(12) **United States Patent** (10) **Patent No.:** **US 6,449,634 B1**

Capiel (45) **Date of Patent:** **Sep. 10, 2002**

(54) **METHOD AND SYSTEM FOR REMOTELY SENSING THE FILE FORMATS PROCESSED BY AN E-MAIL CLIENT**

(75) Inventor: **Gerardo J. Capiel**, San Francisco, CA (US)

(73) Assignee: **Digital Impact, Inc.**, San Mateo, CA (US)

( * ) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/240,096**

(22) Filed: **Jan. 29, 1999**

(51) **Int. Cl.$^7$** ............................................... **G06F 15/16**

(52) **U.S. Cl.** ...................... **709/206**; 709/203; 709/207; 709/217; 709/218; 709/219; 707/104.1; 707/513

(58) **Field of Search** ................................ 709/206, 207, 709/217, 218, 219, 203; 707/10, 101, 102, 103, 104.1, 513

(56) **References Cited**

U.S. PATENT DOCUMENTS

| | | | | | |
|---|---|---|---|---|---|
| 5,848,415 | A | * | 12/1998 | Guck | ............................ 707/10 |
| 5,911,776 | A | * | 6/1999 | Guck | ............................ 709/217 |
| 5,961,590 | A | * | 10/1999 | Mendez et al. | ............. 709/206 |
| 5,973,696 | A | * | 10/1999 | Agranat et al. | ............. 345/307 |
| 5,974,441 | A | * | 10/1999 | Rogers et al. | ............... 709/200 |
| 6,018,774 | A | * | 1/2000 | Mayle et al. | ............... 709/250 |
| 6,092,114 | A | * | 7/2000 | Shaffer et al. | .............. 709/232 |
| 6,253,231 | B1 | * | 6/2001 | Fuji | ........................... 709/206 |

FOREIGN PATENT DOCUMENTS

| | | | | |
|---|---|---|---|---|
| JP | 02000261493 | * | 9/2000 | ........... H04L/12/14 |

* cited by examiner

*Primary Examiner*—Ayaz Sheikh
*Assistant Examiner*—Frantz B. Jean
(74) *Attorney, Agent, or Firm*—Townsend and Townsend and Crew LLP

(57) **ABSTRACT**

The present invention relates to the processing of E-mail messages over a telecommunications network. More particularly, the present invention relates to the detection and monitoring of file formats which can be processed and displayed at an E-mail client. Specific embodiments include, sending an E-mail message to the E-mail client, determining at the E-mail client a file format that the E-mail client can process and display, and indicating to the E-mail sensor server the file format that the E-mail client can process and display. Once the file format is determined, subsequent E-mail messages maybe of the same file format. The file format could be hyper text mark up language (HTML) statements or dynamic HTML(DHTML) statements or Java applets. The method may also include monitoring the status of the E-mail message received at the E-mail client.
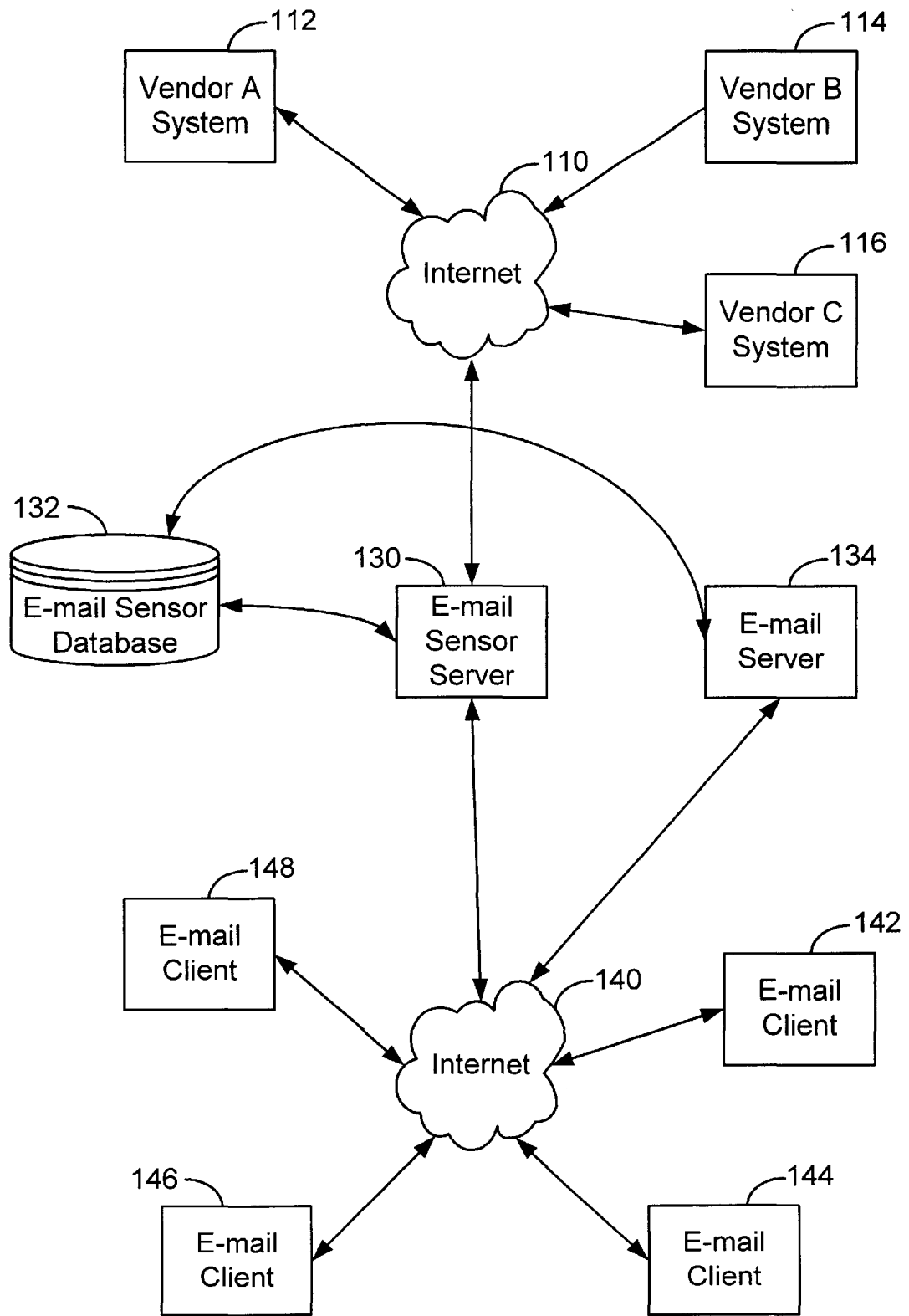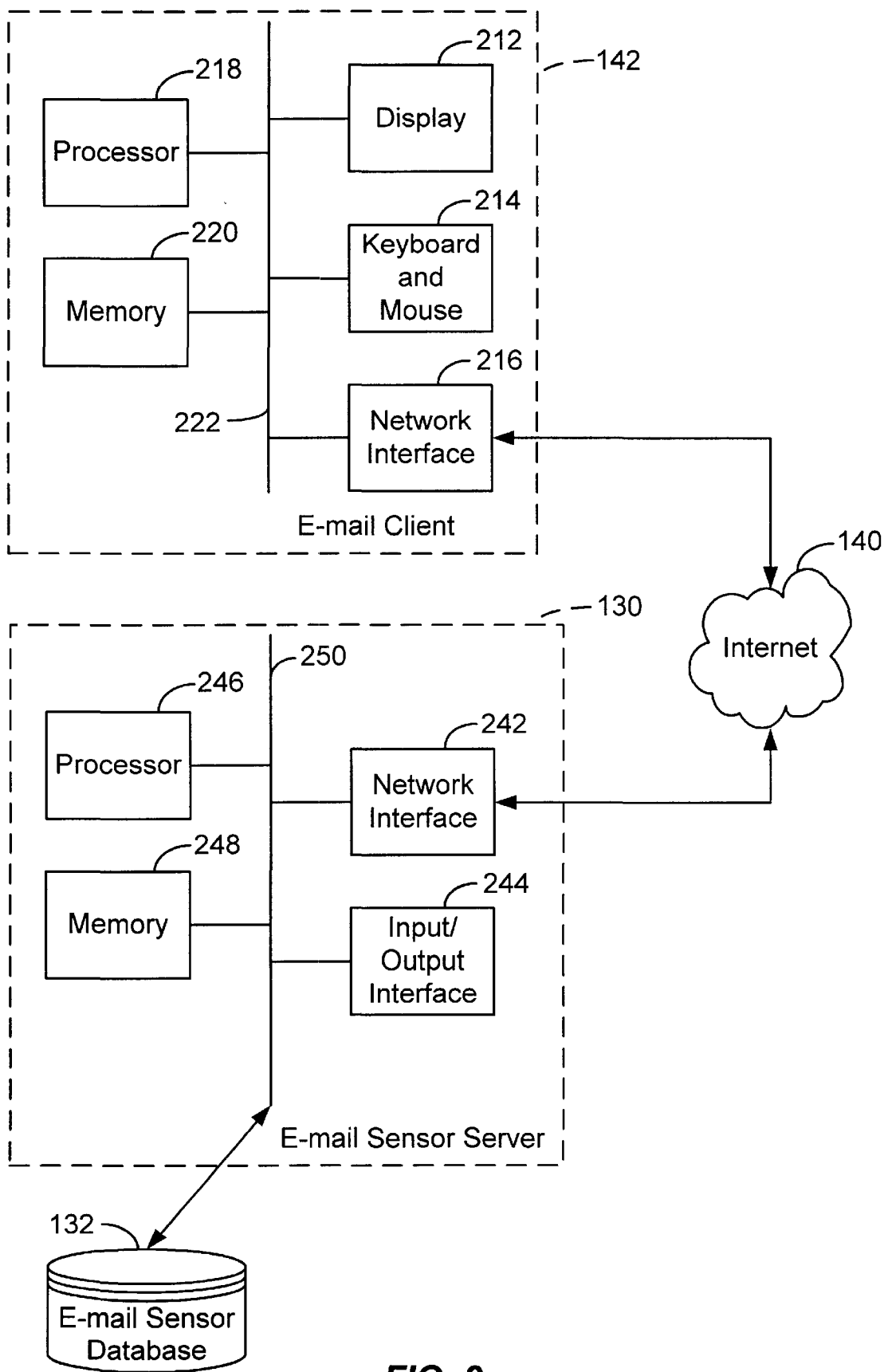
**33 Claims, 10 Drawing Sheets**
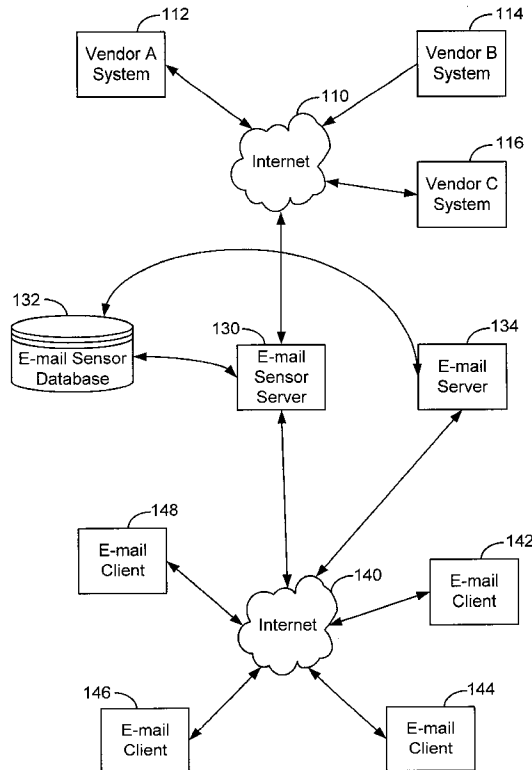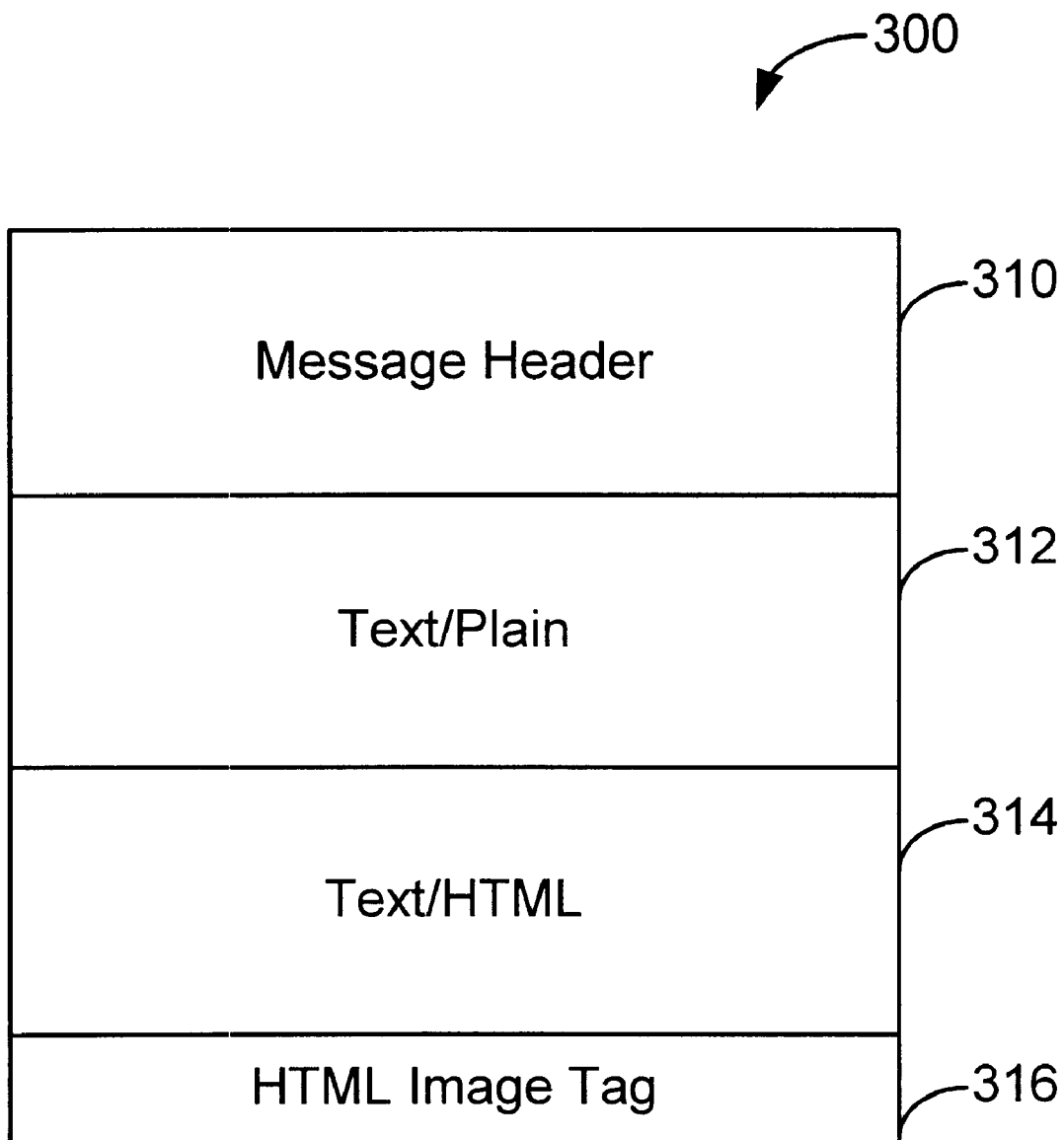
**FIG. 1.**

