100 Peachtree Street
Suite 2800
Atlanta, GA 30309
(404) 815-6500

Daniel S. Young
Laura K. Mullendore
KILPATRICK, TOWNSEND & STOCKTON LLP
1400 Wewatta Street
Suite 600
Denver, CO 80202
(303) 571-4000

/s/ Jack B. Blumenfeld

Jack B. Blumenfeld (#1014)
Rodger D. Smith II (#3778)
1201 North Market Street
P.O. Box 1347
Wilmington, DE 19899
(302) 658-9200
jblumenfeld@mnat.com
rsmith@mnat.com

Attorneys for Defendant AT&T Mobility LLC

POTTER ANDERSON & CORROON LLP

OF COUNSEL:

Josh A. Krevitt
H. Mark Lyon
Alison Watkins
Stuart M. Rosenberg
Neema Jalali
GIBSON DUNN & CRUTCHER LLP
1881 Page Mill Road
Palo Alto, CA 94304
(650) 849-5300

/s/ Richard L. Horwitz

Richard L. Horwitz (#2246)
David E. Moore (#3983)
Hercules Plaza
6th Floor
1313 North Market Street
Wilmington, DE 19899
(302) 984-6000
rhorwitz@potteranderson.com
dmoore@potteranderson.com

Attorneys for Defendants Apple Inc. and AT&T Mobility LLC

SHAW KELLER LLP

OF COUNSEL:

Leo L. Lam
Ashok Ramani
Justina K. Sessions
KEKER & VAN NEST LLP
633 Battery Street
San Francisco, CA 94111
(415) 391-5400

/s/ David Fry

John W. Shaw (#3362)
David Fry (#5486)
800 Delaware Avenue
4th Floor
Wilmington, DE 19801
(302) 559-9623
jshaw@shawkeller.com
dfry@shawkeller.com

Attorneys for Defendants HTC Corp., HTC America Inc., and Exedea Inc.

RICHARDS, LAYTON & FINGER, P.A.

OF COUNSEL:

Eliot D. Williams
Brian Boerman
BAKER BOTTS LLP
30 Rockefeller Plaza
New York, NY 10112
(212) 408-2500

/s/ Anna Shea Gaza

Frederick L. Cottrell, III (#2555)
Anne Shea Gaza (#4093)
920 North King Street
Wilmington, DE 19801
(302) 651-7700
cottrell@rlf.com
gaza@rlf.com

Attorneys for Defendants Huawei Technologies Co., Ltd. and Futurewei Technologies, Inc.

GREENBERG TRAURIG, LLP

OF COUNSEL:

Herbert H. Finn
Eric J. Maiers
Barry Horwitz
GREENBERG TRAURIG, LLP
77 West Wacker Drive
Suite 3100
Chicago, IL 60601
(312) 456-8400

/s/ Gregory Erich Stuhlman

Gregory Erich Stuhlman (#4765)
The Nemours Building
1007 North Orange Street
Suite 1200
Wilmington, DE 19801
(302) 661-7381
stuhlmang@gtlaw.com

Attorneys for Defendants LG Electronics, Inc., LG Electronics USA, Inc., and LG Electronics Mobilecomm USA Inc.

YOUNG, CONAWAY, STARGATT & TAYLOR LLP

OF COUNSEL:

Christine S. Haskett
Michael K. Plimack
COVINGTON & BURLING LLP
One Front Street
San Francisco, CA 94111
(415) 591-6000

/s/ Adam Wyatt Poff

Adam Wyatt Poff (#3990)
Monté Terrell Squire (#4764)
The Brandywine Building
1000 West Street
17th Floor
P.O. Box 391
Wilmington, DE 19899
(302) 571-6600
apoff@ycst.com
msquire@ycst.com

Attorneys for Defendants Samsung Electronics Co. Ltd., Samsung Electronics America, Inc., and Samsung Telecommunications America LLC

RICHARDS, LAYTON & FINGER, PA

OF COUNSEL:

James W. Morando
Cathleen G. Garrigan
Erik C. Olson
FARELLA BRAUN & MARTEL LLP
Russ Building
235 Montgomery Street
17th Floor
San Francisco, CA 94104
(415) 954-4400

/s/ Steven J. Fineman

Steven J. Fineman (#4025)
One Rodney Square
920 North King Street
Wilmington, DE 19801
(302) 651-7700
fineman@rlf.com

Attorneys for Defendant Dell Inc.

PHILLIPS, GOLDMAN & SPENCE, P.A.

OF COUNSEL:

Eric C. Cohen
Michael A. Dorfman
KATTEN MUCHIN ROSENMAN LLP
525 West Monroe Street
Chicago, IL 60661
(312) 902-5200

/s/ Megan C. Haney

John C. Phillips, Jr. (#110)
Megan C. Haney (#5016)
1200 North Broom Street
Wilmington, DE 19806
(302) 655-4200
jcp@pgslaw.com
mch@pgslaw.com

Attorneys for Defendants Kyocera Corp. and Kyocera Wireless Corp.

MORRIS, NICHOLS, ARSHT & TUNNELL LLP

OF COUNSEL:

William H. Boice
Candice C. Decaire
Richard Goldstucker
KILPATRICK, TOWNSEND & STOCKTON LLP
1100 Peachtree Street
Suite 2800
Atlanta, GA 30309
(404) 815-6500

/s/ Jack B. Blumenfeld

Jack B. Blumenfeld (#1014)
1201 North Market Street
P.O. Box 1347
Wilmington, DE 19899
(302) 658-9200
jblumenfeld@mnat.com

Attorneys for Defendant Motorola Mobility LLC

MORRIS, NICHOLS, ARSHT & TUNNELL LLP

OF COUNSEL:

John H. McDowell, Jr.
Benjamin J. Setnick
ANDREWS KURTH LLP
1717 Main Street
Suite 3700
Dallas, TX 75201
(214) 659-4400

/s/ Rodger D. Smith II

Rodger D. Smith II (#3778)
1201 North Market Street
P.O. Box 1347
Wilmington, DE 19899
(302) 658-9200
rsmith@mnat.com

Attorneys for Defendant Sony Ericsson Mobile Communications (USA) Inc.

September 21, 2012

CERTIFICATE OF SERVICE

I hereby certify that on September 21, 2012, I caused the foregoing to be electronically filed with the Clerk of the Court using CM/ECF, which will send notification of such filing to all registered participants.

I further certify that I caused copies of the foregoing document to be served on September 21, 2012, upon the following in the manner indicated:

Steven L. Caponi, Esquire *VIA ELECTRONIC MAIL*
BLANK ROME LLP
1201 North Market Street, Suite 800
Wilmington, DE 19801
Attorneys for Plaintiff

Morgan Chu, Esquire *VIA ELECTRONIC MAIL*
Samuel K. Lu, Esquire
Amir Naini, Esquire
Craig Johnson, Esquire
IRELL & MANELLA LLP
1800 Avenue of the Stars, Suite 900
Los Angeles, CA 90067-4276
Attorneys for Plaintiff

/s/ Jack B. Blumenfeld

Jack B. Blumenfeld (#1014)

EXHIBIT A

REMARKS

This amendment is made in response to the Office Action mailed August 24, 2010. In the Office Action, the drawings were objected to for failing to comply with 37 C.F.R. 1.84(p)(5) for including improper reference characters. Claims 1-25 were rejected on the ground of nonstatutory double patenting over claims of U.S. Patent Nos. 7,461,353 and 7,584,423, and provisionally rejected over the claims of co-pending Application Nos. 11/735,477 and 11/738,486. Claims 1-14 and 20-25 were rejected under 35 U.S.C. § 102(a) as being anticipated by *WEST*: A Web Browser for Small Terminals by Staffan Björk et al. (Viktoria Institute, Göteborg, Sweden, 1999) hereafter *WEST*. Claims 15-19 are rejected 35 U.S.C. § 103(a) as being unpatentable over *WEST*, further in view of The Zoom Browser by Lars Erik Holmquist (Åter till Human IT, 3/1998), hereinafter *Holmquist*. Claims 15-19 were also rejected under 35 U.S.C. § 112, second paragraph, as being indefinite.

Claims 1, 3, 6-8, 10, 12, 15-19, 21, 23, and 25 are amended herein. In particular, independent claims 1 and 15 are amended to more clearly recite elements of their claimed inventions. Claims, 9, 11, 13, 20, 22, and 24 have been canceled. New claims 26-37 have been added. Accordingly, claim 1-8, 10, 12, 14-19, 21, 23, and 25-37 are now pending. In view of the amendments and following remarks, Applicants respectfully assert all claims are in condition for allowance.

<u>Correction of Application Number in Specification</u>

Applicants thank the Examiner for identifying the inadvertent error in identification of U.S. Non-provisional Application No. 09/82**5**,511 in paragraph [0001] of the specification. The application number has been corrected to 09/82**8**,511.

<u>Correction to Drawings</u>

Applicants thank the Examiner for identifying the inadvertent errors in the drawings. Amendment has been made to paragraph [0063] to specifically reference blocks 117, 119, and 121 in FIG. 2C. Reference numbers 32A and 34A have been respectively corrected to 232A and 234A in FIG. 4A. Reference numbers 268" and 270" have been respectfully corrected to 268' and 270' in FIG. 4G. No new matter has been added. Replacement sheets for FIG. 4A and FIG. 4G are submitted herewith.

Discussion of new claim terminology and operations of exemplary embodiments of the invention reduced to practice

In reviewing the above-identified rejections, Applicants believe that the Examiner did not fully comprehend the scope of the original claims. Moreover, Applicants assert the original claims were clearly patentable over the cited references. However, Applicants have chosen to amend the claims herein to provide further clarification to the intended scope of the claims, and to differentiate the claimed inventions from inventions claimed in related patents and applications. Additionally, certain claim elements are discussed in detail below so that their respective claim scope is more clearly understood. Arguments in support of patentability then follow.

Scalable or Scaled Page Layout Information

The terminology "scalable page layout information" is recited in independent claim 1, and an operation comprising "scaling page layout information ..." is recited in independent claims 15 and 30. In order to clarify the scope of this claim terminology, the following discussion is provided.

Independent Claim 1– "scalable page layout information"

Independent claim 1 recites, in part,

> ...
>
> retrieving ... HTML-based Web content associated with the Web page;
>
> *processing, the HTML-based Web content with a rendering engine to generate page layout information corresponding to the original page layout of content on the Web page as interpreted by the rendering engine;*

employing the page layout information to generate ***scalable page layout information****; and*

employing the ***scalable page layout information*** *and/or content derived therefrom to,*

...

The terminology "scalable page layout information" refers to page layout information (that is information used to layout the Web page content) that includes layout objects that are mapped to a virtual coordinate space[1], as discussed in further detail below. The claim element of "*processing, the HTML-based Web content with a rendering engine to generate page layout information corresponding to the original page layout of content on the Web page as interpreted by the rendering engine*" does not restrict the element to a single process or operation, but may include multiple operations. Similarly, the claim element of "*employing the page layout information to generate scalable page layout information*" may be implemented by one or more operations, and may be implemented directly or indirectly, as discussed below.

In accordance with one or more embodiments disclosed in the present application, the processing of the retrieved HTML-based content begins by employing a rendering engine to determine page layout information that includes information from which the location for each displayed object[2] (*i.e.*, HTML element or block having content that is to be displayed) can be determined. In general, the page layout information includes information from which at least one of a datum or bounding box for each object can be determined. In one embodiment, the page layout is interpreted

---

[1] The usage here of "virtual coordinate space" is not to be limiting. In general, a virtual coordinate space may also be commonly referred to as a virtual drawing environment, virtual coordinate system, logical coordinate system, or a virtual drawing system.

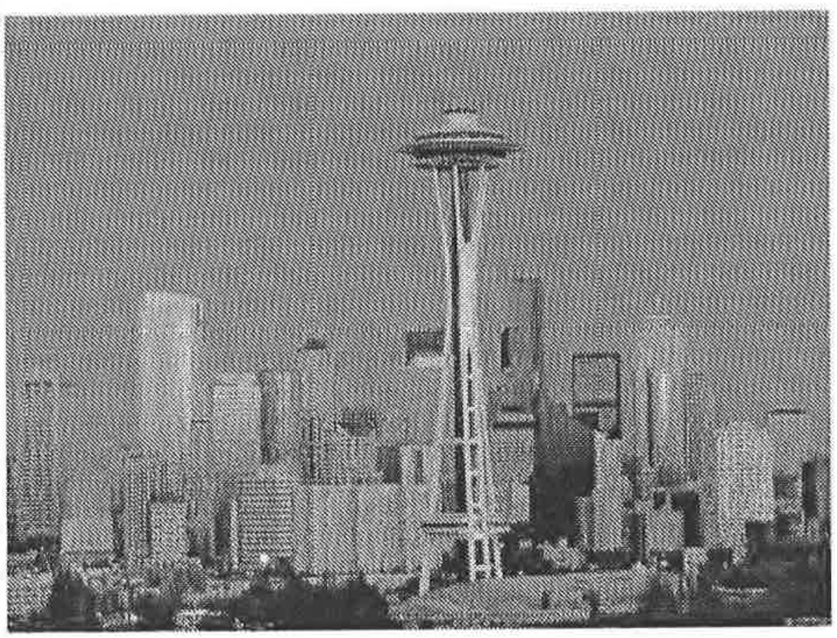
[2] Page layout information may also include information corresponding to objects that are not displayed in rendered content, such as layout container objects, as described below.

using the rendering engine based on a default page width in pixels, and the location of the page layout objects is interpreted by the rendering engine are defined by corresponding pixel locations on the page. Accordingly, at this point the page layout information, such as a datum, is defined by corresponding pixel locations in a two-dimensional (XY) pixel-based coordinate space used by the rendering engine. The page layout information is further processed to generate Simple Vector Format (SVF) drawing commands to map the page layout information from the pixel-based coordinate system to a vector-based virtual coordinate space used by the SVF vector-based drawing model. Under the SVF vector-based drawing model, each coordinate point comprises a vector (*i.e.*, a point vector), and paths for drawing content are defined using vector-based definitions (*i.e.*, a mathematical description of the path using vectors). As such, the vector for each coordinate point can be scaled by applying a scale factor to the point coordinates, resulting in a transformed point (and corresponding scaled vector). In addition, the virtual coordinate space is resolution independent, wherein a resolution of the coordinates employed by the virtual coordinate space is independent of the resolution of the displays of devices used to display content. Accordingly, the same content defined by corresponding SVF drawing commands can be mapped to displays having different resolutions, and rendered at an applicable resolution for each display.

These operations may more easily be understood by an example corresponding to a simple web page, such as shown below. The exemplary web page includes four objects: a text object comprising the text "Seattle Skyline 300x300"; a JPEG image of the Seattle skyline having an original (full scale) size of 300x300 pixels; a text object comprising the text "Seattle Skyline 640x480"; and a JPEG image of the Seattle skyline having an original (full scale) size of 640x480 pixels. The web page also has an interpreted size of 912 pixels wide by 1026 pixels high in the example figures herein, as discussed in further detail below. The HTML for generating this web page is shown following the image. The web page was designed using Microsoft Expression Web 4 software. As such, the upper portion of the code (between the <style ...> and </style> tags comprises Cascading Style Sheet style definitions. For more complex pages, the CSS style definitions may be contained in one or more separate CSS documents. Style definitions may also be defined inline (*e.g.*, when the definition for a given element, such as a line of text, paragraph, *etc.*). All of this corresponds to conventional HTML and CSS standards, which are well-known in the art.

Seattle Skyline 300x300

Seattle Skyline 640x480



Original Web Page Layout and Design

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<meta content="en-us" http-equiv="Content-Language" />
<meta content="text/html; charset=utf-8" http-equiv="Content-Type" />
<title>Seattle Skyline</title>
<style type="text/css">
.auto-style1 {
	margin-left: 360px;
	margin-top: 0px;
}
.auto-style2 {
	margin-left: 72px;
	margin-top: 0px;
}
.auto-style3 {
	margin-top: 50px;
}
.auto-style4 {
	font-size: x-large;
	font-family: "Lucida Sans", "Lucida Sans Regular", "Lucida Grande", "Lucida Sans Unicode", Geneva,
Verdana, sans-serif;
	margin-left: 144px;
	margin-top: 48px;
}
.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;
}
	.auto-style6 {
		border-style: solid;
		border-width: 1px;
	}

</style>
</head>

   <body style="margin-top: 10px">
	<div class="auto-style6">
		<div class="auto-style5" style="width: 284px">
		Seattle Skyline 300x300</div>
		<p class="auto-style3">
		<img alt="" class="auto-style1" height="300" src="seattle%20300x300.jpg" width="300" /></p>
		<p class="auto-style4" style="width: 284px">Seattle Skyline 640x480</p>
		<p>
		<img alt="" class="auto-style2" height="480" src="Seattle%20640x480.jpg" width="640" /></p>
	<div>
   </body>

</html>
```

Example Web Page HTML

The following figure shows the layout information for the page as interpreted by the Mozilla "Gecko" rendering engine.[3] The rendering engine is part of the Firefox browser (current version 3.6). An earlier version (from approximately 2000-2002) of the Gecko rendering engine is employed for the SoftView™ browser screenshots herein.

It is common practice to lay out web page content using a page origin of 0,0[4] at the upper left hand corner of the page. Accordingly, web page rendering engines such as Gecko typically employ a pixel-based coordinate system having an origin at 0,0 with X values increasing to the right, and Y values increasing downward. The coordinates of the pixel-based coordinate system are integer values corresponding to the location of the pixels in a rendered page.

One of the operations performed during page layout is parsing of the HTML to identify the corresponding HTML elements. For "free-form" pages such as this example, the order of the elements affect the page layout (vertically in this case), as objects are laid out relative to other objects, as opposed to having a fixed position. Commercial pages (*e.g.*, nytimes.com, CNN.com, WSJ.com, *etc.*) typically employ one or more cascading style sheets to define a more structured page layout; however, in order to not obscure the page layout concepts taught here a simple web page example is used.

Browser rendering engines typically determine page layout based on the size of the browser window content area (*i.e.*, the portion of the browser application window in which the page content is rendered on a display, such as a desktop computer monitor). In particular, the width of the content area is employed. Depending on the web page design, the width of the content may affect the page layout. In this particular, example, however, there are no objects that are located based on the content area width, as each

---

[3] For further details, see the following section concerning an overview of the SoftView™ browser operation.

[4] Optionally, the page layout datum may be considered at 1, 1, representing the X, Y location of the pixel at the upper left hand corner of a content drawing area used to render page content in a browser.

of the four objects has a defined offset from the left edge of the page, as defined by corresponding HTML elements in the HTML code.

The height of the page is interpreted by aggregating the vertical dimensions of the object layout in the page, as determined by the page layout interpreted by the particular rendering engine used by the browser. Different rendering engines may use different default spacing values and fonts (and/or the default fonts and sizes can be changed by a user), such that the same content may be laid out differently by the different rendering engines. In this particular example, the Gecko rendering engine determined the body block to be 1008 pixels in height. The Firefox browser window width was adjusted such that the body block was interpreted to be 896 pixels wide to simplify the content scaling example herein, as discussed below.

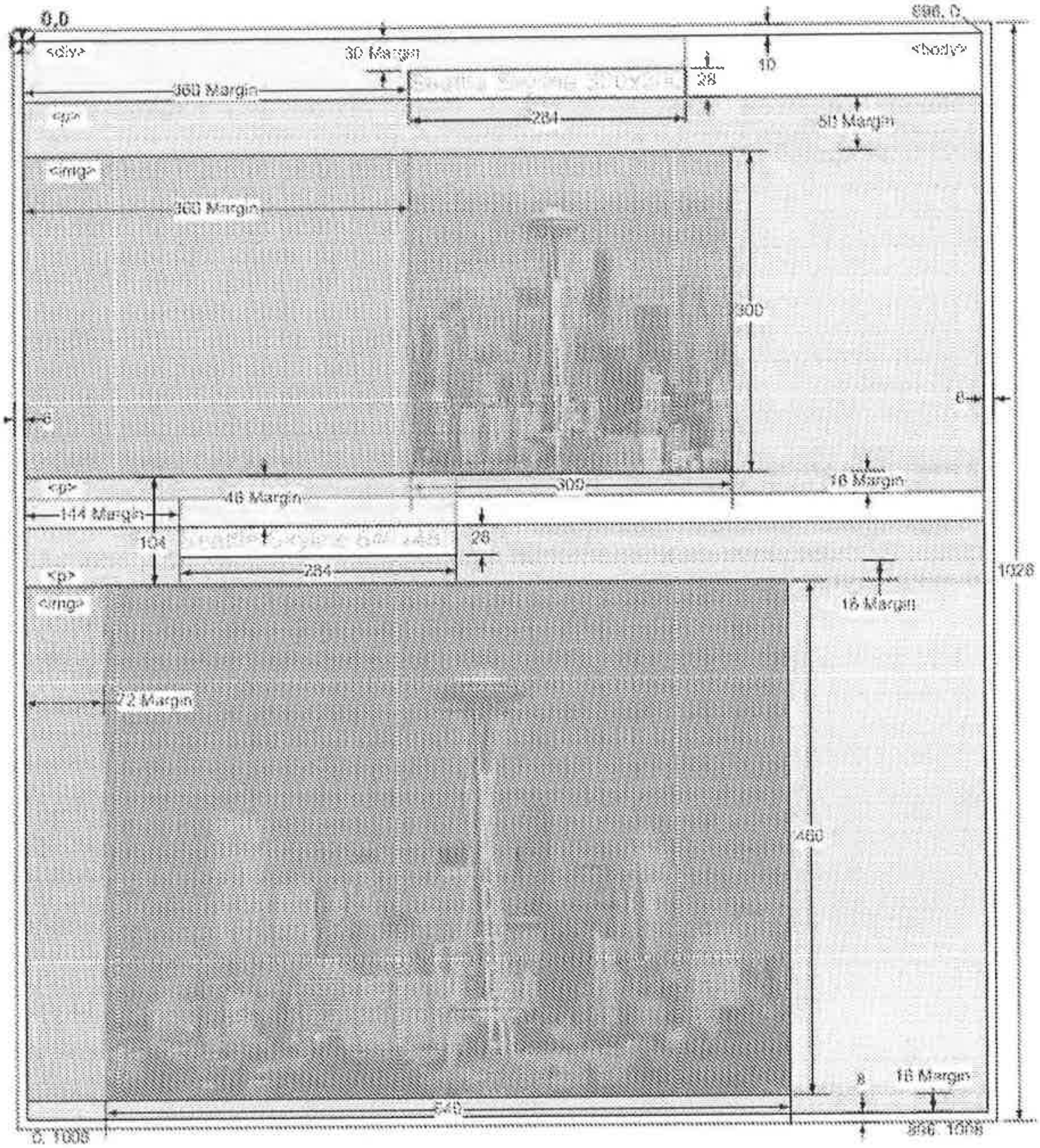
As shown in the page layout below, the Mozilla Gecko rendering engine produces page layout information using container blocks in a nested manner. HTML is coded in a hierarchical manner using HTML tags to define HTML elements (*aka* HTML blocks or objects). As the HTML page source is parsed, the HTML elements are identified and an element hierarchy called a tree is generated. The layout of objects within a given container block are relative to the location of the container block itself, which may further be relative to other container blocks in a layout hierarchy. In general, the html block (content between <html> and </html> tags) comprises the top level container block for the page. However, the top level container block for this example page is effectively the body block, since all content internal to the body block is laid out relative to the body block. Other typical container blocks are defined by <div> elements, paragraph <p>element, *etc.*

It is typical for browsers to apply their own interpreted amount of padding (small vertical and horizontal offsets) around the body block so the edges of a page are not coincident with the edges of the content drawing area. In the case of Firefox, the padding around the body block for this example page for the top, left, right, and bottom is respectfully 10, 8, 8, and 8 pixels. Notably, the 10 pixel top margin offset is explicitly specified by the HTML, while the 8 pixels offsets appear to be default offsets employed by Firefox. Since the content in this example is laid out relative to the body block, the upper left hand corner of the body block container box is defined as the 0, 0 datum for the examples herein rather than the upper left hand corner of the page (again for simpler explanation of the scaling examples).

Under Gecko, the horizontal and vertical position of a given object relative to its container in this example web page is labeled as a "margin." This is in accordance with the CSS box model[5], which may also include parameters defining an optional border and/or and padding around the content bounding box for a given object. Each content bounding box, in turn, is defined by a width and a height. To determine the position of each object relative to a body block datum (*e.g.*, an X, Y pixel coordinate position of 0, 0 in this example), the object's location is determined first by determining the location of its immediate container block, and then applying the applicable offsets in the X and Y directions relative to the container block's box definition (more specifically the X offset is from the container block box's left edge and the Y offset is from the box's top). The container blocks themselves may be nested and/or may be stacked on top of one another vertically; accordingly, their location is determined in a similar manner. The net result is the layout of each object on the page relative to a page datum can be determined by processing the layout information determined by the rendering engine for each object relative to its containing block and aggregating the total X and Y offsets. This is commonly termed as "walking the render tree."

---

[5] See. e.g.. http://www.w3.org/TR/CSS2/box.html



Page Layout (as interpreted by the Mozilla Gecko Rendering Engine)

In the layout, a bounding box for the "Seattle Skyline 300x300" text object is located at an X, Y datum of 360, 30 (corresponding to the upper left-hand corner of the box) and has a width of 284 pixels and a height of 28 pixels. The font and location of this text object is defined by the following HTML,

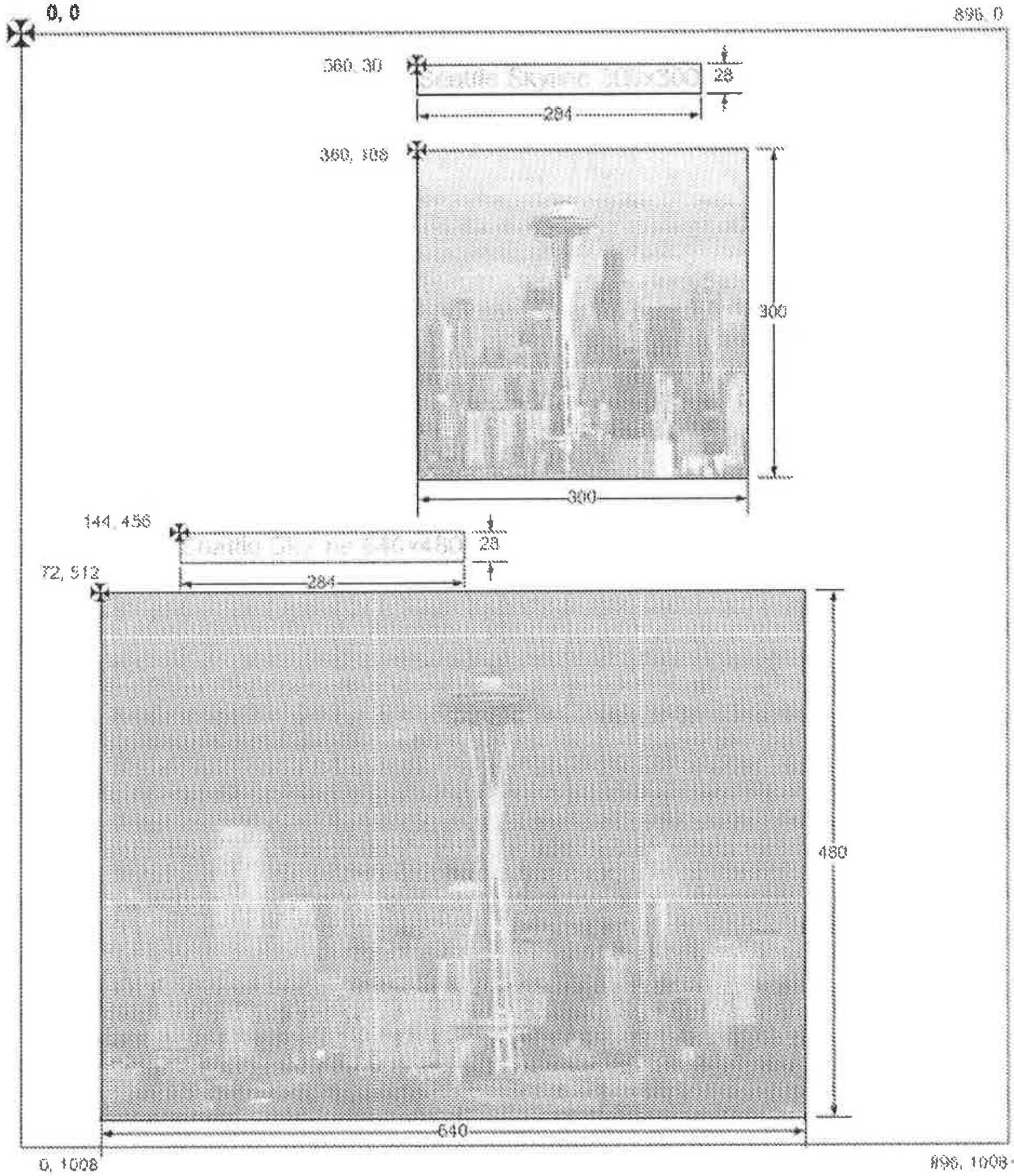
.auto-style5 {

```
        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 a given 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. For simplicity and clarity, upper-left hand corners are used for the datums for the four page objects in this example Web page.



Object location derived from Mozilla Gecko Page Layout

At this stage, selected page layout information is "mapped" to the virtual coordinate space employed by SVF using corresponding SVF drawing commands.

SVF employs a "painters" model under which content is "rendered" to a virtual drawing area (*aka*, virtual drawing space or canvas) using SVF drawing commands, and content may be rendered on top of previously rendered content, much like adding paint to a canvas. SVF also employs the concept of a graphics state including current pen position under which certain types of content are rendered based on current parameters for the graphics state at a location defined by the current pen position. The position of the content is defined by one or more points included in a given drawing command and/or an existing current pen position. When using SVF double (floating point) coordinates[6], each point is defined by corresponding X and Y floating point coordinates in the SVF virtual drawing space, and a given SVF drawing command may include floating point parameters defining a location of one or more coordinate points relating to content to be rendered via processing the drawing command. As such, SVF drawing commands are deemed to have a vector format, and thus an SVF drawing command (as defined by the applicable graphics state when rendered[7]) and its embedded or referenced content comprise vector-formatted content, as used in the present application. Alternatively, SFV drawing commands and their embedded or referenced content may be referred to as vector-based content.

By default, SVF uses a conventional two-dimensional graphics coordinate system under which X values increase to the right, and Y values increase going upward. As is evident from the prior images, the Y axis is flipped from that used by the Gecko rendering engine. SVF also has a provision for flipping the Y axis (so that the coordinate system is the same as that used by Gecko). In one embodiment, the

---

[6] SVF also provides support for using integer coordinates.

[7] The effective rendering instructions provided by an SVF drawing command includes the applicable context of the current graphic state at the time the drawing command is processed. For example, the graphics state context may include such things as current pen position, pen color, pen width, fill color, fill mode, font type (e.g., family and style), font size, font color, etc.

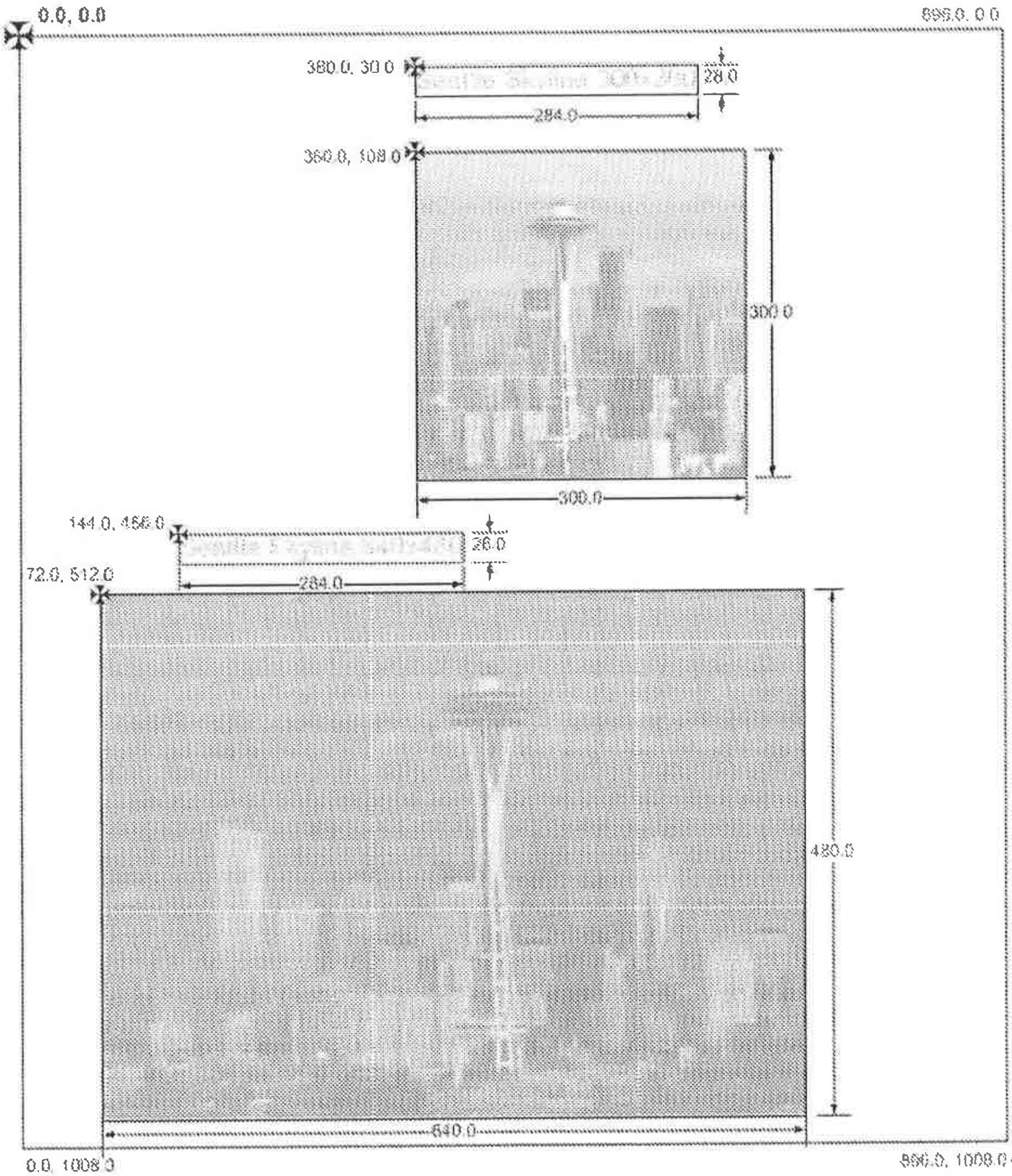
SoftView™ browser employs the flipped SVF coordinate system; however, it shall be understood that the default SVF coordinate system may also be used by flipping and offsetting applicable object coordinates from corresponding coordinates employed by Gecko when mapping objects to the SVF drawing space.

In the following example, the use of SVF using its flipped coordinate system is used. As such, the pixel-based coordinates of the Gecko page layout are mapped to corresponding points in the SVF coordinate system without flipping the Y axis.

In one non-limiting embodiment, the coordinates of the points corresponding to selected datums and bounding box corners are mapped from integer values corresponding to the coordinate locations in the pixel-based coordinates employed by Gecko to corresponding floating point values in the SVF coordinate space. In one embodiment a 1:1 mapping is used, as illustrated in the figures herein; however since SVF employs floating point coordinates, other mappings could be used, as well. For example, the X,Y pixel coordinate of 360, 108 for the 300x300 image of the Seattle skyline is mapped to an X,Y coordinate of 360.0f, 108.0f (where f denotes floating point and is used here for point of illustration to indicate that these are floating point values) in the SVF coordinate space. The page layout information for other page content, such as various text and image objects, is mapped to the SVF vector-based coordinate space in a similar manner using applicable SVF drawing commands[8].

---

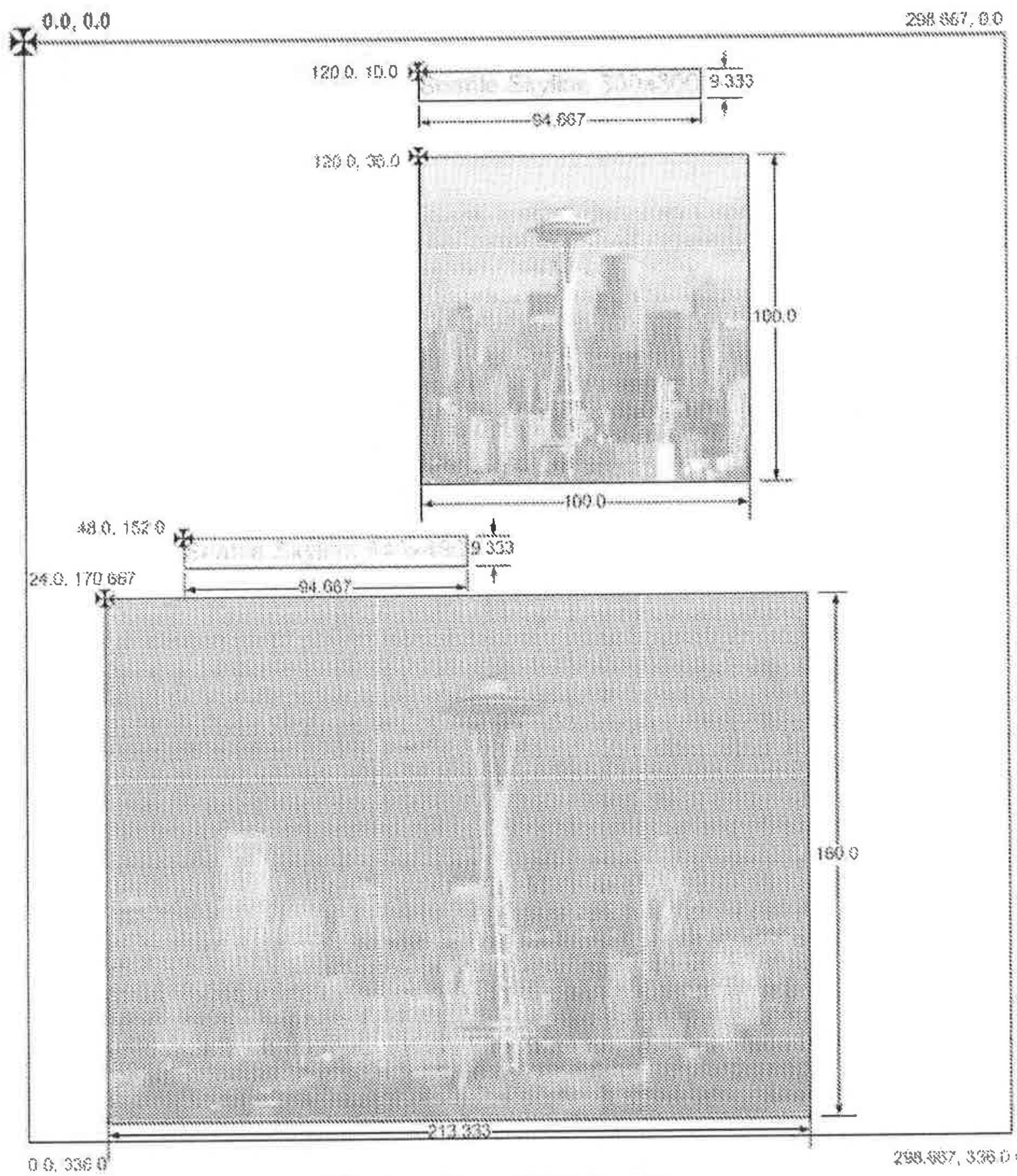
[8]Details of SVF drawing commands are included in the SVF v2.0 specification, which may be accessed at www.svf.org/spec.html. The SVF v2.0 specification has been publically available at this web site since approximately December 1999.



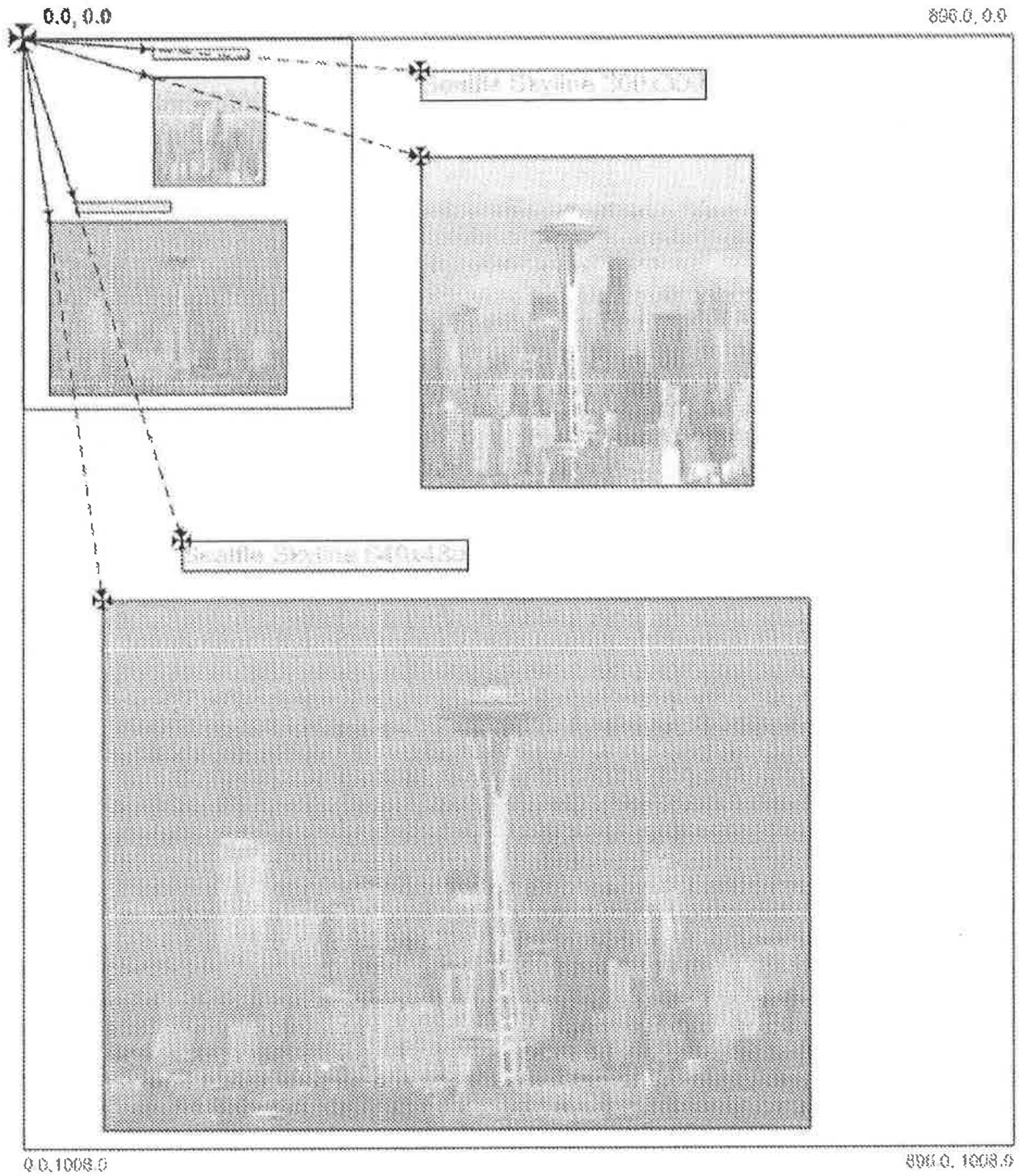
Layout Mapped to SVF Coordinate Space (Flipped Y axis to match Gecko)

As discussed above, the content area of the Firefox browser was adjusted such that the interpreted width of the body block was 896 pixels. The reason for this was so that the following scaling operation can employ simple fractions. Suppose that the mobile device has a pixel resolution of 320 x 320 pixels, such as is the case for the Sony Clie used in the examples of nytimes.com screenshots shown later herein. Of the 320 pixels in the width dimension (when oriented in a portrait position), approximately 304 pixels are used for the content drawing area and 16 are used for a vertical scroll bar. The SoftView™ browser also has a default page width, which can be set by a user. In accordance with this example, the Firefox browser window width was adjusted to illustrate the results of its interpreted page layout as if a default page width set to 912 (pixels) was used, such that the page width would be three times the content area width (3 x 304 = 912).

In some embodiments, users are enabled to view the full width of the web page on this mobile device. In accordance with this example, a scale factor of 304/912 or 1/3 is applied to the page layout coordinate points on the SVF virtual coordinate system to generate a full page-width view having a width of 304 units. The results of this are shown in the next two figures. The first figure shows the floating point coordinate values after scaling, and the second figure 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.