FIG. 2.

FIG. 3.

320 — Dear Gerardo,

Tower is in the Holiday mood - come and join us!

We have gift ideas and boxed sets for every budget.

Here are some holiday hits to get you started:

Nat King Cole - The Christmas Song
**Sale Price: $9.99 - You save: $2.00**

-------------------------------------------------

Thanks for letting us contact you!

If you wish to UNSUBSCRIBE from future mailings, please go to:
http://tower.m0.net/m/u/t.asp?email=gcapiel%40digital-impact.tngi.com

## FIG. 3A.

**TOWER** RECORDS•VIDEO•BOOKS

*What would your Valentine like?*
candy? flowers? MUSIC!

Wed., Jan 27 1999

**PULSE!** Magazine
Featured Listening
Trivia Games: Win $

**Main**
Home
Customer Service
Shopping Basket
Extended Search
Browse
Top 1000 on sale
CDs under $7

**Departments**
Video/DVD
New Releases
Advance Orders
Rock/Pop
Classical
Jazz
International
Blues
Dance/Electronica
Country
Latin
Soundtracks
Labels
Other Formats
Desert Island
Discs
Tower Gift Cards

Store Locator
Corporate
Tower Asia
Tower Europe

**Order by Phone:**
1-800-ASK-TOWER
**Order by Fax:**
1-800-538-6938
**Shop AOL:**
Keyword: Tower

| ▼ Music | ▼ Video | ▼ Classical |

| Artist ▼ | | Search | Classical Search / Video Search |

| The '60s Soundtrack | Collective Soul Dosage | Varsity Blues Soundtrack | 1999 Grammy Nominees |
| ADD TO BASKET | ADD TO BASKET | ADD TO BASKET | ADD TO BASKET |
| $13.99 CD | $13.99 CD | $13.99 CD | $13.99 CD |
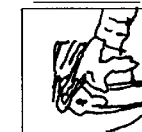
**Framed Miles**
To celebrate the latest Miles Davis releases, The Complete Bitches Brew and Love Songs, we've got 6 framed Miles Davis prints! Enter today to win one for your very own!

**Love for Sale**
Visit our Valentine Shop for a musical treat for your sweetheart this year! What better way to show you care than with a sweet tune?

**Tower's Top 100 for 1998**
It's here, the Top 100 selling albums compiled from our 98 U.S. locations! See if your favorites for last year match up with our stores!.