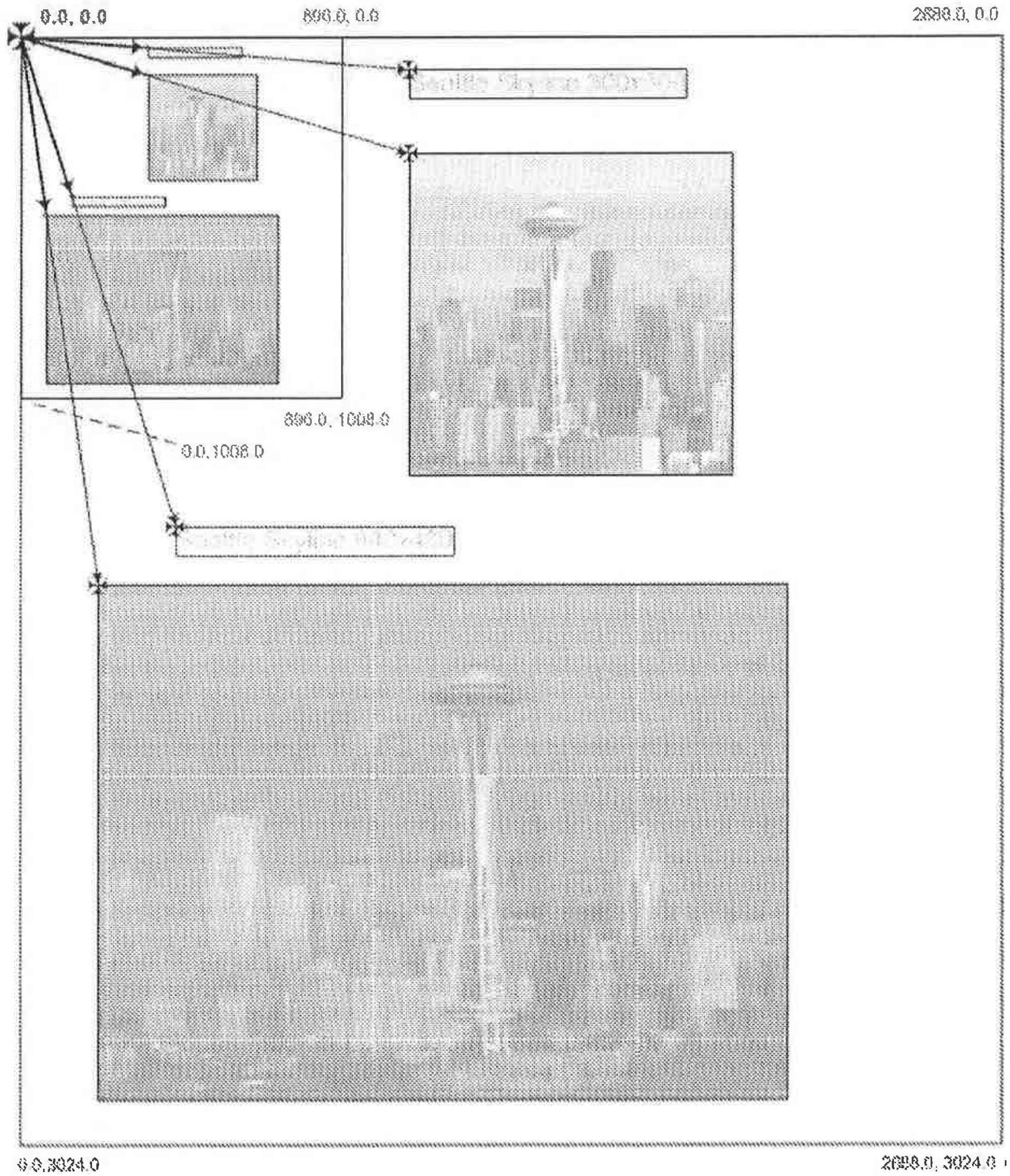


1/3 Scale – Flipped SVF Coordinates



Graphical Depiction of Objects and Layout being Scaled by 1/3 relative to full scale

In addition to scaling at scale factors less than full scale (*i.e.* 1x, or 1 times full scale), the scalable page layout information can be scaled to scale factors greater that 1x. For example, in the following Figure the page layout has been scaled be 3x. This results in generation of a scaled Web page with an effective resolution of *n* times (*i.e.*, *n*x) the full scale page resolution, where *n* corresponds to the scale factor.



Graphical Depiction of Objects and Layout being Scaled by 3 (3x) relative to full scale

Notably, under a vector-based drawing model, scale factors are not limited to integer values or integer fractions, but rather may include substantially any finite value (from a theoretical viewpoint)[9]. From a practical standpoint, the range of scale factors (and zoom levels[10]) will typically be limited to a relatively low value[11], such as 5x or 10x, for example. As screen pixel densities and resolutions increase, the use of higher scale factors will become more prevalent. For example, if the screen pixel density of a display screen doubles, the scale factor must also double to produce content such that it has the same size on respective display screens having the original and doubled pixel densities.

In summary, in one or more non-limiting embodiments a rendering engine is employed to process HTML-based Web page content to interpret the page layout of the Web page display content (*i.e.*, display objects) and selected page layout coordinates are mapped from a pixel-based coordinate system used by the rendering engine to corresponding floating point coordinates in a virtual coordinate space. Once the layout information is defined by the floating point coordinates in the virtual coordinate space, the coordinates of the page layout information can be scaled to produce scaled page layout information. Scaled views of Web page content can then be generated by

---

[9] From a theoretical perspective, the scaling of the page layout information itself will only be limited by the floating-point math limitations of the processor implemented by the device. Similarly, when text content is implemented using vector fonts, such as TrueType, the font glyphs can be scaled indefinitely. However, there are practical limitations to scaling image content in a manner that preserves a reasonable level of quality.

[10] This is not to imply that scale factors and zoom levels are synonymous or must be related on a 1:1 basis. For examples, rendered bitmap content corresponding to a given zoom level may be derived from interpolating bitmap content derived from employing scale factors that are different than the zoom level.

[11] As scale factors are doubled, the amount of memory for rendering corresponding bitmap content increases four-fold (assuming the same amount of memory allocated per pixel). This problem is typically addressed by only rendering bitmap content corresponding to a portion of a full web page at a time when using higher scale factors.

rendering display objects using applicable scale factors at locations defined the scaled coordinates corresponding to each display object.

Overview of operation of ClearView™/SoftView™ browser

New claim language relating to "preserving the original page layout" and "preserving the original page layout and design" of the HTML-based Web content has been added by amendment herein. To more fully understand the scope of this language, a discussion of operations pertaining to an exemplary use case of a mobile device enabled by the presented application is now presented. A similar discussion was first presented in the December 9, 2007 response of Application No. 11/045,757, which is a divisional of the present application; for clarity, much of this description is repeated below, while some details are omitted for brevity. The operations are discussed in the context of the following FIG. 1.

The schematic drawing of FIG. 1 shows an exemplary infrastructure comprising well-known components for facilitating access to and delivery of Web pages. Web page content (*i.e.*, Web content) is served by servers that are accessed via the Internet, also commonly referred to as the World Wide Web (WWW). Accordingly, these servers are typically referred to as "Web" servers. More accurately, they are HTTP (Hypertext Transport Protocol) servers, as they serve content of various types using the HTTP protocol. FIG. 1 shows a pair of exemplary Web servers, including a New York Times (NYT) Web server and an Advertisement (ADV) Web server. It will be appreciated that literally millions of similar Web servers are connected to the Internet across the world, thus forming the World Wide Web. In turn, these Web servers serve billions of Web pages.

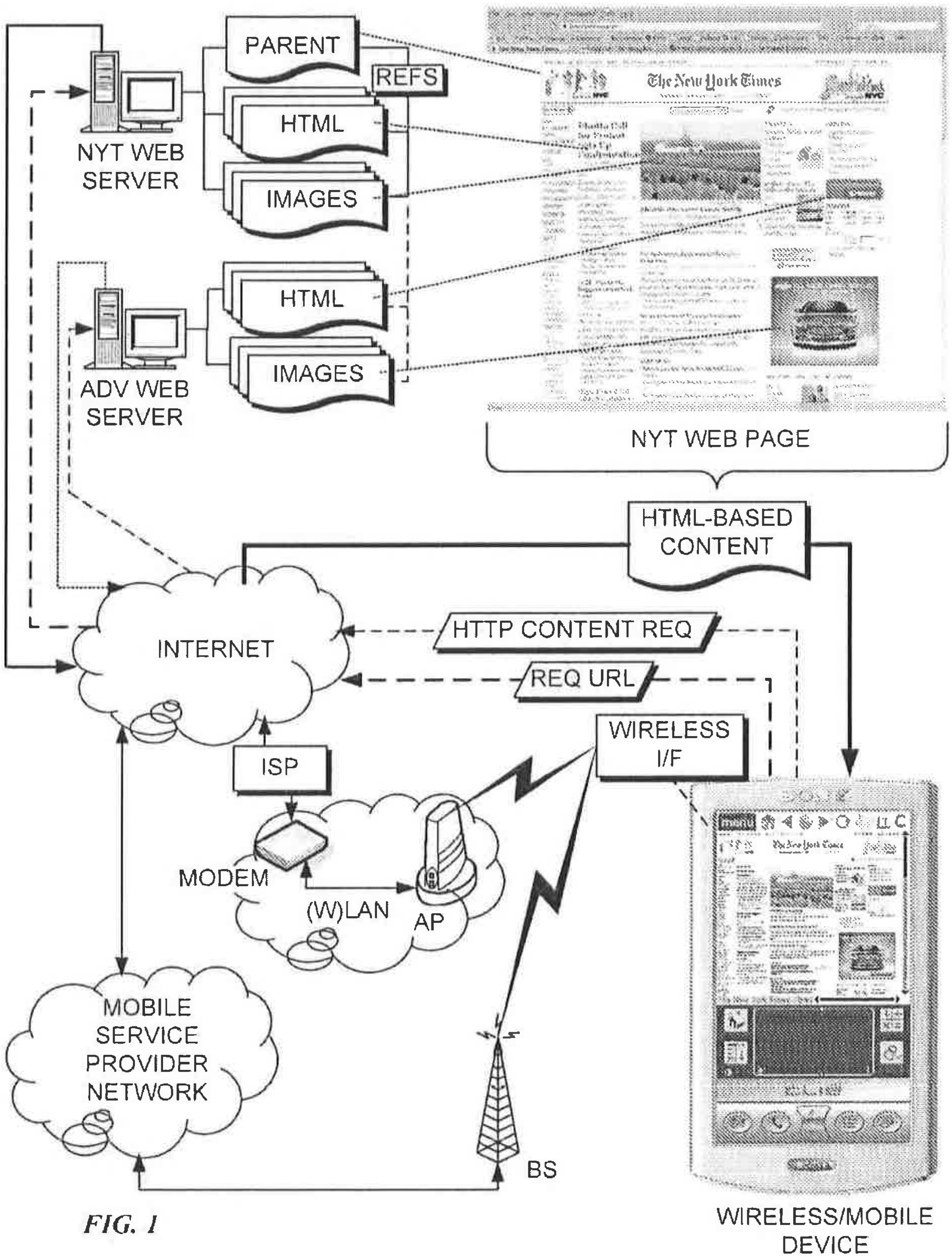

PARENT
REFS
HTML
IMAGES
NYT WEB SERVER
HTML
IMAGES
ADV WEB SERVER
NYT WEB PAGE
HTML-BASED CONTENT
INTERNET
HTTP CONTENT REQ
REQ URL
WIRELESS I/F
ISP
MODEM
(W)LAN
AP
MOBILE SERVICE PROVIDER NETWORK
BS
WIRELESS/MOBILE DEVICE

FIG. 1

To access WWW web servers, users use client devices that are communicatively coupled to the Internet through applicable network infrastructure. In the desktop environment, desktop clients, such as personal computers and workstations, are typically coupled to a Local Area Network (LAN) via an Ethernet link to a LAN host device. (It is noted that some desktop clients may wirelessly connect to a Wireless LAN (WLAN), in a manner similar to that discussed below for wireless clients.) The LAN, in turn is usually connected to the Internet via network infrastructure provided by an Internet Service Provider (ISP). Connection between the LAN and the ISP is typically provided by some type of Modem (*e.g.*, Cable or xDSL Modem) or dedicated hardware (for larger customers, such as businesses). (It is also noted that many individual users still connect to their ISP through a telephone modem.)

Wireless and mobile devices, including those devices covered by the claims herein, typically connect to the Internet in one of the manners illustrated in FIG. 1, or otherwise described in the December 9, 2007 response for Application No. 11/045,757. (Further details are omitted here for brevity.) By way of example but not limitation, wireless access to the Internet and World Wide Web may typically be provided via a mobile service provider (using appropriate radio facilities), or via other types of wireless connections, such as via a WiFi (IEEE 802.11x) or WIMAX connection, for example.

Now that the infrastructure of FIG. 1 has been described, we proceed with discussion of retrieving and processing the Web page content such that the Web page can be accessed via a wireless/mobile device. In the illustrated example, the process is initiated by a user desiring to access the New York Times (*i.e.*, and electronic version of the New York Times published to the Internet on a given day). This is facilitated by a browser in accordance with teaching of the present application running on the wireless/mobile device. The New York Times may be accessed via the Internet by downloading corresponding Web pages from the NYT Web server. More specifically, the New York Times home page may be accessed by entering the URL (Universal Resource Locator) www.nytimes.com via the browser's user interface.

As discussed above and in further detail in the present specification, Web pages comprise HTML-based content which may be stored in one or more documents commonly referred to as HTML documents. In addition, Web pages may include dynamically-generated content. Each Web page has a corresponding main or "parent" HMTL document that includes HTML code defining the Web page content layout, at least at some level. The parent HTML document may reference other HTML documents, as well as other content (such as image content) that further define the layout of content contained in the referenced documents. This may proceed in a hierarchical or nested fashion.

To access the Web page, the browser initiates an HTTP connection with the Web server hosting the Web page, and begins downloading the parent HTML document. Depending on how the Web server and/or Web page is configured, additional content (*i.e.*, beyond that included in the parent HTML document) referenced by the parent HTML document, may be retrieved by the Web page host server and then downloaded to the requesting client device, or a portion of this content may be downloaded by the client device via a separate connection. Generally, content that is hosted by a Web server or Web site is assembled by the Web server and downloaded to the client device. On the other hand, externally-referenced content (that is, content that is not stored on the Web server or Web site), is often left to the client device (*i.e.*, the browser) to retrieve.

An example New York Times home page (dated November 7, 2007, 2:22PM ET), as rendered by the Mozilla Firefox browser running on a desktop or laptop computer (and captured as a screenshot), is shown at the upper right-hand portion of FIG. 1. The same Web page is shown rendered on a Sony Clié using a SoftView™ browser at the lower right-hand portion of FIG. 1.[12] Notably, the same HTML-based content is downloaded by each of the Mozilla Firefox and SoftView™ implementation. Moreover, the same HTML-based content would be retrieved by other desktop browsers, such as Microsoft Internet Explorer, Apple Safari, Google Chrome, and Opera browsers, to render the New York Times home page.

As discussed above, the Parent HTML document typically includes HTML code to define the overall layout of the Web page and its content. For example, the HTML code will define whether the Web page includes frames, and, if so, where those frames are located on the rendered page. The Parent HTML document may also contain Cascaded Style Sheet (CSS) data defining various design aspects of the page, or contain one or more references to other documents containing such CSS data. Various content displayed on the Web page may be stored in the Parent HTML document and/or one or more other HTML documents referenced by the Parent HTML document. If the content is to be rendered in a frame referenced by the Parent HTML document but whose content is not defined within the Parent HTML document, the actual reference to the HTML document storing the content may be in the document defined by the frame reference. For example, for illustration purposes, the content in the column with the heading "U.S. Rejected Aid for Israeli Raid on Iranian Nuclear Site" is depicted to be stored in an HTML document that is hosted by the NYT Web server, but is separate from the Parent HTML document.

Likewise, image content may be stored separate from the Parent HTML document. This is typically done since images, which often contain a large amount of

---

[12] It is noted that the use of the Sony Clie and corresponding SoftView browser screenshots are for illustrative purposes to demonstrate how web page views on a mobile device in accordance with the claims herein might appear. The screenshots were produced using a SoftView proxy server and browser client running on a Palm OS Emulator 3.0a8S1.0, 2000-2001 using a Sony Clie PEG-700 Skin. At the time of filing Application Serial No. 09/878,097, the browser was referred to as "ClearView" rather than "SoftView."

data due to the nature of image data, may require significant download time, especially over a slow connection. By putting image content in (a) separate document(s), the basic page layout and text content can be rendered much faster. Typically, HTML code defining the page layout location of an image on the page may be used to place an image "placeholder" or other indicia on the screen prior to rendering of the image.

As discussed above, various portions of the Web page content may be stored on Web servers that are external to the Web page host server. This is often the case with advertisement content. Rather than have the advertisement content stored locally on each Web server, the advertiser will use an advertisement host site to store and serve the advertisement content. For example, in Fig. 1, image data for rendering the "DON'T MISS HISTORY IN THE MAKING" advertisement is depicted as being stored in an image document on the Advertisement Web server.

Under a typical conventional approach, externally referenced advertisement content is downloaded by the browser directly from the advertisement content host site, rather than from the Web page host site. The network location of the advertisement content host server is identified by parsing the retrieved HTML-based content, and an HTTP GET request is used to download the associated advertisement content from its host server.

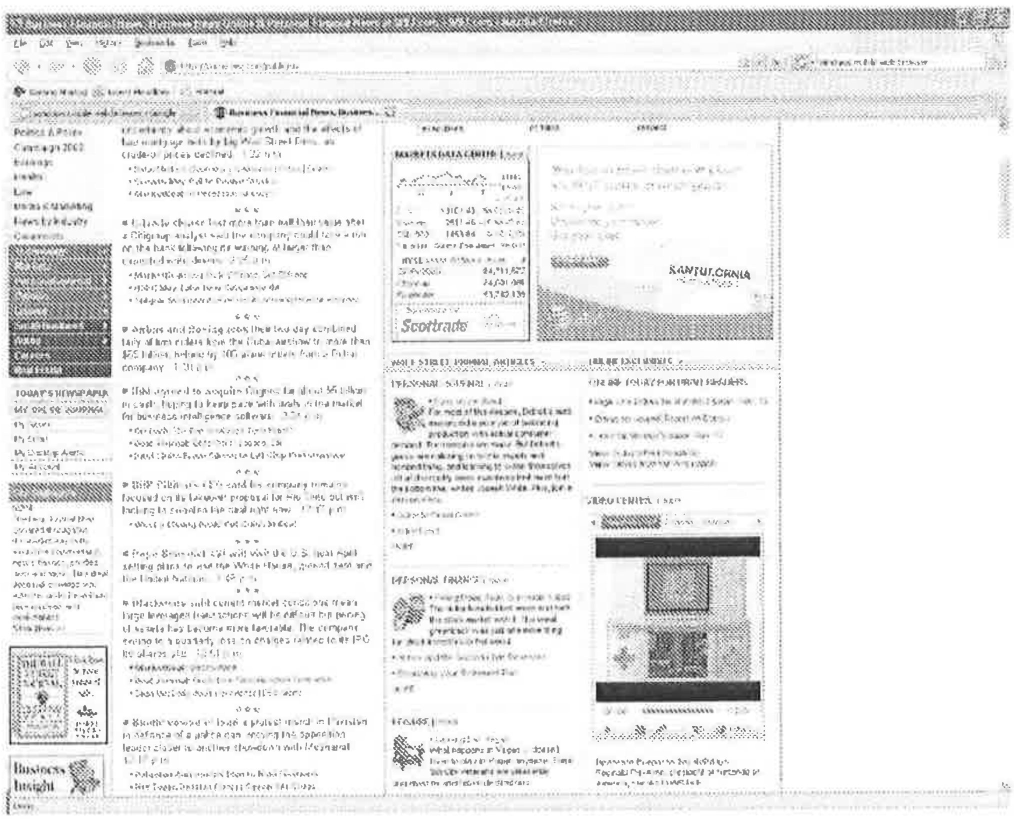
Some Web pages may include "embedded" content hosted by an external site. Oftentimes, such embedded content may be dynamic in nature (that is, may change over time or differ depending on identification of the target user). Generally, embedded content may be retrieved by the browser from an external host site (*e.g.*, advertisement Web server depicted in FIG. 1), or such content may be first retrieved by the Web page host site and served to the browser.

It is common terminology to refer to a browser "retrieving" or "downloading" a Web page. For example, upon entry of a new URL in the browser Web address box of a desktop browser, the browser will download the Web page referenced by the URL. It is well understood that this doesn't imply that all of the content associated with the Web page must be retrieved or downloaded. Some of the content is typically used for search engine or other purposes, such as some Metatag header information, or is otherwise not used for rendering purposes. Other Metatag information may be employed for rendering purposes, such as defining browser compatibility information. In other cases, content may be referenced that is not supported by the requesting browser. For example, "Flash" content typically requires a Flash plug-in viewer (or built-in Flash support provided by some browsers); if the plug-in viewer is not loaded by the browser (or such support isn't built in), the Flash content cannot be displayed. This is also true for TIFF images on the USPTO Web site. Unless an appropriate TIFF plug-in viewer is loaded, the TIFF images will not be displayed.

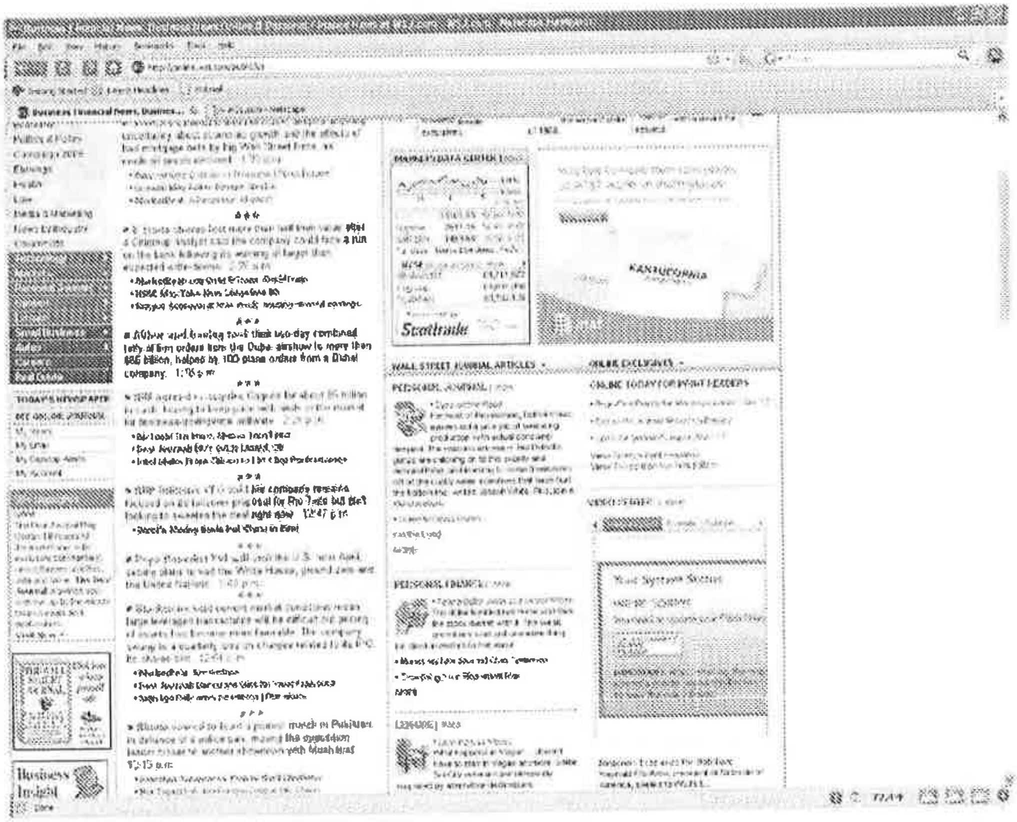
A similar situation exists with Active-X controls. In order to use the Active-X controls, the browser needs to provide support for Active-X controls. Since Active-X controls were developed by Microsoft, all recent versions of Microsoft Internet Explorer provide support for Active-X controls. Meanwhile, browsers from other vendors, such as Apple Safari, Mozilla Firefox, Google Chrome, and Opera, do not support Active-X controls.

When a browser encounters content that is not supported "natively" by the browser, the browser will typically check to see if an appropriate plug-in is available. Depending on the browser and/or particular Web site, if an appropriate plug-in cannot be found, the browser or Web site may apprise the user of the situation and enable the user to download the plug-in. In other instances, the content is simply ignored. Thus, in some cases, the Web page may reference content that is never retrieved when the Web page is retrieved by the browser.

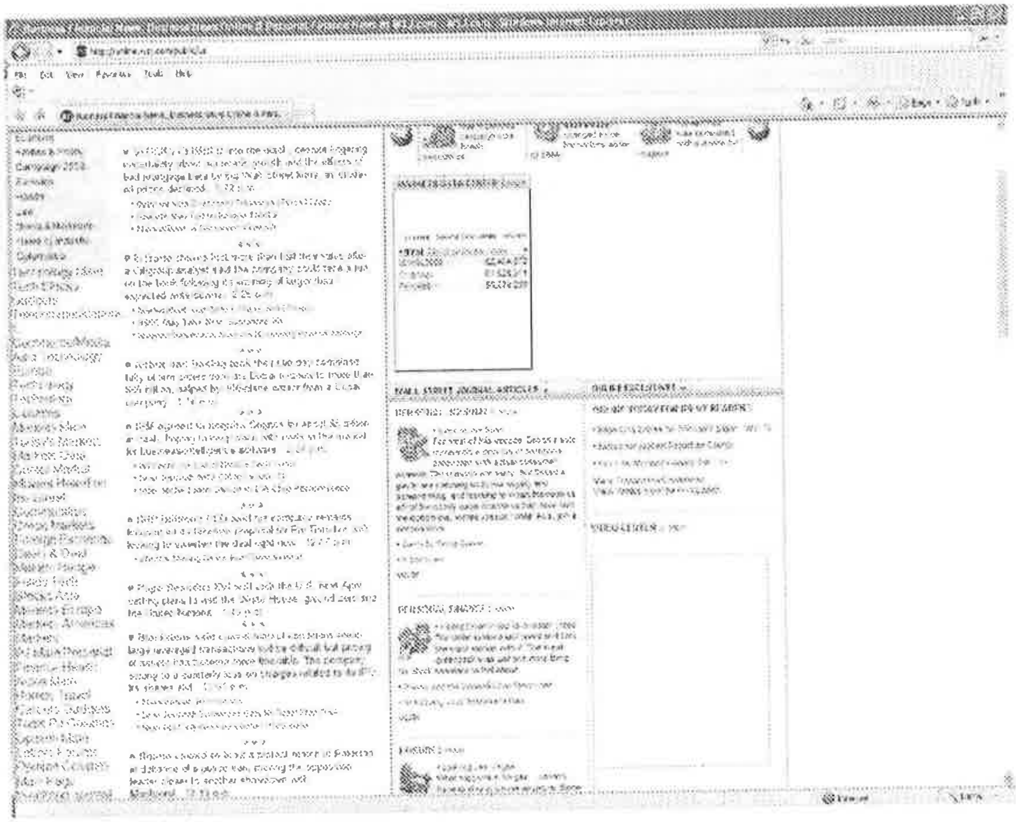
The three screen shots below respectively show the same Web page (WSJ.com home page) rendered on a Netscape Navigator 9 browser, a Mozilla Firefox 2.0 browser, and an Internet Explorer 7 (IE 7) browser. In this particular instance, certain features of the IE7 browser are disabled for security reasons. It is also missing some plug-ins. Each of these browsers is running on the Microsoft Windows XP operating system.



Mozilla Firefox 2.0 Browser



Netscape Navigator 9 Browser



Microsoft Internet Explorer 7 Browser

It will be observed that the Web page is rendered substantially the same by the Netscape Navigator 9 and Firefox 2 browsers, while portions of the Web page are rendered in a different manner by the IE 7 browser (notably the left-hand column). The similarity between Netscape and Firefox is expected since they both use the same Mozilla core rendering code, while IE 7 uses Microsoft's rendering code. It is further noted that some Web pages are coded to account for different browser features. For example, some Web pages will have provisions for Active-X controls for pages to be viewed by Internet Explorer browsers, while possibly including provisions for alternate mechanisms when using other browsers.

This example Wall Street Journal Web page includes various embedded ***non-HTML content*** requiring support of one or more plug-ins[13] or otherwise built in support for rendering ***non-HTML*** content of a particular content type. In particular, the VIDEO CENTER object in the lower right-hand corner requires an Adobe (formerly Macromedia[14]) Flash viewer for rendering Flash content, which uses vector and raster graphics, a native scripting language called ActionScript and bidirectional streaming of video and audio.[15]

It is noted that there is a message in the VIDEO CENTER box in the Web page rendered by the Netscape Navigator 9 browser indicating that the browser needs to update its Flash player. In the case of the Firefox 2 browser, either the appropriate Flash player was found or an appropriate level of support for Flash content is built into the browser. In this case, the Flash .SWF file including data to render a video image of a Nintendo DS console is retrieved from a corresponding host server and rendered by the browser (if it has built-in support) or Flash player, as applicable. In the case of the Netscape Navigator 9 browser, the appropriate Flash player plug-in is not available; accordingly, the video image of the Nintendo DS console is not retrieved.

In the case of the particular IE 7 browser configuration used to obtain the IE7 screen shot, the Flash player is either missing or blocked. As a result, the aforementioned VIDEO CENTER image is missing (just an empty box is rendered, as defined by corresponding HTML). Moreover, the IE 7 browser did not render a

---

[13] As defined by Wikipedia, A **plugin (plug-in, addin, add-in, addon or add-on)** is a computer program that interacts with a host application (a web browser or an email client, for example) to provide a certain, usually very specific, function "on demand". Applications support plugins for many reasons. Some of the main reasons include: enabling third-party developers to create capabilities to extend an application, to support features yet unforeseen, reducing the size of an application, and separating source code from an application because of incompatible software licenses.

[14] Macromedia is now a division of Adobe Systems

[15] For more details on the Adobe Flash Player, see, *e.g.*, http://en.wikipedia.org/wiki/Adobe_Flash_Player

message indicating the Flash player needed to be upgraded. In addition, the source for the AT&T advertisement in the upper right-hand portion is blocked via a security setting, resulting in this portion of the page being rendered using the same background color as the frame it (would be) embedded in.

A point for discussing the foregoing is to make it clear that,

1. Even when rendering the same Web page source content (*i.e.,* the HTML code definition of the Web page), conventional Web browsers may not render the (full scale) Web page identically. Under aspects of embodiments of the invention the overall layout and appearance (design) of the Web page representations defined by the HTML code for the Web page (as interpreted by the rendering/layout engine) are preserved at various zoom levels and panned views. That is, the preservation is relative to how the page layout and design of the Web page content is interpreted by the rendering/layout engine employed for a particular implementation, and is not relative to how the Web page might appear on a particular desktop browser, although they might appear the same or substantially similar if using the same rendering/layout engine.
2. Plug-ins may be required to render ***non-HTML content*** that is embedded within some web pages or used in a separate window launched from a web page. Notably, the plug-in content is not a Web page, but rather a specific type of content requiring a corresponding plug-in application to render the content.

Preservation of Original Page Layout

As discussed above, new claim language has been added by amendment, and includes the language "preserving the original page layout" of the HTML-based Web content. With respect to preserving the original page layout, the page layout (to be preserved) is determined as interpreted by rendering/layout engine components, rather than as a comparison to how the page might be rendered by a particular desktop browser. As described above and in other remarks, browsers often do not render Web pages derived from the same HTML-based definition identically. Accordingly, one of ordinary skill in the browser art would not expect Web pages rendered using an implementation in accordance with the teachings disclosed in the present application to render pages as *exact* scaled replicas of the same page rendered by a conventional desktop browser, such as Internet Explorer or Safari, for example. Due to rendering limitations such as fixed size fonts, renderings of the same page when viewed at different zoom levels may result in small variations, as opposed to an exact scaled version of the same content (*i.e.*, as if viewed by a magnifying glass). While conceivably an implementation might produce this exact (*i.e.*, perfect magnification) result, such results are not required by the scope of the terminology "preserving the original page layout ... of the content."

A similar context exists with respect to "preserving the [original page layout and] design" of the content. Preserving the design of a Web page's HTML-based content corresponds to support for rendering of a Web page at different zoom levels and panned views in accordance with its design as defined by corresponding HTML-based content, which includes such things as type fonts, separator bars, tables, columns, *etc.* (It is noted that in the case of CSS, some design elements affect the page layout, while other relate to object design attributes, such as font color.) Again, the Web page's design is a matter of interpretation by the particular browser (typically via its rendering/layout engine), as, for example, the same content (as defined by its corresponding HTML definition) may be rendered using different colors by different browsers. Similarly, browsers may substitute fonts for original fonts (as defined by corresponding HTML code) that are not supported by the browser or operating system used by a given host device. With respect to the scope of the terminology "preserving the [overall layout and] design" of the content, this refers to preserving the design as interpreted by the rendering engine while at different zoom levels and panned views, as opposed to rendering the content identically to how it is rendered by a particular desktop browser that may interpret the page design differently. As discussed above, some design aspects relating the Cascading Style Sheets may not be implemented or may be implemented incorrectly by some browsers and/or rendering/layout engines. The preservation of the design is relative to the interpretation of the design aspects of the Web page by the rendering/layout engine that is employed.

Examples of SoftView™ Browser Screen Shots

The following pages illustrate various screen shots of the aforementioned nytimes.com Web page as translated by the SoftView™ proxy server and rendered on a SoftView™ browser client running on a Sony Clie emulator[16]. Under the claims herein, operations performed by a proxy server in accordance with these examples would be performed by a mobile and/or wireless device via use of browser software in conjunction with an operating system and potentially other facilities on the device, as would be understood by one skilled in the art. Figs. 2a-2d further include screen shots of the Web page as rendered on a Firefox 3.0.5 desktop browser for comparison purposes.

As can be readily observed, the SoftView™ implementation produces a Web page with substantially similar display content to that rendered by the Firefox browser. However, there are some differences, which are to be expected. For example, since the nytimes.com site uses rotating advertisements, different advertisements will appear at different times, depending on exactly when the screen shots are captured. In addition, since some types of content, such as flash content, were not supported by the SoftView™ implementation employed to take the screen shots, this content is not displayed on the rendered page. For example, the nytimes.com web page includes

---

[16] Palm OS Emulator 3.0a8S1.0, 2000-2001 using a Sony Clie PEG-700 Skin. The emulator was used for the purpose of creating the screen shots - similar page renderings are produced using an actual Sony Clie PDA.

code referencing a flash-based video (*i.e.*, the Play Video content under the VIDEO header). The corresponding area is rendered blank, such as shown in SoftView™ browser screen shot of Fig. 2b. Also, portions of the page layout may be adjusted in position depending on whether the content is supported or was available or successfully retrieved. Adjustments of this type are typically defined by layout rules implemented by the rendering/layout engine. For example, in the SoftView™ implementation an advertisement occupying the position of the "DON"T MISS HISTORY IN THE MAKING" in the desktop browser screenshots was either unable to be retrieved (potentially due to a timeout error or some other undetermined reason) or was of an incompatible type (*e.g.*, a flash advertisement that was one of multiple rotating advertisements used for this area on the nytimes.com home page[17]), and thus content below the missing advertisement is shifted upward in accordance with interpretation of the page by the version of the Mozilla rendering engine employed by the SoftView™ proxy server.

As discussed above, different browsers may render the same HTML-based Web page content slightly differently. Also, since aspects of Web page design and implementation have become more standardized, new types of content and design elements have been added or are otherwise now supported by some rendering/layout engines, and more are likely to be added in the future. As a result, earlier browsers using rendering/layout engines that were developed prior to the definitions of the new types of content and/or design elements or otherwise did not include support for such content or design elements will not display the content or design element(s) or may attempt to display them in an erroneous manner. Typically, when a browser rendering engine encounters content or a CSS element it doesn't recognize, it simply ignores is. In other cases, the HTML-based content may include support for functionality that is

---

[17] It has been observed that the nytimes.com home page uses both external img (image)-based advertisements and flash-based advertisement in this portion of the page.

applicable to a desktop environment, but not applicable to a mobile device. For 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 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.[18] 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.

---

[18] 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.


The New York Times
U.S. Rejected Aid for Israeli Raid on Iranian Nuclear Site
DON'T MISS HISTORY IN THE MAKING.
SONY
menu
CLIE
POWER

Fig. 2a

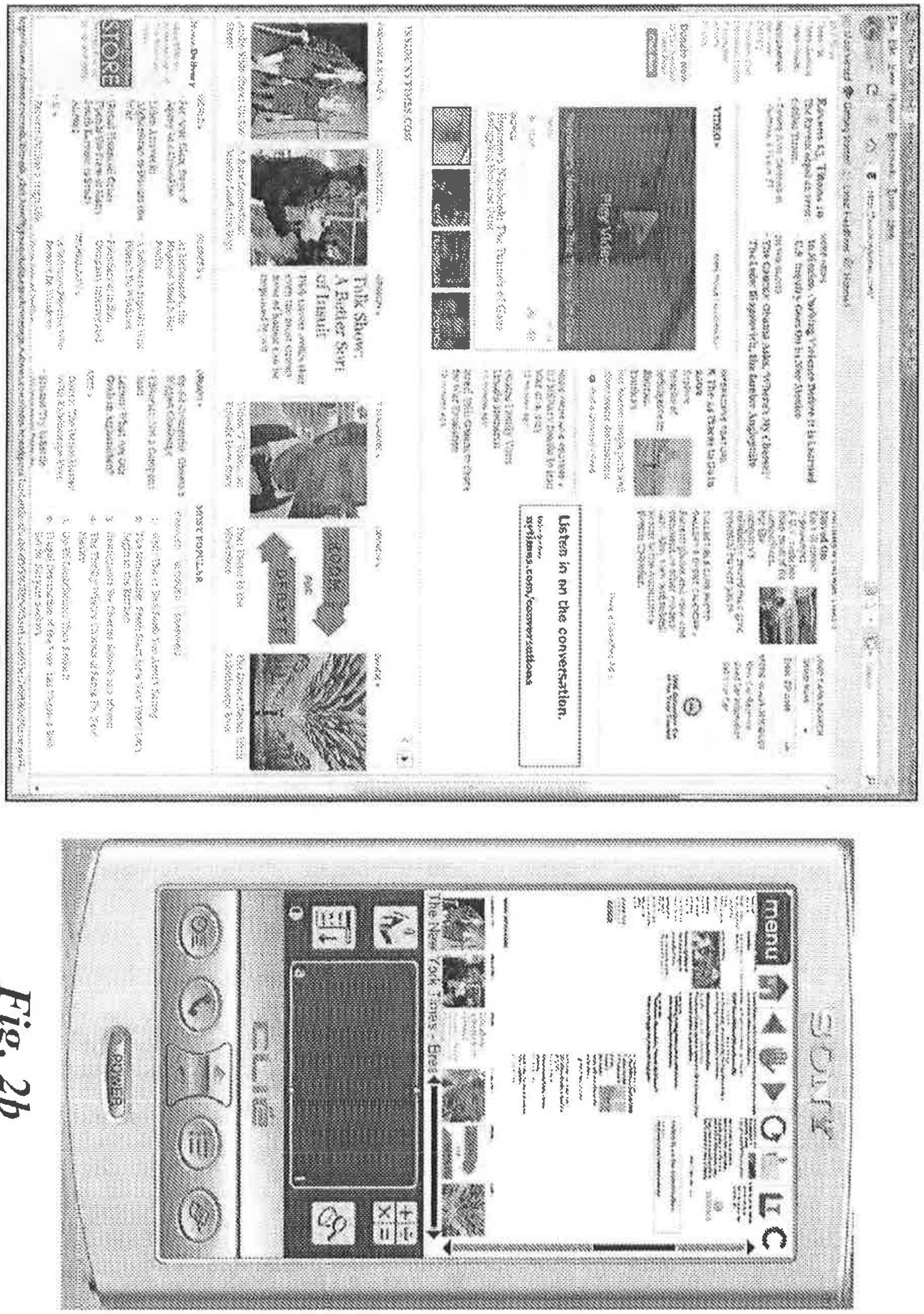


Fig. 2b

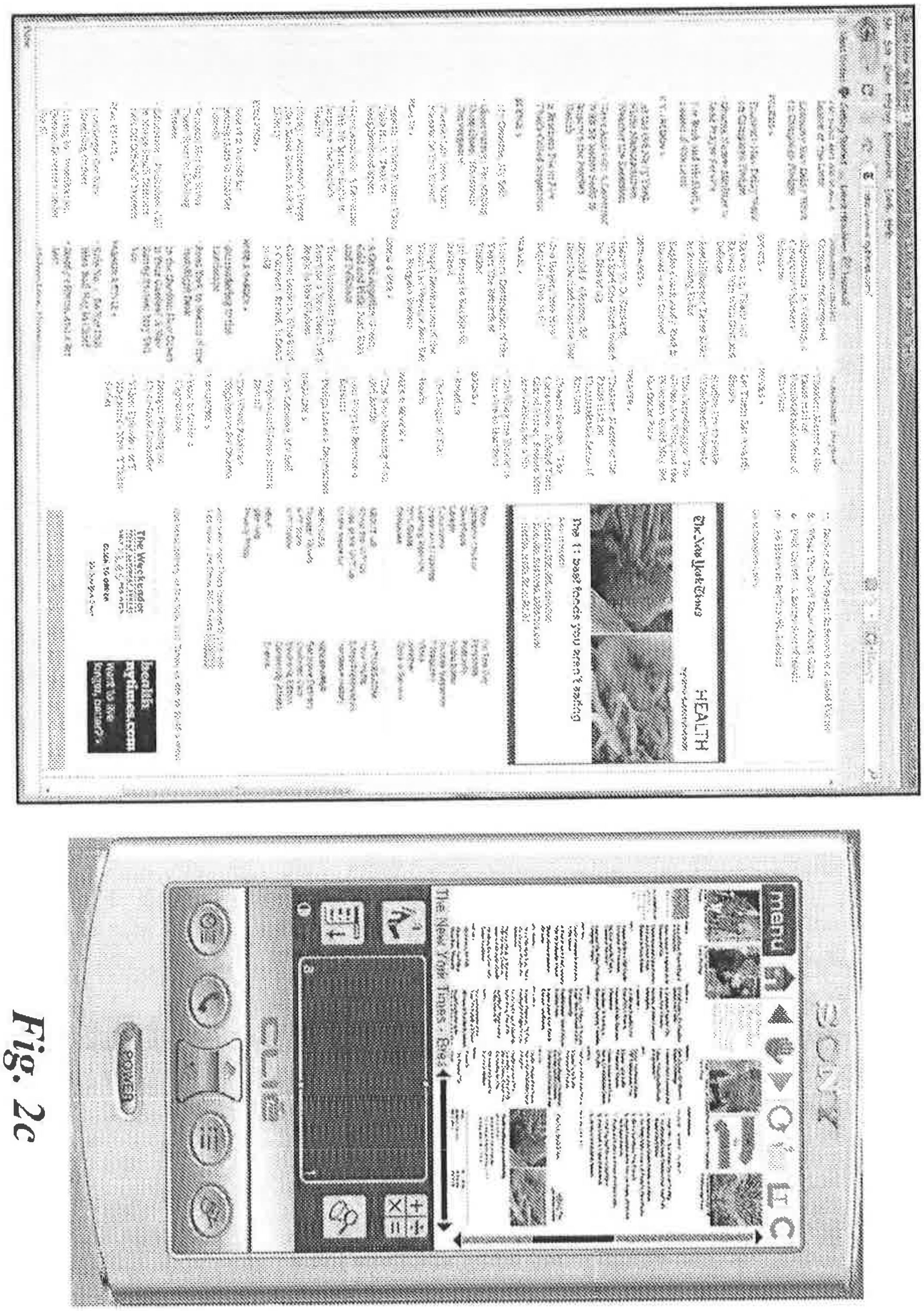

HEALTH
The Weekender
SONY
menu
The New York Times - Brea
POWER

Fig. 2c

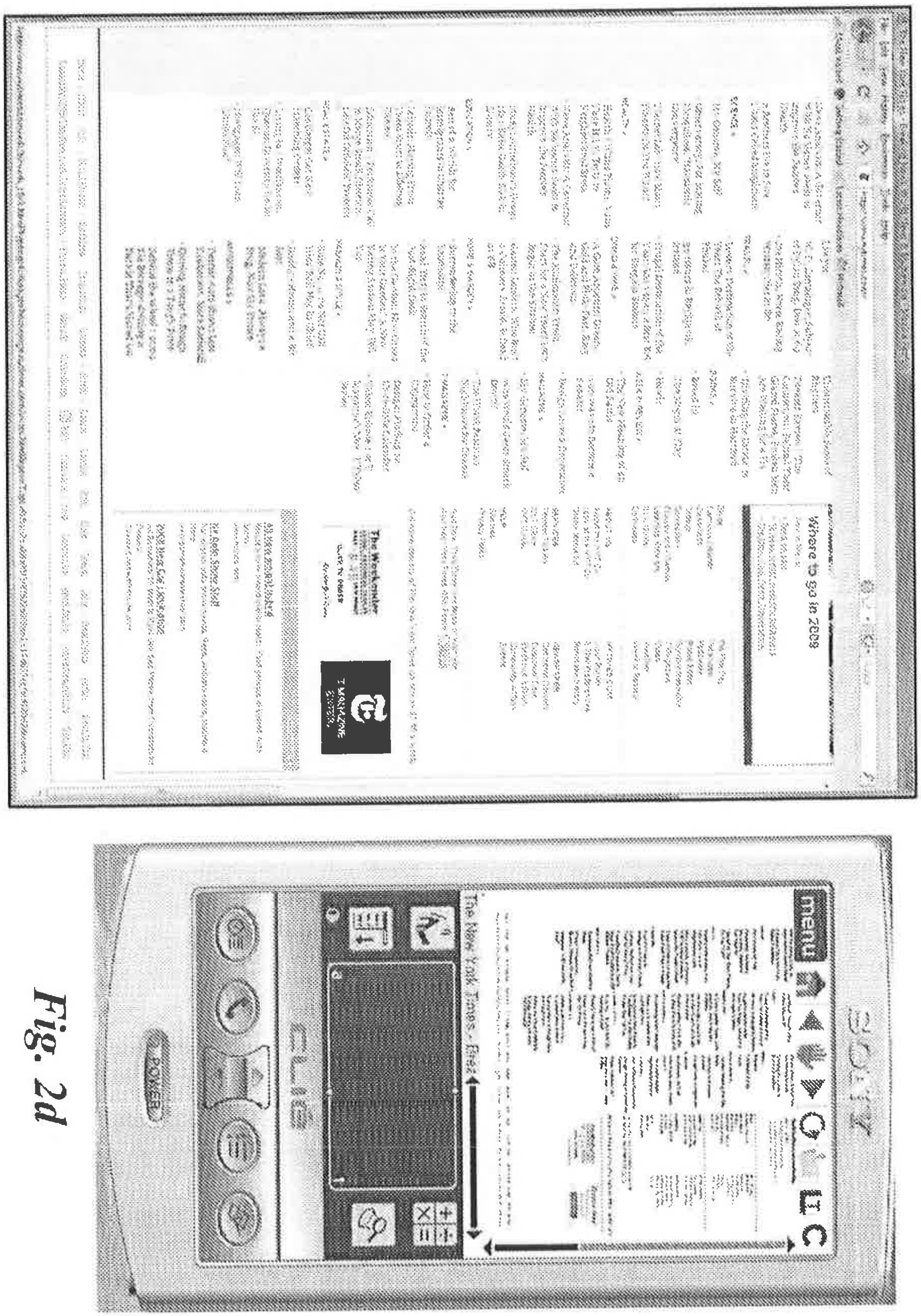


Fig. 2d

Figs. 3a-3f show various portions of the nytimes.com Web page in a full-width view (above) and corresponding zoom views (below) using the SoftView™ implementation. The above view shows a screen capture of the emulator, while the below views comprise screen captures of just the display. As can be readily observed, each zoomed-in view preserves the original page layout and design of the Web page, as interpreted by the SoftView™ implementation.