**John Lee Hooker and B.B. King**
When old friends get together to laugh and reminisce, it's beautiful.

**Looking for Italian Jazz**
Italy has a good helping of straightahead jazz musicians, but this is something else.

**Click Here For How-to-Order Information**

*FIG. 3B.*

*FIG. 4.*

—500

| |
|---|
| Message Header |

—512

| |
|---|
| HTML Comment |

—514

| |
|---|
| Text/HTML |

—516

| |
|---|
| HTML Image Tag |

—518
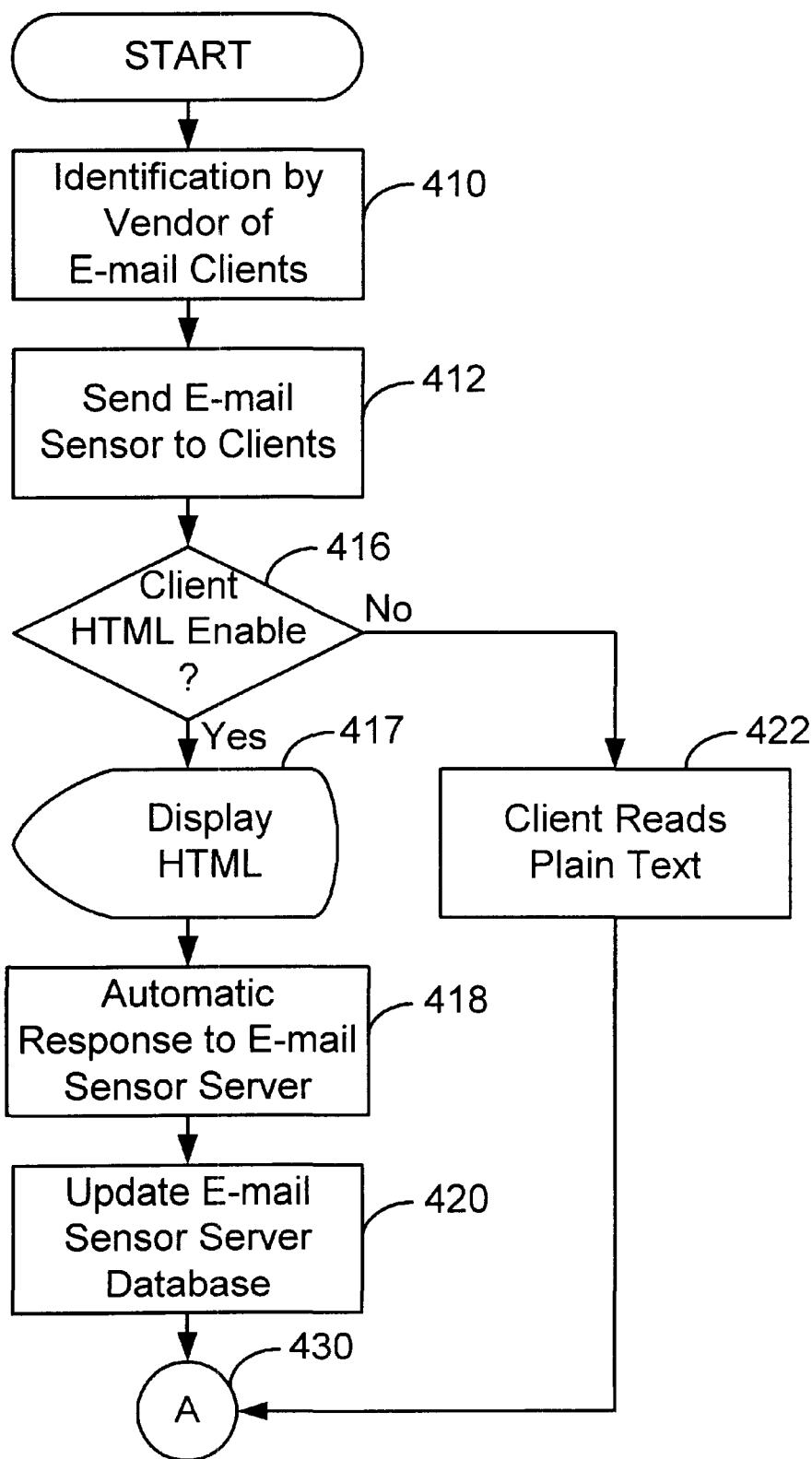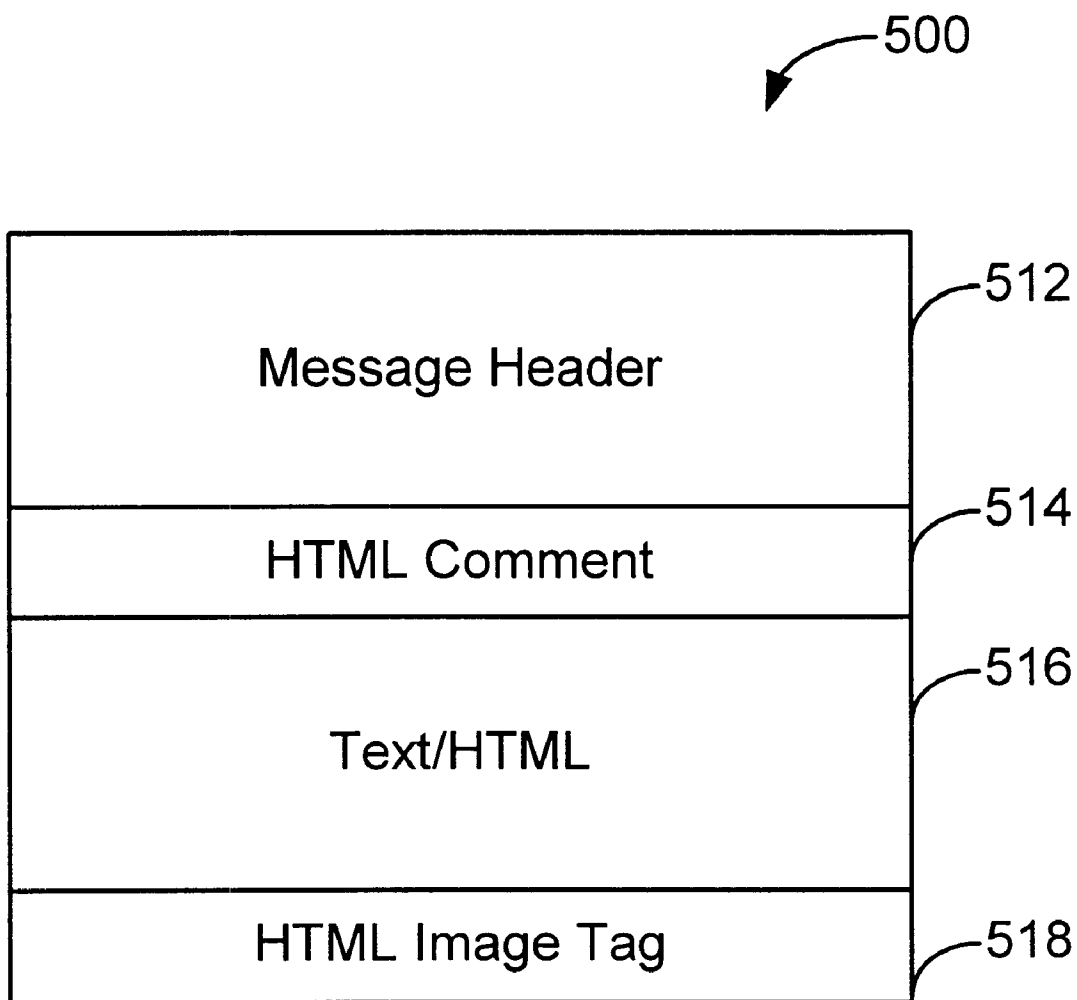
**FIG. 5.**

**TOWER**
RECORDS*VIDEO*BOOKS

**super clearance sale**
www.towerrecords.com

**Dear Valued Customer,**

Everything On Sale!
January 14th-19th

Kick off 1999 with Tower Records!

Shop Tower's Super Clearance sale today for incredible savings on every CD, cassette, video and DVD title!

While at towerrecords.com, be sure to check out New Releases, sample thousands of new tracks in our Listening Station and play Trivia Blitz! for cash prizes!

Find a Tower Records location near you!

If you wish to unsubscribe from future mailings, please click here.

Country   Classical   Jazz   pop   SOUNDTRACKS   Video

*FIG. 5A.*

430

(A)

Request for Second E-mail Sensor Message — 550

Check Database — 552

Client HTML Enabled? — 554

No → Send E-mail Sensor Message (Format 300) — 558

Client Reads Plain Text — 560

End

Yes — 556

Send E-mail Sensor Message (Format 500)