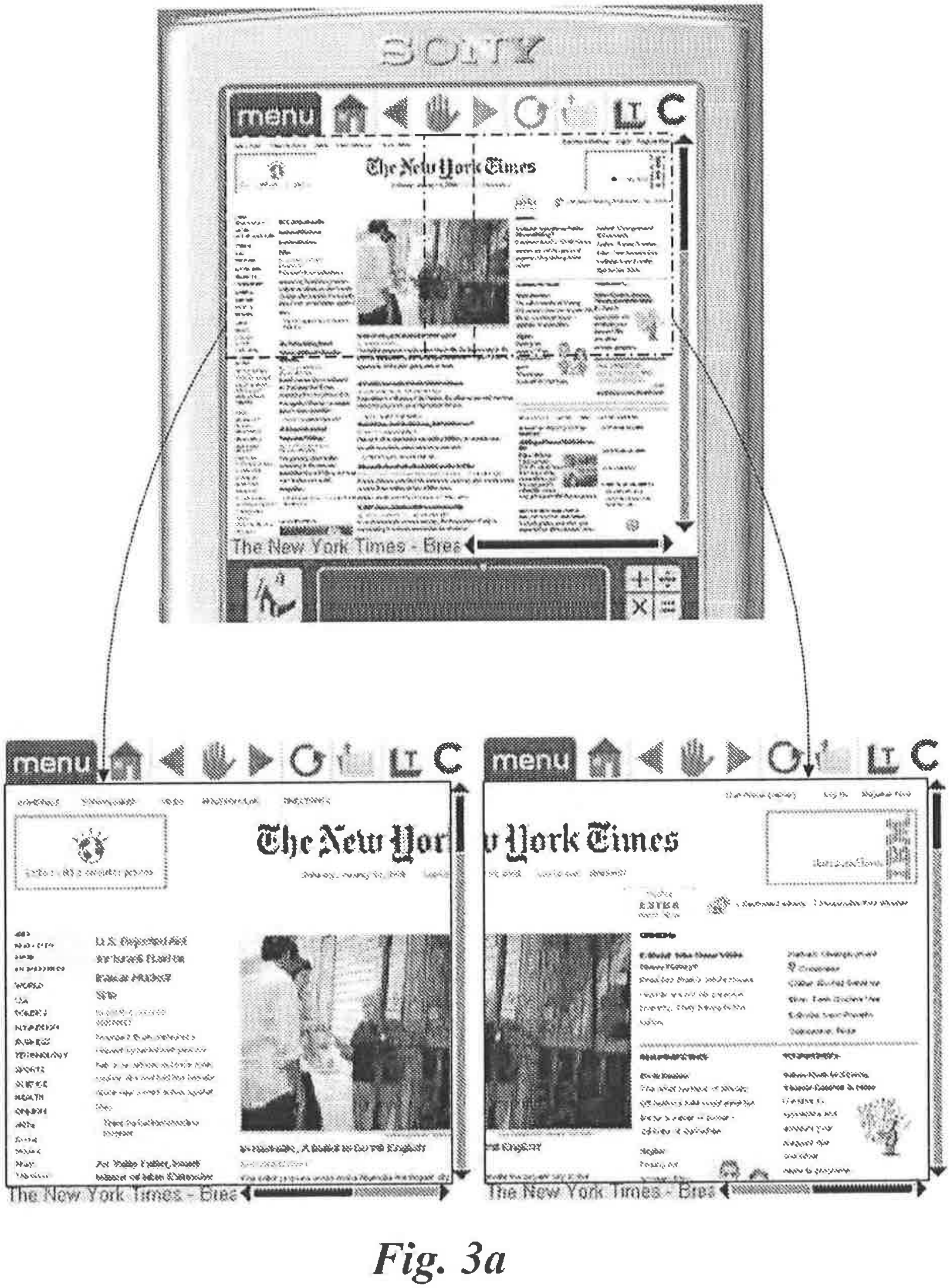

SONY
menu
The New York Times
The New York Times - Bres
menu
The New York Times - Bres
menu
The New York Times - Bres

Fig. 3a


SONY
menu
The New York Times - Bres
menu
The New York Times - Bres
menu
Listen in on the conversation.
The New York Times - Bres

Fig. 3b

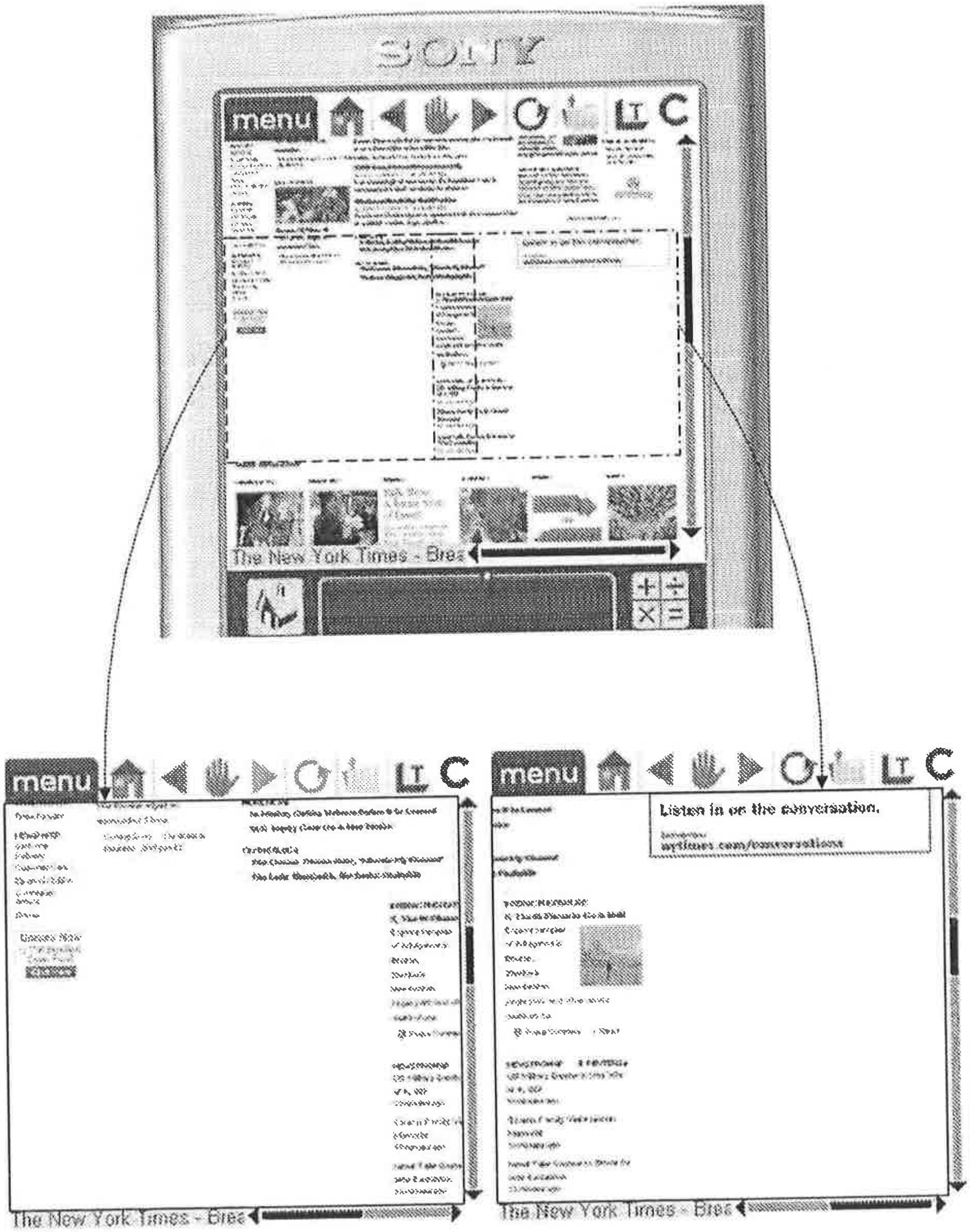

SONY
menu
The New York Times - Bres
menu
The New York Times - Bres
menu
Listen in on the conversation.
nytimes.com/conversations
The New York Times - Bres

Fig. 3c

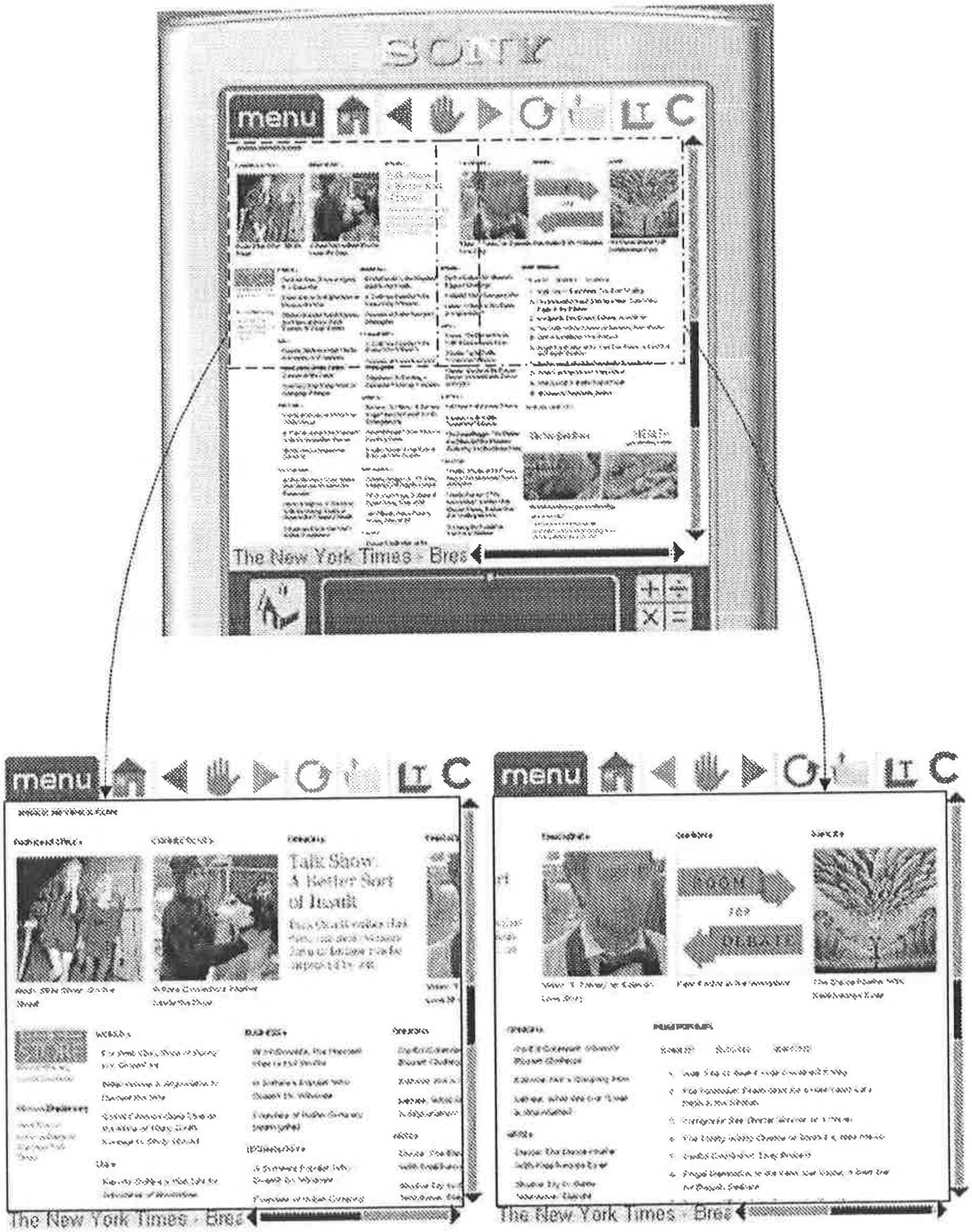

SONY
menu
LT
C
Talk Show:
A Better Sort
of Insult
ROOM
FOR
DEBATE
The New York Times - Bres
The New York Times - Bres
The New York Times - Bres

Fig. 3d

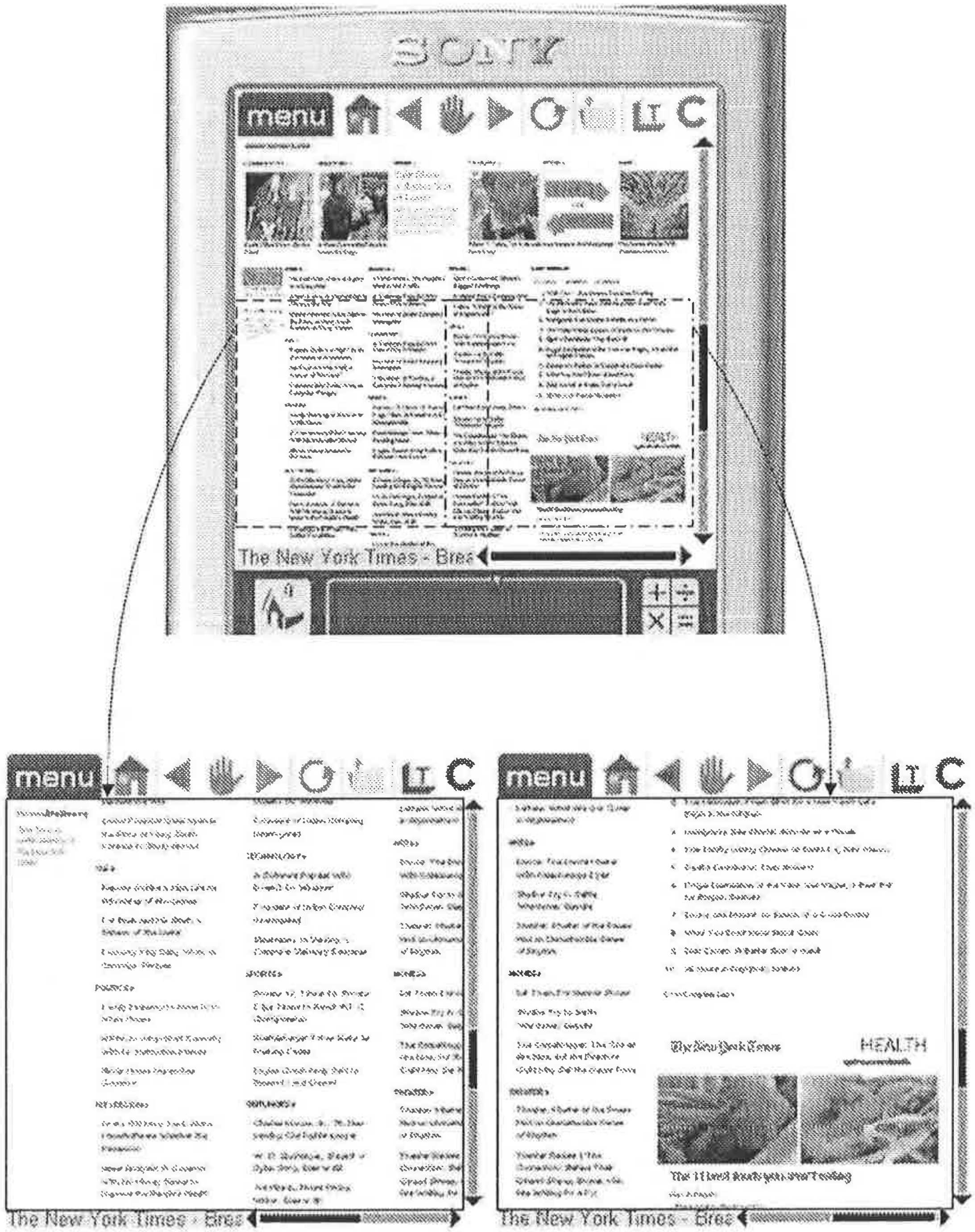

SONY
menu
LT
The New York Times - Bre
menu
LT
The New York Times - Bres
menu
LT
HEALTH
The New York Times - Bres

Fig. 3e

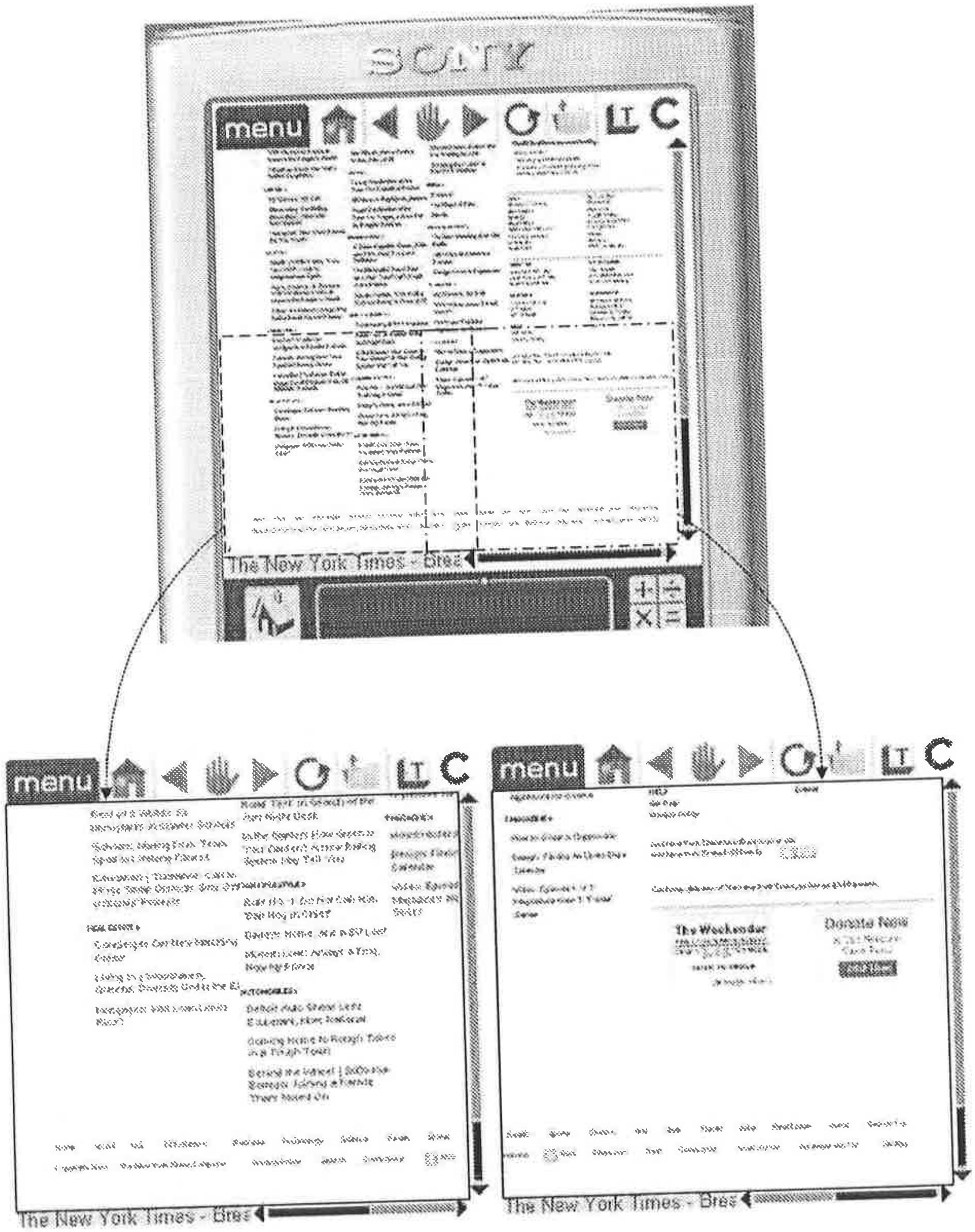

SONY
menu
The New York Times - Bres
menu
The New York Times - Bres
menu
The New York Times - Bres

Fig. 3f

Figs. 4 and 4a show various zoomed-in views of an INSIDE NYTIMES.COM content "ribbon." Figs. 5-7 show selected zoomed-in views of the upper portion of the page, while Figs. 8 and 9 show various zoomed-in portions of the bottom portion of the page. It is noted that the "Ads by Google" portion is missing on the SoftView™ browser screen shots due to a server connection timeout error (which caused this content to not be retrieved).

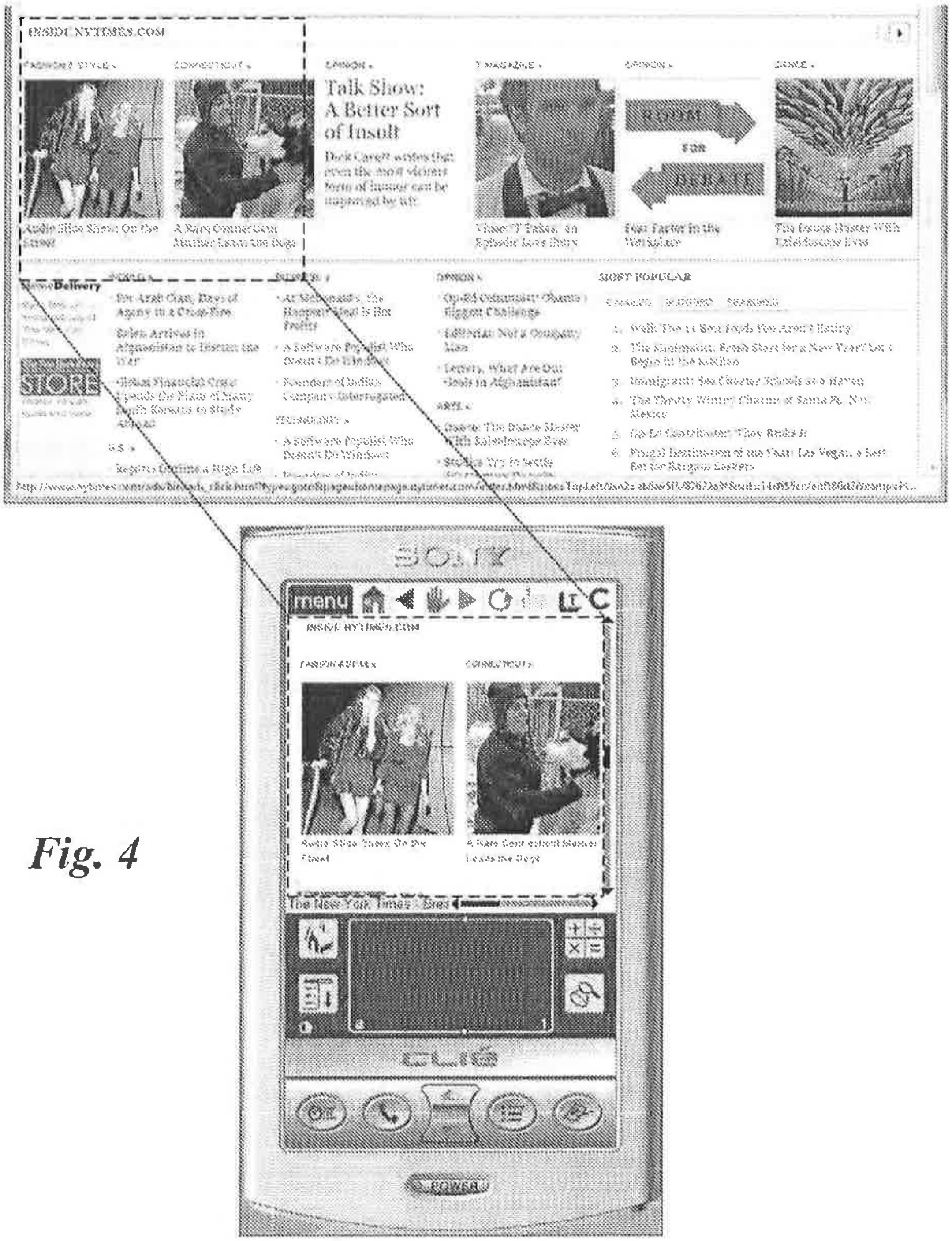

INSIDE NYTIMES.COM
Talk Show:
A Better Sort
of Insult
ROOM
FOR
DEBATE
MOST POPULAR
STORE
SONY
menu

Fig. 4


Play Video
Listen in on the conversation.
nytimes.com/conversations
INSIDE NYTIMES.COM
Talk Show:
A Better Sort
of Insult
MOST POPULAR

Fig. 4a


menu
Talk Show:
A Better Sort
of Insult
Dick Cavett writes that
even the most vicious
form of humor can be
improved by wit.
The New York Times - Bres
menu
The New York Times - Bres

Fig. 5


Fig. 6
The New York Times
U.S. Rejected Aid for Israeli Raid on Iranian Nuclear Site
As Talks Falter, Israel Warns of More Extensive Attacks
In Nashville, A Ballot to Go 'All English'
A Public Servant's Private Stint is at Issue
Detroit Goes for Electric Cars, but Will Drivers?
OPINION »
Editorial: Who Owns White House History?
IN SUNDAY'S TIMES
YOUR MONEY »
When Cash Is Scarce, There's Comfort in Miles
THE LATEST ON DEAL MAKERS AND DEAL BREAKERS
nytimes.com/dealbook
DON'T MISS HISTORY IN THE MAKING
menu
LT
C
The New York Times - Bres


In Nashville, A Ballot to Go 'All English'
As Talks Falter, Israel Warns of More Extensive Attacks
A Gaza War Full of Traps and Trickery
DON'T MISS HISTORY IN THE MAKING.
Ravens 13, Titans 10
menu
NFL PLAYOFFS
The New York Times - Brea

Fig. 6

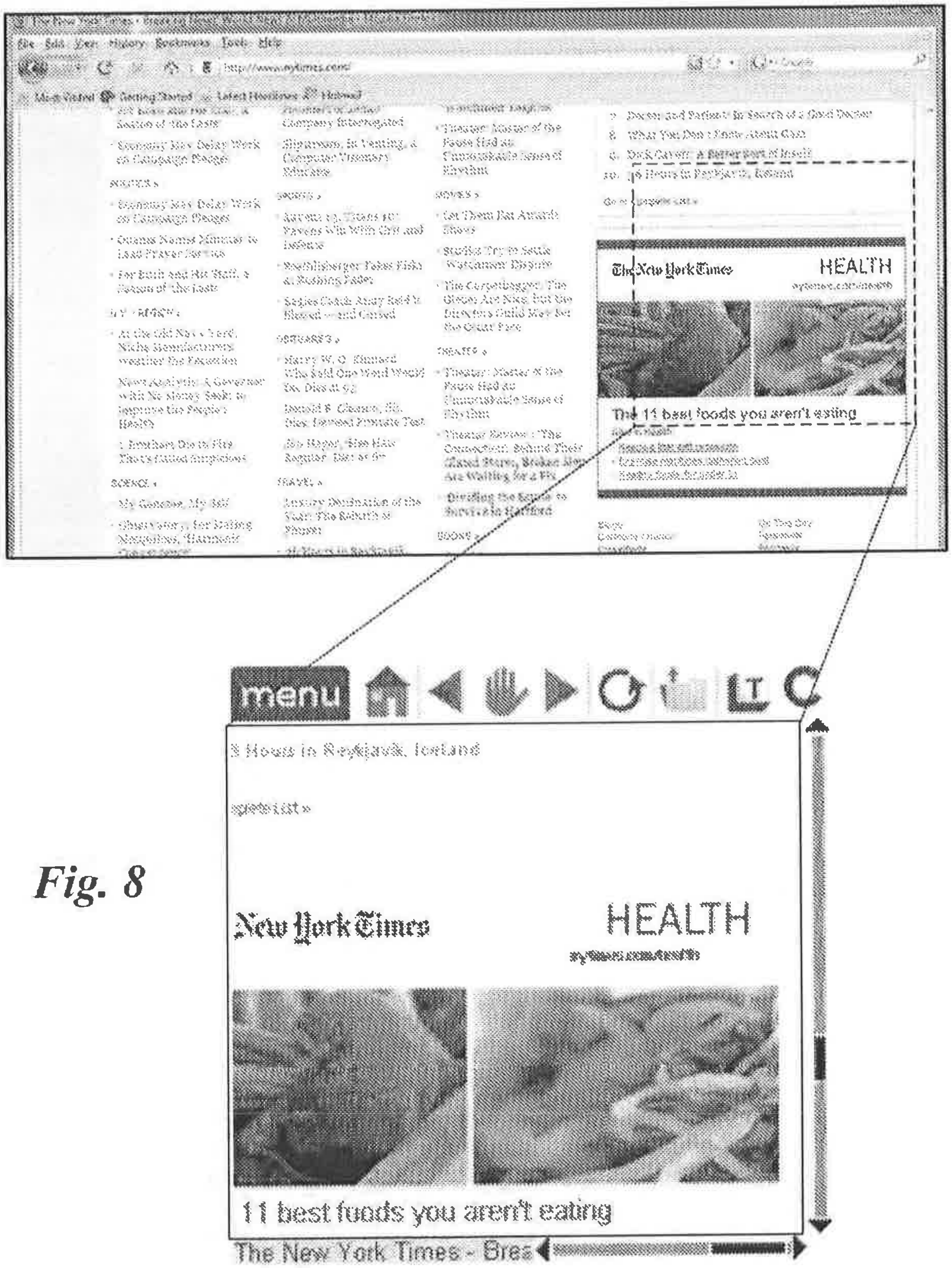

The New York Times HEALTH
The 11 best foods you aren't eating
menu
3 Hours in Reykjavik, Iceland
New York Times
HEALTH
11 best foods you aren't eating
The New York Times - Brea

Fig. 8

Preservation of Functionality

New claim language entered by amendment to claim 30 introduces the term preservation of "functionality." Preserving functionality generally pertains to preserving the interoperability of various HTML-based Web page content, such as hyperlinks and UI (user interface) controls such as input forms defined via corresponding HTML-based code. It is noted that the HTML code defining a Web page's overall layout, functionality and design does not define how a user interaction with the Web content is to be supported, but rather defines the existence of a corresponding function within the Web content to support the interaction. For example, a hyperlink definition within a Web page merely defines a link (hyperlink reference of *href*) to corresponding content, it does not define how the hyperlink associated control is to appear on the screen nor how the hyperlink is to be activated. That is up to the browser's implementation, which varies by browser. For example, some browsers underline text content associated with a hyperlink, while others change the appearance of a pointer when over a control (*e.g.*, text content) associated with a hyperlink (or otherwise change the appearance of such content). Moreover, how the hyperlink is activated is not defined by the corresponding HTML-based definition, but again is left to the browser implementation. Accordingly, preserving content functionality means that functionality defined by corresponding HTML code (*e.g.*, activation of a hyperlink in the present example) is supported, without limiting the particular user interface for how that activation is facilitated.

In the implementation of a zoomable browser, it may be desirable to change user interface behavior depending on a current use and/or view context. For example, the hyperlink controls of a conventional Web page designed to be viewed with a desktop browser are typically activated via the same user interface input (*e.g.*, clicking with a mouse), since all of the hyperlinks controls (on at least well-designed Web pages) are (presumably) designed to be viewable on the desktop browser (at least viewable to most users). In contrast, when the same page is rendered so as to fit on a handheld device's display, corresponding hyperlink controls may not be readable. As a result, it may be advantageous to implement a context-based user interface that may result in a different action for the same user input depending on a current user and/or zoom context. For example, under the zoom to column, image, and paragraph user interface features disclosed in the present application, touching proximate to content associated with a hyperlink control may or may not activate the hyperlink control, depending on a current zoom level. By way of illustration, when touching content proximate to a hyperlink control that is also contained within a column when in a zoomed-out view, such as a full page view, the browser may interpret the input as an input to zoom to the column rather than an input to link to a hyperlinked reference associated with the content, particularly when the content is not readable in the current view.

Enabling access to Web pages from among billions of Web pages

Applicants have amended claims 1 and 15, and included similar language in new claim 33 to recite,

"enabling a user of the mobile device to request access to Web pages *from among billions of Web pages accessible via the World Wide Web* ..."

The scope of the foregoing claim language is intended to convey that under each claimed invention (including this language), a user of the corresponding mobile device is enabled to be able to access Web pages from among billions of Web pages accessible via the World Wide Web *(i.e.*, the Internet), such as a user's favorite Web page or other Web pages selected to be accessed by the user. This language is further meant to preclude coverage of or anticipation by an apparatus or method that requires Web pages to be encoded in a particular manner to provide functionality specific to a corresponding apparatus or technique that is required to support corresponding aspects of the claimed invention.[19] In other words, the claimed inventions can be used to

---

[19] This does not mean the Web pages among the billions of Web pages (corresponding to the claims herein) cannot

access billions of Web pages in their existing form, without requiring any modification or special coding of the Web pages. Meanwhile, any apparatus, technique or scheme, etc., that requires a Web page to be specifically encoded to support corresponding functionality applicable to that apparatus, technique, or scheme will be excluded from the claim coverage (or from being used for anticipation or obviousness purposes) because there were not billions of existing Web pages that were encoded in a similar manner at the time of their respective claimed inventions.

At the same time, the foregoing claim language does not mean the user is enabled to access <u>any</u> Web page accessible via the World Wide Web, as some Web pages may only be accessed via use of the proper security credentials (*e.g.*, userID and password), while pages having display content written in languages that are not supported by the browser and/or operating system implemented on the host device might be accessible but would not be rendered correctly[20].

At the time of the filing of the non-provisional parent application (US 09/878,097 – issued as US 7,210,099) (June 8, 2001) to which the present application claims priority, there were on the order of a several billion web pages associated with the "World Wide Web" and accessible via the Internet, with the specific number being

---

include any content that is defined for mobile devices. For example, the use of Metatag values in Web page code, such as used to define page widths, does not constitute encoding a Web page in a particular manner to provide functionality specific to a corresponding apparatus or technique that is required to support corresponding aspects of the claimed invention.

[20] One of skill in the browser art would recognize that rendering engines do not render Web pages completely by themselves, but rather employ various support functions provided by the host operating system for particular rendering operations. Among these support functions is support for rendering text in various languages. The particular languages that are supported will vary depending on the operating system and/or extensions to the operating system (or otherwise add-on functionality provided by the browser) for rendering text of a particular language. If support for rendering text in a given language via either the operating system or a particular extension is not available, the text content in such a language will not be able to be rendered on pages that include such text content.

somewhat indeterminable. As stated in paragraph [0093] of the present application, ". . . users are enabled to view the entire content of billions of existing Web pages using hand-held devices in a simple and reasonable way." This statement was based on the observation that, when tested, a browser incorporating the principles of the invention disclosed in the present application enabled the test user to browse, zoom, and pan the entirety of nearly every Web page that was tested.[21] Based on the inherent principles and teachings disclosed, this result was expected, as the rendering engine employed by the browser (the Mozilla rendering engine) was based on the same rendering engine used in one of the two most dominant browsers at the time (*i.e.*, the rendering engine used by the Netscape Navigator browser, with the other dominant browser being Microsoft Internet Explorer). (It is respectfully noted that the use of the Mozilla rendering engine in an embodiment in the present disclosure is merely exemplary, and not limiting in any way.) Since Netscape Navigator was a dominant browser at the time, many if not most Web pages were designed to support browsing with Netscape Navigator (either by intent and/or based on good HTML coding practices for manually designed pages, or through use of one of many Web page design tools that generated HTML that could be properly interpreted by Netscape Navigator).

One of skill in the art will recognize that the principles and teachings disclosed in the present application may be applied in a browser implementation employing one of many different rendering engines, such as but not limited to today's version of the Mozilla rendering engine ("Gecko") used by the Firefox browsers, the rendering engine employed by Microsoft Internet Explorer ("Trident" (*aka* MSHTML)), or the Webkit rendering engine use by Apple's Safari browser, Google's Chrome browser, and mobile

---

[21] Of approximately 500 of the most browsed (at the time) Web pages that were tested, only a handful did not work. Of particular note was a Sony Web site that was substantially flash-based (and thus not HTML-based). It is also noted that some Web pages were/are designed to be browsed by a specific browser, such as Internet Explorer; such pages may not render and/or function properly under other browsers.

browsers including Apple Mobile Safari, Google Android, Nokia S60, Palm Web OS browser, and Research in Motion's recently introduced browser. Since each of these rendering engines are capable of rendering the vast majority of today's Web pages, a browser implementing such a rendering engine in combination with the principles and teachings disclosed in the present application would likewise be capable of rendering the vast majority of billions of today's Web pages while preserving the page layout, hyperlink functionality, and design of the Web pages under various zoom levels and panned views.

Plurality of zoom levels; at least one of the zoom levels is at least 4 times full scale; Plurality of scale factors; at least one of the scale factors is at least 4 times full scale

Claim 1 has been amended to recite, in part

"wherein in the multiple zoom levels include zoom levels that are not predefined, and the user is enabled to view the entirety of the HTML-based content of the Web page at a zoom level of at least 4 times full scale, wherein full scale corresponds to a native pixel resolution of the Web page as interpreted by the rendering engine."

In addition, each of claim 15 and new claim 33 recite, in part,

"scaling the page layout information to generate scaled Web page content comprising scaled objects at a plurality of scale factors, wherein the original page layout of the objects as interpreted by the rendering engine is preserved at each scale factor, and at least one of the scale factors is at least 4 times full scale, wherein full scale corresponds to the native pixel resolution of the Web page as interpreted by the rendering engine."

First, as a general comment, Applicants want to reiterate that while a 1:1 relationship between zoom levels and scale factors may exist, there is not a requirement that content displayed at a given zoom level be directly derived by scaling content at an equal scale factor, although such may be the case. However, in the case of the terminology "full scale," a zoom level of 1x (1 times full scale) and a scale factor of 1x (1 times full scale) are equivalent.

The HTML-based content of a Web page includes HTML elements that define the page layout of the objects on the Web page, as discussed above. Among the layout information that is determined by a rendering engine is the size of the Web page, including its Width and Height. The Width of the Web page may be explicitly defined by the HTML-based content, be defined by aggregating layout information, or determined by using a default page width and/or the width of the content area for the browser. The latter technique is employed for pages that do not provide HTML-based content from which a page width can be explicitly determined, such as pages employing one or more variable-width content areas (*e.g.*, such as pages on Wikipedia.com.) Thus, the native pixel resolution of the Web page is the Width and Height resolution of the page in pixels, as interpreted by the rendering engine.

When the aspect ratio of the page is maintained when scaled, there is only one of the Width and Height values that needs to be considered for determining the scale, and that is the Width. At full scale or 1x, the pixel-based resolution of the Web page content presented on the display is the same as the resolution of the page if displayed at its native resolution. On a typical modern desktop monitor, the full page width of a Web page can generally be displayed when the content area of the desktop browser's application window is wide enough. For example, many Web pages are designed to have a native resolution of approximately 980 pixels wide so that they can be displayed at full-width on a monitor having a pixel resolution width of at least 1024 pixels (with consideration that a small portion of the display screen is used for a vertical scroll bar and the window frame).

In the following example, the window of a Firefox 3.6.10 browser has been adjusted such that the width of the body block of the Web page is 960 pixels, and as depicted by the Firefox Firebug inspector, the body block of the Web page has a size of

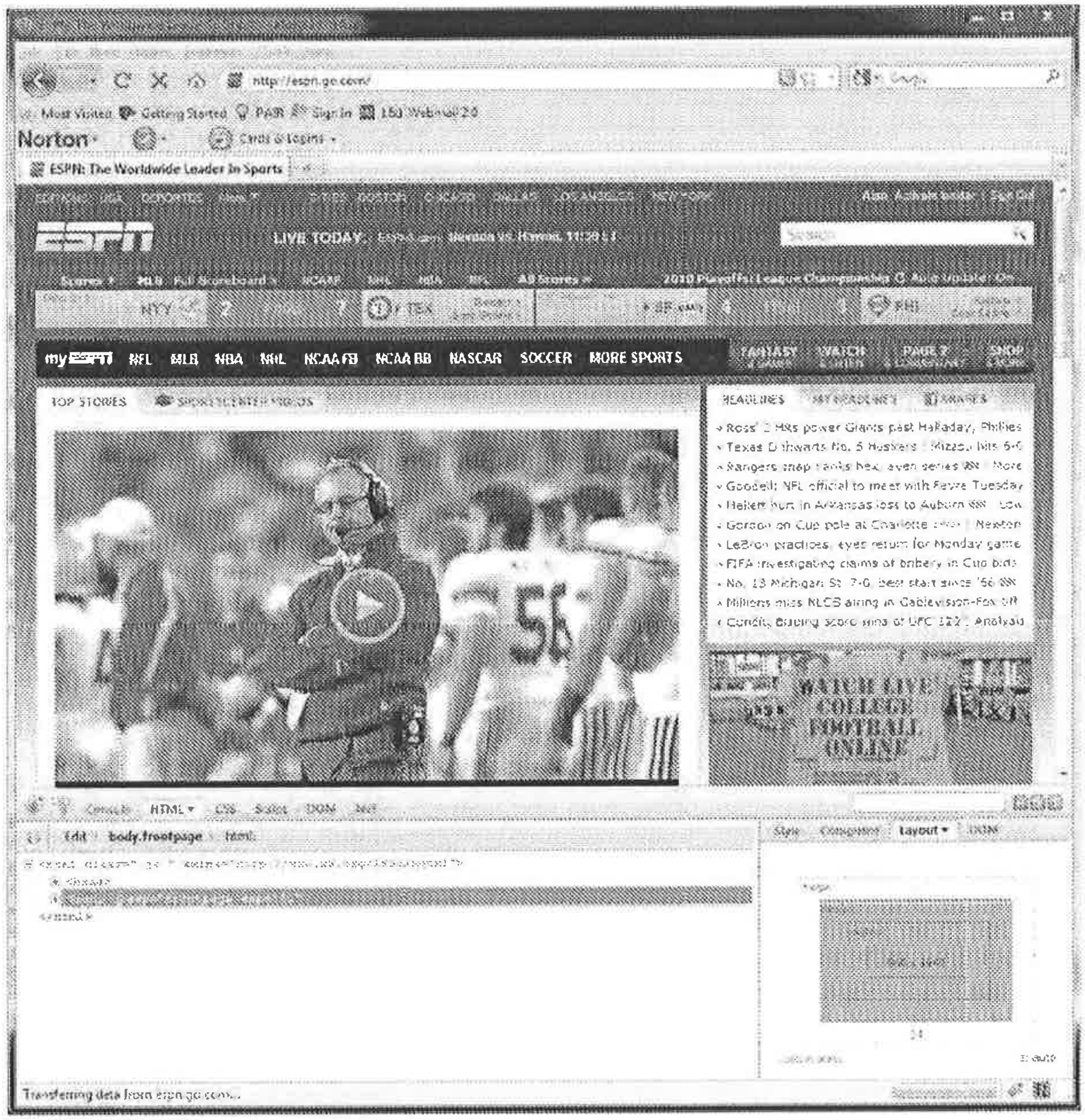
960 x 1948 pixels. For simplicity and point of illustration, we are going to state for this example page that its native resolution is 960 x 1948 – that is, the size of the body block[22].

---

[22] It is noted that some desktop browsers add padding or margin around the body block of Web pages, such that it could be argued the native resolution is actually the size (in pixels) of the body block plus the padding/margin. However, since the padding is not specified by the HTML (it is added by the particular browser implementation), for consistency herein the native resolution will be considered the size of the body block, as interpreted by the rendering engine.


http://espn.go.com/
Norton
ESPN: The Worldwide Leader In Sports
LIVE NOW:
myESPN NFL MLB NBA NHL NCAAFB NCAABB NASCAR SOCCER MORE SPORTS
TOP STORIES
HEADLINES
Ross' 2 HRs power Giants past Halladay, Phillies
Texas D thwarts No. 5 Huskers
Rangers snap Yanks hex, even series
Goodell: NFL official to meet with Favre Tuesday
Mallett hurt in Arkansas loss to Auburn
Gordon on Cup pole at Charlotte
LeBron practices, eyes return for Monday game
FIFA investigating claims of bribery in Cup bids
No. 13 Michigan St. 7-0, best start since '66
Millions miss NLCS airing in Cablevision-Fox tiff
Condit, Bisping score wins at UFC 120
WATCH LIVE COLLEGE FOOTBALL ONLINE
One Big Disappointment
Jim Tressel and Ohio State spent one week at No. 1 before falling to Wisconsin.
Ohio St. Upset
Nebraska Falls
Giants Win
Texas Ties Series
Season-Ticket Rankings
Kobe vs. LeBron
Scouts Inc.
Transferring data from espn.go.com...

ESPN.com Web Page



Firefox Firebug showing body block has a size of 960 x 1948 pixels

Now further suppose that the display resolution of a hand-held device is 320 x 320 pixels (*e.g.*, the resolution of the Sony Clie discussed above). At a full scale or 1x zoom level, a pixel on the display screen maps 1:1 with a corresponding pixel in the native resolution of the web page. This is illustrated in the following figure, were the

320 x 320 block in the upper left hand corner of the page represents the pixel area of the display screen, and the shaded translucent area represents the portion of the display used for viewing Web page content, which is approximately 304 x 272 pixels. In this example configuration[23], the upper portion of the display is used for a Menu, the right hand edge is used for a vertical scroll bar, and the lower portion is used for an information bar and a horizontal scroll bar.

[23] This illustrated configuration is merely exemplary, as other configurations may be employed, including configuration without areas that are allocated for menus, information bars, scroll bars, etc.

Depiction of Web page view on example display configuration

It will be understood that when page content is rendered at full scale, the corresponding zoom level is 1x or 1 times full scale. Similarly, within the scope of applicable claims herein, a scale factor of 1x or 1 corresponds to 1 times full scale. In general, the zoom level referred to in claim 1 and the scale factor referred to in claims 15 and 33 will have a value of *n*x, meaning *n* times the resolution of full scale, wherein the scale *n* is relative to the full scale resolution and associated page layout as interpreted by the rendering engine.

At a zoom level of 4x or a scale factor of 4x (also referred as four times full scale), the effective resolution of the Web page is 4x (*i.e.*, 4 times) the native resolution of the Web page. Thus, in this instant example, the effective resolution at the Web page when scaled using a scale factor of 4x would be 3840 x 7792 pixels. The word "effective" is used in the foregoing statements to convey that this would be the resolution in pixels of the entire Web page if it were to be fully rendered on a display (or to an off-screen bitmap context) at a given point in time. In practice, at a minimum only a portion of the Web page comprising a bitmap having the pixel resolution of the Web content display area for a given device would need to be rendered to the display for a given view. Viewing content corresponding to the full Web page is achieved by panning around the full Web page using the Web content display area as a viewport of the Web page. Rendering of additional off-screen bitmap content for purposes such as to facilitate panning or storing one or more previous views would be optional.

<u>Zoom levels that are not pre-defined</u>

Claim 1 has been amended, in part, to recite,

"wherein in the multiple zoom levels include zoom levels that ***are not predefined***, ..."

Embodiments of the present invention, such as implemented by the SoftView™ browser, enable users to view Web page content at multiple zoom levels, including "arbitrary" zoom levels that are not predefined. For example, an arbitrary zoom level may result from a user selection of an area to zoom on, or a context zoom, such as results from zooming on column, image, or paragraph. Since the user selected area or column, image, or paragraph may be arbitrary in dimension, the corresponding zoom level which is derived as a function of such dimension may likewise be arbitrary. By contrast, a pre-defined zoom level corresponds to zoom level that is defined in advance, such as 100%, 125%, 150%, 200%, 50%, *etc.*

No degradation in quality of the presentation of text content

Claim 1 has been amended to recite, in part,

... wherein there is ***no degradation in quality of the presentation of text content*** *when the zoom level is increased to at least 4 times full scale*. Support for this claim element is based on principles inherent to the use of scalable vector-based fonts, and analogous language in paragraph [0037] of the specification.

Under implementation of aspects of the present disclosure, there is no degradation in quality of the presentation of text content when the zoom level increases (upon completion of a change in zoom levels, as discussed further below). As will be understood by those skilled in the art, the quality of the presentation of text content pertains to the clarity of how text characters and corresponding font glyphs are displayed in Web page views, as perceived by the naked eye.

In some embodiments scalable fonts are used, such as exemplified by, but not limited to, TrueType fonts. Scalable fonts of this class are sometimes termed "vector fonts" or "outline fonts." [24] In brief, the outline of the glyphs used to build the text characters is described by mathematical formulas, such as Bezier curves[25]. Typically, a closed path is described by the mathematical description of the glyph, and the path is then filled. Glyph hinting may be applied to alter the control points that define the outline, with the objective that the rasterizer produces fewer undesirable features on the glyph. Since the glyphs are based on mathematical formulas, they can be scaled with substantially indefinitely, with no degradation in quality.

Although the glyph outlines may be defined using high-precision math, there ultimately is a rasterizing operation that maps the mathematically-defined glyph outlines onto pixels, which reduces the precision and sharpness of corresponding text

---

[24] See http://en.wikipedia.org/wiki/Computer_font for an excellent discussion of computer fonts.

[25] The mathematical definitions are generally referred to a vectors or vector images, hence the terminology vector fonts.

characters. As is well known, various approaches such as anti-aliasing is employed to make the outlines of the characters appear smoother (*i.e.*, less jagged). Such minor discontinuities in the smoothness of the outlines are generally more pronounced, for smaller size fonts and for displays with lower pixel densities.

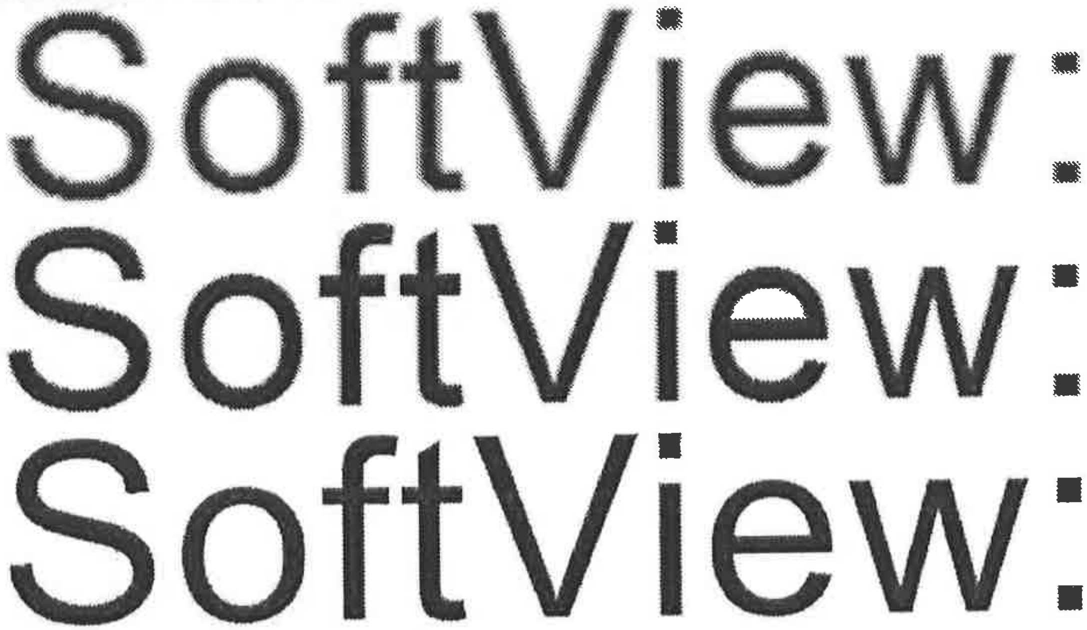
The following screenshots illustrate examples of presentation of text content that result in no degradation in quality of the presentation of text content when the zoom level is increased, and examples of when such degradation results. These screenshots, which are reduced in scale herein to fit the width of the page, where obtained using a desktop SoftView™ browser client implementation running on a version of the Microsoft Windows 7 operating system. It is noted the same proxy server as used in the www.nytimes.com examples presented above was used. However, the text is now rendered by the Windows 7 OS using scalable fonts, as opposed to the text in the Palm emulator examples, which employ a bitmap font at a lower resolution. The text is from a portion of the www.softview.us page.

SoftView:

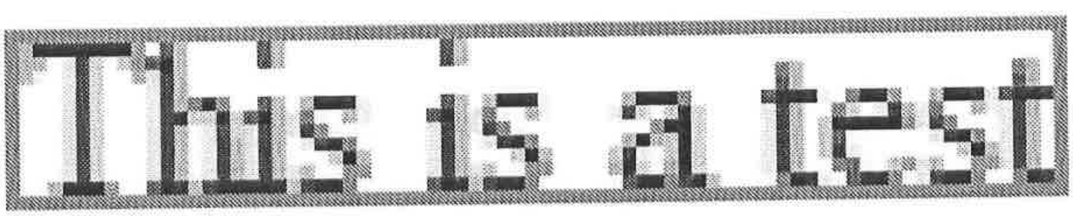
SoftView:

SoftView:

The top example shows the text "SoftView:" at a zoom level of 100%, or 1x (*i.e.*, full scale). The middle shows the text when the browser client was at a zoom level of 200% or 2x, and the bottom shows the text when the browser client was at a zoom level of 400% or 4x. Each of these screenshots comprises a bitmap. In the following representation, the bitmap of the text at 100% has been scaled by a factor of 4x, while the bitmap of the text at 200% has been scaled by a factor of 2x such that the text is now the same size. This illustrates an example of degradation in the quality of the presentation of text content.

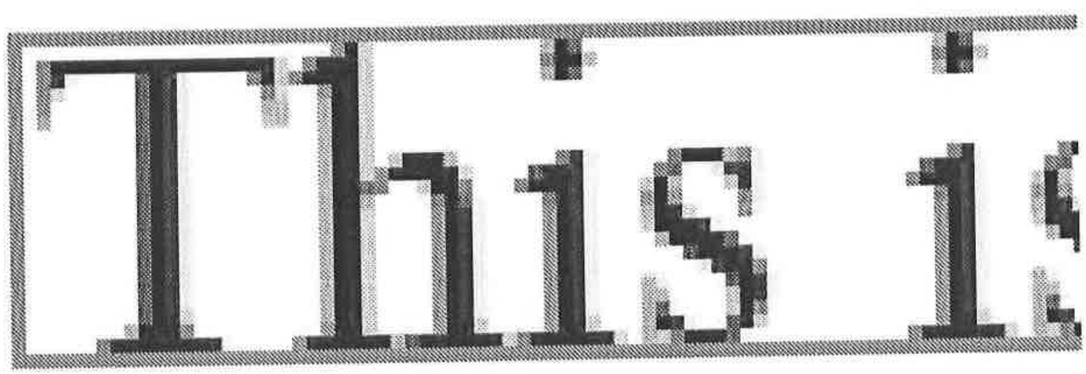


The screenshot of the scaled bitmaps were derived by copying the original 1x and 2x text into Microsoft Paint, changing the zoom level to 400% and 200% respectively, and then capturing new screen shots. It is believed that Microsoft Paint employs a simple bit "blowup" when the zoom level is changed, as demonstrated below:

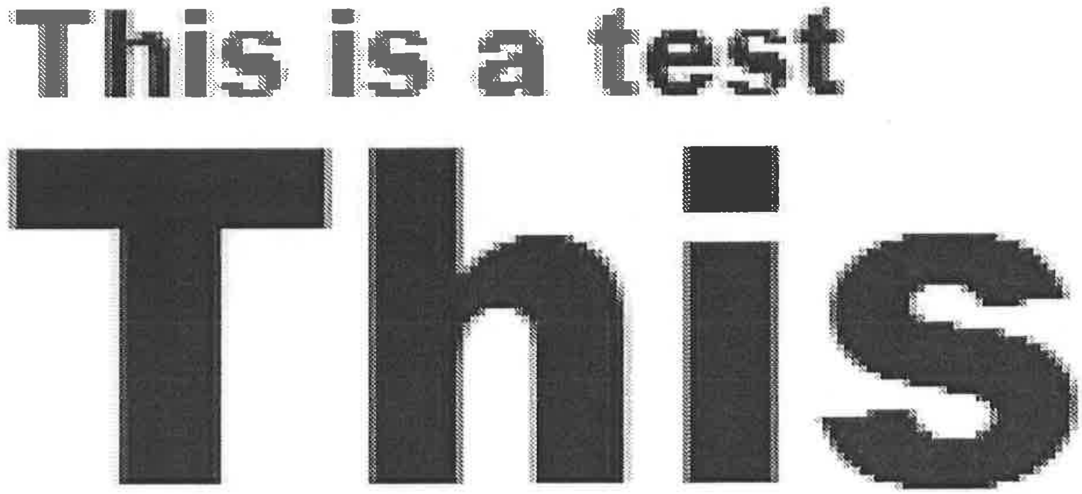


In this screenshot, each square block represents an original pixel of a 12 point Times New Roman font that has been blown up by a factor of 800%. This also shows an example of anti-aliasing, wherein pixels of different colors are selectively employed to make the font outline look smoother to the naked eye.

In general, the relative portion of pixels employed for anti-aliasing of a font glyph to the pixels in the glyph is not proportional with font size. As the font size is increased, the relative portion of pixels employed for anti-aliasing is reduced, since these pixels are usually employed using different colors to augment the outline of the font glyphs. This can be shown graphically by comparing text using various fonts (and corresponding font glyphs) at different sizes. The following screenshots respectively show the same text as above using a 48 point Times New Roman font blown up by 200% and a portion of the text using the 48 point Times New Roman font blown up by 800%:



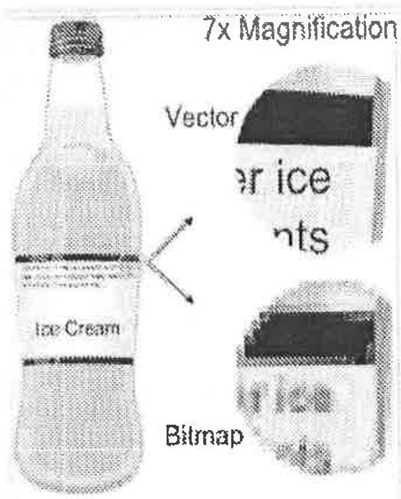
The effect is more pronounced with fonts employing wider glyphs, such as bolded fonts. For example, the following screen shots show 12-point and 48-point Arial Bold font blown up by 800%:



The following screen shots show the same 12-point Arial Bold font blown up by 800% compared with a 48-point Arial Bold font blown up by 200%:

In the above example, as well as the SoftView™ screenshots, the portion of pixels along an axis employed for rendering a font glyph is proportional to the zoom level. As a result, there are more pixels available for rendering the outline of the font glyphs at relatively higher zoom levels, resulting in increased sharpness in text content as the zoom level is increased.

It is generally known to those of skill in the computer graphics art that the use of vector fonts, such as but not limited to TrueType fonts, enable corresponding text content to be scaled substantially indefinitely with no degradation in quality. For example, the following illustration is included on a Wikipedia page concerning vector graphics.[26] It is noted that in this example both the graphical definition of the bottle and the text content on the label is defined by corresponding vector graphic elements; however, the point of inclusion of this example is directed at just the text content. As stated, "raster images are based on pixels and thus scale with loss of clarity, while vector-based images can be scaled indefinitely without degrading quality." In the context of this statement, quality concerns the presentation of the scaled vector-based content when rendered to a display context, such as a computer screen or printer output. In the context of claim 1, this claim element applies to the rendering of text content.



Example showing effect of vector graphics versus raster graphics. The original vector-based illustration is at the left. The upper-right image illustrates magnification of 7x as a vector image. The lower-right image illustrates the same magnification as a bitmap image. Raster images are based on pixels and thus scale with loss of clarity, while vector-based images can be scaled indefinitely without degrading quality

[26] http://en.wikipedia.org/wiki/Vector_graphics. It is noted that in the Bitmap magnification, it is likely the next neighbor interpolation algorithm was used, which generally produces the greatest loss in clarity.

Applicants would like to note several important considerations that shall be understood with respect to the scope of the terminology,

***"no degradation in quality of the presentation of text content** when the zoom level is increased to at least 4 times full scale."*

First, the "no degradation in the quality of the presentation of text content" aspect concerns the quality of the presentation of text content when an operation that results in a change in zoom level is completed. This means that if animation is employed during a zoom operation, the quality of the presentation of text content during the animation is irrelevant to the claim scope.

A second consideration is the claim scope applies to text content corresponding to text objects in the Web pages, and is not applicable to text depicted in images (corresponding to image objects in the Web pages). For example, many advertisements included in Web pages comprise images depicting text, and are stored as image files (*e.g.*, JPEG images, GIF images, bitmap images, etc.) on corresponding servers, such as shown below.[27]

---

[27] The screenshots were captured from the www.nytimes.com Web page using the Firefox 3.6.12 browser on Microsoft Windows 7. The original HTML definition is <img height="90" border="0" width="184" src="http://graphics8.nytimes.com/adx/images/ADS/24/87/ad.248781/movado_sov_left.jpg">.

Original size (full scale)

Scale approximately 3 times full scale

The clarity of text included in such images will generally degrade when the images are scaled (depending on the amount of scaling and the type of bitmap interpolation used), such as illustrated by the blurred text content in the 3x (3 times) full scale image. This is due to aspects of scaling bitmap images, as discussed below.

Third, the "no degradation in quality" corresponds to how the quality of text presentation relative to the text presentation at full scale, as perceived by the naked eye, as considered by one skilled in computer graphics art. As discussed above, the no degradation in quality aspect is a recognized result (to those of skill in the computer graphic art) of using scalable vector fonts.

Several of the references identified during prosecution of the present application and related family members concern techniques for providing Web page content to small client devices such as PDAs under which Web pages are rendered using a conventional Web browser on a server or the like and then bitmap images of portions of the Web page comprising entire views (corresponding to the screen resolution of the small devices) are transmitted to the client device to be replicated on the client device's screen. For example, one technique that was employed proximate to the filing of the earliest application sharing the same disclosure of the present application to which priority is claimed (June 8, 2001)[28] was to generate a bitmap image of a rendered Web page at full scale on a server, and then send portions of the bitmap (or corresponding bitmap content after a color reduction or color space translation was applied) to a client in response to requests from the client. Typically, a zoomed in view resulted in viewing the Web page content at full scale, which as discussed above, corresponds to a zoom level of 1x. The quality of the display content (on the client device) was very good at full scale, since the bitmap portions of the Web page represented by the views sent to the client were derived directly from bitmap content corresponding to the rendering of the Web page at full scale on the server using a conventional Web browser. Thus, at full scale, there is no loss in quality with respect to text content in the original Web page. To zoom out, the full scale bitmap was simply downsampled at the server using a basic interpolation technique, such as next neighbor, which has a very low computational cost, and a corresponding bitmap was sent to the client. This produced a degraded appearance of the Web page when zoomed out at moderately low zoom levels (e.g., .5x), and very degraded appearance at low zoom levels (e.g., .25x). This result is expected, since by definition the resolution of the downsampled images corresponding to the bitmap views sent to the client device is lower that the effective resolution of the full scale views. For example, compare the resolution of a typical image thumbnail to a

---

[28] Applicants do not admit that such techniques qualify as prior art under §102 or §103, and reserve the right to swear behind references teachings such techniques, if applicable.

full image – the thumbnail is achieved by downsampling the bitmap of the full image and has a much lower resolution then the full image.

A similar reduction in quality results when bitmap image content is scaled upward to support viewing content at zoom levels of greater than full scale. In this instance, since the number of pixels to be represented in views at zoom levels greater than full scale is greater than the number of pixels at full scale, there is not enough information in the sampled image to reproduce a scaled image without performing interpolation to "fill in" color values for the extra pixels. The greater the scaling of the bitmap image, the greater the amount of interpolation, leading to degradation in quality, particularly for text content.

By definition, a bitmap image in its original (or raw) form is resolution dependent. In its original form a bitmap image is sometimes referred to as a "sampled" image, meaning the individual pixels represent samples of an original image. This definition works well when considering how original images are digitized (*i.e.*, converted into a digital form). The converting device, such as a digital camera or scanner, acquires samples of the image (using a CCD array, for example) and converts the samples to corresponding bitmap content. For example, a 1 megapixel digital camera operating at its maximum resolution would acquire 1 million samples using its CCD array. Thus, the digitized image in raw form (when processed and rendered to a display) would likewise comprise 1 million pixels.

Generally, storing bitmaps in their original form is inefficient, both in terms of storage size and in terms of transmission speed. As a result, bitmap content is typically converted (compressed) into a more compact form obtained by processing the raw bitmap content with a corresponding algorithm, such as JPEG (Joint Photographic Experts Group), GIF (Graphics Interchange Format), TIFF (Tagged Image File Format) or PNG (Portable Network Graphic). Each of these algorithms employ "lossy" compression, meaning some of the original bitmap content is lost. In addition to storing image content in a compressed form, images stored in each of these formats can be scaled up or down using a corresponding Codec. Since the algorithms are lossy, there is a corresponding loss in quality in the scaled images, with the level of degradation being a function of the amount of scaling, the compression ratio, and the original content itself. Also, aspects of several of these compression algorithms tend to preserve the quality of image content relating to natural scenes better than text content.

Another consideration when translating image content relates to color space mapping. Generally, the sample information for each sample is stored in a corresponding form that is defined by a color space having an associated color resolution. For example, a typical 24-bit sample stores corresponding values for Red, Blue, and Green color 8-bit values. If the color resolution of a destination device differs from the color space used to store the image, a color space mapping is required.

While JPEG, GIF, TIFF and PNG are typically employed for storing bitmap image content in a compressed form, there are other types of bitmap algorithms that are usually used for scaling bitmap content, commonly referred to as bitmap interpolation. These most common of these include next neighbor, bilinear, and bicubic interpolation. More recently, more complex algorithms have been introduced, including, hqx and supersampling algorithms. In addition, other types of algorithms, including wavelet-based algorithms, may be used for scaling image content. The selection of which of these algorithms to employ is generally based on processing "cost" of the algorithm in consideration of the processing and memory resources available on the device and speed at which the scaling needs to be performed. For example, if speed is important, an algorithm that has a lower processing cost may be used. On the other hand, if the most important consideration is quality of the scaled image, a more processor intensive algorithm may be used.

Applicants respectfully note that none of the algorithms identified herein (and certainly none of the algorithms identified in any of the cited references) can be applied to an image bitmap at full scale to produce text content at a zoom level of at least 4 times full scale without degradation of the quality of the text content. This is because there is only a single bitmap (i.e., the full scale rendering of the page on the server) to base the interpolation on, and each these algorithms produce substantial degradation in quality of text when scaling a bitmap by a factor of 4 (which in this instance would correspond to interpolating a bitmap rendered at 1x so as to scale it to 4x). Moreover, because the effective resolution of web pages at 4x is very high and the bandwidths available at the time were very low, it would have been commercially impractical to implement a browser that could zoom at zoom levels up to at least 4x using a proxy server scheme under which the bitmap content is rendered on the server and transmitted to the client (even in compressed form). As discussed above, the effective resolution (X by Y in pixels) of a Web page at 4x comprises 16 times the number of pixels for the Web page at full scale. This would mean to browse the full Web page content at a zoom level of 4x, transmission of 16 times as much content to the client would be required.

The following example shows various levels of degradation resulting from interpolation of an example word "Wiki" using conventional next neighbor, bilinear interpolation, and bicubic interpolation algorithms, plus an addition more complex algorithm (hq2x) when simply doubling the size of text (*i.e.*, 2x) using an example font[29]. The results would be significantly worse at 4x.

---

[29] This image was obtained by capturing a screen shot of the Web page http://en.wikipedia.org/wiki/Image_scaling using monitor with a resolution of 1280 x 1024 pixels.

Scaling methods

[edit]

An image size can be changed in several ways. Consider doubling the size of the following image:

The easiest way of doubling its size is nearest-neighbor interpolation, replacing every pixel with four pixels of the same color:

The resulting image is larger than the original, and preserves all the original detail, but has undesirable jaggedness. The diagonal lines of the W, for example, now show the characteristic "stairway" shape.

Other scaling methods are better at preserving smooth contours in the image. For example, bilinear interpolation produces the following result:

Linear (or bilinear, in two dimensions) interpolation is typically better than the nearest-neighbor system for changing the size of an image, but causes some undesirable softening of details and can still be somewhat jagged. Better scaling methods include bicubic interpolation:

For magnifying computer graphics with low resolution and/or few colors (usually from 2 to 256 colors) the best results will be achieved by hqx or other pixel art scaling algorithms. These produce sharp edges and maintain high level of detail. hq2x:

It is noted that even the best result using hq2x produces a noticeable level of degradation in quality of the text content to the naked eye, such as exemplified by the waviness in the outline of the angled portions of the characters.

By comparison, the following screenshot presents the same "Wiki" text rendered by a Microsoft Windows operating system using Times New Roman font (a scalable vector font)[30], at 36pt. (1x), 72pt. (2x), and 144pt. (4x).

The difference in the result cannot be understated. By enabling a user to view text content using zoom levels of at least 4x without degradation in text quality, browsing Web pages is significantly enhanced. Moreover, as devices implement screens with higher pixel densities, the amount of scaling relative to full scale will need to increase proportionally to provide text at the same size. For example, if the pixel density of a screen is doubled, it requires doubling of the scale factor to render text at

---

[30] Microsoft formerly employed TrueType fonts, but now employs OpenType scalable computer fonts. OpenType is built on TrueType, retaining TrueType's basic structure and adding data structures for prescribing typographic behavior. See, e.g., http://en.wikipedia.org/wiki/OpenType. Screen capture using Microsoft Word (Office 2007) running on Microsoft Windows 7 Ultimate with the same monitor having a 1280 x 1024 resolution.

the same size. Under the resolution-independent approach disclosed in the present application, text content can scaled and rendered to devices of any resolution while maintaining quality of the presentation of the text content.

Applicants want to clarify that the foregoing discussion does not preclude the use of interpolation of bitmap content to achieve the claimed aspect of "no degradation in quality of the presentation of text content" at a zoom level of at least 4 times full scale. Rather such a result cannot be achieved by interpolation of a single bitmap corresponding to Web page content rendered at full scale (when such result is required to be achieved to support browsing of Web pages from among billions of Web pages available on the World Wide Web at a zoom level of at least 4 times full scale). For example, some interpolation techniques may be employed to scale bitmap content by a relatively small amount (e.g., 50%) to produce scaled text content such that degradation in quality of the presentation of text content may be unnoticeable to the naked eye. Moreover, techniques such as trilinear filtering[31] may be employed to produce text content without degradation at a zoom level of at least 4x if the resolution of at least one of the mipmaps[32] used for the trilinear filtering is close enough to the at least 4x zoom level.

---

[31] Also known as trilinear scaling or trilinear interpolation. For an explanation see, e.g., http://en.wikipedia.org/wiki/Trilinear_filtering, http://en.wikipedia.org/wiki/Texture_filtering and http://tech-algorithm.com/articles/trilinear-interpolation-image-scaling/.

[32] For example, when implementing conventional mipmap levels of detail that differ by a factor of 2 (*i.e.*, the level of details of the mipmaps are 2^n), the result of the interpolated text content will be presented without degradation if the mipmap content of at least the closest level of detail is derived by rendering text using scalable vector fonts that are scaled in accordance with the scale factor corresponding to that level of detail. Moreover, if the text content for each level of detail is rendered using scalable vector fonts and the mipmap level of detail spacing is 2, there will be no degradation in quality of the presentation of text content as the zoom level is increased over the mipmap level of detail range that is implemented. For example, if mipmap content is generated at 1x, 2x, 4x, and 8x using scalable vector fonts and trilinear filtering is implemented, corresponding text content can be viewed at zoom levels between 1x and 8x with no degradation in quality.

Scope of the terminology "in real-time"

The term "real-time" is included as a limitation in claim 7. 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 discussed below.

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 to new e-mails as they come in, as compared with waiting until the end of the day or some other time to respond to e-mails in more of a batch manner. In a real time flight tracking context, the data that is provided may actually reflect a tracking position that is several seconds, or even minutes, old.

One of skill in the art would recognize that in a software user-interface context, which is applicable to the present claims, the use of real-time typically means the user is enabled to continue an operation in a non-disrupted manner, meaning the user doesn't have to wait a period of time of significance for the operation to be performed. In this context, real-time is perceived by the user's sense of time.

As defined by SearchSMB.com Definitions[33]

real time

---

[33] http://searchsmb.techtarget.com/sDefinition/0,,sid44_gci214344,00.html

DEFINITION- Also see real-time clock and real-time operating system.

Real time is a level of computer responsiveness that a user senses as sufficiently immediate or that enables the computer to keep up with some external process (for example, to present visualizations of the weather as it constantly changes). *Real-time* is an adjective pertaining to computers or processes that operate in real time. Real time describes a human rather than a machine sense of time.

In the days when mainframe batch computers were predominant, an expression for a mainframe that interacted immediately with users working from connected terminals was *online in real time*.

Applicant wants to make it clear that under the context of the user of in "real-time" as recited in the claims herein, "real-time" does not mean instantaneously. Rather, the operation that is enabled to be performed in "real-time" is performed in a non-disrupted manner, as experienced by the user. For the purpose of a defined time period, "in real time" as used herein means the operation is performed in a few seconds or less.

Terminology "in a fraction of a second"

New dependent claim 31 recites, in part, "dynamically generating scaled Web page content in a fraction of a second." Support for this limitation is provided in paragraph [0037], which recites, in part (emphasis added, text in brackets added),

> 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 [i.e., for zooming], or the window is repositioned [i.e., for panning].

In the foregoing recitation, the "file" refers to the SVF (simple vector format) content, which comprises SVF drawing commands that are processed by the graphics rendering engine to support rapid zooming and panning of content. It will be understood that while the terminology "in real-time" may encompass "a fraction of a second," the scope of real-time shall not be limited to a fraction of a second, but rather have the scope discussed above. It is further noted that the graphics rendering engine discussed in paragraph [0037] is different than the rendering engine employed for interpreting the HTML-based content.

****

Arguments in Support of Patentability of the Amended Claims

Prior to amendment of the pending claims, original claims 1-14 and 20-25 were rejected under 35. U.S.C. § 102(a) as being anticipated by *WEST*: A Web Browser for Small Terminals by Staffan Björk et al. (Viktoria Institute, Göteborg, Sweden, 1999) (*WEST*). A claim is anticipated only if each and every element of the claim is found in a single reference. M.P.E.P § 2131 (citing *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628 (Fed. Cir. 1987)). "The identical invention must be shown in as complete detail as is contained in the claim." M.P.E.P. § 2131 (citing *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226 (Fed. Cir. 1989)). In order to be anticipating, a prior art reference must be enabling so that the claimed subject matter may be made or used by one skilled in the art. Amgen *Inc. v. Hoechst Marion Roussel, Inc.*, 314 F.3d 1313, 1354 (Fed. Cir. 2003); *Helifix, Ltd. v. Blok-Lok, Ltd.*, 208 F.3d 1339, 1346 (Fed. Cir. 2000); *Akzo N.V. v. U.S. Int'l Trade Comm'n*, 808 F.2d 1471, 1479 (Fed. Cir. 1986). Prior art is not enabling so as to be anticipating if it does not enable a person of ordinary skill in the art to carry out the invention. See *Elan Pharms., Inc. v. Mayo Found.*, 346 F.3d 1051, 1057 (Fed. Cir. 2003) (remanding the case to the district court for a determination of whether the prior art reference enabled persons of ordinary skill to make the invention without undue experimentation). To serve as an anticipating reference, the reference must enable that which it is asserted to anticipate. "A claimed invention cannot be anticipated by a prior art reference if the allegedly anticipatory disclosures cited as prior art are not enabled." *Amgen, Inc. v. Hoechst Marion Roussel, Inc.*, 314 F.3d 1313, 1354, 65 USPQ2d 1385, 1416 (Fed. Cir. 2003). *See* *Bristol-Myers Squibb v. Ben Venue Laboratories, Inc.*, 246 F.3d 1368, 1374, 58 USPQ2d 1508, 1512 (Fed. Cir. 2001) ("To anticipate the reference must also enable one of skill in the art to make and use the claimed invention."); *PPG Industries, Inc. v. Guardian Industries Corp.*, 75 F.3d

1558, 1566, 37 USPQ2d 1618, 1624 (Fed. Cir. 1996) ("To anticipate a claim, a reference must disclose every element of the challenged claim and enable one skilled in the art to make the anticipating subject matter.")

Original claims 15-19 were rejected under 35. U.S.C. § 103(a) as being unpatentable over *WEST*, further in view of The Zoom Browser by Lars Erik Holmquist (Åter till Human IT, 3/1998), (*Holmquist*). The Examiner is reminded that to successfully make a prima facie rejection under 35 USC § 103, the Examiner must show that Applicants' claimed subject matter would have been obvious to one of ordinary skill in the art pertinent to Applicants' claimed subject matter at the time it was made. See *KSR International, Co. v. Teleflex, Inc.*, 550 U.S. 398 (decided April 30, 2007). Some of the factors to consider in this analysis include the differences between the applied documents and Applicant's claimed subject matter, along with the level of skill associated with one of ordinary skill in the art pertinent to Applicant's claimed subject matter at the time it was made. See USPTO Memo entitled "Supreme Court decision on KSR Int'l. Co., v. Teleflex, Inc.," (May 3, 2007). One way in which an Examiner may establish a prima facie case of unpatentability under 35 USC § 103 would be to show that three basic criteria have been met. First, the Examiner should show that the applied documents, alone or in combination, disclose or suggest every element of Applicant's claimed subject matter. Second, the Examiner should show that there is a reasonable expectation of success from the proposed combination. Finally, the Examiner should show that there was some suggestion or motivation, either in the applied documents themselves or in the knowledge generally available to one of ordinary skill in the art pertinent to the claimed subject matter at the relevant time, to modify the document(s) or to combine document teachings. The motivation or suggestion to make the proposed combination and the reasonable expectation of success should be found in the prior art, and should not be based on Applicant's disclosure. See *In re* Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991); See

MPEP § 2142; 2143 - § 2143.03 (regarding decisions pertinent to each of these criteria).

Applicants respectfully assert that the rejection of each of the original claims was improper. Although the Applicants have amended the claims herein to more particularly recite features of their claimed inventions, such amendments are not made to overcome any of the §102 or §103 rejections referenced above. However, a discussion of the references used in the §102 or §103 rejections and discussion of the rejections themselves are presented herein to ensure that the teachings of such references as applied to the subject matter claimed herein are clarified.

Discussion of *WEST*

An overview of *WEST* is described in its Abstract, which states (emphasis added),

> We describe ***WEST,*** a **WE**b browser for **S**mall **T**erminals, that aims to solve some of the problems associated with accessing web pages on hand-held devices. Through a novel combination of ***text reduction** and* ***focus+context visualization***, users can access web pages from a very limited display environment, since the system will provide an overview of the contents of a web page *even when it is too large to be displayed in its* ***entirety***. To make maximum use of the limited resources available on a typical hand-held terminal, *much of the most demanding work is done by a proxy server*, allowing the terminal to concentrate on the task of providing responsive user interaction. The system makes use of some interaction concepts reminiscent of those defined in the Wireless Application Protocol (WAP), making it possible to utilize the techniques described here for WAP-compliant devices and services that may become available in the near future.

WEST's approach to the aforementioned problems is to remove content and then totally reconfigure it. The approach is directly opposite the approach used by the SoftView™ browser and as disclosed in the present application and claimed herein. As stated in the *WEST* Introduction (emphasis added),

> The World Wide Web (WWW) currently consists of about half a billion pages, offering users a vast range of informational resources. However, *these pages are almost exclusively* ***designed for use with desktop computers****, i.e. computers with large* ***high resolution screens***, powerful processors, and an abundance of primary and secondary storage.

As further stated in the Introduction (emphasis added),

> The work presented here focuses on this encounter between the WWW and mobile telephony, and more specifically on the need to provide gateways between mobile technologies and existing web resources. Although mobile terminals ***require specially designed formats*** *for optimal usability due to the constraints of the user environment*, it is not likely that all information available on the web will be ***translated into these format*** in advance. Thus, there is need for some kind of automatic on-the-fly ***transformation of existing web content to mobile formats***, in order not to shut mobile users out from the bulk of web resources.

It is clear that the *WEST* authors considered the idea of being able to zoom and pan full page views of Web pages on devices with small screen resolutions as being a bad approach. As stated in the first paragraph in the Interaction in WEST section,

> To give a better idea of how the *WEST* browser works, we will now give a detailed account for how a user may interact with the system. This will take the form of a complete interaction scenario, with an illustration for each screen the user will see.

The example page viewed in a traditional browser on a 160x160 pixel display

> As our example, we have used a page reporting baseball news at the Yahoo Sports site. *The page was comprised mostly of text* – 319 words, or about 1500 characters. There were 15 links to other pages, plus a banner advertisement and a search function. As the figure above will attest, viewing this page on a traditional browser on a 160x160 pixel screen ***presents serious problems***. *Only a very small part of the page would then be available at any time,* ***giving almost no clues to the size or context of the material***.

Applicants agree that viewing a typical Web page designed for the desktop using a traditional browser on a 160x160 pixel screen is problematic *if you do not provide support for true zooming and panning and maintain the visual layout of the content.* In particular, the original implementation of the ClearView™ browser was on a Palm device with a resolution of 160x160 pixels[34]. Accordingly, the approach used by the ClearView™ browser disclosed in the present application and claimed herein and the approach used by *WEST* represent two entirely different (and opposing) approaches using devices with identical screen resolutions.

Among other things, The *WEST* approach clearly would not enable a user to browse an entirety of a Web page (based on its original HTML definition, as interpreted by an HTML rendering engine), as it intentionally removes content, including both text and image content. As stated in the section entitled Pre-processing, Including Card Chunking (emphasis added),

> Proxy servers for real-time pre-processing of web information to be accessed using a mobile terminal is a proven technique used for instance in current web services for palm-sized PDAs. In *WEST, we made use of a **proxy server** to*:
>
> 1. *Filter and **reduce** the contents of web pages in order to adapt them to the capabilities of the mobile browser (this would mean among other things to **get rid of JavaScript, image maps, frames etc.**)*
>
> 2. Convert the reduced web page into *n* sub-pages (cards), each of which can be readily presented on a mobile-sized display (*e.g.* 160x160 pixels). Cards are interlinked to form a deck by arranging them into a suitable reading-order
>
> 3. Produce alternative renderings of these cards corresponding to different levels of detail. Typically a card can be displayed in its full size, in reduced size and minimized. *These alternative renderings **are not necessarily derived from graphical reduction** – in WEST, one alternative when reducing card size is to use automatic text summarization*

---

[34] The screenshots shown in FIGs. 7A, 7B, 8A, 8B, 9A, and 9B are of Web page views rendered by a ClearView™ browser client running on a Palm IIIc emulator having a resolution of 160x160 pixels.

In addition to the foregoing deficiencies, *WEST* absolutely does not employ any page layout information in the context of any of the pending claims. Notably, the only technique disclosed for presenting images of card content is to use graphical compression (which would employ one of the bitmap downsampling techniques discussed above). As stated in the Web Page Rendering section (emphasis added),

> For the graphical presentation of the different cards, each individual card had to be rendered as if it were a web page. However, we were unable to write a full-scale web rendering engine within the constraints of this project. Instead, we used the rendering engine provided by the HotJava Web Browser [31] ***to produce an image of each card as displayed on a screen of the required size (160x160 pixels). The same images where (sic) also graphically compressed to intermediate and thumbnail size. These pre-rendered images were then used by the system for the graphical presentation.***

A bitmap image of a Web page view cannot provide any page layout information. It simply identifies the pixel color values in an XY grid, and provides no information relating to the location of objects relative to the XY grid. Moreover, any image content that is derived from a bitmap alone (*e.g.*, with bitmap interpolation), also cannot convey any page layout information.

Of course, preservation of an original page layout, using any mechanism, is not taught or suggested by *WEST*. *WEST* clearly teaches away from preserving an original page layout by "chunking" the original page content (*i.e.*, breaking it up into cards) and generating view of the card content, such as illustrate below.

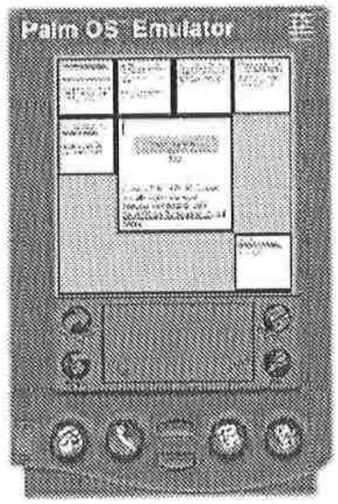


Figure 1: The WEST browser on a simulated Palm OS™ display

Each card is a bitmap image of only a portion of the original Web page content (the chunk allocated to the card), with portions of the entire original text and images not be accessible via any of the cards. By dramatic comparison, under the techniques disclosed in the present application and employed by the ClearView™/SoftView™ browsers (as exemplified by the nytimes.com Sony Clie screenshots presented above), the entire Web page content is enabled to be browsed at multiple zoom levels while preserving the original page layout and design as defined by the HTML-based Web page content (as interpreted by the rendering engine). Thus, users are enabled to browse their favorite Web pages in a familiar manner similar to how such pages are browsed via their desktop browser. Moreover, this browsing functionality is applicable to billions of Web pages available via the World Wide Web.

<u>Discussion of *Holmquist*</u>

The *Holmquist* reference discloses a technique for viewing content on a desktop computer comprising "flip zooming" and focus+context views. Rather than address viewing existing HTML-based Web pages, *Holmquist's* Zoom Browser is directed at enabling viewing of a multi-page document via, among other things, specially-coded

Web pages and a specially-configured desktop browser. In other words, one cannot choose to view an arbitrary page selected by a user from among the billions of Web pages accessible via the Internet (*i.e.*, World Wide Web), but rather the content that is to be viewed under *Holmquist* must be specifically coded to be viewed with the Zoom Browser.

In particular, *Holmquist's* technique employs Java. Java is implemented in a Web browser through use of Java applets (see discussion and Figure 10 on page 10 of *Holmquist*). Java applets are retrieved over a network from a corresponding URL. As such, there needs to be reference in each of *Holmquist's* Web page source code to identify where any Java applets would be retrieved from. Any existing Web page (which would be billions of Web pages) that doesn't have references to the Java applets used for rendering a particular page could not be implemented in a Web browser under *Holmquist*. This means *Holmquist* would be useless for browsing the World Wide Web. Of course, that is not the point of *Holmquist's* Zoom Browser – it is to be able to view the content of a multiple page document more easily, not to surf the Web.

Holmquist's Zoom Browser also has several other significant limitations that cannot be overcome in attempting to support any rejection over both the original claims and now amended claims, either as a standalone reference or in combination with one or more other references. Of significant note, the Zoom Browser could not support in-line pictures (images), or some of the basic layout features of HMTL, such as tables. As stated on page 10 (emphasis added),

> Just like any other browser, the Zoom Browser can be used to load and display documents from the World Wide Web using flip zooming. It supports ***a subset*** *of the Hypertext Markup Language (HTML)*, including hypertext links and *most basic formatting commands*. The Zoom Browser ***does not currently support in-line pictures, or some of the more advanced features of HTML, like tables***, but these could be added in the future.

The Zoom Browser also does not employ any page layout information in the context of the claims herein, which would be clearly understood by one skilled in the HTML browser arts in view of the commonly known meaning of page layout used in performing page layout operations associated with rendering HTML-based Web pages.

Claim 1, as amended, recites (emphasis added),

1. A mobile device, comprising:

a processor;

a wireless communications device operatively coupled to the processor, to facilitate communication with a network via which *the World Wide Web* may be accessed;

a *touchscreen comprising at least a portion of a top surface of the mobile device*;

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 device to perform operations including,

enabling a user of the mobile device to request access to ***Web pages from among billions of Web pages accessible via the World Wide Web***, each Web page including associated HTML-based Web content ***defining an original page layout and-design of content on that Web page, the content including text content and image content***;

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

retrieving, via the wireless communication device, HTML-based Web content associated with a Web page;

processing, the HTML-based Web content with a rendering engine to generate page layout information corresponding to the original page layout of content on the Web page as interpreted by the rendering engine;

employing the page layout information to generate scalable page layout information; and

employing the scalable page layout information and/or content derived therefrom to generate views of the Web page ***to enable the user to browse an entirety of the HTML-based content of the Web page at multiple zoom levels while preserving the original page layout and design of the HTML-based content as interpreted by the rendering engine***,

wherein in the multiple zoom levels include zoom levels that are not predefined, and the user is enabled to view the entirety of the HTML-based content of the Web page at a zoom level of at least 4 times full scale, wherein full scale corresponds to a native pixel resolution of the Web page as interpreted by the rendering engine, and wherein there is no degradation in quality of the presentation of text content when the zoom level is increased to at least 4 times full scale.

Applicants have amended to claim 1 to clarify the intended scope of the claim. For example, as discussed above, the claim element "enabling a user of the mobile device to request access to ***Web pages from among billions of Web pages accessible via the World Wide Web***" is intended to convey that the user is enabled to access Web pages the user desires to access from among billions of Web pages accessible via the World Wide Web. At the same time, this does not mean the user is enabled to access any Web page accessible via the World Wide Web (also see discussion above concerning the intended scope of this terminology).

Notably, this claim element precludes a rejection based on a reference that employs a scheme or technique that requires pages to be specially coded to support corresponding functionality, such as that employed by *Holmquist*. Clearly there are not billions of Web pages on the World Wide Web encoded in the manner employed by *Holmquist*.

With respect to the claim element, "each Web page including associated HTML-based Web content ***defining an original page layout and-design of content on that Web page, the content including text content and image content***," this means that each Web page includes at least some text content and at least one image. Of course, there are Web pages without images and there may be Web pages without text and only one or more images; however, the claimed mobile device is required to be able to browse Web pages that include both text content and image content while also meeting the limitations recited in the other elements of claim 1.

With respect to the element of ***processing, the HTML-based Web content with a rendering engine to generate page layout information corresponding to the original page layout of content on the Web page as interpreted by the rendering engine***;" this corresponds to the conventional use of generating page layout information of a HTML-based Web page by a Web browser (rendering/layout engine), as would be clearly understood by one of ordinary skill in the Web browser art.

With respect to the elements,

employing the page layout information to generate scalable page layout information; and

employing the scalable page layout information and/or content derived therefrom to generate views of the Web page ***to enable the user to browse an entirety of the HTML-based content of the Web page at multiple zoom levels while preserving the original page layout and design of the HTML-based content as interpreted by the rendering engine***

Generally see the discussions above.

With respect to the claimed element of

[and] wherein there is no degradation in quality of the presentation of text content when the zoom level is increased to at least 4 times full scale,

Applicants respectfully asset that such a feature is clearly not taught or fairly suggested by any of the references of record, considered alone or in any combination. Notably, there would be no motivation to implement zoom levels substantially greater than full scale (such as 4x) under the techniques disclosed in the references discussed above employing rendering Web pages as bitmap content on servers and transmitting portions of the bitmap content for rendering on client devices because a) the resolution of the screens was very low (*e.g.*, 160 x 160 pixels) and b) scaled bitmap content based on a single sampled image (*i.e.*, the original bitmap content corresponding to the full scale image) results in text content that is substantially degraded. With respect to the very low image resolution, consider that for Web pages designed to be viewed on monitors with a resolution of at least 800 or 1024 pixels in width (which represented a substantial portion of the Web pages that existed in 2000-2001), the portion of the width of such Web pages that could be displayed in a single view would respectively represent 160/800 (20%) or 160/1024 (15.6%) of the Web page width. At a zoom level of 4x these portions become 5% and ~4%, respectively. Since the primary content on most Web pages comprises text content, and the general purpose of browsing such Web pages is to read the text content, it is important to maintain the clarity of text content. Being able to view text content at full scale on screens with very low resolutions facilitated this need, and there would be no motivation to substantially increase the zoom level above full scale.

With respect to the aspect of no degradation in the quality of the presented text at a zoom level of at least 4x, it is clear that schemes that generate scaled views of content by interpolation of bitmap image content at full scale (only) cannot achieve this result. Accordingly, claim 1 is clearly patentable over each of the corresponding references for at least these reasons.

In view of the arguments presented above, it is clear that claim 1 is patentable over the cited art of record, and is in condition for allowance. In addition, each of the claims that depend either directly or indirectly from claim 1 are likewise in condition for allowance for at least the reasons supporting patentability of claim 1.

While Applicants respectfully assert the rejections of several of the original claims depending either directly or indirectly from claim 1 were improper and unsupported, Applicants assert that such rejections are now moot in view of their respective clarified context in view of the amendment to claim 1. However, there are aspects of some of the rejections for which clarification is warranted.