Client Reads HTML Comment? — 562

Yes → Client Affirmatively Responds That Cannot Read HTML — 566

End

No — 564

Display HTML

End

**FIG. 6.**

Start

Connect to DB — 610

Get Client's Type of Browser (i.e., client's E-mail software) — 612

— 614
Browser in DB? — No

Yes

Create Record in DB — 616

Associate in the DB the Type of Browser with Clients Email address — 618

— 622
Opened for first time? — No

Yes

Increment email open count — 626

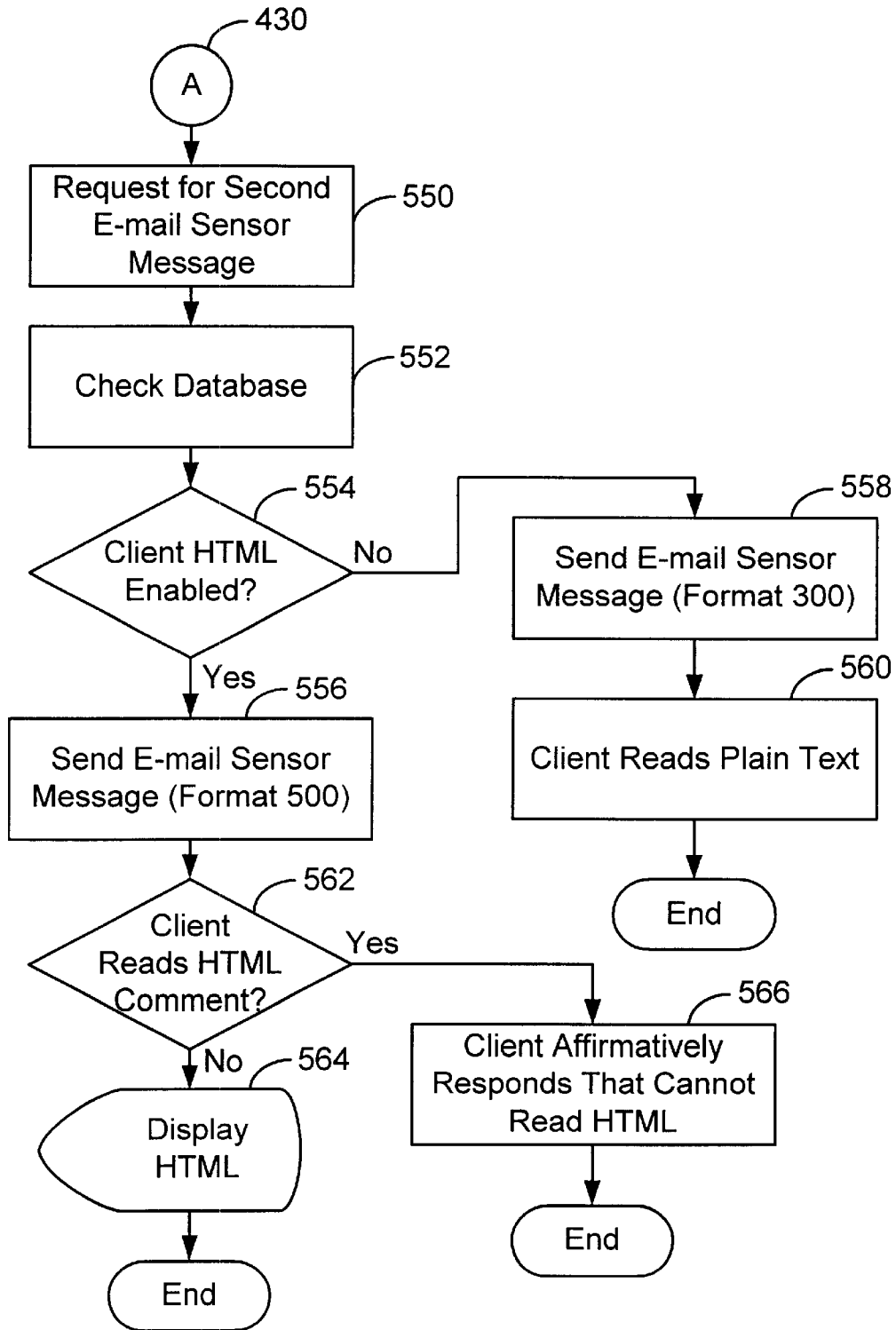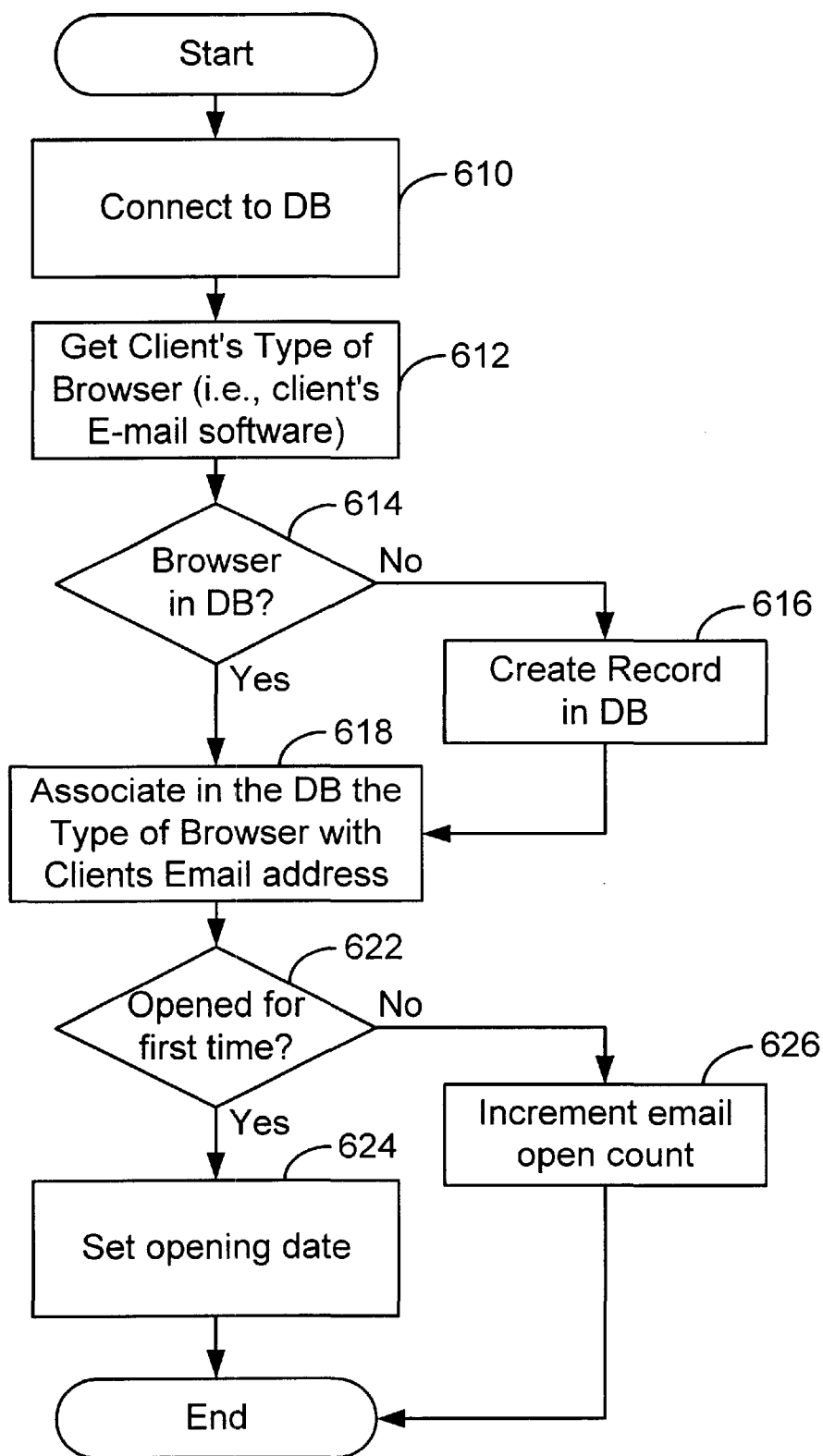Set opening date — 624

End

**FIG. 7.**

1

# METHOD AND SYSTEM FOR REMOTELY SENSING THE FILE FORMATS PROCESSED BY AN E-MAIL CLIENT

## BACKGROUND OF THE INVENTION

The present invention relates to the processing of E-mail messages over a telecommunications network. More particularly, the present invention relates to the detection and monitoring of file formats which can be processed and displayed at an E-mail client.

E-mail has become a major method of communicating information over telecommunication networks; this includes the "Internet" and intranets. There are estimates that about 30–55 million American homes are wired for E-mail communication today, and this number is rapidly growing. This provides a large audience for E-mail advertisers. Thus, through E-mail, vendors can make millions of customers aware of their products. One method is to reach the largest audience possible. However, blanket advertising is akin to junk mail and is ineffective in terms of actual sales.

Targeting advertising to customer profiles has been a method to improve E-mail sales. The challenge has been to identify the audience and tailor the advertising to that audience. Vendor lists of prior customers provide a basis to identify the target audience. Once the customer base is determined, the format of the advertisement is another important factor in increasing sales. Certain sections of the population respond favorably to visual media, e.g., graphics over pure text. This "visual media" group is several times more likely to respond when they receive visual images, then when they receive only pure text. As this "visual media" group represents a significant portion of the E-mail consumers, it is one group to focus on.

Blanket advertising using visual images instead of only text is still ineffective. As the sender, chooses the file format, what the receiver, i.e., user, can process and display may be incompatible. Thus many users, to include members of the "visual media" group, may display the visual images as illegible characters. Thus there is a need to insure the user gets the file format that he/she can display.

## SUMMARY OF THE INVENTION

The present invention describes a method and system for analyzing an E-mail client by an E-mail sensor server. The E-mail client may be, for example, the home personal computer and E-mail client software, Netscape Mail, which a person uses to access his/her E-mail. The E-mail sensor server may be, for example, a merchant's Windows NT server. One goal is to develop a customer base which receives visual advertisements targeted to their interests and that respond by buying the advertised products. The present invention detects through an E-mail sensor message, the file format that an E-mail client can process and display. Thus, those E-mail clients that can display images can receive compatible visual image advertisements and those that cannot receive text. The invention in addition tracks the responses of the E-mail clients to further refine the "visual media" group that responds positively to targeted advertisements with images. Another advantage of the invention is that the invention works with most commercially available E-mail client software, e.g., Qualcomm's Eudora, Novell's Groupwise, Microsoft OutLook/Exchange and Netscape's E-mail client, and hence there is typically no need for users to purchase special software.