For example, with respect to the rejection of original claim 6, the Examiner states,

> **<u>Regarding Claim 6</u>**
>
> Limitation "enabling the Web page to be displayed at different resolutions by, generating scalable content via use of the scalable page layout information; and scaling the scalable content to re-render the display in response to associated user inputs" is shown in page 2 of *West* **The West Browser** section first paragraph and page 4 **Flip Zooming in West** sections second paragraph.

As Applicants noted above, *WEST* does not employ scalable page layout information of any kind, as would be known to one of ordinary skill in the Web browser art. Moreover, also as discussed above, the only technique the *WEST* discloses that could be reasonably be interpreted as scaling anything is the generation of the card images, which are generated by (As stated in the Web Page Rendering section and quoted above, emphasis added).

> For the graphical presentation of the different cards, each individual card had to be rendered as if it were a web page. However, we were unable to write a full-scale web rendering engine within the constraints of this project. Instead, we used the rendering engine provided by the HotJava Web Browser [31] ***to produce an image of each card as displayed on a screen of the required size (160x160 pixels). The same images where (sic) also graphically compressed to***

intermediate and thumbnail size. These pre-rendered images were then used by the system for the graphical presentation.

Accordingly, since there is no use of scalable page layout information disclosed by *WEST*, the rejection of original claim 6 was improper, and the amended version of claim 6 is further patentable over *WEST*.

With respect to the rejection of original claim 7, the Examiner states,

Regarding Claim 7

Limitation "enabling a user to pan a display of the Web page substantially in real-time in response to a corresponding user input" is shown in page 7 of *West* **Presentation and Interaction** section first paragraph.

This cited section of *WEST* states,

Presentation and Interaction

Based on the flip zooming technique, the WEST browser presents each web page as a number of discrete objects, representing individual cards or decks of cards. The user navigates between different objects by using directional buttons or by directly choosing the object to focus upon with a pen or other pointing device. For sequential reading of a whole page, a user would generally switch to a full-screen view and then advance through the cards by pressing a designated "forward" button.

Each view in WEST only presents one level of the hierarchical structure of decks and cards that represent the web page. To move between levels, the user zooms in on the object in focus (usually by clicking or tapping with the pen on it) and will thus go one level deeper into the structure. To go up one level, the user clicks or taps on the "white space" between the objects. This navigation might also be facilitated by the use of "up" and "down" buttons, for moving up and down in the hierarchy, analogous to zooming out and zooming in.

When the user goes down one level in the hierarchy, the focus object takes over the whole screen space to show its content. If the current focus represents a single card, this card will be allowed to fill the screen completely to facilitate reading. In the case of the focus representing a deck, however, a view of all the objects in the deck is presented.

The system provides three different modes in which the material can be viewed: thumbnail, summary and link view. When switching from one view mode to another, the position of the focus in the hierarchical structure is maintained, enabling the user to navigate in a suitable view mode to locate a card, and then change to another mode (typically the thumbnail view) to actually view the card. In the prototype, the user switched between the different views by accessing a pop-up menu.

Notably, there is no mention of "panning" or scrolling in *WEST*. Applicants respectfully assert that the aforementioned technique for viewing cards does not fall within the claim scope of "pan a display of the Web page" as would be understood by someone of ordinary skill in the art. Applicants have further clarified the scope of claim 7 herein to now recite,

7. The mobile device of claim 1, wherein execution of the instructions performs further operations comprising,

for each of the multiple zoom levels, enabling the user to pan a display of the Web page ***at that zoom level*** in real-time in response to a corresponding user input.

Accordingly, the claimed invention of claim 7 requires support for panning a display of Web pages in real-time at multiple zoom levels. It is noted that the terminology "scrolling" is sometimes used to describe operations equivalent to "panning" as used in the claims herein.

With respect to the rejection of original claims 8 and 9, the Examiner states,

<u>**Regarding Claim 8**</u>

Limitation "enabling a user to view a column of the Web page at a higher resolution than a current resolution via a corresponding user interface input" page 4 of *West* **Flip Zooming in West** section **Interaction Example**;

Limitation "the display is re-rendered such that content corresponding to the selected column is displayed substantially across the display" is shown in page 4 of *West* **Flip Zooming in West** section **Interaction Example**.

<u>**Regarding Claim 9**</u>

Limitation "tapping on the column" is shown in page 4 of *West* **Flip Zooming in West** section **Interaction Example**.

First off, it would be somewhat happenstance that any of the thumbnail size images generated by *WEST* would correspond to an entire column, and certainly such a case would not be expected for an arbitrarily selected Web page from among the billions of Web pages available via the World Wide Web. Second, the width of the column in pixels, again by chance, would need to substantially correspond to the width of a predefined bitmap image of content corresponding to the column generated at one of the predefined scale factors employed by *WEST* for generating is bitmap images. It could not be an arbitrary width, as supported by the techniques disclosed in the present application, as the bitmap images employed by *WEST* have predefined resolutions based on the predefined scale factors (*e.g.*, scale factors used in graphical compression to create the predefined intermediate and thumbnail sizes).

With respect to original claims 10-13, the Examiner based the rejections on similar rational to used in rejecting original claims 8 and 9. For similar reasons to those presented above, zooming on images and paragraphs using *WEST* would only fall within the claim scope by happenstance for particular Web pages, and such features could not be facilitated for arbitrarily selected Web pages from among billions of Web pages available via the World Wide Web. Moreover, under WEST content cannot be selected for zooming by tapping on the content when displaying a view of the Web page that occupies the full display. Rather, corresponding thumbnails are selected.

Claim 15 has been amended to now recite (emphasis added).

A mobile device, comprising:

processing means;

wireless communications means to facilitate communication with a network via which Web content may be accessed;

a display;

memory; and

storage means 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, the Web page including associated HTML-based Web content *defining an original page layout and attributes of a plurality of objects,* ***including text objects and image objects****, included in the Web page*;

retrieving, via the wireless communication means HTML-based Web content associated with the Web page;

processing the HTML-based content with a rendering engine to generate page layout information, *wherein the page layout information defines a **layout location** for each of the plurality of objects*;

for each of the plurality of objects,

storing page layout information ***including a bounding box associated with layout of the object***; and

storing information that links the object with its corresponding page layout information,

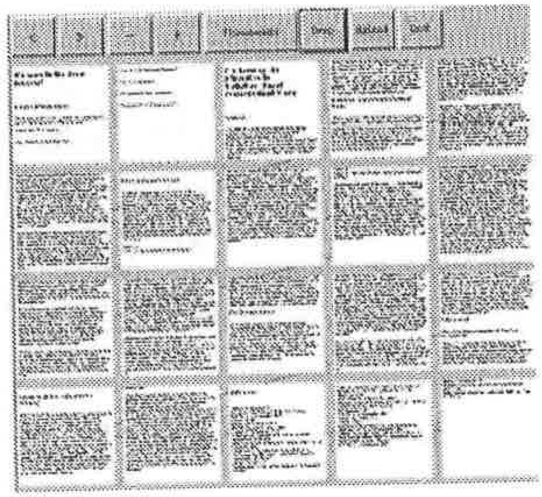
wherein the page layout information further includes information from which a page layout location of each of the bounding boxes can be determined; and

*scaling the page layout information to generate scaled Web page content comprising scaled objects at a plurality of scale factors, wherein the original page layout of the objects as interpreted by the rendering engine is preserved at each scale factor, and at least one of the scale factors **is at least 4 times full scale,** wherein full scale corresponds to the native pixel resolution of the Web page as interpreted by the rendering engine.*

Support for this amendment can generally be found in discussions in the specification with reference to FIGs. 2C, 4A-G, 5, and 6 and teachings of the present application discussed above.

The rejection of original claim 15 is substantially moot, as the elements specifically addressed by the Examiner concerning a primary datum, object datum, and object vector have been removed by amendment. However, Applicants assert that the Examiner's allegations with regard to aspects of the page layout and layout of corresponding objects being disclosed by *Holmquist* is clearly misplaced. The page layout and page layout information concerns the location of where objects are laid out on the Web page at full scale (as interpreted by the rendering engine, as discussed in detail above), and scaled page layout information concerns scaling the original page layout information (*i.e.*, the page layout information at full scale) to generate scaled Web page content, wherein scaled objects are laid out and scaled at layout locations corresponding to the scaled page layout information. For example, see discussion above on concerning scalable page layout information and scaled page layout information. Also see the graphical depictions in the Figures of various screenshots provided herein.

It is further noted that neitner of *WEST* nor *Holmquist* disclose retrieving HTML-based Web content via a mobile device. Under *WEST*, the original HTML content corresponding to a Web page is retrieved by a proxy server, and reformatted by the proxy server into a structure configured to be handled by a mobile device. The mobile device receives the reformatted content from the proxy server. While embodiments of the present application employ a proxy server to perform some operations in those embodiments, at least one of the embodiments disclosed implements all of the operations on the client device (*e.g.*, a mobile device under claims 1 and 15). Moreover, under *Holmquist*, there is no mention whatsoever about using the Zoom Browser on a mobile device, and due to the various limitations of mobile devices at the priority date of U.S. Application No. 09/878,097 (June 8, 2001, to which priority in the present application is claimed) (*e.g.*, screen resolution of 160x160 under the *WEST* implementation on the Palm OS emulator), it would make no sense to even attempt to implement the Zoom Browser on a mobile device at that time. For example, the following screen shot included in *Holmquist* appears to have a resolution of at least 800 x 600 pixels, and likely has a resolution of 1024 x 768 pixels, if not higher.



In Claim 15, as amended, the Web page also includes a plurality of text objects and image objects. As discussed above, *Holmquist* specifically identifies that the implementation of the Zoom Browser could not handle in-line pictures (*i.e.*, images) - that is, image content that is defined in-line in the HTML content of a Web page that includes images.

With respect to the rejection of original claim 16, the Examiner states,

<u>**Regarding Claim 16**</u>

Limitation "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" is shown in page 14 of *Holmquist* **3.3.2 Boxes around pages belongs to the document currently in focus** section;

Limitation "mapping the object vectors and associated bounding boxes to a virtual display in memory" is shown in page 14 of *Holmquist* **3.3.2 Boxes around pages belongs to the document currently in focus** and **3.3.3 Usage-dependent placement of focus pages** sections.

Elements have been added to amended claim 15 relating to generation of bounding boxes. The boxes employed by *Holmquist* have nothing to do with the bounding boxes recited in claim 15. No person of ordinary skill in the Web browser arts would consider the thumbnail images of pages in *Holmquist's* Zoom Browser to be bounding boxes within the scope of claim 15.

Clearly, claim 15 is patentable over *WEST* and *Holmquist*, considered alone or in combination. Additionally, each of claims 16-19, 21-25, and 29-31, which depend either directly or indirectly from claim 15, are likewise in condition for allowance for at least the reasons supporting patentability of claim 15.

Moreover, with further respect to claim 16, and 17, Applicants have amended these claims such that the rejection of original claims 16 and 17 are moot. While not acquiescing to the rejection of original claims 16 and 17, Applicants respectfully assert that none of the references of record, taken alone or in any combination, teach or fairly suggest "mapping the page layout information to a virtual display area comprising a resolution-independent coordinate space, the page layout information that is mapped including an object datum for each object corresponding to where the content for that object is located on a representation of the Web page laid out on the resolution-independent coordinate space." Accordingly, claim 16 and 17 are further patentable and in condition for allowance for at least these reasons.

With respect to the rejection of original claim 18, the Examiner states,

Regarding Claim 18

Limitation "scaling a scalable font to render the text content at different scale factors" is shown in page 13 of *Holmquist* **3.2 The Zoom Browser's display modes: thumbnails, summaries, and alternating** section.

Applicants respectfully note there is nothing about using a scalable font disclosed or fairly suggested by *Holmquist*. According to *Holmquist* (Section 2.1.1 The flip zoom technique, emphasis added),

> For the basic view of a document, it is split up into a number of "pages" of equal length. The pages are laid out on the display in left-to-right, top-to-bottom order. ***The pages are represented by thumbnail sketches***, which provide a context view, but are usually too small to read.

It is most likely the thumbnail sketches are conventional thumbnail images of full size pages there are obtained via bitmap image down-sampling. Moreover, when considering a combination obviousness rejection over *WEST*, it is respectfully noted that Palm OS devices during this timeframe did not support scalable fonts, but rather employed bitmap fonts. It is noted that while images of Palm device emulators are shown in Figures in the present application and herein, the ClearView™/SoftView™ browser was also implemented on Windows CE devices that employed scalable fonts, and the use of scalable fonts is specifically disclosed in the specification of the present application (see paragraph [0096]).

In support of the rejection of original claim 19, the Examiner states,

Regarding Claim 19

Limitation "generating a display list derived, at least in part, via use of the object vectors; and employing the display list to re-render the display of the Web page" is shown in page 11 of *Holmquist* **3.1 Navigating World Wide Web documents** section.

West discloses the same subject matter as claimed in the present application; however, it does not specifically spell out the "datum definition" and "object bounding box" features as claimed in claims 15-19. Nevertheless, these features are well known in the art and one application is disclosed by *Holmquist*.

Although both *West* and *Holmquist* disclose the application of "flip zooming" technology, *Holmquist* discloses the flip zooming in further details with visual aids for navigation and reading coupled with the flip zooming technology in the "The Zoom Browser." Thus, it would have been obvious to one skilled in the art at the time the present invention was made to incorporate the detailed visual aids techniques of *Holmquist* into the small terminal web browser of *West* so that the functions and applications of the *West* browser can be further advanced and more applicable to various platforms.

Applicants respectfully note that neither section 3.1 of *Holmquist* nor the foregoing statements address the use of a display list, as would be understood to one of ordinary skill in the computer graphics arts. Claim 19 has been amended to clarify the intended scope of its invention, and now recites, in part,

generating a display list for rendering corresponding content; and

employing the display list to generate scaled Web page content at the plurality of scale factors.

With respect to support for the amendment to claim 19, Applicants note that one or ordinary skill in the computer graphics arts, would understand that the terminology "display list" corresponds to a list of drawing commands or instructions for rendering corresponding graphic content. For example, as stated on a Wikipedia computer graphics page concerning display lists[35],

A **display list** (or *display file*) is a series of graphics commands that define an output image. The image is created (*rendered*) by executing the commands.

In vector-based drawing environments, such as used for Computer Aided Design (CAD), the display list is sometimes referred to as a display list of vectors[36]. For example, as stated on the computer section of your dictionary.com[37], one of the definitions for a display list is,

In computer graphics, a collection of vectors that are used to display a vector graphic image on screen. The display list is generated from the drawing database.

Having spent many years developing CAD-related software, including advanced award-winning display-list drivers for such products as AutoCAD and their own product VDraft™, Applicants use of the terminology "display list of vectors" and other references to "display list" in the present application would be expected in consideration of the terminology used by those skilled in development of CAD software at the time of the invention. Moreover, terminology relating to the use of graphics files and other references to files (*e.g.*, file manipulation) is consistent with the Wikipedia display list definition above referencing a "display file." In addition, one or ordinary skill in the graphics art would recognize that while the verbiage "file" is used, the actual display list in modern computer devices would generally be stored in memory as a data structure or

---

[35] http://en.wikipedia.org/wiki/Display_list

[36] As stated in paragraph [0092] of the application, "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."

[37] http://computer.yourdictionary.com/display-list

the like. The use of the terminology "file" goes back several decades ago when graphic computer systems had limited volatile memory (e.g., random access memory) available, and thus content corresponding to display lists were often written to and read from a file on disk rather than from memory.

Generally, the drawing commands or instructions in a display list may be in an original or processed/compiled form (*e.g.*, text, source code, object code, *etc.*), and/or may comprise a cached form. Moreover the original or processed/compiled form may include text-based content and/or binary content, and may correspond to an executable or non-executable form. In addition, it is further noted that claim 19 does not require generation of content to a screen or display, but rather the display list is employed to generate scaled Web page content.

With respect to claims 21, 23, and 25, these claims have been amended to now depend either directly or indirectly from claim 15 rather than claim 20, which has been canceled. Further arguments in support of the patentability of claim 21 are presented above for claim 8, which recites similar claim elements. Further arguments in support of the patentability of claim 23 are presented above for claim 10, which recites similar claim elements. Further arguments in support of the patentability of claim 25 are presented above for claim 12, which recites similar claim elements.

<u>Argument in support of patentability of new claims 26-35</u>

New claims 26-35 have been added by this amendment. Claims 26-28 depend from claim 1, and are patentable and in condition for allowance for at least the reasons supporting patentability of claim 1. Claims 29-31 depend from claim 16, and are patentable and in condition for allowance for at least the reasons supporting patentability of claim 16. Claim 32 depends from claim 15, and is patentable and in condition for allowance for at least the reasons supporting patentability of claim 15.

With further respect to claim 29, support for this claim is presented above in the context of employing a vector-based drawing model to layout points on a virtual coordinate space.

With further respect to claim 30, preservation of functionality defined by corresponding HTML-based content is discussed above.

With further respect to claim 31, dynamic generation of scaled Web page content to enable panning views in a fraction of a second is enabled by using the techniques disclosed in the present application, and in particular the limitation regarding "in a fraction of a second" is disclosed in paragraph [0037], as discussed above.

With further respect to claims 32 and 37, support for scaling text content using a scale factor of at least 4 times full scale is inherent to the resolution-independent characteristic of scalable vector fonts.

Claim 33 is a new method claim containing elements that are analogous to corresponding elements in mobile device claim 15. Accordingly, claim 32 is patentable and in condition for allowance for similar reasons supporting the patentability of claim 15 presented above.

Support for new dependent claims 34-36 are generally discussed above in consideration of analogous elements present in claims depending from claim 15. For example, elements of claim 35 are analogous to similar elements in claim 18, and elements in claim 36 are analogous to similar elements in claim 19. With respect to claim 33, support for the terminology "resolution-independent" can be found throughout the specification.

Nonstatutory Double Patenting Rejections

Claims 1-25 were rejected on the grounds of nonstatutory double patenting over claims of U.S. Patent Nos. 7,461,353 and 7,584,423, and provisionally rejected over the claims of co-pending Application Nos. 11/735,477 and 11/738,486. Applicants respectfully assert these rejections are moot in view of the claim amendments herein.

However, Applicants would like to clarify the requirements for supporting a nonstatutory double patenting rejection, and present reasons why the amended claims are patentably distinct from each of these patents and applications.

As a preliminary note, MPEP 804 states the following with regard to obviousness-type double patenting rejections:

> Any obviousness-type double patenting rejection should make clear:
>
> (A) The differences between the inventions defined by the conflicting claims - **a claim in the patent compared to a claim in the application**; and
>
> (B) The reasons why a person of ordinary skill in the art would conclude that the invention defined in the claim at issue is anticipated by, or would have been an obvious variation of, the invention defined in a claim in the patent. (Emphasis added).

In the present office action, the Examiner has not made clear the differences between the claims of the present application and the claims of any of U.S. Patent Nos. 7,461,353 and 7,584,423 or co-pending Application Nos. 11/735,477 and 11/738,486, as there is no comparison between each of the rejected claims and claims in the patent or application for which double patenting is alleged. Although the Examiner has identified allegedly common elements claimed in the present application and these patents and co-pending applications, no information is provided as to the particular claims of these patents and applications that allegedly correspond to the claims of the present application. In particular, the Examiner states,

Double Patenting

3. Claims 1-25 are rejected on the ground of nonstatutory double patenting over claims of U. S. Patent No. 7,461,353 and claims of U. S. Patent No. 7,584,423 since the claims, if allowed, would improperly extend the "right to exclude" already granted in the patents.

The subject matter claimed in the instant application is fully disclosed in the patents and is covered by the patents since the patents and the application are claiming common subject matter, as follows: mobile device; retrieving Web contents; translating the Web contents and scaling the information layout to be displayed across the display; and, enabling zooming and panning in response to user inputs.

Furthermore, there is no apparent reason why applicant was prevented from presenting claims corresponding to those of the instant application during prosecution of the application which matured into a patent. See *In re Schneller*, 397 F.2d 350, 158 USPQ 210 (CCPA 1968). See also MPEP § 804.

4. Claims 1-25 are provisionally rejected on the ground of nonstatutory double patenting over claims of copending Application No. 11/735,477 and claims of copending Application No. 11/738,486. This is a provisional double patenting rejection since the conflicting claims have not yet been patented.

The subject matter claimed in the instant application is fully disclosed in the referenced copending applications and would be covered by any patent granted on those copending applications since the referenced copending applications and the instant application are claiming common subject matter, as follows: mobile device; retrieving Web contents; translating the Web contents and scaling the information layout to be displayed across the display; and, enabling zooming and panning in response to user inputs.

Furthermore, there is no apparent reason why applicant would be prevented from presenting claims corresponding to those of the instant application in the other copending applications. See *In re Schneller*, 397 F.2d 350, 158 USPQ 210 (CCPA 1968). See also MPEP § 804.

Notably, the Examiner has provided no basis for why one of ordinary skill in the art would conclude that the pending claims are either anticipated by the claims of these above-referenced patents and application or an obvious variation thereof. Accordingly, Applicants respectfully request the Examiner make clear the items listed in (A) and (B) as cited above with respect to <u>each</u> claim or withdraw the double patenting rejection of that claim. Applicants respectfully note that Examiner is required to provide the reasons supporting the rejection of <u>each</u> claim for which a nonstatutory double-patenting rejection is issued, not just independent claims or otherwise selected claims and generalizations being applied to other claims.

Applicants further want to point out previous restrictions concerning the claimed subject matter for this patent and related family members. As is well known, subject matter that is subject to restriction cannot later be rejected based on the grounds of nonstatutory double patenting rejection over the application/patent the subject matter was restricted from. See 35 U.S.C. § 121.

The present application was subject to restriction from Application No. 11/045,757, which issued as 7,461,353. In particular, the restriction (dated August 15, 2007) stated,

Election/Restrictions

Restriction to one of the following inventions is required under 35 U.S.C. 121:

I. Claims 40-70, drawn to display processing, classified in class 715, subclass 526.

II. Claims 71-150, drawn to boundary processing in class 715, subclass 521.

The inventions are distinct, each from the other because of the following reasons:

Inventions, II and I are related as subcombinations disclosed as usable together in a single combination. The subcombinations are distinct from each other if they are shown to be separately usable. In the instant case, invention (I) has separate utility such as display processing; invention (II) has separately utility such as boundary processing control by tough-sensitive display and clipping portions (See MPEP § 806.05(d)).

By way of example and not limitation, each of the independent claims 40, 71, 99, and 128 pending at the time of the restriction are presented below.

40. 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 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 device to perform operations including, enabling a user to request access to a Web page having an original format comprising HTML-based Web content defining an original page layout of content on the Web page;

retrieving at least a portion of the HTML-based Web content associated with the Web page;

translating the at least a portion of the HTML-based Web content to produce scalable page layout information; and

employing the scalable page layout information and/or data derived therefrom to,

render at least a portion of the Web page on the display using a first scale factor; and

re-render the Web page in response to associated user inputs to enable a user to zoom in and out a display of the Web page.

while independent claim 71 recited,

71. A wireless device, comprising:

processing means,

wireless communications means, to facilitate wireless communication with a network via which Web content may be accessed;

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 a Web page, the Web page including associated Web content having an original format defining an original page layout and attributes of content on the Web page;

retrieving and translating at least a portion of the Web content from its original format into scalable content that supports a scalable resolution-independent display of the content that substantially retains the original page layout and attributes of the content defined by its original format when rendered; and

employing the scalable content to render at least a portion of the Web page on the display using a first scale factor.

Independent claim 99 recited,

99. A mobile device, comprising:

a processor,

a wireless communications device, to facilitate wireless communication with a network via which Web content may be accessed;

a touch-sensitive 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 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 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 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 zoom in and out a display of the Web page.

Independent claim 128 recited,

128. A mobile device, comprising:

processing means;

wireless communications means, to facilitate wireless communication with a network via which Web content may be accessed;

a touch-sensitive 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 of content on the Web page;

retrieving and processing at least a portion of the HTML-based Web content to produce scalable content; and

employing the scalable content and/or data derived therefrom to,

render 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 zoom in and out a display of the Web page.

The foregoing claims were restricted, based (in view of a discussion with the Examiner, Quoc A. Tran) on a differentiation between scalable page layout information (recited in claim 40) as compared to scalable content and its recited respective uses in each of independent 71, 99, and 128. It is further noted that claim 40 corresponds to original claim 1 of the present application, and was restricted out from each of claims 71-150 of the '757 application, which were elected for further prosecution. Applicants acknowledge that further claims were added to the '757 application, and the elements of some of the claims 71-150 were amended after the restriction; however, none of the claims in 7,461,353 include language reciting "scalable page layout information," "scaling page layout information," or even "page layout information." Accordingly, in view of the previous restriction identifying scalable page layout information as being patentably distinct from scalable content and the existence of the terminology "scalable page layout information" or "scaling page layout information" in each of the independent claims of the present application, by definition of 35 U.S.C. § 121 it is not possible to support a nonstatutory double patenting rejection over any of the claims in U.S. Patent No. 7,461,353.

With respect to U.S. Patent No. 7,584,243 (which corresponds to U.S. Application No. 11/045,649), subject matter of this patent was restricted out from the original claims of 09/878,097, from which the original claims of 7,461,353 were likewise restricted (there was a three-way restriction). In particular, the subject matter of the '243 patent pertains to operations performed on the network side (only) in connection with a proxy server – thin client embodiment of the invention. Moreover, the claimed operations do not include any operations that are explicitly performed by the client. In stark contrast, under each of the pending claims, all of the recited operations are performed by a mobile device. Operations such as scaling page layout information or employing page layout information for any purpose is not present in any of the claims of the '243 patent. Accordingly, each of the pending claims is patentably distinct from each of the claims in the '243 patent. As an additional note, in *Boehringer Ingelheim Int'l GMBH v. Barr Labs, Inc.*, No. 2009-1032 (Fed. Cir., Jan. 25, 2010) (slip op.), the Federal circuit held that the safe harbor provision of 35 U.S.C. § 121 applies to a proper divisional of a divisional (which corresponds to the fact pattern here). The present application is a divisional of 11/735,477, which is a divisional of 09/878,097 and 11/045,649.

U.S. Application No. 11/735,477 is co-pending and has yet to be examined. Applicants acknowledge this is a provisional nonstatutory double patenting rejection. As is standard USPTO procedure, if the only remaining rejection in a first application is a provisional nonstatutory double patenting rejection over claims in a co-pending second application, and the co-pending second application has not received a notice of allowance, the first application should be issued a notice of allowability with the provisional nonstatutory double-patenting rejection removed.

U.S. Application No. 11/738,486 issued on November 9, 2010 as U.S. Patent No. 7,831,926. Applicants acknowledge that various claims in the '926 patent pertain to aspects of the generation and use of page layout information, including scalable vector-based page layout information. However, each of the pending independent claims 1, 15, and 32 include at least one claim element that is not analogous to any similar element in any of the claims of the '926 patent. In particular, claim 1 includes the claim elements,

enabling a user of the mobile device to request access to ***Web pages from among billions of Web pages accessible via the World Wide Web, each Web page*** *including associated HTML-based Web content defining an original page layout* and *design of content on that Web page, the content including text content and image content*;

...

wherein in the multiple zoom levels ***include zoom levels that are not predefined***, and the user is *enabled to view the entirety of the HTML-based content of the Web page at a zoom level of* ***at least 4 times full scale, wherein full scale corresponds to a native pixel resolution of the Web page as interpreted by the rendering engine, and wherein there is no degradation in quality of the presentation of text content when the zoom level is increased to at least 4 times full scale***.

while each of independent claims 15 and 32 include the elements,

enabling a user of the mobile device to request access to ***Web pages from among billions of Web pages accessible via the World Wide Web, each Web page*** *including associated HTML-based Web content defining an original page layout of a plurality of objects, including text objects and image objects, included in the Web page;*

...

scaling page layout information to generate scaled Web page content at a plurality of scale factors, wherein the original page layout of the objects as interpreted by the rendering engine is preserved at each scale factor, and at least one of the scale factors is at least 4 times full scale, wherein full scale corresponds to a native pixel resolution of the Web page as interpreted by the rendering engine.

Applicants respectfully assert that each of independent claims 1, 15, and 32 include elements that clearly make their respective claimed inventions patentably distinct over all of the claims in U.S. Patent No. 7,831,926.

The last statement the Examiner makes concerning no apparent reason why the applicant would be prevented from presenting claims corresponding to those of the instant application in the other co-pending applications is clearly in error and unsupported. As discussed above, the subject matter of the present application was restricted out from the subject matter elected to be prosecuted in U.S. Application No. 11/045,757. Also as discussed above, the subject matter of 11/045,757 was restricted out from the subject matter of U.S. Application No. 09/878,097 in a three-way restriction. Thus, there is ample evidence that an attempt was made to present this subject matter in another co-pending application, and aspects of the subject matter leading to the restrictions were not only presented in 11/045,757, they were also presented in an application as far back as in the original claims of 09/878,097 filed on June 8, 2001. But for the undue delay on the part of the USPTO in examining each of these applications and the corresponding restrictions that were issued, the subject matter of the present application could have been examined years ago. Moreover, the art unit policy of placing divisional application in the queue as if they are completely unrelated new applications and the long delays to a first examination of such divisional applications is highly prejudicial to Applicants and further exacerbated this situation.

Finally, Applicants further note that the filing of terminal disclaimers in various related (by common ancestor) applications does not provide evidence of agreement with or acquiesce to corresponding non-statutory double-patenting rejections in those applications, but rather may have been entered solely for the purpose of expediting prosecution.

Status of co-pending cases

Below is a table showing the status of co-pending applications that are related to the present application (status of present application shown in bold).

Application No.	**Filing Date**	**Relationship**	**Status**
60/211,345	06-12-2000	Provisional Application	Expired
60/217,345	07-11-2000	Provisional Application	Expired
09/828,511	04-07-2001	Original Non-provisional application	Abandoned
09/878,097	06-08-2001	Continuation-in-part of 09/828,511	7,210,099
11/045,649	01-28-2005	Divisional of 09/878,097	7,584,423
11/045,757	01-28-2005	Divisional of 09/878,097	7,461,353
11/735,477	04-15-2007	Continuation of 09/878,097	Awaiting First Action
11/735,482	04-15-2007	Continuation of 09/878,097	To be issued as 7,844,889

			on Nov. 30 2010
11/738,486	04-21-2007	Continuation of 09/878,097	7,831,926
11/738,932	04-23-2007	Continuation of 09/878,097	7,823,083
11/868,124	**10-05-2007**	**Divisional of 11/045,757**	**Non-final Action**
12/326,092	12-01-2008	Continuation of 11/045,757	Awaiting First Action
12/941,106	11-08-2010	Continuation of 11/738,486	Awaiting First Action

Status of Patent Litigation

As previously disclosed, in a complaint filed May 10, 2010, the assignee of the present application SoftView LLC has asserted one or more claims in U.S. Patent 7,461,353 (U.S. Application No. 11/045,757). The present application is a divisional of 11/045,757).

On November 10, 2010, SoftView LLC filed a motion to leave to amend its complaint so as to add U.S. Patent 7,831,926 to the complaint, along with a proposed amended complaint and brief in support of the motion to leave to amend the complaint. The '926 patent issued from U.S. Application Serial No. 11/738,486, which is a continuation U.S. Application Serial No. 09/878,097 (issued as U.S. Patent 7,210,099). 11/045,757 is a divisional of 09/878,097. A copy of this motion, along with a proposed amended complaint are provided electronically herewith and listed on the PTO Form PTO/SB/08b filed herewith.

Conclusion

In consideration of the claim amendments and remarks submitted herewith, Applicants respectfully assert that all pending claims are in condition for allowance. If, however, the Examiner believes that there are any unresolved issues requiring adverse action in any of the claims now pending in the application, it is requested that the Examiner telephone R. Alan Burnett at (425) 417-4729 or (425) 562-0923 so that appropriate arrangements can be made for resolving such issues as expeditiously as possible.

Respectfully submitted,

LAW OFFICE OF R. ALAN BURNETT, PS

Date: *November 24, 2010* *|S| R. Alan Burnett*
R. Alan Burnett
Reg. No. 46,149

4108 131st Ave SE
Bellevue, WA 98006

EXHIBIT B

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	11045757
	Filing Date	2005-01-28
	First Named Inventor	Rohrabaugh
	Art Unit	2176
	Examiner Name	Tran, Quoc A
	Attorney Docket Number	005207.P001XD

	3	20010047428		2001-11-29	Hunter, Kevin D.	
	4	20040049598		2004-03-11	Tucker et al.	

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

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
	1							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

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
	1	BENJAMIN B. BEDERSON ET AL., Pad++: A Zoomable Graphical Sketchpad For Exploring Alternate Interface Physics, September 19, 1995, http://www.cs.unm.edu/pad++	☐
	2	BENJAMIN B. BEDERSON ET AL., A Zooming Web Browser, SPIE 1996, http://www.cs.umd.edu/hcil/jazz/learn/papers/spie-96-webbrowser.pdf	☐
	3	Specification for Simple Vector Format (SVF) v1.1 Jan. 16, 1995	☐
	4	Specification for Simple Vector Format (SFV) v2.0 Dec. 6, 2000, http://www.svf.org/spec.html	☐

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /QT/

Specification for the Simple Vector Format (SVF) v1.1

Overview

SVF is designed to be a simple format for describing vector images. The basic drawing objects include points, lines, circles, arcs, bezier curves and text. Features of the format include layers (for controlling the visibility of objects), hyperlinks (for allowing the user to click on a portion of the drawing to perform an action), notifications (for sending messages when the user has passed a certain zoom level), fills, and the ability to override the default colors.

The file is broken into 3 sections, the intro, the header, and the main body. The intro is simply a text string identifying the file as SVF. The header includes general information useful for displaying or manipulating the image like layers, extents, and colors. The main body describes how to draw the image and hyperlinks.

Objects are drawn in the order they occur in the file, so if you wish to draw overlapping objects the latter object will be drawn on top of previous objects.

Command parameters are integers (coordinates, colors, etc.). Most command parameters can be different lengths (1 to 8 bytes with the 1 and 2 byte versions currently implemented and the 8 byte floating point version implemented for scale and base point), allowing the programmer to create more efficient files. The two most significant bits of the command byte determine the length (00 - 1, 01 - 2, 10 - 4, 11 - 8). So, for example, the 1 byte form of the Point command is hex 10 and the 2 byte form is hex 90. Numbers are stored bigendian.

As with most vector formats, the origin (0,0) is in the lower left corner with increasing x coordinates moving to the right and increasing y coordinates moving up. Coordinates are specified as positive integers; offsets can also be negative.

The programming examples reference code from svfout.c (with prototypes included in svfout.h and defines in svfcom.h) which handles outputting the commands as described in the syntax document. These functions automatically handle outputting the appropriate form of the command based on the required length. Note that 'fp' is a file pointer (or reference).

The file intro

The file starts out with text which describes the file as SVF and lists a version number. The following code accomplishes this and then closes the file when the file is finished. Note that SVFOpen writes a null-terminated text string (e.g. "SVFv1.1") to begin the file and SVFClose simply closes the file.

```
FILE *fp = SVFOpen("name.svf");

        ...

SVFClose(fp);
```

Drawing lines

The simplest way to draw lines is with the MoveTo and LineTo commands. MoveTo moves the current point to a new location without drawing. It is used to specify the starting point for most drawing commands. LineTo draws a line from the current point to the specified position, updating the current point in the process.

The following code draws a line from (10,10) to (50,30).

```
SVFOutputMoveTo(fp,10,10);

SVFOutputLineTo(fp,50,30);
```

These two lines combined with the two lines listed in the file intro section would create a valid SVF file which simply drew a line.

Defaults

Most settings have defaults. The default drawing color is white (color index 255); the default background color is black (color index 0). The current point is initially at (0,0). The current layer is 0. Text height starts out at 10. Pen width defaults to 0. Fill mode is off. Default extents are (0,0)-(255,255).

Image extents

Since the size of the image can change, you may need to include the image's size (or extents) so the program viewing it will know how to display it initially. The default is (0,0)-(255,255). Extents information is required if your extents differ from the default. For example, if your image was entirely contained in the square (0,0)-(5000,5000), the following code would convey this.

```
SVFOutputExtents(fp,0,0,5000,5000);
```

Polylines

Polylines are a more efficient method for drawing a series of connected lines. Note that polylines start drawing from the current point. The value for the number of points allowed in the polyline takes up the same number of bytes as the coordinate values (limit 255 for 1 byte coordinate values, etc.).

Unlike the other convenience functions, polylines require that the programmer decide whether to use the short form (0-255) or long form (0-65535) of the command. Each full polyline command consists of the command describing how many points will be specified followed by each of the points. The final parameter specifies how many bytes each value requires. A polyline displayed with fill mode on is displayed as a filled polygon. The final line segment is always an implicit line from the first point (current pen position) to the last point. Polyline link regions follow the same rules as drawing a polygon.

```
SVFOutputMoveTo(fp,100,200);
SVFOutputPolylineStart(fp,3,2); /* 3 pts in polyline */
SVFOutputPolyPoint(fp,200,100,2);
SVFOutputPolyPoint(fp,300,200,2);
SVFOutputPolyPoint(fp,200,300,2);
```

Relative lines and polylines

MoveTo, LineTo, and Polyline also have a relative form, where each point describes an offset from the current point. The offsets can be positive or negative. Thus the previous polyline example could be rewritten using the 1 byte form of relative polylines.

```
SVFOutputMoveTo(fp,100,200);
SVFOutputRelPolylineStart(fp,3,1);
SVFOutputPolyPoint(fp,100,-100,1);
SVFOutputPolyPoint(fp,100,100,1);
SVFOutputPolyPoint(fp,-100,100,1);
```

Circles and arcs

Circles and arcs are drawn with their center at the current point A filled arc looks like a piece of pie while a plain arc just consists of the curved arc segment. Arc link regions also behave like a piece of pie.

The following example draws a circle with radius 20 with a center at (10,10).

```
SVFOutputMoveTo(fp,10,10);
SVFOutputCircle(fp,20);
```

Arcs are specified by radius and start and end angles, the arc being drawn counter clockwise. The 1 byte form of the arc command describes the angles ranging from 0 to 255 in 255/360 degree increments. The 2 byte form angles range from 0 to 65535 in 65535/360 degree increments.

This code draws an arc centered at (10,10) with a radius of 20 starting at 30/360 degrees sweeping

FH_DEF000213

Motorola PX 1026_172

counterclockwise to 240/360 degrees.

```
SVFOutputMoveTo(fp,10,10);
SVFOutputArc(fp,20,30,240);
```

A problem with the arc convenience function comes from the fact that most of the functions automatically determine the number of bytes used by its parameters. If you wanted an arc of radius 200 starting at 10 out of 65535 angle units and continuing to 243 out of 65535 angle units, the command would assume that since 200, 10, and 243 are all under 256, the 1 byte form should be used when the 2 byte form was actually required. To circumvent this problem, you can use SVFOutputArcLength to force a different version of the command.

Text

Text can be displayed using a default system dependent courier-like font. Text is drawn using the current text height (default is 10). A width can also be specified with the text. If this value is 0, the text will be drawn using the font's default width. If a width is specified, the text will be scaled to fit. This is suggested if the width of the text is important because the font metrics may differ between systems. The current point is used for the baseline of the text. Descenders may go below this baseline. Text in the SVF file is null-terminated.

```
SVFOutputTextHeight(fp,100);
SVFOutputMoveTo(fp,10,10);
SVFOutputText(fp,0,"Hello world");       /* text width 0 */
```

Points and rectangles

You can draw points. You can also draw rectangle outlines or filled areas (dpending on the fill mode) which start at the current point. Rectangles do not update the current point.

```
SVFOutputPoint(fp,50,50);
SVFOutputMoveTo(fp,10,10);
SVFOutputRectangle(fp,100,100);
```

Bezier curves

Cubic Bezier curves are supported. The first end point is the current point. The command specifies two control points followed by the second end point. This command updates the current point. Note that Bezier curves do not go through the control points. For link regions and filled regions, a Bezier curve region is defined as the area between the curve and a line connecting its two end points.

The following code draws a cubic bezier curve from (10,10) to (240,20) using the control points (105,110) and (200,100).

```
SVFOutputMoveTo(fp,10,10);
SVFOutputBezier(fp,105,110,200,100,240,20);
```

Colors

There is a default color table consisting of 256 colors which will be reasonable for most images. Alternately, you can specify your own color table.

If you work with the default colors, the following constants can be used for the basic colors:

`SVFC_Black`	0
`SVFC_White`	255
`SVFC_LightGray`	219
`SVFC_Gray`	146
`SVFC_DarkGray`	73
`SVFC_Red`	224
`SVFC_Green`	28
`SVFC_Blue`	3
`SVFC_Yellow`	252
`SVFC_Cyan`	31
`SVFC_Magenta`	227

```
SVFOutputSetColor(fp,SVFC_Yellow);
```

The color definition can be thought of as the byte rrrgggbb where each color specification (red, green, or blue) is the most significant bits of the desired intensity. For example, if you want the RGB value 34,200,145 (00100010, 11001000, 10010001 in binary), where each value ranges form 0 (black) to 255 (pure color) the appropriate color index would be 00111010 in binary, or 58 in decimal.

You can use SVFGetClosestColor to turn an RGB value into an index referencing the default color table.

SVFOutputSetColor sets the current color for drawing objects until the next color change. SVFOutputBackground defines which color is used for the background. The default background color is black (0) and the default drawing color is white (255). SVFOutputTransparent sets a color to be transparent (objects drawn with that color are drawn with the background color of the target window). This overrides the definition for that color in the color table. Note that these commands use a color index into the image's color table. There can only be a single background color and a single transparent color.

See svfxampl.c for an example of how to output your own color table.

In order to define an object which doesn't appear (useful for hyperlink regions which don't display), you can set the current color to invisible. Objects are not drawn until a non-invisible color is set.

```
SVFOutputInvisible(fp);
```

Pen width

By default, the pen width is 0, which means that all lines will be drawn 1 pixel wide. Alternately a pen width (in SVF coordinates) can be specified. Note that filled regions and link regions are unaffected by pen width.

```
SVFOutputSetPenWidth(fp,100);
```

Filled regions

To display filled regions, you can turn the fill mode on (the default is off). This affects polylines, rectangles, circles, arcs, and bezier curves. All regions are displayed filled until the fill mode is turned off.

```
SVFOutputFillMode(fp,SVF_ON);

SVFOutputMoveTo(fp,100,100);

SVFOutputCircle(fp,50);          /* circle is filled */

SVFOutputRectangle(fp,30,20);    /* rectangle is filled */

SVFOutputFillMode(fp,SVF_OFF);

SVFOutputRectangle(fp,20,30);    /* only outline is drawn */
```

Hyperlinks

Hyperlinks are regions defined in the image where the user can click in order to perform an action, similar to clicking on links in a Web document to bring up different information. Hyperlinks are enabled or disabled based on the layer they are on.

Links are defined in the main body. You set whether the following link regions will be 1 dimension or 2 dimension, the link action, and optional explanatory text. The link regions are displayed like any other entity unless the current color is invisible. On the Web, the action would be a URL which will be activated by the link. If this text is an empty string, the associated link is inactive. The explanatory text may be used by a browser to give the user more information about the link. If the associated layer is turned off, the link is deactivated.

A 1 dimensional link means that the user would select the line or curve. A 2 dimensional link is chosen by picking inside the region (as for a circle or polygon region). A point link is selected by picking close to the point. Text links are defined as the bounding box around the text.

Note that after a call to SVFOutput1dLink or SVFOutput2dLink, all following objects are links using the specified action until another link definition or an SVFOutputEndLink is found. SVFOutputEndLink is actually a call to SVFOutput1dLink with two NULL parameters.

```
SVFOutput2dLink(fp,"http://www.university.edu/",NULL);

SVFOutputMoveTo(fp,10,10);       /* triangle link region */

SVFOutputPolylineStart(fp,2,2);

SVFOutputPolyPoint(fp,500,500);
```

```
SVFOutputPolyPoint(fp,250,10);


SVFOutput1dLink(fp,"URL","select the circle outline");

SVFOutputMoveTo(fp,50,50);

SVFOutputCircle(fp,20);


SVFOutputEndLink(fp);
```

The link syntax allows links to activate a URL to bring up another document or toggle layers.

Link regions are checked in the reverse order they are defined so the image display matches how links are activated. If regions overlap, the first region selected will be the activated link.

Layers

Layers give you the ability to group objects together so the user can turn on or off the display of portions of the drawing. For example, if you drew a map with coastlines, rivers, and cities, you may want to allow the user to turn on and off the display of the city names so they can more easily view the information they are interested in. Each of these groups could be placed on separate layers. Hyperlinks are enabled or disabled based on the layer they are on.

The first step in using layers is to specify a layer table in the header section. This is simply a list of layer names and the initial display status (on or off) of each layer. The following example creates 3 layers called Coastline, Rivers, and Cities. If SVF_ON and SVF_OFF are constants representing 1 and 0 respectively, all objects on the first two layers would initially be displayed, but objects on the Cities layer would not be displayed. These settings can be changed when the user views the image.

```
SVFOutputLayerTable(fp,3);      /* 3 entries */

SVFOutputLayerEntry(fp,SVF_ON,"Coastline");

SVFOutputLayerEntry(fp,SVF_ON,"Rivers");

SVFOutputLayerEntry(fp,SVF_OFF,"Cities");
```

In the main body you can switch layers as often as you like. After you switch layers, all subsequent objects are part of that layer until you switch layers again.

```
SVFOutputSetLayer(fp,0);        /* Coastline layer... */

SVFOutputMoveTo(fp,10,10);

SVFOutputLineTo(fp,50,50);

SVFOutputSetLayer(fp,2);        /* Cities layer... */

SVFOutputLineTo(fp,30,50);
```

Note that the status of layers has no affect on commands which change settings: SetColor, SetLayer, TextHeight, and FillMode. The current point is updated by the appropriate commands regardless of whether the object is on a layer which is displayed or not.

Notifications

With notifications, actions can be performed when a certain magnification has been reached. These actions are the same as hyperlink actions and have the same syntax. So, for example, when the user has zoomed in a ways, layers could be turned on so extra information could be displayed. Or information describing the current view can be passed back so a new image could be sent based on where the user was examining in the original image.