Specific embodiments of the method of the present invention include, sending an E-mail message to the E-mail client;

2

determining at the E-mail client a file format that the E-mail client can process and display and indicating to the E-mail sensor server the file format that the E-mail client can process and display. In one specific embodiment, the E-mail sensor server may only determine if a particular file format can be processed and displayed at the E-mail client, and not if it cannot be processed and displayed at the E-mail client. The processing may further determine if hyper text mark up language (HTML) statements or dynamic HTML(DHTML) statements or Java applets can be executed by the E-mail client. Specific embodiments may include the E-mail client executing an HTML image tag having a call to an E-mail server sensor program. The E-mail server sensor program may save information about the E-mail client, such as the E-mail client software type, to an E-mail sensor server database. These embodiments may also include monitoring the status of the E-mail message received at the E-mail client.

These and other embodiments of the present invention, as well as its advantages and features, are described in more detail in conjunction with the text below and the attached figures.

## BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a simplified representation of the E-mail sensor system of the present invention;

FIG. 2 illustrates a simplified block diagram of one specific connection of a specific embodiment of the present invention;

FIG. 3 shows a specific embodiment of the E-mail sensor format;

FIG. 3A shows a simplified display of an E-mail sensor message;

FIG. 3B shows the visual images that results by the user "clicking" on a hyperlink in FIG. 3A;

FIGS. 4 illustrates the process by which an E-mail sensor message is sent to the E-mail client;

FIG. 5 shows the format of the second E-mail sensor message.

FIG. 5A shows an example of a display of the second E-mail sensor message;

FIG. 6 shows a specific embodiment of the present invention of sending a second E-mail sensor message to the E-mail client;

FIG. 7 shows a simplified specific embodiment of the E-mail sensor server.

## DESCRIPTION OF THE SPECIFIC EMBODIMENTS

Specific embodiments of the present invention provide a method and system for determining at the E-mail sensor server, the file format that an E-mail client can display. If the file format allows visual images to be displayed, then the E-mail sensor server may update its sensor database and may send subsequent files of the same format to the E-mail client. In addition, the E-mail client may send to the E-mail sensor server, the type and version of the E-mail client software and status information on the opening and re-openings of the E-mail sensor message. If visual images cannot be displayed, the E-mail sensor server may send only textual messages to the client.

E-mail Sensor System

FIG. 1 shows a simplified representation of the E-mail sensor system of the present invention. The E-mail sensor

server **130** is connected both to the vendor systems through Internet **110** and to the E-mail clients through Internet **140**. Internet **110** and Internet **140** belong to the same global Internet and are separated only for convenience of illustration. The E-mail sensor server **130**, runs the sensor server program which communicates with the E-mail sensor database **132**. E-mail server **134** is an additional server which sends E-mail to and receives E-mail from the E-mail clients. E-mail server **134** is also connected to the E-mail sensor database **132**. In another embodiment E-mail server **134** and E-mail sensor server **130** may run on one computer. Internet **110** connects several vendor systems, for example vendor A system **112**, vendor B system **114**, and vendor C system **115**, to the E-mail sensor server **130**. Internet **140** connects several E-mail clients, for example E-mail clients **142**, **144**, **146**, and **148**, to the E-mail sensor server **130** and Email server **134**. Thus the E-mail sensor server **130** serves as a middleman information collection point between the vendor systems **112**, **114**, **116** and the E-mail clients **142**, **144**, **146**, **148**. This information is stored by the E-mail sensor server **130** in its E-mail sensor database **132**.

In a specific embodiment of the present invention, the vendor's **112**, **114**, and **116** may have their own hypertext markup language (HTML) documents which include visual images. The vendor may for example, have his own World Wide Web home site with the site's web pages containing these HTML documents. Vendor A **112**, for example, would then pass it's HTML document information along with its E-mail client customer list through internet **110**, to the E-mail sensor server **130** which would then store the information in the E-mail sensor database **132**. The E-mail server **134** would access the database **132** and include Vendor A's HTML information in the E-mail sensors sent out to all E-mail clients listed on Vendor A's customer list. For illustration, let this be E-mail clients **142** and **144**. If E-mail client **142** can process and execute HTML statements, E-mail client **142** will send a response back to E-mail sensor server **130**. In a specific embodiment, the response is implemented by the E-mail client software executing a HTML "image tag" statement which calls a program on the E-mail sensor server. The E-mail sensor server may then update its E-mail sensor database **132**.

The E-mail client display includes any hyperlinks to Vendor's A HTML document and any visual images that are the result of executing the HTML statements. For example, an E-mail client by "clicking" on a HTML hyperlink displayed at the E-mail client's computer could then display a vendor's Web page (This is later shown in FIGS. 3A and 3B). Future E-mail messages to E-mail client **142** from E-mail sensor server **130** may contain HTML content including visual images and hyperlink's back to Vendor A. The hyperlinks to Vendor A are all channeled through E-mail sensor server **130**. If, for example, E-mail client **144** cannot process and execute HTML, but only text, then E-mail client **144** may display only the textual information included in the E-mail sensor. Future E-mail messages sent to E-mail client **144** may contain only textual information.

In other alternative embodiments, the information transferred between vendor's **112**, **114**, **116**, E-mail sensor server **130** and E-mail clients **142**, **144**, **146**, **148** may include dynamic hypertext mark up (DHTML) statements and/or Java applets. DHTML is a combination of HTML style sheets and scripts that allows documents to be animated. In these alternative embodiments an E-mail sensor server program may be called by the E-mail client executing a HTML "image tag" statement, in the case of DHTML, or by the E-mail client executing a Java applet triggered by the E-mail

sensor message. The E-mail sensor server program would also receive the E-mail client's software type and version from the E-mail client. This information could be used to tailor the type and version of DHTML and Java that the E-mail client can process and display.

FIG. **2** illustrates a simplified block diagram of one specific connection of a specific embodiment of the present invention. FIG. **2** shows a connection between the E-mail sensor server **130** and the E-mail sensor database **132** through the Internet **140** to one of the E-mail clients **142**. The E-mail sensor server includes a processor **246**, a memory **248**, a network interface **242**, and an input/output interface **244**. These are all connected together through internal bus **250**. The processor **246** may contain one or more central processing units (CPU), for example a Pentium II, a Motorola 68000 or a UltraSparc processor. The memory includes both volatile memory, for example, RAM, and nonvolatile memory, for example, hard disk, and ROM. The input/output interface may include a CRT display, keyboard, and mouse. The network interface **242** connects the E-mail sensor server **130** to the Internet **140** and may include, for example, a modem or an Ethernet card. The E-mail sensor server **130** is connected through connection **252** to its E-mail sensor database **132**. The E-mail sensor database **132** includes information on the E-mail client profile, E-mail client software type and version, whether or not the E-mail client can process and display HTML, and the E-mail client's message status. The sensor database may be a relational SQL database implemented for example on a personal computer or on a UNIX server. Not shown is the E-mail server **134** which has hardware similar to the E-mail sensor server **130**. This is a typical E-mail server **134** with E-mail server software customized to append a unique HTML image tag. The E-mail server is connected to database **132** and Internet **140**.

One specific embodiment of the E-mail client **142** shows a processor **218**, memory **220**, a display **212**, a keyboard and mouse **214**, and a network interface **216** connected together through an internal bus **222**. The network interface **216**, which may for example be a modem, connects the E-mail client **142** to the Internet **140**. The processor **218** may, for example, be a Pentium II. The memory **220** includes both volatile memory, for example RAM, and nonvolatile memory, for example, a hard disk. Thus in this specific embodiment the E-mail client **142** can be represented by a personal computer with a Pentium processor **218** which executes E-mail client software stored in **220**. Examples of E-mail client software are Qualcomm's Eudora, Novell's Groupwise, Microsoft OutLook/Exchange and Netscape's E-mail client.

E-mail Sensor

FIG. **3** shows a specific embodiment of the E-mail sensor format **300**. This format **300** represents an E-mail message which is sent from the E-mail server **134** through the Internet **140** to the E-mail client **142**. The E-mail sensor format **300** includes an E-mail message header **310**, a section of plain text **312**, a section of text in HTML **314**, and an HTML image tag **316**. For example, FIG. **3A** shows a simplified display of an E-mail sensor message that a user may view at his/her Netscape E-mail client on his/her home PC. FIG. **3B** shows the visual images that results by the user "clicking" on the "Tower" hyperlink **320** to the Tower Records home page.

FIG. **4** illustrate the process by which an E-mail sensor is sent to the E-mail clients. In FIG. **4** the vendor identifies a list of its E-mail clients (step **410**). These E-mail clients are stored in the E-mail sensor database **132**. The E-mail server

**134** then sends an E-mail sensor message **300** to the E-mail clients (step **412**). The E-mail sensor message **300** is processed at the E-mail client by hardware similar to E-mail client processor **218** using E-mail client software stored in its memory **220**. The E-mail client software determines if it can process and display HTML (step **416**). If the client is HTML enabled (step **416**) then the plain text part **312** is ignored and the HTML part **314** is displayed at the E-mail client **142** (step **417**). In addition, if the client is HTML enabled (step **416**), then an automatic response is sent to the E-mail sensor server **130** (step **418**). This response is accomplished by the E-mail client software executing the HTML image tag **316**. This image tag is a call to a E-mail sensor server program described below. The E-mail client also passes to the sensor server program information, such as the type and version of the E-mail client's software. At step **420** the E-mail sensor server **130** updates its sensor database **132** with the retrieved information. The next step **430** is to go to node A and will be described with respect to FIG. **6**.

If the client is not HTML enabled (step **416**) then the client reads the plain text (step **422**) and goes to node A (step **430**). The HTML part **314** and the HTML image tag **316** may appear as unreadable marks at the end of the text part **312**. The amount and complexity of the HTML sent in this first Email sensor message is kept small and simple to reduce the unreadable marks. see FIG. **3A**). The E-mail sensor message in this specific embodiment only determines if the E-mail client can process and display HTML and does not determine if only pure text can be processed and displayed. Thus the E-mail sensor server does not get an automatic response telling it that the E-mail client cannot process and display HTML.

Not shown in FIG. **4** or FIG. **6** is that the client is typically given the opportunity through a line of text in the message to unsubscribe from the service. If the client decides to unsubscribe then all further service may be stopped. This line that allows the client to unsubscribe may be present in all E-mail messages sent to the client.

A specific example of the E-mail sensor message of the format of FIG. **3** sent to the E-mail client **142** by the E-mail server **134** is given below:

Return-Path: <Vendor@app1.merchantmail.net>

Received: from app4.merchantmail.net (app4.merchantmail.net [206.79.71.110])
 by ws10.digital-impact.tngi.com (8.8.7/8.8.7) with SMTP id WAA29568
 for <gcapie1@digital-impact.tngi.com>; Mon, Nov. 30, 1998 22:41:51-0800

Date: Mon, Nov. 30, 1998 22:41:51-0800

Message-Id: <199812010641.WAA29568@ws10.digital-impact.tngi.com>

From: Vendor <Vendor@merchantmail.net>

Reply-to: Vendor <Vendor@merchantmail.net>

To: gcapie1@digital-impact.tngi.com

Mime-version: 1.0

Subject: Holiday Music From Vendorrecords.com

Errors-to: Vendor@app1.merchantmail.net

Content-type: multipart/alternative;

boundary="141511991.912494616853.root@app6.merchantmail.net"

X-cid: 10424522

Status:

--141511991.912494616853.root@app6.merchantmail.net

Content-Type: text/plain

Dear Gerardo,

Vendor is in the Holiday mood—come and join us! http://Vendor.m0.net/m/s.asp?H10424522X17652

We have gift ideas and boxed sets for every budget. http://Vendor.m0.net/m/s.asp?H10424522X17690

Here are some holiday hits to get you started:
 Nat King Cole—The Christmas Song
 ** Sale Price: $9.99—You save: $2.00 **
 For more info: http://Vendor.m0.net/m/s.asp?H10424522X17653
 - - -
 Thanks for letting us contact you!

If you wish to UNSUBSCRIBE from future mailings, please go to:

http://Vendor.m0.net/m/u/t.asp?E-mail=gcapie1%40digital-impact.tngi.com.

--141511991.912494616853.root@app6.merchantmail.net

Content—Type: text/html

<HTML><BODY>

Dear Gerardo, <BR>

<BR>

<a href="http://Vendor.m0.net/m/s.asp?H10424522X17651">Vendor</a> is in the <a href="http://Vendor.m0.net/m/s.asp?H10424522X17652">Holiday mood</a>—come and join us!<BR>

<BR>

We have <a href="http://Vendor.m0.net/m/s.asp?H10424522X17690">gift ideas</a> and

<a href="http://Vendor.m0.net/m/s.asp?H10424522X17691">boxed sets</a> for every budget.<BR>

<BR>

Here are some holiday hits to get you started: <BR>

<BR>

<a href="http://Vendor.m0.net/m/s.asp?H10424522X17653">Nat King Cole—The Christmas Song</a><BR>

** Sale Price: $9.99—You save: $2.00**<BR>

- - - <BR>

Thanks for letting us contact you! <BR>

<BR>

If you wish to UNSUBSCRIBE from future mailings, please go to: <BR>

<a href="http://Vendor.m0.net/m/u/t.asp?E-mail=gcapie1% 40digital-impact.tngi.com">http://Vendor.m0.net/m/u/t.asp?E-mail=gcapie1% 40digital-impact.tngi.com</a>. <BR>

<BR><BR>

<img width='1' height='1' src="http://Vendor.m0.net/mm/logopen02.asp?catid=10424522&E-mail=gcapie1@digital-impact.tngi.com" alt=" ">

<BR></BODY></HTML>

--141511991.912494616853.root@app6.merchantmail.net--.

The example above shows a multi-purpose internet mail extensions (MIME) E-mail message having four parts. The HTML displayed at the E-mail client is shown in FIG. **3A**. The first part between "Return-Path" and "Status" is part of the E-mail message header **310**. The "boundary" text line is:

**7**

"--141511991.912494616853.
root@app6.merchantmail.net." and the first "boundary" text
line is between the message header and the second part **312**.
In the first part **310**, the MIME version is given: "MIME_
version: 1.0." MIME allows the E-mail sensor server to send
a variety of data types using E-mail, including sound files,
picture files, textual data, video files, and messages consist-
ing of multiple parts. Both the E-mail server **134** and E-mail
clients, **142**, **144**, **146**, and **148**, should have MIME com-
pliant E-mail software. Most commercial E-mail client soft-
ware packages are MIME compliant. In this specific
example, both the E-mail server **134** and E-mail client **142**
have MIME compliant E-mail software used to process the
E-mail messages. The content-type "multipart/alternative"
means that if the E-mail client **142** can process and display
HTML, then the third part **314** and fourth part **316** of E-mail
message format **300** will be executed and displayed and the
second part **312** is ignored. If the E-mail client **142** cannot
process and display HTML, then the second part **312** is
displayed as human readable textual information and the
third **314** and fourth **316** parts maybe displayed and maybe
unreadable.

The second **312** and third **314** parts are also separated by
the same "boundary" text line and both contain the same
content. The second part **312** in the above E-mail sensor
message example starts with "Content-Type: text/plain", and
ends with the "boundary" text line. The second part **312**
contains ASCII text which can be displayed by most E-mail
clients. The third part **314** starts with "Content-Type: text/
html" and ends with the line before "<img width='1' height=
'1'". The third part **314** is the HTML version of the second
part **312** with additions to include HTML hyperlinks to
vendor documents and visual images.

The fourth part **316** includes a HTML image tag that has
the following format:

    <img width='1' height='1'
    src="http://sensorserver.domain.com/
      sensorserverprogram?
catid=uniqueEmailcode&email=emailaddress">
The line <img width='1' height ='1'" is a small image
square that is displayed by the E-mail client **142** near the end
of the E-mail sensor server's **130**, execution of the "sensor
server program" given in the above HTML line beginning
with "src=". The E-mail sensor sever program, in the
example E-mail sensor message shown above is "mm/
logopen 02.asp".

The E-mail address of each recipient, i.e., "email=," is
included as a parameter in the above HTML "src=" state-
ment along with a unique E-mail code, i.e., "catid=", that is
unique to every E-mail delivered. In the E-mail sensor
message example given above, email=gcapie1@digital-
impact.com, which represents, in this example, the Internet
address of E-mail client **142**, and catid=10424522, which
represents the unique identifier assigned to this E-mail
sensor message. When multiple messages are sent to the
same recipient, the unique E-mail code will change to
identify different messages. In an alternative DHTML
embodiment the HTML image tag would still be present.
Since the image tag passes to the E-mail sensor server the
type and version of the E-mail client software, the type and
version of DHTML that can be processed by the E-mail
client can be determined. Thus the correct type and version
of DHTML can be sent to the E-mail client.

In another alternative embodiment of the HTML image
tag shown above, the E-mail sensor message may include a
Java-related tag in place of the HTML image tag. This Java
"object" tag in the HTML document specifies the applet to

**8**

be run on the E-mail client **142**. When the HTML document
is executed on the E-mail client, the Java "object" tag is
executed and the applet is downloaded from the Uniform
Resource Locator (URL) specified in the tag, for example,
the E-mail sensor server **130**. The applet is executed and a
connection is established back to the E-mail sensor server
**130**. The E-mail client' software type, i.e., "browserType",
and E-mail address, i.e., "urlinfo", is sent from the E-mail
client to the E-mail sensor server. An example applet fol-
lows:

    import java.applet.*;
    import java.awt.*;
    import java.net.*;
    import java.util.*;
    import java.io.*;

```
public class MailSensor extends Applet
{
        static public void main (String argv[ ])
    {
            new MailSensor ( );
    }
    public void MailSensor ( )
    {
            String urlInfo = this.getParameter ("urlInfo");
            String browserType = this.getParameter
            ("browserType");
            urlInfo = urlInfo + "&browserType=" + browserType;
      try {
            URL url = new URL (urlInfo);
            try {
                    url getContent ( )
            }
            catch (IOException e) { ; }
      }
      catch (MalformedURLException e2) { ; }
    }
}
```

Other alternative embodiments which perform the same
functions as the Java applet or HTML image tag could be
written in Active X, VBscripts, or JavaScript.