A notification can be triggered by zooming in or zooming out. This is specified by the width of the view. If, for example, a file was 1000 units on a side and you wanted to turn on the "Cities" layer after the current view was less than or equal to 300 units on a side, and have the layer turned back off if they zoomed out again, the following code would accomplish this:

```
SVFOutputNotifyTable(fp,1,1);    /* 1 zoomout and 1 zoomin */

        /* first entry - notify when view <= 300 units wide */

SVFOutputNotifyEntry(fp,"layeron=Cities",300);

        /* second entry - notify when view >= 300 units wide */

SVFOutputNotifyEntry(fp,"layeroff=Cities",300);
```

Note that when you initiate the notify table, the two numbers you pass indicate the number of entries in the zoomin portion of the table and the the zoomout portion respectively.

Notifications get executed before the image is redisplayed. If more than one notification gets triggered, the first one defined in the table is called first, then the others in order.

MIME type

The MIME type for SVF files is *image/vnd.svf*. The standard file extension is .SVF or .svf.

Miscellaneous information

An image can contain a suggestion for how big it would like to appear. You can specify the suggested width of the image in millimeters. The display program is not required to honor the request. The SVF Plug-in for Netscape currently ignores this value.

```
SVFOutputWidth(fp,100); /* image should be 100 mm wide */
```

An SVF file can contain other data which isn't necessarily picked up by a viewer so extra information can be embedded in the file. A name can be added in the header section. Arbitrary data can be stored in the main body of the file.

```
SVFOutputName(fp,"some text");

SVFOutputData(fp,20,ptr);        /* output 20 bytes of data */
```

Scale and base point can be added to the image. This information is not used for display. It would be used if the original coordinate system needed to be recreated from the integer values stored in the file. An SVF coordinate pair would first be added to the base point, then each value would be multiplied by the scale factor. The base point corresponds to the origin (0,0) of the SVF file. This command is the only one that currently uses the 8 byte floating point version of the commands.

```
SVFOutputTransform(fp,3.214,1.92,-4.3); /* scale, basex, basey */
```

A flag can be specified which indicates that a display program is free to rearrange entities within a layer. This is useful for certain optimizations. Note that it is still valid to display entities in the order they occur in the file. 0 is the default. 1 indicates you can rearrange within a layer.

```
SVFOutputSortByLayer(fp,1);
```

Related pages...

*SVF Technical Information
*SVF Syntax
*HTML Syntax for hyperlinks and embedding SVF images in HTML documents

scotts@softsource.com

Specification for Simple Vector Format (SVF) v2.0

About SVF

Design Criteria

Specification

SVF XML

API

SVF Products

Table of Contents

Overview

SVF is designed to be a simple format for describing vector images. The basic drawing objects include points, lines, circles, arcs, bezier curves, text, ellipses, and bitmaps. Features of the format include layers (for controlling the visibility of objects), hyperlinks (for allowing the user to click on a portion of the drawing to perform an action), notifications (for sending messages when the user has passed a certain zoom level), fills, the ability to override the default colors, object identifiers (for linking information to a database or the original drawing), custom user data and saved views. All text is stored in UTF-8 encoded Unicode.

The file is broken into 3 sections, the intro, the header, and the main body. The intro is simply a text string identifying the file as SVF and the version number. The header includes general information useful for displaying or manipulating the image like layers, extents, and colors. The main body describes how to draw the image and hyperlinks.

Objects are drawn in the order they occur in the file, so if you wish to draw overlapping objects, the later objects will be drawn on top of previous objects. Objects are drawn using the concept of a current state. An SVF file starts in a default state which can be changed in the file. Some commands start from the current pen position and may modify it.

As with most vector formats, the origin (0,0) is in the lower left corner with increasing x coordinates moving to the right and increasing y coordinates moving up. Using the Flags header command you can instead have the origin in the upper left, with y coordinates increasing as you move down. Coordinates are specified as positive integers or double precision floating point; offsets are signed integers or double precision floating point.

Header

The header contains information necessary for displaying and organizing the graphic information in the body. All header information should be specified before the body. All header commands are optional and should occur only once if at all (though most files will at least include Extents). Valid header commands include all tokens from 1 to 15 plus Extension. The Extension command is special and may appear multiple times.

- Background
- BaseURL
- ColorTable
- DataTypes
- Extension
- Extents
- Flags
- FontTable
- InitialView
- LayerTable
- Name
- NextObjectID
- NotifyTable
- Transform
- Transparent
- ViewTable
- Width

Body

Body commands occur after the header. Unlike header commands, each body command may appear multiple times. Valid body commands consist of all tokens from 16 on up plus Extension.

- 1dLink
- 2dLink
- Arc
- Bezier
- Circle
- Data
- Data2
- Ellipse
- EllipseArc
- Extension
- FillMode
- Invisible
- LineTo
- Link
- MoveTo
- Object
- Point
- Polyline
- RelLineTo
- RelMoveTo
- RelPolyline
- Rectangle
- SetColor
- SetFixedWidth
- SetFont
- SetLayer
- SetLineType
- SetObjectID
- SetPenWidth
- Text
- TextAngle
- TextFlags
- TextHeight

Command Structures

Tokens

An SVF *token* or command is a single byte. This is generally followed by command specific information - numbers, text, etc. The two most significant bits of the token indicate how many bytes numbers will take, 1, 2, 4 or 8 (00 - 1, 01 - 2, 10 - 4, 11 - 8). For example, the token for the Point command is 16 (binary 0001 0000). 0001 0000 indicates that the point coordinates will be 1 byte each; 0101 0000 indicates 2 bytes each; 1001 0000 indicates 4 bytes; and 1101 0000 indicates 8 bytes.

Some commands may require an extra byte to describe the token. These commands start with an XHeader token (value 15) followed by the specific token. For example, DataTypes consists of an XHeader token (15) followed by the DataTypes token (1). The precision of this specific command is based on the high bits of the DataTypes token.

Numbers

The precision numbers and counts are stored at depends on the associated token. If it is stored in 1, 2 or 4 bytes, it generally should be interpreted as a positive integer. 8 byte numbers generally should be interpreted as a double precision floating point number. Use the Transform command to convert integer coordinates into floating point values. Raw integer coordinates are refered to as *SVF Integer Coordinates*. Floating point coordinates and transformed integer coordinates are known as *SVF Transformed Coordinates*. All numbers are stored bigendian. In this document, numbers listed for a command that have a green background depend on the token for their precision.

Offsets follow the same rules as numbers for how they're stored. The only difference is that integers are signed values instead of unsigned.

Angles follow the same rules as numbers for how they're stored. 1, 2 and 4 byte angles can be converted to radians using the formula *angle * 2 * PI / MAXINT* where *MAXINT* is the maximum value for the appropriate size integer. For example, a 2 byte integer would use the formula *angle * 2 * PI / 0xffff*. Angles stored as floating point are stored in radians. Angles start from the positive x axis and go counter clockwise.

IDs follow the same rules as numbers for how they're stored with the exception that the 8 byte version is interpreted as an 8 byte integer. Thus they are always unsigned integers.

Strings

All text in an SVF file, including layer names and text objects, are stored using the UTF-8 encoding of Unicode. This allows for fully international text. An advantage of UTF-8 is that standard ASCII (values 127 and lower) is also valid UTF-8. Most other European characters can be encoded in 2 bytes and others in 3 bytes

Unicode characters from 0-127 are stored in 1 byte. Characters from hex 80 to 7ff (decimal 128 to 2047) are stored in 2 bytes and hex 800 to ffff (decimal 2048 to 65535) are stored in 3 bytes.

```
0000-007F   0xxxxxxx

0080-07FF   110xxxxx 10xxxxxx

0800-FFFF   1110xxxx 10xxxxxx 10xxxxxx
```

- For a complete description of Unicode, see unicode.org.
- For a more complete description of UTF-8, see The Internet Engineering Task Force's UTF-8 specification.

Link Actions

Hyperlinks and notifications can link to external documents or perform various actions. These are stored as a string and represent a URL.

Defaults

Until explicitly set in the SVF image, the following defaults are assumed.

extents	(0,0)-(255,255)
drawing color	white (color index 255)
background color	black (color index 0)
transparent color	none
available colors	256 colors (see colors) (or RGB)
pen position	(0,0)
layer	entry 0 (if there are layers)
fill mode	off
text height	10 integer units
text rotation	0 (horizontal)
text font	font entry 0 or monospaced if no fonts
pen width	0
linetype	0 (continuous)

Graphic objects

Most graphic objects start drawing from the current pen position and many also update it. To update the current point without drawing anything, you can use MoveTo or RelMoveTo. MoveTo moves the pen to the specified absolute coordinates. RelMoveTo specifies a relative offset. It is useful when a relative move takes less precision.

MoveTo	**(body)**		
type	token	number	number
value	17	x coordinate	y coordinate

updates current point

RelMoveTo	**(body)**		
type	token	offset	offset
value	20	x offset	y offset

updates current point

Changing the color and other properties of graphic objects is described in the graphic object properties section.

Lines

Lines are drawn from the current pen position and also update the current position. You can use either LineTo or RelLineTo. As with RelMoveTo, RelLineTo is useful when relative coordinates can be stored in less space than absolute coordinates.

LineTo	(body)		
type	token	number	number
value	18	x coordinate	y coordinate

uses current point
updates current point

RelLineTo	(body)		
type	token	offset	offset
value	21	x offset	y offset

uses current point
updates current point

Polylines

Polylines are a more efficient method for drawing a series of connected lines. They are drawn from the current pen position and also update the current position. The value for the number of points allowed in the polyline takes up the same number of bytes as the coordinate values (limit 255 for 1 byte coordinate values, etc.).

A polyline displayed with fill mode on is displayed as a filled polygon. The final line segment of the polygon is always an implicit line from the first point (current pen position) to the last point. Polyline link regions follow the same rules as drawing a polygon.

As with moves and lines, polylines can be specified using relative coordinates for more efficiency.

Polyline	(body)			
type	token	count	{ number	number }
value	19	count	x coordinate	y coordinate

uses current point
updates current point
uses fill mode

RelPolyline	(body)			
type	token	count	{ offset	offset }
value	22	count	x offset	y offset

uses current point
updates current point
uses fill mode

Points and rectangles

Points are simply a dot drawn at the given location

You can also draw rectangle outlines or filled areas (depending on the fill mode) which start at the current

FH_DEF000225

Motorola PX 1026_184

point. If the port origin is in the lower left (the default), the current point is interpreted as the lower left of the rectangle. If the port origin is in the upper left, the current point becomes the upper left corner of the rectangle.

Point	(body)		
type	token	number	number
value	16	x coordinate	y coordinate

Rectangle	(body)		
type	token	number	number
value	23	width	height

uses current point
uses fill mode

Circles and arcs

Circles and arcs are drawn with their center at the current point A filled arc looks like a piece of pie while a plain arc just consists of the curved arc segment. Arc link regions also behave like a piece of pie.

Arcs are specified by radius and start and end angles, the arc being drawn counter clockwise.

Circle	(body)	
type	token	number
value	24	radius

uses current point
uses fill mode

Arc	(body)			
type	token	number	angle	angle
value	25	radius	start angle	end angle

uses current point
uses fill mode

Ellipses and elliptical arcs

Ellipses and elliptical arcs follow similar rules to circles and arcs. They are specified using the height and width of the ellipse. Ellipses can be rotated.

The angles for elliptical arcs are actually the parameters for the parametric equation of an ellipse. An easy way to think of it is to picture the angles in a circular arc, then squish (scale) the arc to make an elliptical arc.

Ellipse	(body)			
type	token	number	number	angle
value	41	width	height	rotation

uses current point

uses fill mode
added with SVF2

EllipseArc	**(body)**					
type	token	number	number	angle	angle	angle
value	42	width	height	rotation	start parameter	end parameter

uses current point
uses fill mode
added with SVF2

Bezier curves

Cubic Bezier curves are supported. The first end point is the current point. The command specifies two control points followed by the second end point. This command updates the current point. Note that Bezier curves do not go through the control points. For link regions and filled regions, a Bezier curve region is defined as the area between the curve and a line connecting its two end points.

Bezier	**(body)**						
type	token	number	number	number	number	number	number
value	26	control point 1 x	control point 1 y	control point 2 x	control point 2 y	end point x	end point y

uses current point
updates current point
uses fill mode

Text

Specific text properties include height, width, rotation, font. The current point is used for the lower left corner of the text. Descenders may go below this baseline. Text is drawn using the current text height (default is 10). Text can be Unicode.

A width can also be specified with the text. If this value is 0, the text will be drawn using the font's default width. If a width is specified, the text will be scaled to fit. This is suggested if the width of the text is important because the font metrics may differ between systems.

Text can be drawn using different fonts. A list of fonts can be included in the header section of the file, then referenced by index in the body.

Text	**(body)**		
type	token	number	string
value	27	width	text

uses current point

TextHeight	**(body)**	
type	token	number
value	28	height

TextAngle	(body)	
type	token	angle
value	43	text rotation

added with SVF2

SetFont	(body)	
type	token	number
value	38	font table index

added with SVF2

To make subsequent text bold or italics, set the appropriate bit in the value passed to the TextFlags command.

By default, text is drawn using the current point as the location for the baseline. Alternately, you can have the current point be interpreted as the point below text descenders.

By default, when you specify the height of text, this is considered to be the height of a capital M. Alternately, you can have text height be the *cell height*, i.e. character height including ascenders, descenders and internal leading.

TextFlags	(body)	
type	token	number
value	48	flags

added with SVF2

bits	C constant	Java constant	meaning
00000001	SVFTF_Bold	eTF_Bold	Bold if set, regular if not
00000010	SVFTF_Italics	eTF_Italics	Italics if set, regular if not
00000000	SVFTF_BasePointBaseLine	eTF_BasePointBaseLine	Text basepoint is at the baseline
00000100	SVFTF_BasePointBottom	eTF_BasePointBottom	Text basepoint is below the descenders
00001000	SVFTF_BasePointReserved1	eTF_BasePointReserved1	*reserved for future use*
00001100	SVFTF_BasePointReserved2	eTF_BasePointReserved2	*reserved for future use*
00000000	SVFTF_TextHeightM	eTF_TextHeightM	Text height is the height of an M
00010000	SVFTF_TextHeightCell	eTF_TextHeightCell	Text height is the cell height
00100000	SVFTF_TextHeightReserved1	eTF_TextHeightReserved1	*reserved for future use*
00110000	SVFTF_TextHeightReserved2	eTF_TextHeightReserved2	*reserved for future use*
01000000	SVFTF_Underline	eTF_Underline	underline

Bitmaps and other objects

Bitmaps can be displayed from an SVF file. The bitmap resides in an external file and is referenced using an absolute or relative URL. The lower left corner uses the current pen position. You are required to specify the width (in SVF coordinates) of the area the bitmap should be displayed into. If the height is 0, the aspect ratio of the bitmap is used to determine the actual height. If the height is specified, the bitmap is scaled to fit in the given box. If the port origin is in the lower left (the default), the current point is interpreted as the lower left of the rectangle. If the port origin is in the upper left, the current point becomes the upper left corner of the rectangle.

A MIME type can be specified to describe what type of object is embedded. An SVF viewer must a least support *image/jpeg* for JPEG bitmaps but may support others. A MIME type doesn't have to be specified for common bitmap formats.

If the object can't be display for some reason (the file isn't found or the SVF viewer doesn't support the given MIME type), alternate text can be displayed instead. It will be displayed from the current point using the current text properties.

Object	**(body)**					
type	token	number	number	string	string	string
value	40	width	height	URL of object	MIME type	alternate text

uses current point
added with SVF2

Graphic object properties

Graphic objects in SVF are drawn using the idea of a current state. Instead of specifying a color, layer, etc. for every object, you only specify when the current value changes. The objects that are drawn use whatever value was last set.

Hyperlinks

Hyperlinks are regions defined in the image where the user can click in order to perform an action, similar to clicking on links in a Web document to bring up different information. Hyperlinks are enabled or disabled based on the layer they are on.

Links are defined in the main body. You set whether the following link regions will be 1 dimension or 2 dimension, the link action, and optional explanatory text. The Link command also allows information about where on the image the user picked. The link regions are displayed like any other entity unless the current color is invisible. On the Web, the action would be a URL which will be activated by the link. If this text is an empty string, the associated link is inactive. The explanatory text may be used by a browser to give the user more information about the link. If the associated layer is turned off, the link is deactivated.

A 1 dimensional link means that the user would select the line or curve. A 2 dimensional link is chosen by picking inside the region (as for a circle or polygon region). A point link is selected by picking close to the point. Text links are defined as the bounding box around the text.

Note that all objects after a 1dLink, 2dLink or Link command are hyperlinks until another link definition or an empty link is found. If both the URL and alternate text are empty, subsequent objects are not hyperlinks.

SVF Script allows links to activate a URL to bring up another document or toggle layers.

Link regions are checked in the reverse order they are defined so the image display matches how links are activated. If regions overlap, the first region found will be the activated link.

1dLink	**(body)**		
type	token	string	string
value	35	link action	title

2dLink	**(body)**		
type	token	string	string
value	36	link action	title

If you wish to specify the target or have information about the location of where the user picked in a hyperlink area, use the Link command. This is essentially an extended version of the 1dLink and 2dLink commands; the *link action* and *title* parameters have the same meaning. If the SVFLF_IsQuery flag is set, this link command will cause a query string of the form *url?<x>,<y>* or *url?svflink=x:<x>,y:<y>* to be passed back to a server (e.g. *http://whatever.com/file.html?43,50* or *http://whatever.com/file.html?svflink=x:43.23,y:50.98*), where *url* is the *link action* and *x,y* is the point the user picked.

If the scale and basepoint are 0 for this command, the coordinates are passed back as SVF Transformed Coordinates. If scale and basepoint are specified, SVF Transformed Coordinates are run through the additional transformation: (x',y') = ((x + basex) * xscale, (y + basey) * yscale). The current view width and pixel width are likewise considered to be in SVF Transformed Coordinates. If the additional transformation is specified, width' = (width + basex) * xscale.

Optionally, you can specify the *target* of where the hyperlinked file will be put. This can be *_self*, *_parent*, *_top*, *_blank*, or the name of an HTML frame. *target* may be a null string. See frame targets in the HTML documentation for more information.

Link	**(body)**								
type	token	number	number	number	number	byte	string	string	string
value	47	xscale	yscale	basex	basey	flags	link action	title	target

added with SVF2

The *flags* parameter is summarized in the following table. If the low bit is set, it's a 2 dimensional link - otherwise it's 1 dimensional. If the second bit is set, a query string will be added on to the URL passed back to the server. If the third bit is set, all coordinates should be rounded to the nearest integer and the query string will look like *url?<x>,<y>*. If the third bit is not set, the query string will look like *url?svflink=x:<x>,y:<y>*.

bit	C constant	Java constant	meaning
00000001	SVFLF_2D	eLF_2D	2D if set, 1D if not
00000010	SVFLF_IsQuery	eLT_IsQuery	add query string to URL
00000100	SVFLF_ShortForm	eLT_ShortForm	round coordinates to nearest integer

Layers

Layers give you the ability to group objects together so the user can turn on or off the display of portions of the drawing. For example, if you drew a map with coastlines, rivers, and cities, you may want to allow the user to turn on and off the display of the city names so they can more easily view the information they are interested in. Each of these groups could be placed on separate layers. Hyperlinks are enabled or disabled based on the layer they are on.

The first step in using layers is to specify a layer table in the header section. This is simply a list of layer names and the initial display status (on or off) of each layer. The status can be changed when the user views the image.

In the main body you can switch layers as often as you like. After you switch layers, all subsequent objects are part of that layer until you switch layers again. Layer numbers used for SetLayer start referencing the layer table from 0.

Note that the status of layers has no affect on commands which change settings: SetColor, SetLayer, TextHeight, and FillMode. The current point is updated by the appropriate commands regardless of whether the object is on a layer which is displayed or not.

LayerTable	**(header)**			
type	token	count	{ byte	string }
value	5	count	on or off	layer name

SetLayer	**(body)**	
type	token	number
value	30	layer table index

Colors

There is a default color table consisting of 256 colors which will be reasonable for most images. Alternately, you can specify your own color table which allows any number of colors, or simply reference all colors as RGB values.

If you work with the default colors, the following constants can be used for the basic colors:

```
SVFC_Black        0
SVFC_White        255
SVFC_LightGray    219
SVFC_Gray         146
SVFC_DarkGray     73
SVFC_Red          224
SVFC_Green        28
SVFC_Blue         3
SVFC_Yellow       252
```

```
SVFC_Cyan            31
SVFC_Magenta         227
```

The color definition can be thought of as the byte rrrgggbb where each color specification (red, green, or blue) is the most significant bits of the desired intensity. For example, if you want the RGB value 34,200,145 (00100010, 11001000, 10010001 in binary), where each value ranges form 0 (black) to 255 (pure color) the appropriate color index would be 00111010 in binary, or 58 in decimal.

SetColor sets the current color for drawing objects until the next color change. Background defines which color is used for the background. The default background color is black (index 0 in the default color table) and the default drawing color is white (index 255 in the default color table). Transparent sets a color to be transparent (objects drawn with that color are drawn with the background color of the target window). This overrides the definition for that color in the color table. Note that these commands use a color index into the image's color table. There can only be a single background color and a single transparent color.

Alternately, you can use RGB values when setting colors using the 4 byte versions of SetColor and Background. The upper byte is reserved for future use and should be set to 255. The next byte is the red component, followed by the green component with the blue component in the lower byte. (Note that this also puts a theoretical limit on the size of a color table, but having more than 64k worth of color entries wouldn't be very practical anyway.) Because SVF is designed so it can be displayed as it's loaded without having to make an initial pass through the file, an SVF viewer may choose to use the default palette and simply find the closest match for any RGB value specified if display is on a palette system.

In order to define an object which doesn't appear (useful for hyperlink regions which don't display), you can set the current color to invisible using Invisible. Subseqent objects will not be drawn until the next SetColor.

ColorTable	**(header)**				
type	token	count	{ byte	byte	byte }
value	3	count	red (0-255)	green (0-255)	blue (0-255)

Background	**(header)**	
type	token	number
value	4	color table index or RGB value

Transparent	**(header)**	
type	token	number
value	9	color table index

SetColor	**(body)**	
type	token	number
value	29	color table index or RGB value

Invisible	**(body)**
type	token

value	37

Filled regions

To display filled regions, you can turn the fill mode on (the default is off). This affects polylines, rectangles, circles, arcs, ellipses, elliptical arcs, and bezier curves. All regions are displayed filled until the fill mode is turned off.

FillMode	**(body)**	
type	token	number
value	32	0 - off, 1 - on

Pen width

By default, the pen width is 0, which means that all lines will be drawn 1 pixel wide. You can specify a width in SVF coordinates so that it will scale with the image by using SetPenWidth. Or you can have the pen width remain fixed by using SetFixedWidth and specifying the width in hundredths of a millimeter. Note that filled regions and link regions are unaffected by pen width.

SetPenWidth	**(body)**	
type	token	number
value	31	width in svf coordinates

SetFixedWidth	**(body)**	
type	token	number
value	45	width in hundredths of a millimeter

added with SVF2

Linetypes

SVF linetypes are used for lines, polylines, circles, arcs, ellipses, elliptical arcs and bezier curves. They do not change scale when zooming in or out on the drawing. The actual length in pixels of the dashes and spaces may vary per platform. Note that filled regions and link regions are unaffected by linetypes.

value	C enum	Java constant
0	SVFL_Solid	eLT_Solid
1	SVFL_Dash	eLT_Dash
2	SVFL_Dot	eLT_Dot
3	SVFL_DashDot	eLT_DashDot
4	SVFL_DashDotDot	eLT_DashDotDot

SetLineType	**(body)**

FH_DEF000233

Motorola PX 1026_192

type	token	number
value	44	linetype

added with SVF2

Object IDs

Object identifiers (or object IDs) can be assigned to an object or group of objects for linking information to a database or the original graphic information. It could also be used for grouping objects together. Object IDs can be up to 8 bytes; the 8 byte version of the command is interpreted as an integer not a floating point double. This information does not affect display in any way.

SetObjectID sets the object id that is inherited by subsequent objects. NextObjectID is a header command which sets the id to be used when a new object is created.

NextObjectID	**(header)**	
type	token	id
value	13	object id

added with SVF2

SetObjectID	**(body)**	
type	token	id
value	39	object id

added with SVF2

Fonts

Text can be displayed with different fonts. A table of fonts is placed in the header, then text can reference those fonts using an index into the table. Due to the fact that different systems support different fonts, you can list alternate fonts to use if a font isn't found as well as the general font type, or font "family". Multiple fonts in a single font table entry are separated by a comma.

value	C enum	Java constant	meaning
0	SVFF_Other	eFF_Other	Font family other than built in SVF types
1	SVFF_Serif	eFF_Serif	Font with serifs, e.g. Times New Roman
2	SVFF_SansSerif	eFF_SansSerif	Font with no serifs, e.g. Arial or Helvetica
3	SVFF_Monospaced	eFF_Monospaced	Fixed width font, e.g. Courier
4	SVFF_Symbol	eFF_Symbol	Symbol font, e.g. Dingbats
5	SVFF_Cursive	eFF_Cursive	Cursive font e.g. Zapf-Chancery
6	SVFF_Decorative	eFF_Decorative	Decorative font, e.g. Old English

FontTable	**(header)**			
type	token	count	{ byte	string }

FH_DEF000234

Motorola PX 1026_193

value	12	count	font family	font names

added with SVF2

Clipping

SVF has support for any arbitrary objects serving as clipping boundaries. You specify how many boundary objects there will be in the ClippingObjects command, then follow that with the SVF objects that will serve as the boundary. After the boundary objects, all further objects will be clipped to the boundary until you change boundaries or specify 0 clipping objects, which effectively turns off clipping. *Note: currently viewers are only required to support a single rectangular boundary.*

Clipping objects are displayed the same as other objects. If you don't wish to have them display, use the Invisible token first. Once you have finished with the clipping objects, call SetColor to turn display back on.

ClippingObjects	**(body)**	
type	token	count
value	50	number of objects

Image properties

Extents

You should describe the image's size (or extents) in the header section so the program viewing it will know how to display it initially. The extents should include everything in the image. The default is (0,0)-(255,255). Extents information is required if your extents differ from the default.

Extents	**(header)**				
type	token	number	number	number	number
value	2	xmin	ymin	xmax	ymax

Transformations

Scale and base point can be stored in an image which describe how to convert integer SVF coordinates back to the coordinate system of the original drawing or to the double coordinates stored in the SVF file. If (sx,sy) is a point in SVF integer coordinates and (ox,oy) is the corresponding point in the original coordinates (or SVF double coordinates), then (ox,oy) = ((sx + basex) * scale, (sy + basey) * scale). The base point corresponds to the origin (0,0) of the SVF file.

Transform	**(header)**			
type	token	number	number	number
value	10	scale	basex	basey

Notifications

With notifications, actions can be performed when a certain magnification has been reached. These actions

FH_DEF000235

Motorola PX 1026_194

are the same as hyperlink actions and have the same syntax. So, for example, when the user has zoomed in a ways, layers could be turned on so extra information could be displayed. Or information describing the current view can be passed back so a new image could be sent based on where the user was examining in the original image.

A notification can be triggered by zooming in or zooming out. This is specified by the width of the view. If, for example, a file was 1000 units on a side and you wanted to turn on the "Cities" layer after the current view was less than or equal to 300 units on a side, and have the layer turned back off if they zoomed out again, you would use a link action of "svfscript:layers.visible(true, Cities)" and a threshhold of 300 for the zoomin notification. The zoomout notification would have a link action of "svfscript:layers.visible(false, Cities)" and a threshhold of 300.

Note that when you start the notify table, the two numbers you pass indicate the number of entries in the zoomin portion of the table and the the zoomout portion respectively.

Notifications get executed before the image is redisplayed. If more than one notification gets triggered, the first one defined in the table is called first, then the others in order.

NotifyTable	**(header)**						
type	token	count	count	{ Name	Width }	{ Name	Width }
value	8	zoomin count	zoomout count	link action	threshhold	link action	threshhold

View descriptions

By default, an SVF image is displayed to its extents. Alternately, you can specify an intial view to be used. The view is described using the opposite corners of the view window.

InitialView	**(header)**				
type	extended token	number	number	number	number
value	15,2	xmin	ymin	xmax	ymax

added with SVF2

View information can be stored in the header which can later be used to easily change the view of the image. A view is given a name which can be used to identify it later. The opposite corners of the desired view are stored in the view table.

ViewTable	**(header)**			
type	token	count	{ string	Extents }
value	14	count	name	size

added with SVF2

Miscellaneous

An image can contain a suggestion for how big it would like to appear. You can specify the suggested width of the image in millimeters. The display program is not required to honor the request.

Width	**(header)**

FH_DEF000236

Motorola PX 1026_195

type	token	number
value	7	millimeters

The SVF image can be given a name (in the header section).

Name	**(header)**	
type	token	string
value	1	name

A flag can be specified which indicates that a display program is free to rearrange entities within a layer. This is useful for certain display optimizations. Note that it is still valid to display entities in the order they occur in the file. 0 is the default. 1 indicates you can rearrange within a layer.

By default, the origin in considered to be the lower left corner and coordinates increase as you move up or to the right. This is how Cartesion coordinates work. Alternately you can specify a flag which indicates that the origin will be in the upper left corner, with coordinates increasing as you move down or to the right.

Flags	**(header)**	
type	token	number
value	11	flag

value	**C constant**	**Java constant**	**meaning**
1	SVFF_RearrangeWithinLayer	eF_RearrangeWithinLayer	can rearrange objects within a layer
2	SVFF_OriginUpperLeft	eF_OriginUpperLeft	origin for coordinates in upper left

If a BaseURL is specified, it is used to resolve all relative URLs in the file. This applies to notifications, hyperlinks, bitmaps, and other objects.

BaseURL	**(header)**	
type	extended token	string
value	15,3	URL

Extensions

Custom user data

Arbitrary data can be added to an SVF file. First a general description of the application is placed in the header section. Later, the arbitrary data can reference this table. An entry in the *data type table* consists of four strings: the company name, a description of the data type, comments and a URL which can point to a web page (this could be used to supply extra information on the data, for example). The data type description is the only required information; the other strings are optional.

DataTypes table entries are then referenced by Data2 when you store arbitrary data in the file. It allows your application to identify data that it knows how to interpret.

The company name *SVF* is reserved for standard SVF extension data types.

DataTypes	**(header)**					
type	extended token	count	{ string	string	string	string }
value	15, 1	count	company	type	comments	URI

added with SVF2

Data2	**(body)**			
type	token	number	count	{ byte }
value	46	data type table index	byte count	user data

added with SVF2

SVF 1.1 had custom user data but did not have a data types table. For simple backwards compatibility, Data can be used to store arbitrary data that doesn't have an associated data type.

Data	**(body)**		
type	token	count	{ byte }
value	34	byte count	user data

Supporting future commands

The Extension command allows future versions of SVF to add commands that can be skipped by an older SVF reader. It simply consists of the Extension token followed by the number of bytes in the following command. If the SVF reader recognizes the token, it can interpret the command, otherwise it knows how many bytes to skip to get to the next valid command. Thus if SVF3 adds a new command, it could wrap it in an Extension to write it back to SVF2 without having to throw away data. An SVF3 reader would then be able to properly read the new command even though it's stored in an SVF2 file. Note that versions of SVF prior to 2.0 do not support the Extension command.

Extension			
type	token	count	{ byte }
value	0	byte count	future command

added with SVF2

Syntax differences between SVF1.1 and SVF2

In general, SVF2 is simply SVF1.1 with additional commands added, making a program which needs to read both versions easy to write. The only situation where you need to know the version to read information properly is text strings. SVF2 stores all text (layer names, graphical text, etc.) in UTF-8 encoded Unicode but SVF1.1 only used ASCII. If all the text is simply lower ASCII, the difference isn't noticeable, but for other characters this is important. See strings for more information.

In SVF2, the 4 byte version of the SetColor and Background commands specify an RGB value. The 1 and 2 byte versions indicate an index into the color table. In SVF1.1, these commands always specified a color index.

See Changes to SVF for a complete description of the changes between SVF1.1 and SVF2.

Stream Status

The stream status flags affect how a reader treats the incoming SVF stream. They don't directly affect display. Once a reader encounters the *End* flag, it can close the file or connection. Note that this is optional, so readers should not rely on it being present. The *Reset* flag indicates that all previous data should be discarded. Entirely new content will be sent.

StreamStatus	**(body)**	
type	token	number
value	49	flag

value	C constant	Java constant	meaning
1	eSVFS_End	eS_End	end of file
2	eSVFS_Reset	eS_Reset	discard previous data and start over

MIME type

The MIME type for SVF files is *image/vnd.svf*. The standard file extension is .svf or .SVF.

A MIME (Multipurpose Internet Mail Extensions) type is a string which describes data types such as JPEG bitmaps (*image/jpeg*), HTML (*text/html*), SVF (*image/vnd.svf*) or XML (*text/xml*).

- More information on MIME types
- Registered MIME types

EXHIBIT C

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

NOTICE OF ALLOWANCE AND FEE(S) DUE

59860 7590 08/08/2008

LAW OFFICE OF R. ALAN BURNETT
4108 131ST AVE. SE
BELLEVUE, WA 98006

EXAMINER	
TRAN, QUOC A	
ART UNIT	PAPER NUMBER
2176	

DATE MAILED: 08/08/2008

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/045,757	01/28/2005	Gary B. Rohrabaugh	7342.P001XD	4819

TITLE OF INVENTION: RESOLUTION INDEPENDENT VECTOR DISPLAY OF INTERNET CONTENT

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	$720	$300	$0	$1020	11/10/2008

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PTOL-85 (Rev. 08/07) Approved for use through 08/31/2010.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

59860 7590 08/08/2008

LAW OFFICE OF R. ALAN BURNETT
4108 131ST AVE. SE
BELLEVUE, WA 98006

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)

(Signature)

(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/045,757	01/28/2005	Gary B. Rohrabaugh	7342.P001XD	4819

TITLE OF INVENTION: RESOLUTION INDEPENDENT VECTOR DISPLAY OF INTERNET CONTENT

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	YES	$720	$300	$0	$1020	11/10/2008

EXAMINER	ART UNIT	CLASS-SUBCLASS
TRAN, QUOC A	2176	715-517000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

❑ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

❑ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,

(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1________________________

2________________________

3________________________

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ❑ Individual ❑ Corporation or other private group entity ❑ Government

4a. The following fee(s) are submitted:

❑ Issue Fee

❑ Publication Fee (No small entity discount permitted)

❑ Advance Order - # of Copies ____________________

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**

❑ A check is enclosed.

❑ Payment by credit card. Form PTO-2038 is attached.

❑ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number ____________ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)

❑ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.

❑ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature ______________________________ Date __________________________

Typed or printed name ______________________________ Registration No. ______________________

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.
11/045,757	01/28/2005	Gary B. Rohrabaugh	7342.P001XD	4819

59860 7590 08/08/2008

LAW OFFICE OF R. ALAN BURNETT
4108 131ST AVE. SE
BELLEVUE, WA 98006

EXAMINER	
TRAN, QUOC A	
ART UNIT	**PAPER NUMBER**
2176	

DATE MAILED: 08/08/2008

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)

(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 376 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 376 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

PTOL-85 (Rev. 08/07) Approved for use through 08/31/2010.

Notice of Allowability	Application No. 11/045,757	Applicant(s) ROHRABAUGH ET AL.	
	Examiner Quoc A. Tran	Art Unit 2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *Applicant's amendments/remarks filed 05/20/2008* .
2. ☒ The allowed claim(s) is/are *71-92, 94-242,244-335,337-389 and 391-393- renumbering as 1-319 (see claims indexing for claims renumbering details)* .
3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
   1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date See Continuation Sheet
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☐ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

/Doug Hutton/
Supervisory Patent Examiner
Technology Center 2100

Continuation of Attachment(s) 3. Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date: 1/22/08;10/28/08;10/05/08;9/20/07;9/20/07;9/18/07;9/17/07;5/6/07 and 1/28/08.

DETAILED ACTION

In response to Applicant's Response after NonFinal Office Action filed 05/20/2008, originally filed 01/28/2005, which claimed priority of 09/828,511 filed 04/07/2001 from provisional 60/211,019 dated ***06/12/2000*** (SoftSource). It is noted amendments to the claims are accepted and entered. Also, the "***Terminal Disclaimer"*** for copending patent application number 09/878,097 was filed on 06/08/2008 and was approved on 06/30/2008.

Claims ***71-92; 94-242; 244-335; 337-389 and 391-393*** are allowed (see reason for allowance for details).

Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

Response to Arguments

Applicant’s arguments filed 05/20/2008 have been fully considered and they are persuasive.

Allowable Subject Matter

Claims 71-92; 94-242; 244-335; 337-389 and 391-393 are allowed.

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. Addtionally, 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.

The dependent claims, further limiting the independent claims, are also allowed.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Quoc A. Tran whose telephone number is 571-272-8664. The examiner can normally be reached on Mon through Fri 8AM - 5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Doug Hutton can be reached on (571)272-4137. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Quoc A, Tran/
Patent Examiner

/Doug Hutton/
Doug Hutton
Supervisory Primary Examiner
Technology Center 2100

EXHIBIT D

Notice of Allowability	Application No. 11/738,486	Applicant(s) ROHRABAUGH ET AL.	
	Examiner QUOC A. TRAN	Art Unit 2176	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *Amendments/Remarks filed 04/21/2010 and telephone interviewed on 05/18/2010*.
2. ☒ The allowed claim(s) is/are *1-6, 8--13, 15-16, 19-20, 22, 21, 24, 26, 89, 7, 93-94, 25,90- 92, 95, 27-37, 39-40, 43-44, 46-47, 45, 48-50, 96, 51-63, 65-68, 72-73, 69, 70-71, and 74-88 (renumbering as 1-88 respectively)* .
3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
   1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date 04/21/2010
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☒ Interview Summary (PTO-413), Paper No./Mail Date 06/02/2010 .
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

/DOUG HUTTON/
Supervisory Patent Examiner, Art Unit 2176

Examiner-Initiated Interview Summary	**Application No.** 11/738,486	**Applicant(s)** ROHRABAUGH ET AL.	
	Examiner QUOC A. TRAN	**Art Unit** 2176	

All Participants:

(1) QUOC A. TRAN (USPTO).

(2) Allan Burnett (Attorney).

(3) ____.

(4) ____.

Status of Application: Pending

Date of Interview: 06/02/2010

Time: 2PM

Type of Interview:

☒ Telephonic
☐ Video Conference
☐ Personal (Copy given to: ☐ Applicant ☐ Applicant's representative)

Exhibit Shown or Demonstrated: ☐ Yes ☒ No
If Yes, provide a brief description: .

Part I.

Rejection(s) discussed:
No

Claims discussed:
1, 27, 51, 79 and 90

Prior art documents discussed:
No

Part II.

SUBSTANCE OF INTERVIEW DESCRIBING THE GENERAL NATURE OF WHAT WAS DISCUSSED:
See Continuation Sheet

Part III.

☒ It is not necessary for applicant to provide a separate record of the substance of the interview, since the interview directly resulted in the allowance of the application. The examiner will provide a written summary of the substance of the interview in the Notice of Allowability.

☐ It is not necessary for applicant to provide a separate record of the substance of the interview, since the interview did not result in resolution of all issues. A brief summary by the examiner appears in Part II above.

/Quoc A. Tran/
Examiner, Art Unit 2176

(Applicant/Applicant's Representative Signature – if appropriate)

Continuation of Substance of Interview including description of the general nature of what was discussed: The Examiner is invited to contract the undersigned for any reason related to the advancement of this case. Applicant's Attorney agrees to the following: amending claims 1, 27, 51, 79 and 90. The claims have been amended in accordance with the substance of the telephone interview. Favorable consideration of the pending claims and passing them allowance is agreed upon. .

DETAILED ACTION

In response to Amendments/Remarks filed 04/21/2010 and telephone interviewed on 06/02/2010, the examiner's amendment was authorized by attorney of record Allan Burnett, Attorney for Applicants. The current patent application originally filed 04/21/2007.

It is noted the current application is a continuation of 09/878,097 now is US Patent 7,210,099 filed 06/08/2001 issued 04/24/2007, which is a continuation in part of ***09/828,511***, which Claims Priority from Provisional Application 60/211,019 filed ***06/12/2000***, which claims Priority from Provisional Application 60/217,345 filed 07/11/2000.

- Claims 1-13, 15-16, 19-22, 24-37, 39-40, 43-63, and 65-96 are pending.
- ***Claims 1, 27, 51, 79 and 90 are currently amended***.
- Claims 14, 17, 18, 23, 38, 41, 42 and 64 were previously cancelled.
- Claims 1-13, 15-16, 19-22, 24-37, 39-40, 43-63, 65-89 and 91-96 were Previously Presented and/or Original.
- Claims 1; 27; 51 and 79 are independent claim.

In addition, it is noted the Obviousness-Type Double Patent to claims 1-88, and the 112 rejections to claims 1; 11; 13; 15-16; 25; 27; 29; 35; 37; 39; 40; 49; 56; 61; 63; 65; 66; 75; 84; 86 and 87, set forth in the previous office action dated 04/01/2010 is hereby withdrawn, in light of Terminal Disclaimer filed 04/21/2010 and Approved 05/01/2010 and further view of the amendment filed 04/21/2010 (@ Pages 3-26).

Also, the Examiner acknowledges apllicant's amendments to the specification filed 04/21/2010 (see the Amendments to the Specification paper filed 04/21/2010 @ page 2).

Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

EXAMINER'S AMENDMENT

The application has been amended as follows:

In the Claims:

- Please replace the claims with the following claim set:

1. (***Currently Amended***) 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 ~~attributes~~ 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 ~~a display~~ 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. (Original) The mobile device of claim 1, wherein the device comprises a mobile phone.

3. (Previously Presented) The mobile device of claim 1, wherein the device comprises one of a hand-held device or a palm-held device.

4. (Original) 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. (Previously Presented) The mobile device of claim 1, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

6. (Original) 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. (Previously Presented) The mobile device of claim 89, 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.

8. (Previously Presented) 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.

9. (Original) 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.

10. (Original) 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.

11. (Previously Presented) The mobile device of claim 10, wherein execution of the instructions performs further operations comprising enabling the display of the Web content to be panned in real-time.

12. (Original) 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.

13. (Previously Presented) 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.

14. (Canceled)

15. (Previously Presented) 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.

16. (Previously Presented) 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.

17. (Canceled)

18. (Canceled)

19. (Previously Presented) 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.

20. (Previously Presented) 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.

21. (Previously Presented) 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.

22. (Previously Presented) The mobile device of claim 20, wherein execution of the instructions performs further operations comprising:

mapping the object vectors to a virtual display area in memory.

23. (Cancelled)

24. (Original) 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.

25. (Previously Presented) The mobile device of claim 89, 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. (Original) 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.

27. (***Currently Amended)*** 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 ~~attributes~~ design of content on the Web page;

retrieving ~~Web~~ HTML-based content associated with the Web page;

translating at least a portion of the ~~Web~~ HTML-based content from its original format to produce translated content ~~into~~ including scalable vector-based content that supports a scalable resolution-independent ~~display~~ representation of the HTML-based content that ~~retains~~ 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.

28. (Original) The mobile phone of claim 27, 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.

29. (Previously Presented) The mobile phone of claim 28, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

30. (Original) The mobile phone of claim 27, 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.

31. (Previously Presented) The mobile phone of claim 27, 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.

32. (Original) The mobile phone of claim 27, 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.

33. (Original) The mobile phone of claim 27, 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.

34. (Original) The mobile phone of claim 27, 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.

35. (Previously Presented) The mobile phone of claim 34, wherein execution of the instructions performs further operations comprising enabling the display of the Web content to be panned in real-time.

36. (Original) The mobile phone of claim 27, 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.

37. (Previously Presented) The mobile phone of claim 27, 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.

38. (Canceled)

39. (Previously Presented) The mobile phone of claim 27, 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.

40. (Previously Presented) The mobile phone of claim 27, 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.

41. (Canceled)

42. (Canceled)

43. (Previously Presented) The mobile phone of claim 27, 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. (Previously Presented) The mobile phone of claim 27, 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. (Previously Presented) The mobile phone of claim 27, 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.

46. (Previously Presented) 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.

47. (Previously Presented) The mobile phone of claim 46, 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.

48. (Original) The mobile phone of claim 27, 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. (Previously Presented) The mobile phone of claim 27, 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. (Original) The mobile phone of claim 27, wherein at least a portion of the instructions comprise Java-based instructions configured to be executed on a Java virtual machine.

51. (***Currently Amended***) 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 ~~a display~~ 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.

52. (Original) The mobile device of claim 51, wherein the device comprises a mobile phone.

53. (Original) The mobile device of claim 51, wherein the device comprises one of a Personal Digital Assistant (PDA) or pocket PC.

54. (Original) The mobile device of claim 51, 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.

55. (Original) The mobile device of claim 54, 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.

56. (Previously Presented) The mobile device of claim 51, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

57. (Original) The mobile device of claim 51, 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.

58. (Original) The mobile device of claim 51, wherein at least a portion of the scalable content comprises scalable vector-based content.

59. (Original) The mobile device of claim 51, 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.

60. (Original) The mobile device of claim 51, 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.

61. (Previously Presented) The mobile device of claim 60, wherein execution of the instructions performs further operations comprising enabling the display of the Web content to be panned in real-time.

62. (Original) The mobile device of claim 51, 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.

63. (Previously Presented) The mobile device of claim 51, 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.

64. (Canceled)

65. (Previously Presented) The mobile device of claim 51, 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. (Previously Presented) The mobile device of claim 51, 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. (Original) 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. (Original) The mobile device of claim 51, 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. (Previously Presented) The mobile device of claim 51, 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.

70. (Previously Presented) The mobile device of claim 51, 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.

71. (Original) The mobile device of claim 70, 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.

72. (Previously Presented) 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.

73. (Previously Presented) The mobile device of claim 72, 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.

74. (Original) The mobile device of claim 51, 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. (Previously Presented) The mobile device of claim 51, 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. (Original) The mobile device of claim 51, wherein at least a portion of the instructions comprise Java-based instructions configured to be executed on a Java virtual machine.

77. (Original) The mobile device of claim 51, wherein a portion of the HTML-based Web content comprises XML-based content.

78. (Previously Presented) The mobile device of claim 51, wherein a portion of the HTML-based Web content comprises cascading style sheet data.

79. (Previously Presented) 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 ~~a display~~ 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. (Original) The mobile device of claim 79, wherein the processing means includes a general-purpose processor.

81. (Original) The mobile device of claim 79, wherein at least a portion of the programmed circuit means is embodied as a special-purpose processor.