In a specific embodiment from node A (step **430**), a
second E-mail sensor message is prepared to be sent to the
E-mail client. If the E-mail client can process and display
HTML then FIG. **5** shows the format **500** of the second
E-mail sensor message. FIG. **5A** shows an example of a
display of the second E-mail sensor message of format **500**.
Note that it contains more complicated HTML statements
and visual images than the first E-mail sensor message.
Format **500** is like format **300** of FIG. **3**, except the Message
Header **512** is of "Content-Type: text/html" and Text/Plain
**312** is replaced by a HTML comment **514**. An example of
the message header **512** is:

Received: from gatekeeper.townsend.com
  ([10.1.31.150])
  by mailhub.townsend.com; Wed, Jan. 27, 1999
    18:26:37-0800

Received: from ws23. digital-impact.tngi.com
  (ws23.digital-impact.tngi.com
[207.214.203.120])
  by gatekeeper.townsend.com (2.5 Build 2630
    (Berkeley 8.8.6)/8.8.4) with ESMTP
    id SAA06538 for <kk@townsend.com>; Wed, Jan.
      27, 1999 18:23:32-0800

Received: from app1.merchantmail.net
(nobody@ws23.digital-impact.tngi.com
[207.214.203.120])
   by ws23.digital-impact.tngi.com (8.8.7/8.8.7) with
    SMTP id SAA05227
   for <kk@townsend.com>; Wed, Jan. 27, 1999
    18:24:43-0800
Date: Wed, Jan. 27, 1999 18:24:43-0800
Message-Id: <199901280224.SAA05227@ws23.digital-
   impact.tngi.com>
From: Tower Records <tower@merchantmail.net>
Reply-to: Tower Records <tower@merchantmail.net>
To: kk@townsend.com
Subject: Super Clearance Sale
Errors-to: tower@merchantmail.net
Content-Type: text/html
X-cid: 32578156
Mime-version: 1.0
An example of the HTML comment **514** is:
   <!--<BR>
   * * *

   <BR>
   NOTICE: If you can read this, we are sending you email
    with <BR> the wrong format. Sorry for the inconve-
    nience. To insure that <BR> we get it right, please send
    an email to:<BR>
   <BR>
   nohtml@merchantmail.net<BR>
   <BR>
   You don't have to put anything special anywhere in the
    message. <BR>
   With your help, we can make sure your email is the best
    possible! <BR>
   * * * <BR>
   <BR>
   -->
This HTML comment **514** is not displayed if the E-client can process and display HTML. If the first E-mail sensor was incorrect and the E-mail client cannot process or display HTML, then this comment is displayed. The user at the E-mail client sends an E-mail message to the E-mail server to have subsequent messages sent with only a format similar to the message header **310** and Text/Plain **312**. In an alternative embodiment, the second E-mail sensor message would have format **500** of FIG. **5** without the HTML comment **514**. And thus would assume the first E-mail sensor message was correct in determining that the E-mail client could process and display HTML.

In the specific embodiment, FIG. **6** shows sending a second E-mail message, including the E-mail sensor from the E-mail server **134** to the E-mail client **142**. This second E-mail message is sent after the first E-mail sensor message. All subsequent E-mail messages will go through routine of FIG. **6**. FIG. **6** starts at node A **430** which is a continuation from FIG. **4**. The E-mail server first gets a request for another E-mail message (step **550**). The E-mail server **134** checks the E-mail sensor database **132** to determine if the E-mail client can process and display HTML (step **552**). If the E-mail client is HTML enabled (step **554**) then the E-mail server **134** sends the second E-mail sensor message of format **500** to the E-mail client **142** (step **556**). If the user at the E-mail client display **212** can read the HTML comment **514**, then the user sends an E-mail message to the

E-mail address given in the HTML comment **514** to have subsequent messages sent with only pure text. The E-mail server **134** then updates the database **132** for this E-mail client. Otherwise, the E-mail server will continue to send E-mail messages of format **500**. If the user at the E-mail client display **212** cannot read the HTML comment **514** then the E-mail client **142** displays the HTML part **516** of the E-mail message, and using the HTML image tag **518**, reports the E-mail message status information back to the E-mail sensor server **130**. The HTML image tag contains a call to the sensor server program stored on the E-mail sensor server **130**. The sensor server program then updates the database **132** with, for example, the date and time the E-mail message was opened.

If the E-mail client is not HTML enabled (step **554**) then the E-mail sensor server **130** again sends another E-mail sensor message of format **300** to the E-mail client **142** (step **558**). The E-mail client then displays the plain text part **312** of the E-mail sensor message (step **560**). In this case, although the HTML **314** and HTML image tag **316** portions of the E-mail sensor message may be displayed, they are typically unreadable. The E-mail client may continue to receive in subsequent messages, E-mail sensor messages of format **300**.

E-mail Sensor Server

FIG. **7** shows a simplified specific embodiment of the E-mail sensor server, when the E-mail client is HTML enabled. Upon receiving information from the HTML image tag **316** from the E-mail client **142**, the E-mail sensor server **130** accesses the sensor database **132** (step **610**). The E-mail sensor server **130** gets the type and version of E-mail client's software (step **612**). If the E-mail client's software type is not in the E-mail sensor database **132** (step **614**) then a new record is created in the sensor database **132** (step **616**). This normally occurs when the E-mail sensor first executes the HTML image tag in an HTML enabled E-mail client **142**. The type and version of the E-mail client's software, i.e., "browser-type", is sent as environmental variables from the E-mail client **142** to the E-mail sensor server program. This is then entered into the E-mail sensor database **132**. If the browser type is in the sensor database (step **614**) then the type of browser is associated with the E-mail client's E-mail address (step **618**). Since the E-mail message at the client has been opened, the E-mail sensor server **130** then determines if the E-mail client's message has been opened for the first time (step **622**). This typically occurs when the E-mail sensor is first received by an HTML enabled E-mail client **142** and the HTML image tag executed. If yes, then the E-mail sensor server sets the opening date and time in the E-mail sensor database **132** (step **624**). If the E-mail message has been opened for a second or greater time (step **622**) then an E-mail counter is incremented in the E-mail sensor database **132** (step **626**). This typically occurs every time the E-mail client software re-opens the E-mail message and executes the HTML image tag which again calls the sensor server program on the E-mail sensor server **130** and increments the counter.

In another specific embodiment a JavaScript or Java applet could be used to monitor how long a time interval the E-mail message is open. The JavaScript or Java applet would run on the E-mail client and send to the E-mail sensor server when the E-mail message is opened and when it is closed. This time can then be used to calculate the interval the E-mail message was open and the time can be stored in the database **132**. In another alternative embodiment the JavaScript or Java applet could poll the E-mail sensor server at predetermined intervals as long as the E-mail message is

open. These polling times can then be used to calculate the interval the E-mail message was open and the time can be stored in the database **132**.

As a specific example of the E-mail sensor server program, an example of a Visual Basic Script that runs on the E-mail sensor server **130** executing the flowchart of FIG. **7** is given below: <%@ LANGUAGE="VBSCRIPT"

```
    'disable caching
    Response.Expires=0
%>
<OBJECT RUNAT=Server ID=oConn PROGID=
    "ADODB.Connection"></OBJECT>
<!-- #include file="i_global01.asp" -->
<%
Call Main ( )
'- - -
Sub Main ( )
    Response.ContentType="image/JPG"
    If Request.QueryString("E-mail")=" " Then
        Call LogError("ERROR 534: No E-mail address
            passed:"+
Request.QueryString("E-mail"))
        Exit Sub
    End If
    oConn.open Session ("DIDB_ConnectionString")
    browserType=Request.ServerVariables("HTTP_
        User_Agent")
    If browserType="" Then
browserType="HTML"
    End If
    Set getClientIdCmd=Server.CreateObject
        ("ADODB.Command")
getClientIdCmd.ActiveConnection=oConn
getClientIdCmd.CommandType=1
getClientIdCmd.CommandText="select em_client_id
    from E-mail_clients where name=?"
Set E-mailNameParm=getClientIdCmd.CreateParameter
    ("E-mailparm", 8, 1)
getClientIdCmd.Parameters.Append E-mailNameParm
    getClientIdCmd(0)=browserType
    Set oRs=getClientIdCmd.Execute
    If oRs.EOF Then
        Set            oRs=Server.CreateObject
            ("ADODB.Recordset")
        oRs.Open "E-mail_clients",oConn, 1, 3
        oRs.AddNew
        oRs("name")=browserType
        oRs("html")=1
        oRs.Update
    End if
    E-mailId=oRs.Fields.Item("em_client_id")
    oRs.Close
    'Now update the E-mail address with the E-mail client
        type information
    Set updateE-mailMetaCmd=Server.CreateObject
        ("ADODB.Command")
updateE-mailMetaCmd.ActiveConnection=oConn
updateE-mailMetaCmd.CommandType=1
updateE-mailMetaCmd.CommandText="update
    member_E-mails set em_client_id=
?, modified_on=getdate( ) where E-mail=? and target_
    address=1"
```

```
    Set            clientIdParm=updateE-
        mailMetaCmd.CreateParameter("clientidparm", 2, 1)
updateE-mailMetaCmd.Parameters.Append clientIdParm
    Set            E-mailParm=updateE-
        mailMetaCmd.CreateParameter("E-mailparm", 8, 1)
updateE-mailMetaCmd.Parameters.Append E-mailParm
    updateE-mailMetaCmd(0)=E-mailId
    updateE-mailMetaCmd(1)=Trim(Request.QueryString
        ("E-mail"))
    updateE-mailMetaCmd.Execute
    'Log when the catalog was first opened and how many
        times since
    If Request.QueryString("catid")< >" " Then
        SQLQuery="select opened,opened_count from
            catalogs    where    catalog_id="+
            Request.QueryString("catid")
        Set oRs=oConn.Execute(SQLQuery)
    If Not oRs.EOF Then
        If oRs.Fields("opened").ActualSize=0 Then
            SQLQuery="update catalogs set opened=
                getdate( ),
opened_count=1    where    catalog_id="+
    Request.QueryString("catid")
        Else
            newCount=oRs.Fields("opened_count").Value+1
            If newCount>255 Then newCount=255
            SQLQuery="update catalogs set opened_count="+
CStr(newCount)+"where    catalog_id="+
    Request.QueryString("catid")
        End If
        oRs.Close
        On Error Resume Next
        Set oRs=oConn.Execute(SQLQuery)
        If Err.Number < >0 Then
            Call LogError("ERROR 532: Could not updated
                when
catalog was opened:"+Request.QueryString("catid"))
        End If
        oRs.Close
    Else
        Call LogError("ERROR 533: Could not locate cata-
            log
information:"+Request.QueryString("catid"))
        End If
        oRs.Close
    End If
    Response.BinaryWrite Application("IMG_
        CONTENT")
End Sub
'- - -
%>
```