82. (Original) 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. (Original) 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. (Previously Presented) The mobile device of claim 79, wherein the display of the Web page is re-rendered in real-time to effect zooming operations.

85. (Original) 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. (Previously Presented) 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. (Previously Presented) 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. (Previously Presented) 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.

89. (Previously Presented) 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[[;]] .

90. (***Currently Amended***) 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,

$$X' = X * SF \ [[+ \Delta X]] \ ;$$

$$Y' = Y * SF \ [[+ \Delta Y]] \ ;$$

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~~, ΔX is a translation along the X axis, and ΔY is a translation along the Y axis~~.

91. (Previously Presented) The mobile device of claim 90, 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.

92. (Previously Presented) 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.

93. (Previously Presented) The mobile device of claim 7, 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.

94. (Previously Presented) The mobile device of claim 93, 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.

95. (Previously Presented) 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.

96. (Previously Presented) The mobile device of claim 27, 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.

Information Disclosure Statement

The signed and dated copies of applicant's IDS, which were filed on 04/21/2010, is attached to this Office Action.

Allowable Subject Matter

Claims 1-13, 15-16, 19-22, 24-37, 39-40, 43-63, and 65-96 are allowed.

The following is a statement of reasons for the indication of allowable subject matter: Under the broadest reasonable interpretation of the claimed limitation consistence with the Applicant's Specification, the claimed invention advantageously provides a method and device for creating resolution-independent vector display of Intemet content to allow it to be scaled (zoomed) larger and smaller for better viewing or to fit any resolution or screen size, which are designed for a client device (e.g., desktop computers, a "small-screen" device, such as a cell phone, mobile device and/or a PDA).

In particular, the claimed invention takes HTML-based Web content in its original page layout 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 that for enabling Web pages in their ***original HTML-based content form to be accessed via mobile devices***, viewed at various zoom levels by zooming in and out views of the Web pages and interacted with via the mobile devices in a manner the ***preserves the original page layout, functionality*** (including preservation of hyperlink functionality), ***and design*** of the Web page content (as defined by its HTML-based content) (See also the currently amended independent claims 1, 27, 51 and 79 as recited above.)

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Quoc A. Tran whose telephone number is 571-272-8664. The examiner can normally be reached on Mon through Fri 8AM - 5PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Doug Hutton can be reached on (571)272-4137. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Quoc A. Tran/
Examiner, Art Unit 2176

/DOUG HUTTON/
Supervisory Patent Examiner, Art Unit 2176

EXHIBIT E

Attorney Docket No. 005207.P001XD *Patent*

Certificate of Electronic Filing

I hereby certify that this correspondence is being Electronically Filed via EFS

on May 20, 2008

Date of Deposit

R. Alan Burnett

Name of Person Filing Correspondence

/S/ R. Alan Burnett May 20, 2008

Signature Date

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)	
	)	
Rohrabaugh et al.	)	Examiner: Tran, Quoc A.
	)	
Serial No. 11/045,757	)	Art Unit: 2176
	)	
Filed: June 8, 2001	)	
	)	
For: SCALABLE DISPLAY OF INTERNET CONTENT ON MOBILE DEVICES	)	

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL AMENDMENT

Sir:

Applicants request the Examiner to enter the following amendment and to reconsider all pending claims in view of the amendment and the following remarks.

Amendments begin on page 2. Remarks begin on page 65.

AMENDMENT

In the Specification

Please amend paragraph [0008] as follow:

[0008] According to additional aspects of the invention, methods and software for enabling support for resolution-independent scalable display of Web content [[is]] 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 ~~access millions~~ browse Web pages from among literally billions of Web pages ~~on various devices having different screen resolutions~~ while providing a full Web browsing experience.

In the Claims

This listing of claims replaces all prior versions and listing of claims in the application. Amendments or cancellations of any claims are done without prejudice, waiver and/or disclaimer. Applicants reserve the right to claim the subject matter of any amendment and/or cancellation in a continuing application.

1-70. (Cancelled)

71. (Currently Amended) 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 ~~and attributes~~, 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 ~~display~~ representation of the Web page that ~~substantially retains~~ preserves the original page layout, functionality and ~~attributes~~

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 ~~the original~~ a width of the Web page is rendered to fit ~~substantially~~ across the display~~,~~

~~wherein the rendered Web page comprises a scaled representation of the original Web page that substantially preserves the original page layout and attributes of the Web page content~~.

72. (Previously Presented) The wireless device of claim 71, 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.

73. (Currently Amended) The wireless device of claim 72, wherein the display of the Web page is re-rendered ~~substantially~~ in real-time to effect zooming operations.

74. (Previously Presented) The wireless device of claim 71, 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.

75. (Previously Presented) The wireless device of claim 71, 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.

76. (Currently Amended) The wireless device of claim 71, 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 ~~and attributes~~, functionality, and design of the Web page content are ~~substantially~~ preserved at each of the different resolutions.

77. (Previously Presented) The wireless device of claim 71, 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.

78. (Currently Amended) The wireless device of claim 71, wherein execution of the instructions performs further operations comprising enabling a user to pan a ~~display~~ view of the Web page in response to a corresponding user input.

79. (Currently Amended) The wireless device of claim 78, wherein execution of the instructions performs further operations comprising enabling the ~~display~~ view of the Web page to be panned ~~substantially~~ in real-time.

80. (Currently Amended) The wireless device of claim 71, 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.

81. (Previously Presented) The wireless device of claim 71, 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.

82. (Previously Presented) The wireless device of claim 81, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

83. (Previously Presented) The wireless device of claim 71, 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.

84. (Previously Presented) The wireless device of claim 71, 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.

85. (Previously Presented) The wireless device of claim 84, wherein the content of the paragraph is reformatted to fit characteristics of the display when the display is

re-rendered.

86. (Previously Presented) The wireless device of claim 71, 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.

87. (Previously Presented) The wireless device of claim 71, 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.

88. (Currently Amended) The wireless device of claim 71, wherein execution of the instructions performs further operations comprising:

parsing markup language code corresponding to the ~~received~~ retrieved Web ~~content~~ 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.

89. (Previously Presented) The wireless device of claim 88, 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.

90. (Previously Presented) The wireless device of claim 89, wherein execution of the instructions performs further operations comprising:

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

91. (Currently Amended) The wireless device of claim 90, wherein execution of the instructions performs further operations comprising:

enabling a user to view the Web page at a user-selectable zoom level and ~~pan~~ 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 ~~pan~~ 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.

92. (Previously Presented) The wireless device of claim 71, 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.

93. (Cancelled)

94. (Previously Presented) The wireless device of claim 71, wherein at least a portion of the instructions comprise Java-based instructions.

95. (Previously Presented) The wireless device of claim 71, wherein the device comprises a mobile phone.

96. (Previously Presented) The wireless device of claim 71, wherein the device comprises one of a Personal Digital Assistant (PDA) or hand-held computer.

97. (Previously Presented) The wireless device of claim 71, wherein the network comprises a mobile service provider network.

98. (Previously Presented) The wireless device of claim 71, wherein a portion of the scalable content comprises vector-based content.

99. (Currently Amended) A mobile <u>hand-held</u> 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 <u>hand-held</u> 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 ~~and attributes~~, 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-rendering~~ re-render the display ~~Web page~~ in response to associated user inputs ~~to enable the user to zoom in and out a display of the Web page~~ to enable the Web page to be ~~viewed~~ browsed at various zoom levels and panned views while ~~substantially~~ preserving the original page layout ~~and attributes~~, functionality, and design of the Web page content at each zoom level and panned view.

100. (Currently Amended) The mobile hand-held device of claim 99, wherein the device comprises a mobile phone.

101. (Currently Amended) The mobile hand-held device of claim 99, wherein the device comprises one of a Personal Digital Assistant (PDA) or hand-held computer.

102. (Currently Amended) The mobile hand-held device of claim 99, 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.

103. (Currently Amended) The mobile hand-held device of claim 102, wherein the user interface input enables the user to define ~~a window~~ an area of a current view of the Web page on which to zoom in on.

104. (Currently Amended) The mobile hand-held device of claim 99, wherein the display of the Web page is re-rendered ~~substantially~~ in real-time to effect zooming operations.

105. (Currently Amended) The mobile hand-held device of claim 99, 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.

106. (Currently Amended) The mobile hand-held device of claim 99, wherein at least a portion of the scalable content comprises scalable vector-based content.

107. (Currently Amended) The mobile hand-held device of claim 99, 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.

108. (Currently Amended) The mobile hand-held device of claim 99, wherein execution of the instructions performs further operations comprising enabling a user to pan a ~~display~~ view of the Web page in response to a corresponding user input made via the display.

109. (Currently Amended) The mobile hand-held device of claim [[108]] 99, wherein execution of the instructions performs further operations comprising enabling

the ~~display~~ panned view of the Web page to be panned ~~substantially~~ in real-time.

110. (Currently Amended) The mobile hand-held device of claim 99, 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.

111. (Currently Amended) The mobile hand-held device of claim 99, 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 ~~displayed substantially~~ rendered to fit across the display.

112. (Currently Amended) The mobile hand-held device of claim 111, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

113. (Currently Amended) The mobile hand-held device of claim 99, 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 ~~displayed substantially~~ rendered to fit across the display.

114. (Currently Amended) The mobile hand-held device of claim 99, 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 ~~displayed substantially~~ rendered to fit across a display area of the display.

115. (Currently Amended) The mobile hand-held device of claim 114, wherein the content of the paragraph is reformatted to fit characteristics of the display area when the display is re-rendered.

116. (Currently Amended) The mobile hand-held device of claim 99, 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.

117. (Currently Amended) The mobile hand-held device of claim 99, 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.

118. (Currently Amended) The mobile hand-held device of claim 99, wherein execution of the instructions performs further operations comprising:

parsing HTML-based code corresponding to the ~~received~~ retrieved Web ~~content~~ 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.

119. (Currently Amended) The mobile hand-held device of claim 118, 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.

120. (Currently Amended) The mobile hand-held device of claim 119, 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.

121. (Currently Amended) The mobile hand-held device of claim 120, wherein execution of the instructions performs further operations comprising:

enabling a user to view the Web page at a user-selectable zoom level and ~~pan~~ 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 ~~pan~~ 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.

122. (Currently Amended) The mobile <u>hand-held</u> device of claim 99, 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.

123. (Currently Amended) The mobile <u>hand-held</u> device of claim 99, 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 ~~substantially~~ <u>to fit</u> across a <u>browser</u> display area of the display; and

employing the scale factor to render the <u>browser</u> display area.

124. (Currently Amended) The mobile <u>hand-held</u> device of claim 99, wherein at least a portion of the instructions comprise Java-based instructions.

125. (Currently Amended) The mobile <u>hand-held</u> device of claim 99, wherein a portion of the HTML-based Web content comprises XML code.

126. (Currently Amended) The mobile <u>hand-held</u> device of claim 99, wherein a portion of the HTML-based Web content comprises cascaded style sheet data <u>defining aspects of the Web page design that are preserved at each zoom level and panned view</u>.

127. (Currently Amended) The mobile hand-held device of claim 99, wherein ~~a~~

~~portion of the scalable content comprises vector-based content~~ the network comprises a Local Area Network or Wide Area Network.

128. (Currently Amended) 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 ~~and attributes~~, 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 ~~Web page~~ in response to associated user inputs made via the display to enable ~~the user to zoom in and out a display of~~ the Web page to be browsed at various zoom levels and panned views while preserving~~,~~

~~wherein~~ the original page layout ~~and attributes~~, functionality, and design of the Web page content ~~are substantially preserved regardless of a~~ at each zoom level and panned view of the Web page.

129. (Previously Presented) The mobile device of claim 128, wherein the processing means includes a general-purpose processor.

130. (Previously Presented) The mobile device of claim 128, wherein the processing means includes a special-purpose processor.

131. (Previously Presented) The mobile device of claim 128, 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.

132. (Currently Amended) The mobile device of claim 131, wherein the user interface input enables the user to define ~~a window~~ an area of a current view of the Web page on which to zoom in on.

133. (Currently Amended) The mobile device of claim 128, wherein the display of the Web page is re-rendered ~~substantially~~ in real-time to effect zooming operations.

134. (Currently Amended) The mobile device of claim 128, wherein execution of the instructions performs further operations comprising enabling a user to pan a ~~display~~ view of the Web content in response to a corresponding user interface input made via the display.

135. (Currently Amended) The mobile device of claim [[134]] 128, wherein execution of the instructions performs further operations comprising enabling the ~~display~~ view of the Web content to be panned ~~substantially~~ in real-time.

136. (Currently Amended) The mobile device of claim 128, 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 ~~displayed substantially~~ rendered to fit across the display.

137. (Previously Presented) The mobile device of claim 128, 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.

138. (Previously Presented) The mobile device of claim 128, wherein the network comprises a mobile service provider network.

139. (Currently Amended) The ~~wireless~~ mobile device of claim 128, wherein the device comprises a mobile phone.

140. (Currently Amended) The ~~wireless~~ mobile device of claim 128, wherein the device comprises one of a Personal Digital Assistant (PDA) or hand-held computer.

141. (Currently Amended) The ~~wireless~~ mobile device of claim 128, wherein a portion of the scalable content comprises vector-based content.

142. (Previously Presented) The mobile device of claim 128, wherein the processing means includes logic circuitry programmed with a portion of the instructions.

143. (Currently Amended) 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 <u>hand-held</u> 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<u>, functionality, and design</u> of content on the Web page ~~and defining an original width and height of the Web page~~;

<u>in response to a user request of the Web page,</u>

retrieving the Web page via the wireless communications interface;

rendering the Web page such that <u>a width of</u> the Web page is rendered to fit ~~substantially~~ across the display; and

re-rendering the <u>display</u> ~~Web page~~ in response to associated user inputs to enable ~~the user to zoom in and out a display of the Web page to enable~~ the Web page to be ~~viewed~~ <u>browsed</u> at various zoom levels <u>and panned views</u> while ~~substantially~~ preserving the original page layout ~~and attributes~~<u>, functionality, and design</u> of the Web page content at each zoom level <u>and panned view</u>.

144. (Currently Amended) The mobile <u>hand-held</u> device of claim 143, 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.

145. (Currently Amended) The mobile <u>hand-held</u> device of claim 144, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

146. (Currently Amended) The mobile <u>hand-held</u> device of claim 143, 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.

147. (Currently Amended) The mobile hand-held device of claim 143, 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.

148. (Currently Amended) The mobile hand-held device of claim 147, wherein the content of the paragraph is reformatted to fit characteristics of the display when re-rendered.

149. (Currently Amended) The mobile hand-held device of claim 143, wherein execution of the instructions performs further operations comprising enabling a user to pan a ~~display~~ view of the web page while in a zoomed state under which a portion of the web page is displayed.

150. (Currently Amended) The mobile hand-held device of claim 143, 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.

151. (Currently Amended) The mobile hand-held device of claim 143, wherein the display of the Web page is re-rendered ~~substantially~~ in real-time to effect zooming operations.

152. (Currently Amended) The mobile hand-held device of claim 144, wherein the corresponding user input comprises tapping on the column via the display.

153. (Currently Amended) The mobile hand-held device of claim 146, wherein the corresponding user input comprises tapping on the image via the display.

154. (Currently Amended) The mobile hand-held device of claim 147, wherein the corresponding user input comprises tapping on the paragraph via the display.

155. (Currently Amended) The mobile hand-held device of claim 143, wherein execution of the instructions performs further operations comprising enabling a user to pan a ~~display~~ view of the Web page in response to a corresponding user input made via the display.

156. (Currently Amended) The mobile hand-held device of claim [[155]] 143, wherein execution of the instructions performs further operations comprising enabling the ~~display~~ view of the Web page to be panned ~~substantially~~ in real-time.

157. (Currently Amended) The mobile hand-held device of claim 143, 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.

158. (Currently Amended) The mobile hand-held device of claim 143, wherein a portion of the HTML-based Web content comprises XML code.

159. (Currently Amended) The mobile hand-held device of claim 143, 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.

160. (Currently Amended) The mobile hand-held device of claim 143, wherein the network comprises a mobile service provider network.

161. (Currently Amended) The mobile hand-held device of claim 143, wherein the device comprises a mobile phone.

162. (Currently Amended) The mobile hand-held device of claim 143, wherein the device comprises one of a Personal Digital Assistant (PDA) or hand-held computer.

163. (Currently Amended) The mobile device of claim 143, wherein the ~~device comprises one of a notebook computer or laptop computer~~ network comprises a Local Area Network or a Wide Area Network.

164. (Previously Presented) The wireless device of claim 71, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

165. (Currently Amended) The mobile hand-held device of claim 99, wherein the device comprises one of a notebook computer or laptop computer.

166. (Previously Presented) The mobile device of claim 128, wherein the device comprises one of a notebook computer or laptop computer.

167. (Previously Presented) The wireless device of claim 81, wherein the corresponding user input comprises tapping on the column via the display.

168. (Previously Presented) The wireless device of claim 83, wherein the corresponding user input comprises tapping on the image via the display.

169. (Previously Presented) The wireless device of claim 84, wherein the corresponding user input comprises tapping on the paragraph via the display.

170. (Currently Amended) The mobile hand-held device of claim 111, wherein

the corresponding user input comprises tapping on the column via the display.

171. (Currently Amended) The mobile hand-held device of claim 113, wherein the corresponding user input comprises tapping on the image via the display.

172. (Currently Amended) The mobile hand-held device of claim 114, wherein the corresponding user input comprises tapping on the paragraph via the display.

173. (Previously Presented) The mobile device of claim 136, wherein the corresponding user input comprises tapping on the image via the display.

174. (Currently Amended) 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 ~~and attributes~~, 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 ~~display~~ representation of the Web page that ~~substantially retains~~ preserves the original page layout, functionality and ~~attributes~~ 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 ~~and attributes~~, functionality, and design of the Web page content are ~~substantially~~ preserved under both the first and second scale factors.

175. (Currently Amended) The wireless device of claim 174, wherein the display is re-rendered ~~substantially~~ in real-time.

176. (Currently Amended) The wireless device of claim 174, wherein the device comprises ~~one of~~ a ~~Personal Digital Assistant (PDA) or~~ hand-held ~~computer~~ device.

177. (Currently Amended) The wireless device of claim 174, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

178. (Currently Amended) The wireless device of claim 174, wherein execution of the instructions performs further operations comprising enabling a user to pan a ~~display~~ view of the Web page in response to a corresponding user input.

179. (Currently Amended) The wireless device of claim 178, wherein execution of the instructions performs further operations comprising enabling the ~~display~~ view of the Web page to be panned ~~substantially~~ in real-time.

180. (Currently Amended) 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 ~~and attributes~~, 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 ~~display~~ representation of the Web page that ~~substantially retains~~ preserves the original page layout, functionality and ~~attributes~~ 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 ~~the original~~ a width of the Web page is rendered to fit ~~substantially~~ across the display~~,~~

~~wherein the rendered Web page comprises a scaled representation of the original Web page that substantially preserves the original page layout and attributes of the Web page content~~.

181. (Previously Presented) The method of claim 180, 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.

182. (Currently Amended) The method of claim 181, wherein the display of the Web page is re-rendered ~~substantially~~ in real-time to effect zooming operations.

183. (Previously Presented) The method of claim 180, 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.

184. (Currently Amended) The method of claim 180, ~~performs~~ 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.

185. (Currently Amended) The method of claim 180, 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 ~~and attributes~~, functionality, and design of the Web page content are ~~substantially~~ preserved at each of the different resolutions.

186. (Previously Presented) The method of claim 180, further comprising returning the display of the Web page to a previous view in response to a corresponding user input.

187. (Currently Amended) The method of claim 180, further comprising enabling a user to pan a ~~display~~ view of the Web page in response to a corresponding user

input.

188. (Currently Amended) The method of claim 187, further comprising enabling the ~~display~~ view of the Web page to be panned ~~substantially~~ in real-time.

189. (Currently Amended) The method of claim 180, 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.

190. (Currently Amended) The method of claim 180, 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 ~~displayed substantially~~ rendered to fit across the display.

191. (Previously Presented) The method of claim 190, wherein the corresponding user input comprises tapping on the column via the display.

192. (Previously Presented) The method of claim 190, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

193. (Currently Amended) The method of claim 180, 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 ~~displayed substantially~~ rendered to fit across the display.

194. (Previously Presented) The method of claim 193, wherein the corresponding user input comprises tapping on the image via the display.

195. (Currently Amended) The method of claim 180, 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 ~~displayed substantially~~ rendered to fit across the display.

196. (Currently Amended) The method of claim [[132]] 195, wherein the corresponding user input comprises tapping on the paragraph via the display.

197. (Currently Amended) The method of claim [[132]] 195, wherein the content of the paragraph is reformatted to fit characteristics of the display when the display is re-rendered.

198. (Previously Presented) The method of claim 180, 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.

199. (Previously Presented) The method of claim 180, 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.

200. (Currently Amended) The method of claim 180, further comprising:

parsing markup language code corresponding to the ~~received~~ retrieved Web ~~content~~ 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.

201. (Previously Presented) The method of claim 200, 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.

202. (Previously Presented) The method of claim 201, further comprising:

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

203 (Currently Amended) The method of claim 202, further comprising:

enabling a user to view the Web page at a user-selectable zoom level and ~~pan~~ <u>panned view</u> by,

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and ~~pan~~ <u>panned view</u> 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.

204. (Previously Presented) The method of claim 180, wherein the scalable content includes scalable text content, the method further comprising scaling a scalable font to render the scalable text content.

205. (Previously Presented) The method of claim 180, wherein the method is facilitated, at least in part, via execution of Java-based instructions.

206. (Previously Presented) The method of claim 180, wherein the device comprises a mobile phone.

207. (Currently Amended) The method of claim 180, wherein the device comprises ~~one of~~ a ~~Personal Digital Assistant (PDA) or~~ hand-held ~~computer~~ device.

208. (Previously Presented) The method of claim 180, further comprising accessing the Internet via a wireless connection to retrieve the Web page.

209. (Previously Presented) The method of claim 180, wherein a portion of the scalable content comprises vector-based content.

210. (Currently Amended) The method of claim 180, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

211. (Currently Amended) 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 ~~and attributes~~, 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 ~~Web page~~ in response to associated user inputs ~~to enable the user to zoom in and out a display of the Web page~~ to enable the Web page to be ~~viewed~~ browsed at various zoom levels and panned views while ~~substantially~~ preserving the original page layout ~~and attributes~~, functionality, and design of the Web page content at each zoom level and panned view.

212. (Currently Amended) The method of claim 211, wherein the hand-held device comprises a mobile phone.

213. (Currently Amended) The method of claim 211, wherein the hand-held device comprises one of a Personal Digital Assistant (PDA) or hand-held computer.

214. (Previously Presented) The method of claim 211, 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.

215. (Currently Amended) The method of claim 214, wherein the user interface input enables the user to define ~~a window~~ an area of a current view of the Web page on which to zoom in on.

216. (Currently Amended) The method of claim 211, wherein the display of the Web page is re-rendered ~~substantially~~ in real-time to effect zooming operations.

217. (Previously Presented) The method of claim 211, 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.

218. (Previously Presented) The method of claim 211, wherein at least a portion of the scalable content comprises scalable vector-based content.

219. (Previously Presented) The method of claim 211, further comprising returning the display of the Web page to a previous view in response to a corresponding user input made via the display.

220. (Currently Amended) The method of claim 211, further comprising enabling a user to pan a ~~display~~ view of the Web page in response to a corresponding user input made via the display.

221. (Currently Amended) The method of claim [[220]] 211, further comprising enabling the ~~display~~ view of the Web page to be panned ~~substantially~~ in real-time.

222. (Currently Amended) The method of claim 211, 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.

223. (Previously Presented) The method of claim 211, 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.

224. (Previously Presented) The method of claim 223, wherein the corresponding user input comprises tapping on the column via the display.

225. (Previously Presented) The method of claim 223, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

226. (Previously Presented) The method of claim 211, 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.

227. (Previously Presented) The method of claim 226, wherein the corresponding user input comprises tapping on the image via the display.

228. (Previously Presented) The method of claim 211, 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.

229. (Previously Presented) The method of claim 228, wherein the corresponding user input comprises tapping on the paragraph via the display.

230. (Previously Presented) The method of claim 228, wherein the content of the paragraph is reformatted to fit characteristics of the display area when the display is re-rendered.

231. (Previously Presented) The method of claim 211, 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.

232. (Currently Amended) The method of claim 211, <u>wherein the hand-held device includes dynamic memory having at least a portion employed for rendering purposes, the method further</u> 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.

233. (Currently Amended) The method of claim 211, further comprising:

parsing HTML-based code corresponding to the ~~received~~ <u>retrieved</u> Web ~~content~~ <u>page</u> 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.

234. (Previously Presented) The method of claim 233, 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.

235. (Currently Amended) The method of claim 234, wherein the <u>hand-held</u> 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.

236. (Currently Amended) The method of claim 235, further comprising:

enabling a user to view the Web page at a user-selectable zoom level and ~~pan~~ <u>panned view</u> by,

determining a first scale factor and offset in response to one or more corresponding user inputs defining a user-selectable zoom level and ~~pan~~ <u>panned view</u> 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.

237. (Previously Presented) The method of claim 211, wherein the scalable content includes scalable text content, the method further comprising scaling a scalable font to render the scalable text content.

238. (Currently Amended) The method of claim 211, 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 ~~substantially~~ across a browser display area of the display; and

employing the scale factor to render the browser display area.

239. (Previously Presented) The method of claim 211, wherein the method is facilitated, at least in part, via execution of Java-based instructions.

240. (Previously Presented) The method of claim 211, wherein a portion of the HTML-based Web content comprises XML code.

241. (Currently Amended) The method of claim 211, 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.

242. (Previously Presented) The method of claim 211, wherein a portion of the scalable content comprises vector-based content.

243. (Cancelled)

244. (Currently Amended) 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 ~~and defining an original width and height of 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 ~~to fit substantially across~~ on the display; and

re-rendering the Web page in response to associated user inputs to the hand-held device to enable ~~the user to zoom in and out a display of~~ the Web page to be browsed at various zoom levels and panned views while preserving~~,~~

~~wherein~~ the original page layout, functionality, and design of the Web page content ~~is substantially preserved regardless of a~~ 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.

245. (Previously Presented) The method of claim 244, 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.

246. (Previously Presented) The method of claim 245, wherein the corresponding user input comprises tapping on the column via the display.

247. (Previously Presented) The method of claim 245, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

248. (Previously Presented) The method of claim 244, 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.

249. (Previously Presented) The method of claim 248, wherein the corresponding user input comprises tapping on the image via the display.

250. (Previously Presented) The method of claim 244, 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.

251. (Previously Presented) The method of claim 250, wherein the corresponding user input comprises tapping on the paragraph via the display.

252. (Previously Presented) The method of claim 250, wherein the content of the paragraph is reformatted to fit characteristics of the display when re-rendered.

253. (Currently Amended) The method of claim 244, further comprising enabling a user to pan a ~~display~~ 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.

254. (Previously Presented) The method of claim 244, further comprising returning the display of the Web page to a previous view in response to a corresponding user input.

255. (Currently Amended) The method of claim 244, wherein the display of the Web page is re-rendered ~~substantially~~ in real-time to effect zooming operations.

256. (Currently Amended) The method of claim 244, further comprising enabling a user to pan a ~~display~~ view of the Web page in response to a corresponding user input made via the display.

257. (Currently Amended) The method of claim [[256]] 244, further comprising enabling the ~~display~~ view of the Web page to be panned ~~substantially~~ in real-time.

258. (Previously Presented) The method of claim 244, 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.

259. (Previously Presented) The method of claim 244, wherein a portion of the HTML-based Web content comprises XML code.

260. (Currently Amended) The method of claim 244, 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.

261. (Previously Presented) The method of claim 244, wherein the ~~wireless connection comprises~~ Web page is retrieved via a wireless connection to one of a mobile service provider network, local area network, or wide area network.

262. (Previously Presented) The method of claim 244, wherein the device comprises a mobile phone.

263. (Currently Amended) The method of claim 244, wherein the device comprises ~~one of a Personal Digital Assistant (PDA) or~~ a hand-held ~~computer~~ device.

264. (Currently Amended) The method of claim 244, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

265. (Currently Amended) 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 ~~and attributes~~, 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 ~~display~~ representation of the Web page that ~~substantially retains~~ preserves the original page layout, functionality and ~~attributes~~ 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 ~~and attributes~~, functionality, and design of the Web page content are ~~substantially~~ preserved under both the first and second scale factors.

266. (Currently Amended) The method of claim 265, wherein the display is re-rendered ~~substantially~~ in real-time.

267. (Currently Amended) The method of claim 265, wherein the device comprises ~~one of~~ a ~~Personal Digital Assistant (PDA) or~~ hand-held ~~computer~~ device.

268. (Previously Presented) The method of claim 265, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

269. (Currently Amended) The method of claim 265, further comprising enabling a user to pan a ~~display~~ view of the Web page in response to a corresponding user input.

270. (Currently Amended) The method of claim 269, further comprising enabling the ~~display~~ view of the Web page to be panned ~~substantially~~ in real-time.

271. (Currently Amended) 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 ~~and attributes~~, 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 ~~display~~ representation of the Web page that ~~substantially retains~~ preserves the original page layout, functionality and ~~attributes~~ 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 ~~the original~~ a width of the Web page is rendered to fit ~~substantially~~ across the display~~,~~

~~wherein the rendered Web page comprises a scaled representation of the original Web page that substantially preserves the original page layout and attributes of the Web page content~~.

272. (Previously Presented) The machine-readable medium of claim 271, 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.

273. (Currently Amended) The machine-readable medium of claim 272, wherein the display of the Web page is re-rendered ~~substantially~~ in real-time to effect zooming operations.

274. (Previously Presented) The machine-readable medium of claim 271, 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.

275. (Previously Presented) The machine-readable medium of claim 271, 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.

276. (Currently Amended) The machine-readable medium of claim 271, 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 ~~and attributes~~, functionality, and design of the Web page content are ~~substantially~~ preserved at each of the different resolutions.

277. (Previously Presented) The machine-readable medium of claim 271, 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.

278. (Currently Amended) The machine-readable medium of claim 271, wherein execution of the instructions performs further operations comprising enabling a user to pan a ~~display~~ view of the Web page in response to a corresponding user input.

279. (Currently Amended) The machine-readable medium of claim 278, wherein execution of the instructions performs further operations comprising enabling the ~~display~~ view of the Web page to be panned ~~substantially~~ in real-time.

280. (Currently Amended) The machine-readable medium of claim 271, 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.

281. (Previously Presented) The machine-readable medium of claim 271, 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.

282. (Previously Presented) The machine-readable medium of claim 281, wherein the corresponding user input comprises tapping on the column via the display.

283. (Previously Presented) The machine-readable medium of claim 281, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

284. (Previously Presented) The machine-readable medium of claim 271, 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.

285. (Previously Presented) The machine-readable medium of claim 284, wherein the corresponding user input comprises tapping on the image via the display.

286. (Previously Presented) The machine-readable medium of claim 271, 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.

287. (Previously Presented) The machine-readable medium of claim 286, wherein the corresponding user input comprises tapping on the paragraph via the display.

288. (Previously Presented) The machine-readable medium of claim 286, wherein the content of the paragraph is reformatted to fit characteristics of the display when the display is re-rendered.

289. (Previously Presented) The machine-readable medium of claim 271, 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.

290. (Previously Presented) The machine-readable medium of claim 271, 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.

291. (Currently Amended) The machine-readable medium of claim 271, wherein execution of the instructions performs further operations comprising:

parsing markup language code corresponding to the ~~received~~ retrieved Web ~~content~~ 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.

292. (Previously Presented) The machine-readable medium of claim 291, 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.

293. (Previously Presented) The machine-readable medium of claim 292, wherein execution of the instructions performs further operations comprising:

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

294. (Currently Amended) The machine-readable medium of claim 293, wherein execution of the instructions performs further operations comprising:

enabling a user to view the Web page at a user-selectable zoom level and ~~pan~~ 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 ~~pan~~ 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.

295. (Previously Presented) The machine-readable medium of claim 271, 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.

296. (Previously Presented) The machine-readable medium of claim 271, wherein at least a portion of the instructions comprise Java-based instructions.

297. (Previously Presented) The machine-readable medium of claim 271, wherein

the device comprises a mobile phone.

298. (Currently Amended) The machine-readable medium of claim 271, wherein the device comprises ~~one of~~ a ~~Personal Digital Assistant (PDA) or~~ hand-held ~~computer~~ device.

299. (Previously Presented) The machine-readable medium of claim 271, wherein the Web page is accessed via a mobile service provider network.

300. (Previously Presented) The machine-readable medium of claim 271, wherein a portion of the scalable content comprises vector-based content.

301. (Previously Presented) The machine-readable medium of claim 271, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

302. (Previously Presented) The machine-readable medium of claim 271, wherein the instructions are embodied as a Web browser.

303. (Currently Amended) 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 ~~and attributes~~, 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 <u>display</u> ~~Web page~~ in response to associated user inputs to enable ~~the user to zoom in and out a display of~~ the Web page <u>to be browsed at various zoom levels and panned views while preserving</u>~~,~~

~~wherein~~ the original page layout ~~and attributes~~<u>,</u> <u>functionality, and design</u> of the Web page content ~~are substantially preserved regardless of a~~ <u>at each</u> zoom level <u>and panned view</u> of the Web page[[.]]<u>,</u>

<u>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</u>.

304. (Previously Presented) The machine-readable medium of claim 303, wherein the device comprises a mobile phone.

305. (Currently Amended) The machine-readable medium of claim 303, wherein the device comprises ~~one of~~ a ~~Personal Digital Assistant (PDA) or~~ hand-held ~~computer~~ <u>device</u>.

306. (Previously Presented) The machine-readable medium of claim 303, 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.

307. (Currently Amended) The machine-readable medium of claim 306, wherein the user interface input enables the user to define ~~a window~~ <u>an area</u> of a current view of the Web page on which to zoom in on.

308. (Currently Amended) The machine-readable medium of claim 303, wherein the display of the Web page is re-rendered ~~substantially~~ in real-time to effect zooming

operations.

309. (Currently Amended) The machine-readable medium of claim 303, 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.

310. (Previously Presented) The machine-readable medium of claim 303, wherein at least a portion of the scalable content comprises scalable vector-based content.

311. (Previously Presented) The machine-readable medium of claim 303, 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.

312. (Currently Amended) The machine-readable medium of claim 303, wherein execution of the instructions performs further operations comprising enabling a user to pan a ~~display~~ <u>view</u> of the Web page in response to a corresponding user input <u>made via the display</u>.

313. (Currently Amended) The machine-readable medium of claim [[312]] <u>303</u>, wherein execution of the instructions performs further operations comprising enabling the ~~display~~ <u>view</u> of the Web page to be panned ~~substantially~~ in real-time.

314. (Currently Amended) The machine-readable medium of claim 303, 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 <u>of the Web page</u> having a different aspect ratio.

315. (Currently Amended) The machine-readable medium of claim 303, 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 ~~displayed substantially~~ <u>rendered to fit</u> across the display.

316. (Previously Presented) The machine-readable medium of claim 315, wherein the corresponding user input comprises tapping on the column via the display.

317. (Previously Presented) The machine-readable medium of claim 315, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

318. (Currently Amended) The machine-readable medium of claim 303, 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 ~~displayed substantially~~ <u>rendered to fit</u> across the display.

319. (Previously Presented) The machine-readable medium of claim 318, wherein the corresponding user input comprises tapping on the image via the display.

320. (Currently Amended) The machine-readable medium of claim 303, 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 ~~displayed substantially~~ rendered to fit across a browser display area of the display.

321. (Previously Presented) The machine-readable medium of claim 320, wherein the corresponding user input comprises tapping on the paragraph via the display.

322. (Previously Presented) The machine-readable medium of claim 320, wherein the content of the paragraph is reformatted to fit characteristics of the display area when the display is re-rendered.

323. (Previously Presented) The machine-readable medium of claim 303, 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.

324. (Currently Amended) The machine-readable medium of claim 303, 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.

325. (Currently Amended) The machine-readable medium of claim 303, wherein execution of the instructions performs further operations comprising:

parsing HTML-based code corresponding to the ~~received~~ retrieved Web ~~content~~ 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.

326. (Previously Presented) The machine-readable medium of claim 325, 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.

327. (Currently Amended) The machine-readable medium of claim 326, ~~further comprising~~ 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.

328. (Currently Amended) The machine-readable medium of claim 327, wherein execution of the instructions performs further operations comprising:

enabling a user to view the Web page at a user-selectable zoom level and ~~pan~~ 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 ~~pan~~ 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.

329. (Previously Presented) The machine-readable medium of claim 303, 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.

330. (Currently Amended) The machine-readable medium of claim 303, 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 ~~substantially~~ across a browser display area of the display; and

employing the scale factor to render the browser display area.

331. (Previously Presented) The machine-readable medium of claim 303, wherein at least a portion of the instructions comprise Java-based instructions.

332. (Previously Presented) The machine-readable medium of claim 303, wherein a portion of the HTML-based Web content comprises XML code.

333. (Currently Amended) The machine-readable medium of claim 303, 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.

334. (Previously Presented) The machine-readable medium of claim 303, wherein a portion of the scalable content comprises vector-based content.

335. (Previously Presented) The machine-readable medium of claim 303, wherein the device comprises one of a desktop computer, notebook computer or laptop computer.

336. (Cancelled)

337. (Currently Amended) 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 ~~and defining an original width and height of 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 ~~substantially~~ across the display; and

re-rendering the Web page in response to associated user inputs to enable ~~the user to zoom in and out a display of the Web page to enable~~ the Web page to be ~~viewed~~ browsed at various zoom levels and panned views while ~~substantially~~ preserving the original page layout ~~and attributes~~, functionality, and design of the Web page content at each zoom level <u>and panned view</u>.

338. (Previously Presented) The machine-readable medium of claim 337, 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.

339. (Previously Presented) The machine-readable medium of claim 338, wherein the corresponding user input comprises tapping on the column via the display.

340. (Previously Presented) The machine-readable medium of claim 338, wherein the content of the column is reformatted to fit characteristics of the display when the display is re-rendered.

341. (Previously Presented) The machine-readable medium of claim 337, 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.

342. (Previously Presented) The machine-readable medium of claim 341, wherein the corresponding user input comprises tapping on the image via the display.

343. (Previously Presented) The machine-readable medium of claim 337, 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.

344. (Previously Presented) The machine-readable medium of claim 343, wherein the corresponding user input comprises tapping on the paragraph via the display.

345. (Previously Presented) The machine-readable medium of claim 343, wherein the content of the paragraph is reformatted to fit characteristics of the display when re-rendered.

346. (Previously Presented) The machine-readable medium of claim 337, 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.

347. (Previously Presented) The machine-readable medium of claim 337, 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.

348. (Currently Amended) The machine-readable medium of claim 337, wherein the display of the Web page is re-rendered ~~substantially~~ in real-time to effect zooming operations.

349. (Currently Amended) The machine-readable medium of claim 337, wherein execution of the instructions performs further operations comprising enabling a user to pan a ~~display~~ view of the Web page in response to a corresponding user input made via the display.

350. (Currently Amended) The machine-readable medium of claim [[349]] 337, wherein execution of the instructions performs further operations comprising enabling the ~~display~~ view of the Web page to be panned ~~substantially~~ in real-time.

351. (Previously Presented) The machine-readable medium of claim 337, 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.

352. (Previously Presented) The machine-readable medium of claim 337, wherein a portion of the HTML-based Web content comprises XML code.

353. (Currently Amended) The machine-readable medium of claim 337, 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.

354. (Currently Amended) The machine-readable medium of claim 337, 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.

355. (Currently Amended) The machine-readable medium of claim 337, wherein the wireless device comprises a mobile phone.

356. (Currently Amended) The machine-readable medium of claim 337, wherein the wireless device comprises ~~one of~~ a ~~Personal Digital Assistant (PDA) or~~ hand-held ~~computer~~ device.

357. (Currently Amended) The machine-readable medium of claim 337, wherein the wireless device comprises one of a notebook computer or laptop computer.

358. (Previously Presented) The machine-readable medium of claim 337, wherein the instructions are embodied as a Web browser.

359. (Currently Amended) 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 ~~and attributes~~, 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 ~~display~~ representation of the Web page that ~~substantially retains~~ preserves the original page layout, functionality and ~~attributes~~ design of the content defined by its original format when scaled and rendered; and

employing the scalable content to render the Web page [[in]] 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 display of~~ the Web page on the Web browser,

wherein the original page layout ~~and attributes~~, functionality, and design of the Web page content are ~~substantially~~ 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.

360. (Currently Amended) The machine-readable medium of claim 359, wherein

the display is re-rendered ~~substantially~~ in real-time.

361. (Currently Amended) The machine-readable medium of claim 359, wherein the Web browser is configured to be installed on a <u>hand-held</u> device ~~comprising one of a Personal Digital Assistant (PDA) or hand-held computer~~.

362. (Previously Presented) The machine-readable medium of claim 359, wherein the Web browser is configured to be installed on at least one of a desktop computer, notebook computer or laptop computer.

363. (Currently Amended) The machine-readable medium of claim 359, wherein execution of the instructions performs further operations comprising enabling a user to pan a ~~display~~ <u>view</u> of the Web page in response to a corresponding user input.

364. (Currently Amended) The machine-readable medium of claim 363, wherein execution of the instructions performs further operations comprising enabling the ~~display~~ <u>view</u> of the Web page to be panned ~~substantially~~ in real-time.

365. (Currently Amended) The wireless device of claim 81, wherein the display is re-rendered such that content corresponding to the selected column is ~~displayed substantially~~ <u>rendered to fit</u> across the display.

366. (Currently Amended) The wireless device of claim 83, wherein the display is re-rendered such that the image is ~~displayed substantially~~ <u>rendered to fit</u> across the display.

367. (Currently Amended) The wireless device of claim 84, wherein the display is re-rendered such that content corresponding to the selected paragraph is ~~displayed substantially~~ <u>rendered to fit</u> across the display.

368. (Currently Amended) The mobile <u>hand-held</u> device of claim 144, wherein

the display is re-rendered such that content corresponding to the selected column is ~~displayed substantially~~ rendered to fit across the display.

369. (Currently Amended) The mobile hand-held device of claim 146, wherein the display is re-rendered such that the image is ~~displayed substantially~~ rendered to fit across the display.

370. (Currently Amended) The mobile hand-held device of claim 147, wherein the display is re-rendered such that content corresponding to the selected paragraph is ~~displayed substantially~~ rendered to fit across the display.

371. (Currently Amended) The method of claim 223, wherein the display is re-rendered such that content corresponding to the selected column is ~~displayed substantially~~ rendered to fit across the display.

372. (Currently Amended) The method of claim 226, wherein the display is re-rendered such that the image is ~~displayed substantially~~ rendered to fit across the display.

373. (Currently Amended) The method of claim 228, wherein the display is re-rendered such that content corresponding to the selected paragraph is ~~displayed substantially~~ rendered to fit across the display.

374. (Currently Amended) The method of claim 245, wherein the display is re-rendered such that content corresponding to the selected column is ~~displayed substantially~~ rendered to fit across the display.

375. (Currently Amended) The method of claim 248, wherein the display is re-rendered such that the image is ~~displayed substantially~~ rendered to fit across the display.

376. (Currently Amended) The method of claim 250, wherein the display is re-rendered such that content corresponding to the selected paragraph is ~~displayed substantially~~ rendered to fit across the display.

377. (Currently Amended) The machine-readable medium of claim 281, wherein the display is re-rendered such that content corresponding to the selected column is ~~displayed substantially~~ rendered to fit across the display.

378. (Currently Amended) The machine-readable medium of claim 284, wherein the display is re-rendered such that the image is ~~displayed substantially~~ rendered to fit across the display.

379. (Currently Amended) The machine-readable medium of claim 286, wherein the display is re-rendered such that content corresponding to the selected paragraph is ~~displayed substantially~~ rendered to fit across the display.

380. (Currently Amended) The machine-readable medium of claim 338, wherein the display is re-rendered such that content corresponding to the selected column is ~~displayed substantially~~ rendered to fit across the display.

381. (Currently Amended) The machine-readable medium of claim 341, wherein the display is re-rendered such that the image is ~~displayed substantially~~ rendered to fit across the display.

382. (Currently Amended) The machine-readable medium of claim 343, wherein the display is re-rendered such that content corresponding to the selected paragraph is ~~displayed substantially~~ rendered to fit across the display.

383. (Currently Amended) The wireless device of claim 71, wherein the device enables a user to ~~view~~ browse, zoom, and pan ~~the HTML-based Web page content of~~

~~substantially any Web page~~ billions of Web pages in a manner that ~~substantially~~ preserves the original layout ~~and attributes~~, functionality, and design of the HTML-based Web page content of each Web page.

384. (Currently Amended) The mobile device of claim 99, wherein the device enables a user to ~~view~~ browse, zoom, and pan ~~the HTML-based Web page content of substantially any Web page~~ billions of Web pages in a manner that ~~substantially~~ preserves the original layout ~~and attributes~~, functionality, and design of the HTML-based Web page content of each Web page.

385. (Currently Amended) The mobile hand-held device of claim 143, wherein the device enables a user to ~~view~~ browse, zoom, and pan ~~the HTML-based Web page content of substantially any Web page~~ billions of Web pages in a manner that ~~substantially~~ preserves the original layout ~~and attributes~~, functionality, and design of the HTML-based Web page content of each Web page.

386. (Currently Amended) The method of claim 211, further comprising enabling a user to ~~view~~ browse, zoom, and pan ~~the HTML-based Web page content of substantially any Web page~~ billions of Web pages in a manner that ~~substantially~~ preserves the original layout ~~and attributes~~, functionality, and design of the HTML-based Web page content of each Web page.

387. (Currently Amended) The method of claim 265, further comprising enabling a user to ~~view~~ browse, zoom, and pan ~~the HTML-based Web page content of substantially any Web page~~ billions of Web pages in a manner that ~~substantially~~ preserves the original layout ~~and attributes~~, functionality, and design of the HTML-based Web page content of each Web page.

388. (Currently Amended) The machine-readable medium of claim 271, wherein

execution of the instructions enables a user to ~~view~~ browse, zoom, and pan ~~the HTML-based Web page content of substantially any Web page~~ billions of Web pages in a manner that ~~substantially~~ preserves the original layout ~~and attributes~~, functionality, and design of the HTML-based Web page content of each Web page.

389. (Currently Amended) The machine-readable medium of claim 337, wherein execution of the instructions enables a user to ~~view~~ browse, zoom, and pan ~~the HTML-based Web page content of pan substantially any Web page~~ billions of Web pages in a manner that ~~substantially~~ preserves the original layout ~~and attributes~~, functionality, and design of the HTML-based Web page content of each Web page.

390. (Cancelled)

391. (Currently Amended) 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, ~~and a plurality of hyperlinks~~ 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 ~~display~~ view of the Web page;

activate ~~any viewable~~ a currently displayed hyperlink to an external reference while at ~~any~~ a given zoom level and ~~pan position~~ panned view ~~by tapping on the hyperlink~~, 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.

392. (Currently Amended) The hand-held wireless device of claim 391, wherein the Web page comprises HTML-based Web page content defining an original page layout ~~and attributes~~, functionality, and design of the Web page content, and wherein the browser renders the Web page such that the original page layout ~~and attributes~~, functionality, and design of the Web page are ~~substantially~~ preserved at any selectable zoom level.

393. (Currently Amended) The hand-held wireless device of claim 392, wherein the user is enabled to ~~view~~ browse, zoom, and pan ~~the HTML-based Web page content of substantially any Web page~~ billions of Web pages in a manner that ~~substantially~~ preserves the original layout ~~and attributes~~, functionality, and design of the HTML-based Web page content of each Web page.

REMARKS

This Amendment is in response to the Office Action mailed October 23, 2007. In the Office Action,

In the Amendment, claims 73, 76, 78, 79, 88, 91, 99-127, 128, 133-136, 139-141, 143-163, 165, 170-172, 174-180, 182, 184, 185, 187-190, 193, 195-197, 200, 203, 207, 210, 211-213, 215, 216, 220-222, 232, 233, 235, 236, 238, 141, 144, 153, 255-257, 260, 263-267, 269-271, 273, 276, 278-280, 291, 294, 298, 303, 305, 307-309, 312-315, 318, 320, 324, 325, 327, 328, 330, 333, 337, 348-350, 353-357, 359-361, 363-389, and 391-393 have been amended to clarify the claimed invention. Claims 243, 336, and 390 have been cancelled. Claims 71-92 and 94-242, 244-335, 337-389, and 391-393 are now pending. No new matter has been added, and all claims are supported by the original disclosure of 09/878,097 and other priority applications incorporated therein by reference (Application Serial Nos. 60/217,345, 60/211,019, and 09/828,511). Entry of this amendment is respectfully solicited.

Examiner Interview

An in-person examiner interview was conducted at the USPTO on May 5, 2008. The attendees included Examiner Quoc A. Tran, Primary Examiner Rachna Desai, Inventor Gary Rohrabaugh, and attorney representative R. Alan Burnett.

Demonstration of Device

During the interview, a demonstration of a device and software based on the underlying teachings of the claimed invention was presented. The demonstration device was a Toshiba Pocket PC running a version of the SoftView™ browser, as discussed in the response to Office Action filed December 9, 2007. Inventor Gary Rohrabaugh demonstrated the SoftView™ browser's ability to scale and render Web pages to fit the Toshiba's display, selectively zoom on user-defined windows, images, columns, and paragraphs, and generally zoom and pan Web pages and performing browser functions such as navigation via hyperlinks while preserving the layout, functionality, and design of the Web pages in a manner similar to desktop browser such as Internet Explorer, Firefox, Netscape Navigator, *etc.* Claims corresponding to each of these features are included in the present application.

Discussion of 35 U.S.C. § 103 Rejections

A discussion of the rejections under 35 U.S.C. §103(a) as being unpatentable over *Chithambaram*, in view of *Roy* was conducted. In connection with the discussion was a video demonstrating how the Autodesk MapGuide technology disclosed in *Chithambaram* and *Roy* works (in addition, see further discussion below). The video shows a desktop browser display of various MapGuide sites, and clearly demonstrates that the MapGuide implementation employs an embedded application (plug-in) that operates separately from the browser. The video shows the tracking of packets (using a packet-sniffer utility) received from the MapGuide host site, and demonstrates that the data delivered to the MapGuide plug-in does not comprise HTML-based content, but rather comprises proprietary MapGuide data and related data associated with HTTP Requests and Responses. There was a further discussion of this art as applied to independent claim 71 in particular, where Applicant Rohrabaugh and Representative Burnett made clear that even when the client was considered to be a desktop, the combination of *Chithambaram* and *Roy* fails the prima facie obviousness test for at least the reason that there is no generation of scalable content based on HTML-based content, and that the only content that could be construed as scalable was MapGuide data, which is received by the desktop client in a scalable form to begin with.

Obviousness-type Double Patenting

A pending provisional obviousness-type double patenting rejection was also discussed. Applicants asserted that the present claims are not obvious over the issued claims of the parent 7,210,099 patent claims. Examiner Tran said he would need to reconsider this rejection in view of his new understanding of the claims and arguments presented in response to the current Office Action. Applicants respectfully request the Examiner to consider in detail the arguments made in the December 9, 2007 response, as well as the amendments to the pending claims in reassessing this rejection. Applicants have chosen not to file a terminal disclaimer at this time.

Rejections under 35 U.S.C. § 112, Second Paragraph – use of "Substantially"

In the office action of October 23, 2007, Examiner Tran rejected a number of claims reciting the term "substantially" under 35 U.S.C. § 112, Second Paragraph as rendering the claims indefinite. During the interview, it became clear that Examiner Tran was construing the term "substantially" in an extremely broad manner that was much broader than the intended claim language. By way of example, Examiner Tran took a piece of paper and asked, "is this substantial?" He then folded the paper and asked "is this substantial?" He folded the paper one more time (so it was now a quarter of its original size), and again asked "is this substantial?" Moreover, both Examiners Tran and Desai identified that the use of "substantially" in the context of the recited claim language was not explicitly defined in the specification. When representative Burnett pointed out that there is a significant portion of US patents that include the word "substantially" in at least one claim, many of which do not use the word "substantially" anywhere in the specification outside of the claims, Examiner Tran indicated those were examined by other examiners, and not him. To illustrate how the use of "substantially" can be supported via drawings alone, representative Burnett presented a copy of US 5,956,025 to Goulden *et al*. In particular, each of claims 3, 4, 11, and 12 recite, in part "wherein the respective first are comprises a band *substantially across the display*." The support for this claim element is via the drawing figures, as the term "substantially" is not present in the 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:

NYT Web page as rendered on a Mozilla Firefox desktop browser running under the Microsoft Windows XP operating system

Sony Clié running Palm OS

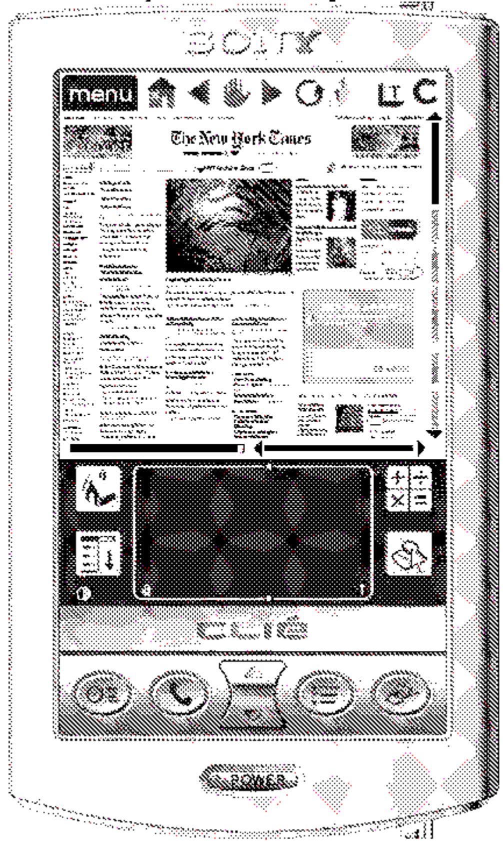


Image 1A

Pocket PC running Windows CE OS

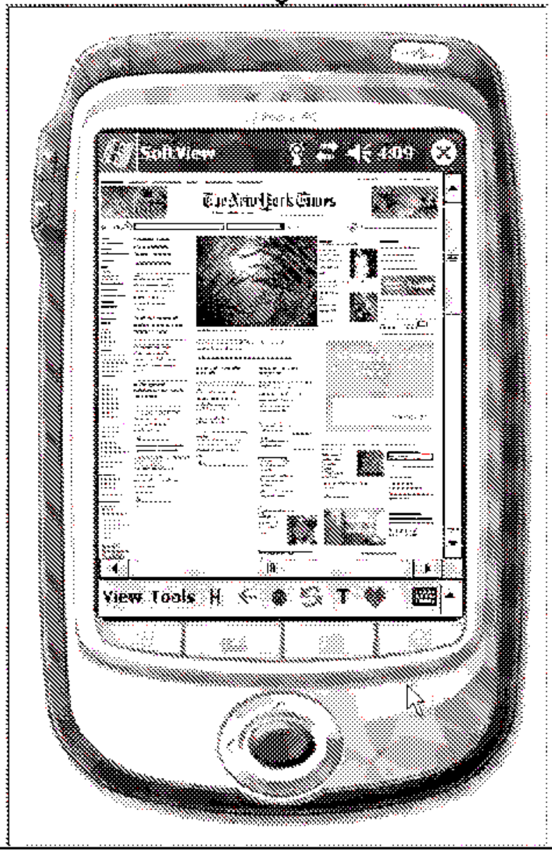


Image 1B

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.

Scope of the terminology "the content corresponding to the selected column is rendered to fit across the display"

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.

Scope of the terminology "the display is re-rendered such that the image is rendered to fit across the display"

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)

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 to new e-mails as they come in, as compared with waiting until the end of the day or some other time to respond to e-mails in more of a batch manner. In a real time flight tracking context, the data that is provided may actually reflect a tracking position that is several seconds, or even minutes, old.

One of skill in the art would recognize that in a software user-interface context, which is applicable to the present claims, the use of real-time typically means the user is enabled to continue an operation in a non-disrupted manner, meaning the user doesn't have to wait a period of time of significance for the operation to be performed. In this context, real-time is perceived by the user's sense of time.

As defined by SearchSMB.com Definitions[3]

real time

DEFINITION- Also see real-time clock and real-time operating system.

Real time is a level of computer responsiveness that a user senses as sufficiently immediate or that enables the computer to keep up with some external process (for example, to present visualizations of the weather as it constantly changes). *Real-time* is an adjective pertaining to computers or processes that operate in real time. Real time describes a human rather than a machine sense of time.

In the days when mainframe batch computers were predominant, an expression for a mainframe that interacted immediately with users working from connected terminals was *online in real time*.

The inclusion of "substantially' in the use of a "substantially in real-time" context (as recited in the claims prior to the instant amendments) was to differentiate the claim from meaning it occurs instantaneously, which would be an erroneous interpretation under the proper use context. Rather, the operation is performed in a non-disrupted manner, as experienced by the user. For the purpose of a defined time period, "in real time" as used herein means the operation is performed in a few seconds or less.

---

[3] http://searchsmb.techtarget.com/sDefinition/0,,sid44_gci214344,00.html

Under the Examiner's interpretation of the term "substantially" in general (as discussed above), the prior claims reciting "substantially in real-time" did not have the foregoing claim scope. Accordingly, there is no equivalence lost due to *Festo* estoppel, as the intended scope of these claims is the same as argued in the December 9, 2007 response.

Discussion of new claim terminology, "preserves the original page layout, functionality, and design of the Web page content."

In accordance with teachings disclosed in the present application (and its related applications incorporated herein), users of various devices, from handheld devices with small screens, to desktop PC's and laptops, to very large screen devices, are enabled to view and interact with Web pages in a manner independent of the screen resolution of such devices' built-in or associated display, while preserving the look and feel (*i.e.,* functionality) of browsing such pages with a conventional desktop browser. As a result, users are enabled to access millions of Web pages on various devices having different screen resolutions while providing a full Web browsing experience similar to that experienced when browsing the same Web pages using a desktop browser.

In order to clarify this result, Applicants have amended many of the claims to recite, in part, "preserves the original page layout, functionality, and design of the [HTML-based Web page] content." For example, amended independent claim 1 now recites (emphasis added),

71. 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***;

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.

A discussion of operations pertaining to an exemplary use case of a device enabled by the presented application was presented in the December 9, 2007 response; for clarity, much of this description is repeated below, while some details are omitted for brevity. The operations are discussed in the context of the following FIG. 1.

The schematic drawing shows an exemplary infrastructure comprising well-known components for facilitating access to and delivery of Web pages. Web page content (*i.e.,* Web content) is served by servers that are accessed via the Internet, also commonly referred to as the World Wide Web (WWW). Accordingly, these servers are typically referred to as "Web" servers. More accurately, they are HTTP (Hypertext Transport Protocol) servers, as they serve content of various types using the HTTP protocol. FIG. 1 shows a pair of exemplary Web servers, including a New York Times

(NYT) Web server and an Advertisement (ADV) Web server. It will be appreciated that literally millions of similar Web servers are connected to the Internet across the world, thus forming the World Wide Web.

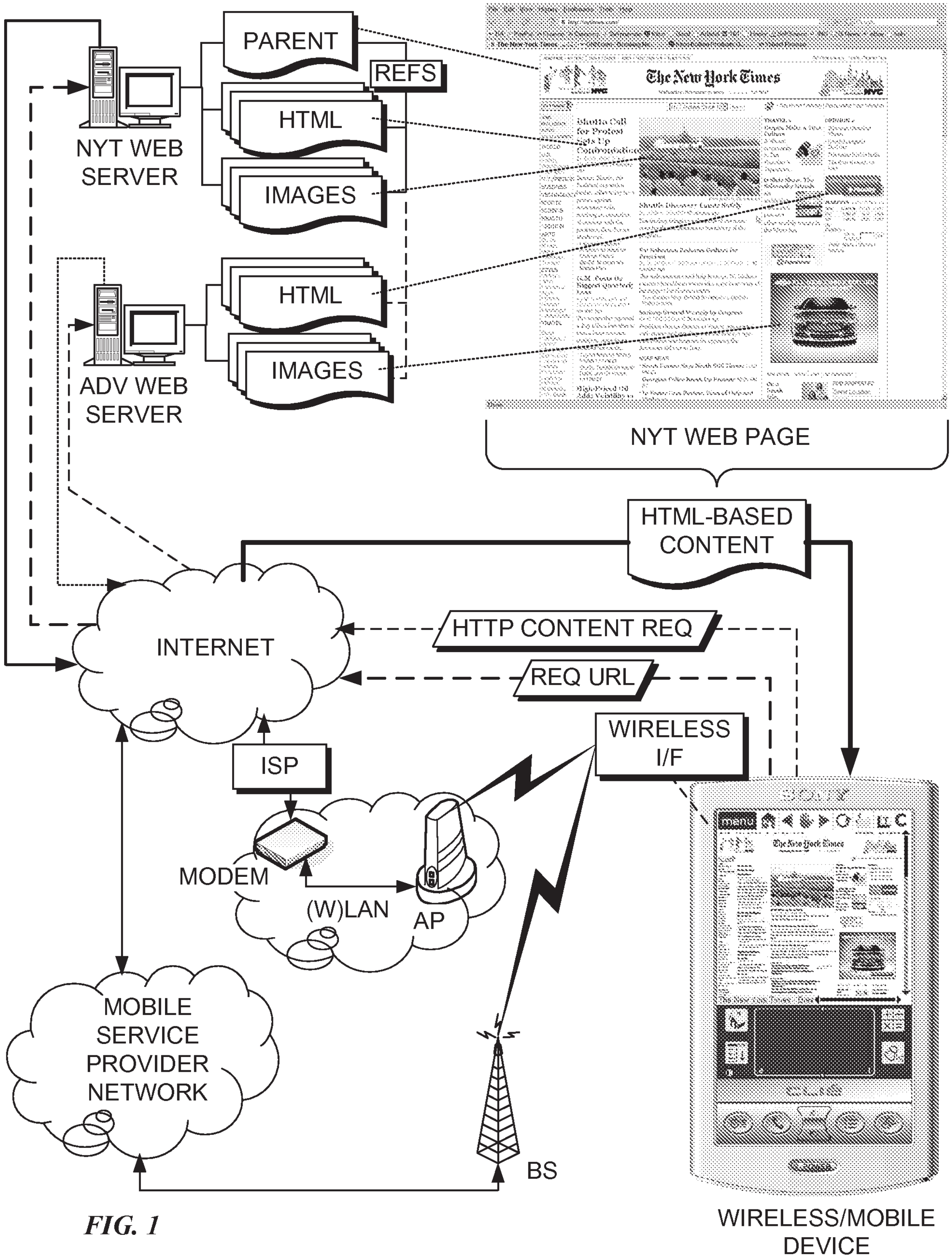

PARENT
REFS
HTML
IMAGES
NYT WEB SERVER
HTML
IMAGES
ADV WEB SERVER
NYT WEB PAGE
HTML-BASED CONTENT
HTTP CONTENT REQ
REQ URL
INTERNET
ISP
WIRELESS I/F
MODEM
(W)LAN
AP
MOBILE SERVICE PROVIDER NETWORK
BS
WIRELESS/MOBILE DEVICE

FIG. 1

To access WWW web servers, users use client devices that are communicatively coupled to the Internet through applicable network infrastructure. In the desktop environment, desktop clients, such as personal computers and workstations, are typically coupled to a Local Area Network (LAN) via an Ethernet link to a LAN host device. (It is noted that some desktop clients may wirelessly connect to a Wireless LAN (WLAN), in a manner similar to that discussed below for wireless clients.) The LAN, in turn is usually connected to the Internet via network infrastructure provided by an Internet Service Provider (ISP). Connection between the LAN and the ISP is typically provided by some type of Modem (*e.g.,* Cable or xDSL Modem) or dedicated hardware (for larger customers, such as businesses). (It is also noted that many individual users still connect to their ISP through a telephone modem.)

Wireless and mobile devices, including those devices covered by the claims herein, typically connect to the Internet in one of the manners illustrated in FIG. 1, or otherwise described in the December 9, 2007 response. (Further details are omitted here for brevity.) By way of example but not limitation, wireless access to the Internet may typically be provided via a mobile service provider, or via other types of wireless connections, such as via WiFi or WIMAX connection, for example.

Now that the infrastructure of FIG. 1 has been described, we proceed with discussion of retrieving and processing the Web page content such that the Web page can be accessed via a wireless/mobile device. In the illustrated example, the process is initiated by a user desiring to access the New York Times (*i.e.,* and electronic version of the New York Times published to the Internet on a given day). This is facilitated by a browser in accordance with teaching of the present application running on the wireless/mobile device. The New York Times may be accessed via the Internet by downloading corresponding Web pages from the NYT Web server. More specifically, the New York Times home page may be accessed by entering the URL (Universal

Resource Locator) www.nytimes.com via the browser's user interface.

As discussed above and in further detail in the present specification, Web pages comprise HTML-based content which may be stored in one or more documents commonly referred to as HTML documents. In addition, Web pages may include dynamically-generated content. Each Web page has a corresponding main or "parent" HMTL document that includes HTML code defining the Web page content layout, at least at some level. The parent HTML document may reference other HTML documents, as well as other content (such as image content) that further define the layout of content contained in the referenced documents. This may proceed in a hierarchical or nested fashion.

To access the Web page, the browser initiates an HTTP connection with the Web server hosting the Web page, and begins downloading the parent HTML document. Depending on how the Web server and/or Web page is configured, additional content (*i.e.,* beyond that included in the parent HTML document) referenced by the parent HTML document, may be retrieved by the Web page host server and then downloaded to the requesting client device, or a portion of this content may be downloaded by the client device via a separate connection. Generally, content that is hosted by a Web server or Web site is assembled by the Web server and downloaded to the client device. On the other hand, externally-referenced content (that is, content that is not stored on the Web server or Web site), is often left to the client device (*i.e.,* the browser) to retrieve.

An example New York Times home page (dated November 7, 2007, 2:22PM ET), as rendered by the Mozilla Firefox browser running on a desktop or laptop computer, is shown at the upper right-hand portion of FIG. 1. The same Web page is shown rendered on a Sony Clié using a SoftView™ browser at the lower right-hand portion of FIG. 1. Notably, the same HTML-based content defining the page layout, functionality, and design of the Web page content is downloaded by each of the Mozilla

Firefox and SoftView™ browsers. Moreover, the same HTML-based content would be retrieved by other desktop browsers, such as Microsoft Internet Explorer, Apple Safari, and Opera browsers, to render the New York Times home page.

As discussed above, the Parent HTML document typically includes HTML code to define the overall layout of the Web page and its content. For example, the HTML code will define whether the Web page includes frames, and, if so, where those frames are located on the rendered page. Various content displayed on the Web page may be stored in the Parent HTML document and/or one or more other HTML documents referenced by the Parent HTML document. If the content is to be rendered in a frame referenced by the Parent HTML document but whose content is not defined within the Parent HTML document, the actual reference to the HTML document storing the content may be in the document defined by the frame reference. For example, for illustrated purposes, the content in the column with the heading "Bhutto Call for Protest Sets Up Confrontation" is depicted to be stored in an HTML document that is hosted by the NYT Web server, but is separate from the Parent HTML document.

Likewise, image content may be stored separate from the Parent HTML document. This is typically done since images, which often contain a large amount of data due to the nature of image data, make require significant download time, especially over a slow connection. By putting image content in (a) separate document(s), the basic page layout and text content can be rendered much faster. Typically, HTML code defining the page layout location of an image on the page may be used to place an image "placeholder" or other indicia on the screen prior to rendering of the image.

As discussed above, various portions of the Web page content may be stored on Web servers that are external to the Web page host server. This is often the case with advertisement content. Rather than have the advertisement content stored locally on each Web server, the advertiser will use an advertisement host site to store and serve the advertisement content. For example, in FIG. 1, image data for rendering the "All New Chevy Malibu" advertisement is depicted as being stored in an image document on the Advertisement Web server.

Typically, externally referenced advertisement content is downloaded by the browser directly from the advertisement content host site, rather than from the Web page host site. The network location of the advertisement content host server is identified by parsing the retrieved HTML-based content, and an HTTP GET request is used to download the associated advertisement content from its host server.

Some Web pages may include "embedded" content hosted by an external site. For example, the New York Times Web page includes embedded content provided by Fidelity. Oftentimes, such embedded content may be dynamic in nature (that is, may change over time or differ depending on identification of the target user). Generally, embedded content may be retrieved by the browser from an external host site (*e.g.,* advertisement Web server depicted in FIG. 1[4]), or such content may be first retrieved by the Web page host site and served to the browser.

It is common terminology to refer to a browser "retrieving" or "downloading" a Web page. For example, upon entry of a new URL in the browser Web address box, the browser will download the Web page referenced by the URL. It is well understood that this doesn't imply that all of the content associated with the Web page must be retrieved or downloaded. Some of the content is typically used for search engine purposes, such as Metatag header information, or is otherwise not used for rendering purposes. In other cases, content may be referenced that is not supported by the requesting browser. For example, "Flash" content typically requires a Flash plug-in viewer (or built-in Flash support provided by some browsers); if the plug-in viewer is not loaded by the browser (or such support isn't built in), the Flash content cannot be

---

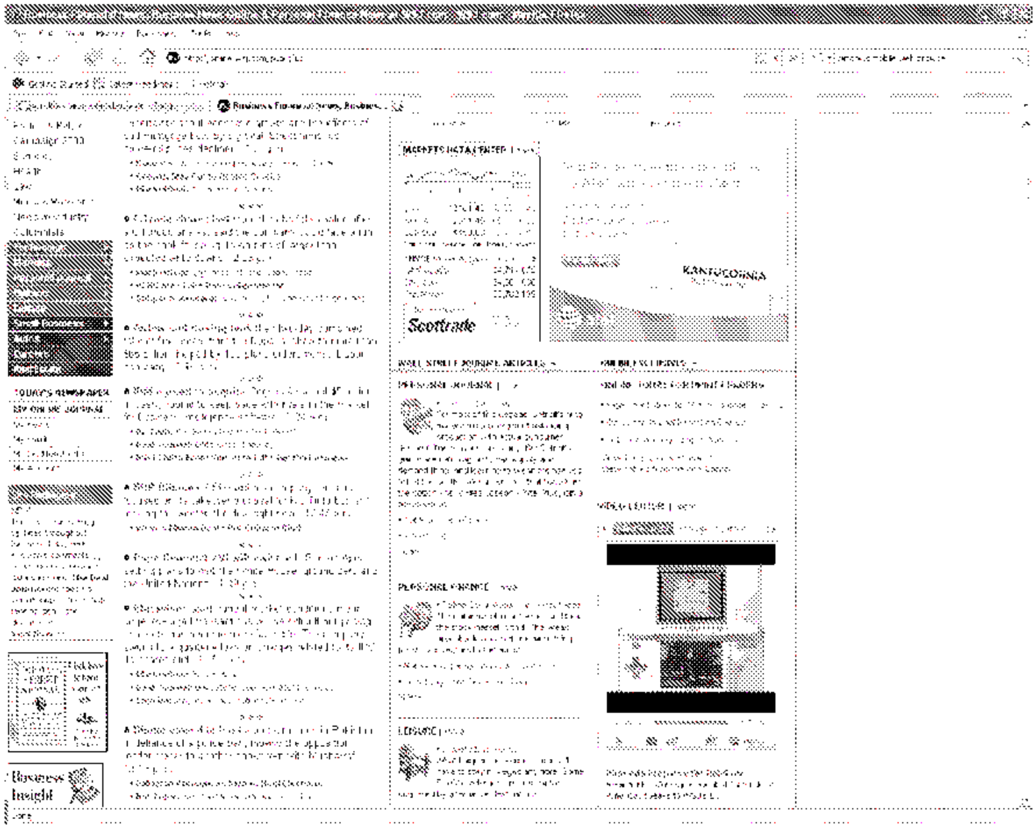
[4] It is noted that it would be likely the Fidelity content would be hosted by its own server that would be separate from the advertisement server; however, for simplicity, the advertisement is used for illustrative purposes.

displayed. This is also true for TIFF images on the USPTO Web site. Unless the proper TIFF plug-in viewer is loaded, the TIFF images will not be displayed.

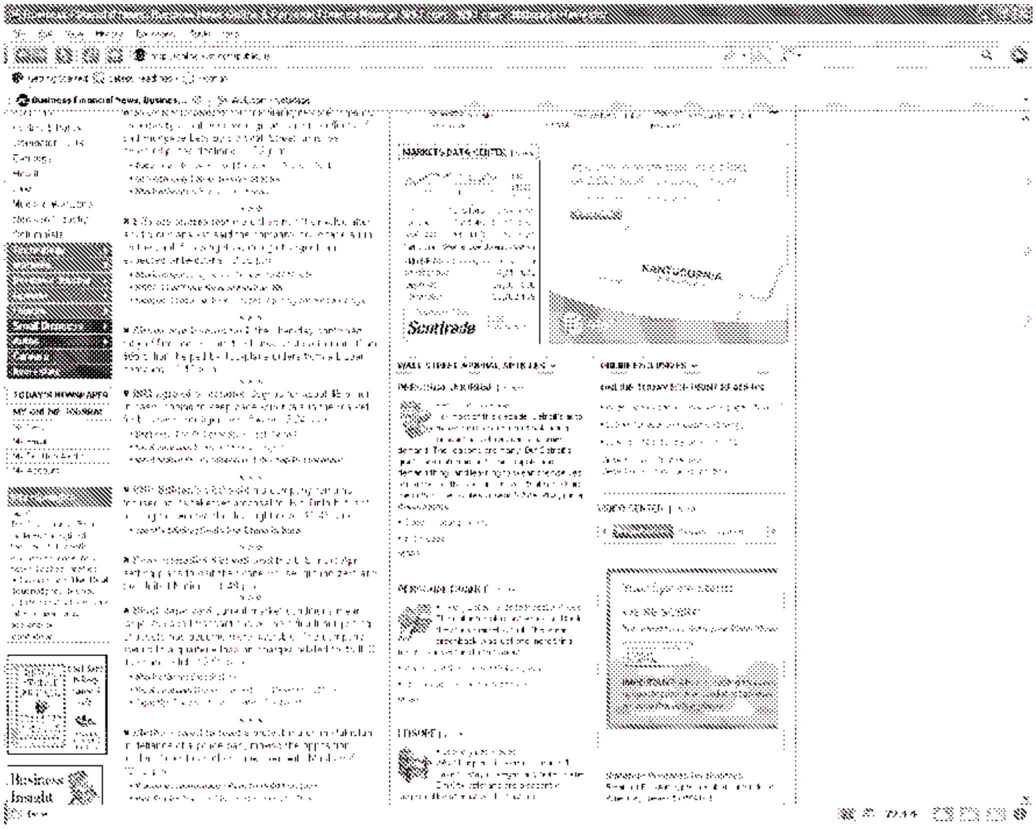
A similar situation exists with Active-X controls. In order to use the Active-X controls, the browser needs to provide support for Active-X controls. Since Active-X controls were developed by Microsoft, all recent versions of Microsoft Internet Explorer provided support for Active-X controls. Meanwhile, browsers from other vendors, such as Apple Safari, Mozilla Firefox, and Opera, do not support Active-X controls.

When a browser encounters content that is not supported “natively” by the browser, the browser will typically check to see if an appropriate plug-in is available. Depending on the browser and/or particular Web site, if an appropriate browser cannot be found, the browser or Web site may apprise the user of the situation and enable the user to download the plug-in. In other instances, the content is simply ignored. Thus, in some cases, the Web page may reference content that is never retrieved when the Web page is retrieved by the browser.

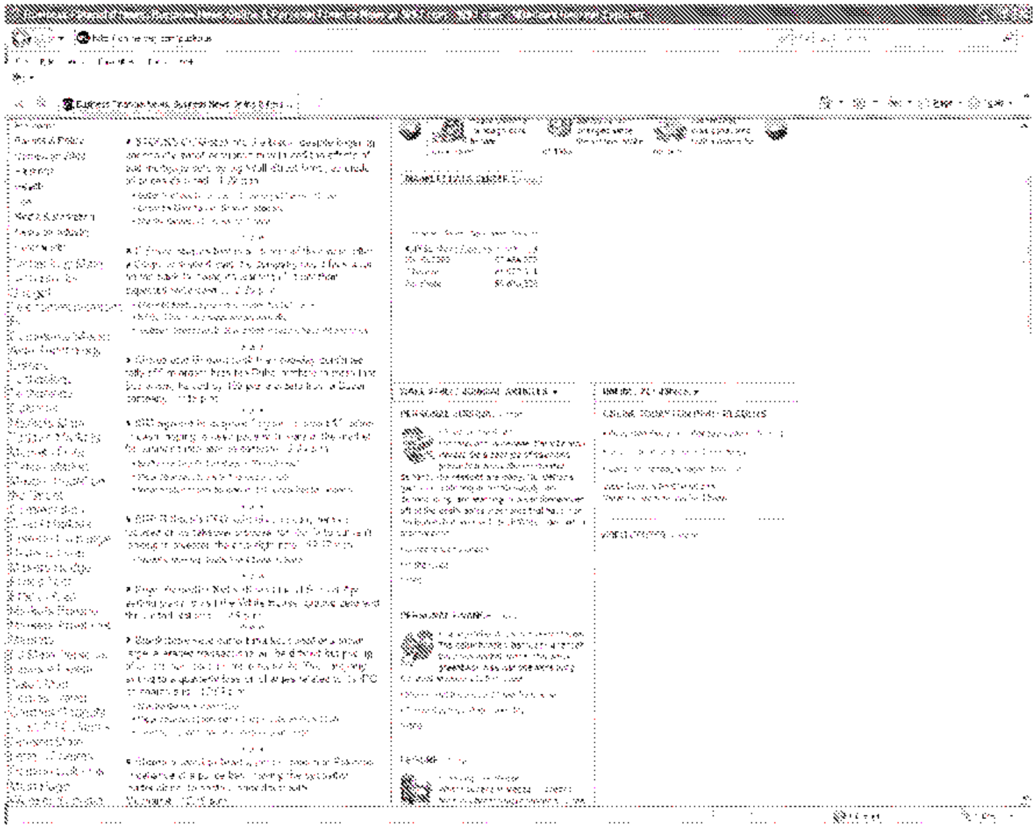
The three screen shots below respectively show the same Web page rendered on a Netscape Navigator 9 browser, a Mozilla Firefox 2.0 browser, and an Internet Explorer 7 (IE 7) browser. In this particular instance, certain features of the IE7 browser are disabled for security reasons. It is also missing some plug-ins. Each of these browsers is running on the Microsoft Windows XP operating system.



Mozilla Firefox 2.0 Browser



Netscape Navigator 9 Browser



Microsoft Internet Explorer 7 Browser

It will be observed that the Web page is rendered substantially the same by the Netscape Navigator 9 and FireFox 2 browsers, while portions of the Web page are rendered in a different manner by the IE 7 browser (notably the left-hand column). The similarity between Netscape and Firefox is expected since they both use the same Mozilla core rendering code, while IE 7 uses Microsoft's rendering code. It is further noted that some Web pages are coded to account for different browser features. For example, some Web pages will have provisions for Active-X controls for pages to be viewed by Internet Explorer browsers, while possibly including provisions for alternate mechanisms when using other browsers.

This example Wall Street Journal Web page includes various embedded ***non-***

HTML content requiring support of one or more plug-ins[5] or otherwise built in support for rendering ***non-HTML*** content of a particular content type. In particular, the VIDEO CENTER object in the lower right-hand corner requires an Adobe (formerly Macromedia[6]) Flash viewer for rendering Flash content, which uses vector and raster graphics, a native scripting language called ActionScript and bidirectional streaming of video and audio.[7]

It is noted that there is a message in the VIDEO CENTER box in the Web page rendered by the Netscape Navigator 9 browser indicating that the browser needs to update its Flash player. In the case of the Firefox 2 browser, either the appropriate Flash player was found or an appropriate level of support for Flash content is built into the browser. In this case, the Flash .SWF file including data to render a video image of a Nintendo DS console is retrieved from a corresponding host server and rendered by the browser (if it has built-in support) or Flash player, as applicable. In the case of the Netscape Navigator 9 browser, the appropriate Flash player plug-in is not available; accordingly, the video image of the Nintendo DS console is not retrieved.

In the case of the particular IE 7 browser configuration used to obtain the IE7 screen shot, the Flash player is either missing or blocked. As a result, the aforementioned VIDEO CENTER image is missing (just an empty box is rendered, as defined by corresponding HTML). Moreover, the IE 7 browser did not render a message indicating the Flash player needed to be upgraded. In addition, the source for

---

[5] As defined by Wikipedia, A **plugin** (**plug-in**, **addin**, **add-in**, **addon** or **add-on**) is a computer program that interacts with a host application (a web browser or an email client, for example) to provide a certain, usually very specific, function "on demand". Applications support plugins for many reasons. Some of the main reasons include: enabling third-party developers to create capabilities to extend an application, to support features yet unforeseen, reducing the size of an application, and separating source code from an application because of incompatible software licenses.

[6] Macromedia is now a division of Adobe Systems

[7] For more details on the Adobe Flash Player, see, *e.g.*, http://en.wikipedia.org/wiki/Adobe_Flash_Player

the AT&T advertisement in the upper right-hand portion is blocked via a security setting, resulting in this portion of the page being rendered using the same background color as the frame it (would be) embedded in.

A point for discussing the foregoing is to make it clear that,

1. Even when rendering the same Web page source content (*i.e.,* the HTML code definition of the Web page), conventional Web browsers may not render the (non-scaled) Web page identically. Scaling Web pages may also result in alternation of the page layout. However, under aspects of embodiments of the invention (such as claimed in claim 71) the overall layout, functionality and appearance (design) of the scaled Web pages defined by the HTML code for the Web page are preserved.
2. Plug-ins may be required to render ***non-HTML content*** that is embedded within some web pages or used in a separate window launched from a web page. Notably, the plug-in content is not a Web page, but rather a specific type of content requiring a corresponding plug-in application to render the content.

The new claim language introduces the term "functionality." Preserving functionality generally pertains to preserving the interoperability of various HTML-based Web page content, such as hyperlinks and UI controls such as input forms defined via corresponding HTML-based code. It is noted that the HTML code defining a Web page's overall layout, functionality and design does not define how a user interaction with the Web content is to be supported, but rather defines the existence of a corresponding function within the Web content to support the interaction. For example, a hyperlink definition within a Web page merely defines a link (hyperlink reference of *href*) to corresponding content, it does not define how the hyperlink associated control is to appear on the screen nor how the hyperlink is to be activated. That is up to the browser's implementation, which varies by browser. For example, some browsers underline text content associated with a hyperlink, while others change the appearance of a pointer when over a control (*e.g.,* text content) associated with a hyperlink (or otherwise change the appearance of such content). Moreover, how the hyperlink is activated is not defined by the corresponding HTML-based definition, but again is left to the browser implementation. Accordingly, preserving content functionality means that functionality defined by corresponding HTML code (*e.g.,* activation of a hyperlink in the present example) is supported, without limiting the particular user interface for how that activation is facilitated.

In the implementation of a zoomable browser, it may be desirable to change user interface behavior depending on a current use and/or view context. For example, the hyperlink controls of a conventional Web page designed to be viewed with a desktop browser are typically activated via the same user interface input (*e.g.,* clicking with a mouse), since all of the hyperlinks controls (on at least well-designed Web pages) are (presumably) designed to be viewable on the desktop browser (at least viewable to most users). In contrast, when the same page is rendered so as to fit on a handheld device's display, corresponding hyperlink controls may not be readable. As a result, it may be advantageous to implement a context-based user interface that may result in a different action for the same user input depending on a current user and/or zoom context. For example, under the zoom to column, image, and paragraph user interface features disclosed in the present application, touching proximate to content associated with a hyperlink control may or may not activate the hyperlink control, depending on a current zoom level. By way of illustration, when touching content proximate to a hyperlink control that is also contained within a column when in a zoomed-out view, such as a full page view, the browser may interpret the input as an input to zoom to the column rather than an input to link to a hyperlinked reference associated with the content, particularly when the content is no readable in the current view.

Preserving the design of the Web page's HTML-based content corresponds to rendering the Web page at different zoom levels and panned view in accordance with its original design, which includes such things as type fonts, separator bars, tables, *etc.* Again, the Web page's design is a matter of interpretation by the particular browser, as, for example, the same content (as defined by its corresponding HTML definition) may be rendered using different colors by different browsers. Similarly, browsers may substitute fonts for fonts (as defined by corresponding HTML code) that are not supported by the browser or operating system. With respect to the scope of the terminology "preserving the [overall layout, functionality, and] design" of the content, this refers to preserving the design as interpreted by the browser[8] while at different zoom levels and panned views, as opposed to rendering the content identically to how it is rendered by a particular desktop browser that may interpret the page design differently.

With further respect to dependent claims 126, 159, 241, 260, 333, and 353, design aspects of a Web page as defined by cascaded style sheet (CSS) data included in the Web page definition are also preserved.

A similar context exists with respect to "preserving the overall layout [, functionality, and design]" of the content. Again, the page layout (to be preserved) is determined as interpreted by the browser, rather than as a comparison to how it is rendered by a particular desktop browser. As described above and in other remarks, browsers often do not render Web pages derived from the same HTML-based definition identically. Accordingly, one of ordinary skill in the browser art would not expect Web pages rendered using a browser in accordance with the teachings disclosed in the present application (*e.g.,* the SoftView™ browser) to render pages as *exact* scaled replicas of the same page rendered by another browser, such as Internet Explorer or Safari, for example. Also as discussed previously, due to rendering limitations such as

[8] More particularly, differences in page interpretation will generally be a function of the browser's rendering engine (*a.k.a.* layout engine).

fixed size fonts, renderings of the same page when viewed at different zoom levels may result in small variations, as opposed to an exact scaled version of the same content (as if viewed by a magnifying glass). While there are implementations that may produce this exact result, such results are not required by the scope of the terminology “preserving the overall layout … of the content.”

Each of claims 174, 180, 265, 271, and 359 include claim elements that likewise recite “preserves the overall layout, functionality, and design” of the [HTML-based Web] content, while each of claims 99, 143, 211, and 337 recite “while preserving the overall layout, functionality, and design” of the [HTML-based Web] content. Accordingly, the claim scope of the related elements in each of these claims is to be interpreted in a similar manner that that discussed above with respect to the use of similar terminology in claim 71.

Each of claims 76, 185, and 276 recite, “wherein the original page layout, functionality, and design of the Web page content are preserved at each of the different resolutions,” while each of claims 128, 244, and 303 recite, “wherein the original page layout, functionality, and design of the Web page content are preserved regardless of a zoom level of the Web page.” Accordingly, the claim scope of the related elements in each of these claims is to be interpreted in a similar manner that that discussed above with respect to the use of similar terminology in claim 71.

<u>Discussion of new claim terminology, “enables [enabling] a user to view, zoom, and pan the HTML-based Web page content of billions of Web pages in a manner that preserves the original layout, functionality, and design of the Web page content . . .”</u>

Each of claims 244, 303, 359, 383-389, and 393 have been amended to now recite, in part, “enables” or “enabling” “a user to view, zoom, and pan the HTML-based Web page content of billions of Web pages in a manner that preserves the original layout, functionality, and design of the Web page content . . .” The scope of the terminology “in a manner that preserves the original layout, functionality, and design of the Web page content” should be in accordance with that discussed above for similar terminology in claim 71. In addition, in each of these claims the terminology “substantially any Web page” has been replace by “billions of Web pages” to be more definite.

At the time of the filing of the non-provisional parent application (US 09/878,097 – issued as US 7,210,099) (mid-2001) to which the present application claims priority, there were on the order of a several billion web pages associated with the “World Wide Web” and accessible via the Internet, with the specific number being somewhat indeterminable. As stated in paragraph [0093] of the present application, “. . . users are enabled to view the entire content of billions of existing Web pages using hand-held devices in a simple and reasonable way.” This statement was based on the observation that, when tested, a browser incorporating the principles of the invention disclosed in the present application enabled the test user to browse, zoom, and pan nearly every Web page that was tested, while preserving the original page layout, functionality, and design of the Web page.[9] Based on the inherent principles and teachings disclosed, this result was expected, as the rendering engine employed by the browser (the Mozilla rendering engine) was based on the same rendering engine used in one of the two most dominant browsers at the time (i.e., the rendering engine used by the Netscape Navigator browser). (It is respectfully noted that the use of the Mozilla rendering engine in an embodiment in the present disclosure is merely exemplary, and not limiting.) Since Netscape Navigator was a dominant browser at the time, many if not most Web pages were designed to support browsing with Netscape Navigator

[9] Of approximately 500 of the most browsed (at the time) Web pages that were tested, only a handful did not work. Of particular note was a Sony Web site that was entirely flash-based (and thus not HTML-based). It is also noted that some Web pages were/are designed to be browsed by a specific browser, such as Internet Explorer; such pages may not render and/or function properly under other browsers.

(either by intent and/or based on good HTML coding practices for manually designed pages, or through use of one of many Web page design tools that generated HTML that could be properly interpreted by Netscape Navigator).

One of skill in the art will recognize that the principles and teachings disclosed in the present application may be applied in a browser implementation employing one of many different rendering engines, such as but not limited to today's version of the Mozilla rendering engine (code-named "Gecko") used by the Firefox and Netscape Navigator browsers, the rendering engine employed by Microsoft Internet Explorer (code-named "Trident" (*aka* MSHTML)), or the Webkit rendering engine use by Apple's Safari browser. Since each of these rendering engines are capable of rendering the vast majority of today's Web pages[10], a browser implementing such a rendering engine in combination with the principles and teachings disclosed in the present application would likewise be capable of rendering the vast majority of billions[11] of today's Web pages while preserving the page layout, functionality, and design of the Web pages under various zoom levels and panned views.

Conclusion

In view of the amendments and the remarks above, Applicant respectfully submits that this application is in condition for allowance. If, however, the Examiner believes that there are any unresolved issues requiring adverse action in any of the

---

[10] One of skill in the browser art would recognize that rendering engines do not render Web pages completely by themselves, but rather employ various support functions provided by the host operating system for particular rendering operations. Among these support functions is support for rendering text in various languages. The particular languages that are supported will vary depending on the operating system and/or extensions to the operating system (or otherwise add-on functionality provided by the browser) for rendering text of a particular language. If support for rendering text in a given language via either the operating system or a particular extension is not available, the text content in such a language will not be able to be rendered on pages that include such text content.

[11] Depending on the source, it is estimated there are currently 16-48 billion Web pages available via the Internet.

claims now pending in the application, it is requested that the Examiner telephone R. Alan Burnett at (425) 417-4729 or (425) 562-0923 so that appropriate arrangements can be made for resolving such issues as expeditiously as possible.

Respectfully submitted,

LAW OFFICE OF R. ALAN BURNETT, PS

Date: May 20, 2007 /S/ R. Alan Burnett

R. Alan Burnett
Reg. No. 46,149

4108 131st Ave SE
Bellevue, WA 98006

EXHIBIT F

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:
MAILED
AUG 08 2011
CENTRAL REEXAMINATION UNIT

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)

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.

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 (<u>parent</u> 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 (<u>parent</u> 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): (<u>parent</u> 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 <u>did not</u> 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.

<u>Claim Terms and other Meaningful Citations</u>

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) OR (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."

Third Party Requester's Proposed Grounds of Rejections based on the Pad++ References

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 a point and a magnification (scale or zoom). "The point specifies the portion of the surface that will appear at the center of the view." 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:

ZD

ADK

EXHIBIT G

Attorney Docket No. 005207.P001XD *Patent*

Certificate of Electronic Filing

I hereby certify that this correspondence is being Electronically Filed via EFS

on December 9, 2007

Date of Deposit

R. Alan Burnett

Name of Person Filing Correspondence

/S/ R. Alan Burnett December 9, 2007

Signature Date

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)
	)
Rohrabaugh et al.	) Examiner: Tran, Quoc A.
	)
Serial No. 11/045,757	) Art Unit: 2176
	)
Filed: June 8, 2001	)
	)
For: SCALABLE DISPLAY OF INTERNET CONTENT ON MOBILE DEVICES	)

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT AND RESPONSE TO OFFICE ACTION

Sir:

Responsive to the Office Action mailed October 23, 2007, the Applicant requests the Examiner to enter the following amendments and to reconsider all pending claims in view of the amendment and the following remarks.

Amendments begin on page 2. Remarks begin on page 24.