When the HTML image tag is executed at the E-mail client **142**, a request is made to a computer located at Internet address "sensorserver.domain.com" to run program "sensor server program" with parameters "E-mail address" and "unique mail code". In this specific example, the internet address "sensorserver.domain.com" is for the E-mail sensor server **130**, the "sensor server program" is that given in the visual basic script above, the "E-mail address" is that of E-mail client **142**, and the "unique mail code" is "X-cid: 10424522" as given in the example of the E-mail sensor message above.

The sensor server program in shown above, updates three relational database tables. Table 1 illustrates three simplified sensor database tables that may be stored in E-mail sensor database **132**. The "E-MAIL_CLIENTS" table has the type of the E-mail client software that E-mail client **142** uses to

display the HTML E-mail. This type is stored in "name". The "E-mail address" parameter identifying the E-mail client 142, is stored in "E-mail" in the "MEMBER_E-MAILS" table. In the "CATALOGS" table: the "unique E-mail code" is stored in "catalog-id;" (in the E-mail sensor message example above, catalog id=10424522); the date and time the E-mail client first opens the E-mail message is stored in "opened"; and the count of number of times the E-mail message is opened is stored in "open_count".

client, if the processing determines that the E-mail client can process and display a web page.

6. The method of claim 1 wherein the indicating to the E-mail sensor server comprises the E-mail client executing a HTML image tag comprising a call to an E-mail server sensor program.

7. The method of claim 1 wherein the E-mail message is a multi-purpose internet mail extensions (MIME) compliant E-mail message.

TABLE 1

| CATALOGS TABLE | | | MEMBER_E-MAILS TABLE | | | E-MAIL_CLIENTS TABLE | | |
|---|---|---|---|---|---|---|---|---|
| Column Name | Type | Length | Column Name | Type | Length | Column Name | Type | Length |
| catalog_id | | 4 | member_id | int | 4 | em_client_id | smallint | 2 |
| campaign_id | int | 4 | E-mail | varchar | 250 | name | char | 250 |
| member_id | int | 4 | low_E-mail | varchar | 250 | ms_dhtml | bit | 1 |
| | | | em_client_id | smallint | 2 | ns_dhtml | bit | 1 |
| catalog_type | char | 1 | modified_on | smalldatetime | 4 | inline_images | bit | 1 |
| catalog_url | char | 64 | target_address | bit | 1 | html | bit | 1 |
| status | tinyint | 1 | valid | bit | 1 | browser_based | bit | 1 |
| mailed | datetime | 8 | bounced_cnt | tinyint | 1 | | | |
| opened | datetime | 8 | last_bounced | smalldatetime | 4 | | | |
| opened_count | tinyint | 1 | mp_override | bit | 1 | | | |
| bounced_cnt | tinyint | 1 | domain | varchar | 250 | | | |

Conclusion

In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. Other embodiments will be apparent to those of ordinary skill in the art. For example, the E-mail sensor message format may include DHTML rather than HTML, the Email sensor server program may be written in Java, C++ or Perl rather than Visual Basic script, or the E-mail sensor may include Active X or JavaScript to determine if visual images and text can be displayed. Thus, it is evident that various modifications and changes may be made thereunto without departing from the broader spirit and scope of the invention as set forth in the appended claims and their full scope of equivalents.

What is claimed is:

1. A method for sending and tracking E-mail messages, said method comprising;

sending an E-mail message from an E-mail server to an E-mail client over a public network, said E-mail message including a first portion in a text format and a second portion that includes an image;

determining, at the E-mail client, whether the E-mail client can process and display said image included in said E-mail message;

if the E-mail client can process and display said image, indicating to the E-mail sensor server that the E-mail client can process and display said image and displaying the image on a display coupled to the E-mail client; and

if the E-mail client cannot process and display said image, displaying the text portion of the E-mail message on the display.

2. The method of claim 1 wherein the image included in the second portion is a mark up language statement.

3. The method of claim 1 wherein the E-mail message includes a Java applet.

4. The method of claim 1 wherein the indicating further comprises indicating to the E-mail sensor server if the E-mail client can execute a Java applet.

5. The method of claim 1 further comprising sending a second E-mail message containing a web page to the E-mail

8. The method of claim 7 wherein the E-mail message comprises an E-mail header comprising a multipart-alternative content-type field.

9. The method of claim 8 wherein the text portion of the E-mail message comprises a plain text part and wherein the second portion further comprises a mark up language text part.

10. The method of claim 9 wherein said plain text part and said mark up language text part are both advertisements for a same product.

11. The method of claim 9 wherein said mark up language text part is an HTML statement.

12. The method of claim 9 wherein said mark up language part is a DHTML statement.

13. The method of claim 9 wherein said image is a 1×1 pixel.

14. The method of claim 7 wherein the E-mail message further comprises an HTML image tag.

15. The method of claim 14 wherein the HTML image tag comprises a unique message identifier and the E-mail client's Internet address.

16. The method of claim 1 further comprising monitoring, with the E-mail server, the status of the E-mail message received at the E-mail client.

17. The method of claim 16 wherein the monitoring the status comprises logging the date the E-mail message is first opened at the E-mail client in a database accessible to the E-mail server.

18. The method of claim 16 wherein the monitoring the status comprises monitoring the total number of times the E-mail message is opened by the E-mail client and tracking said number in a database accessible to the E-mail server.

19. The method of claim 1 wherein said image is a 1×1 pixel.

20. The method of claim 1 wherein the public network is the Internet.

21. A method for sending and tracking E-mail messages, said method comprising;

sending an E-mail message from an E-mail server to an E-mail client over a public network, said E-mail message including a mark up language text portion and an

image tag, said image tag comprising an image, a unique message identifier and the E-mail client's E-mail address;

receiving and opening said E-mail message at said E-mail client;

determining, at said E-mail client, whether the E-mail client can process and display said image;

if the E-mail client can process and display said image, sending a response to the E-mail server over the public network, said response including said unique message identifier and the E-mail client's E-mail address;

receiving said response at said E-mail server; and

storing in a database accessible to the E-mail server an indication that said message was received and opened.

22. The method of claim 21 wherein said storing step includes storing said unique message identifier and associating said identifier with said client's E-mail address.

23. The method of claim 22 wherein a response is generated from said E-mail client to said E-mail server each time said E-mail client opens said E-mail message and said storing step further comprises storing a count that tracks the number of times said E-mail client opens said E-mail message.

24. The method of claim 21 wherein said response further comprises a type and version of an E-mail program executing on said E-mail client, and wherein said storing step further includes storing said type and version of said E-mail program in said database.

25. The method of claim 21 wherein said mark up language text part is an HTML statement.

26. The method of claim 21 wherein said mark up language part is a DHTML statement.

27. The method of claim 21 wherein said image in said image tag is an HTML statement.

28. The method of claim 21 wherein said image is a 1×1 pixel.

29. The method of claim 21 wherein the public network is the Internet.

30. A method for sending and tracking E-mail messages, said method comprising;

sending an E-mail message from an E-mail server to an E-mail client over a public network, said E-mail message including a mark up language text portion and a java object tag, said java object tag comprising a unique message identifier and the E-mail client's E-mail address;

receiving and opening said E-mail message at said E-mail client;

determining, at said E-mail client, whether the E-mail client can process said java object tag;

if the E-mail client can process said java object tag, sending a response to the E-mail server over the public network, said response including said unique message identifier and the E-mail client's E-mail address;

receiving said response at said E-mail server; and

storing in a database accessible to the E-mail server, said unique message identifier, the client's E-mail address.

31. The method of claim 30 wherein said mark up language text part is an HTML statement.

32. The method of claim 30 wherein said mark up language part is a DHTML statement.

33. The method of claim 30 wherein the public network is the Internet.

* * * * *