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

INVENTOR: Gerardo J. Capiel, San Francisco, CA (US)

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.

APPL. NO.: 09/240,096

FILED: Jan. 29, 1999

INTERNATIONAL CLASSES .......................... G06F 15/16

UNITED STATES CLASSES ............................ 709/206; 709/203; 709/207; 709/217; 709/218; 709/219; 707/104.1; 707/513

FIELD OF SEARCH ................................. 709/206, 207, 709/217, 218, 219, 203; 707/10, 101, 102, 103, 104.1, 513

REFERENCES CITED

U.S. PATENT DOCUMENTS
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
6,253,231 B1 * 6/2001 Fuji ......................... 709/206

FOREIGN PATENT DOCUMENTS

CITED BY EXAMINER

Primary Examiner—Ayaz Sheikh
Assistant Examiner—Franz B. Jean
Attorney, Agent, or Firm—Townsend and Townsend and Crew LLP

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 may be 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. 2.
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.
What would your Valentine like?
candy? flowers? MUSIC!

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

The '60s Soundtrack
Collective Soul Dosage
Varsity Blues Soundtrack

1999 Grammy Nominees

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

c 1996-1999 MTS, Inc./Tower Records
Send any questions or comments to feedback@towerrecords.com.
Thanks for shopping Tower!
Pricing at towerrecords.com applies for online purchases only. Sale pricing may not apply in Tower retail stores.
START

Identification by Vendor of E-mail Clients

Send E-mail Sensor to Clients

Client HTML Enable?

Yes

Display HTML

Automatic Response to E-mail Sensor Server

Update E-mail Sensor Server Database

No

Client Reads Plain Text

FIG. 4.
FIG. 5.
Dear Valued Customer,

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

Request for Second E-mail Sensor Message

Check Database

Client HTML Enabled?

Yes

Send E-mail Sensor Message (Format 500)

Client Reads HTML Comment?

Yes

Client Affirmatively Responds That Cannot Read HTML

No

Display HTML

End

End

Send E-mail Sensor Message (Format 300)

Client Reads Plain Text

End

FIG. 6.
Start

Connect to DB 610

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

Browser in DB? 614

Yes

Associate in the DB the Type of Browser with Client's Email address 618

No

Create Record in DB 616

Opened for first time? 622

Yes

Set opening date 624

No

Increment email open count 626

End

FIG. 7.
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 telecommunication 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, than 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; 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) 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 system;

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 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 server sensor 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’s 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.

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 server sensor 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 the E-mail 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
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

<!DOCTYPE HTML>
<html><head>
<title>Body</title>
</head>
<body>

Dear Gerardo,

We have a new offer: http://Vendor.m0.net/m/s.asp?H10424522X17690*—gift ideas (A) and
http://Vendor.m0.net/m/s.asp?H10424522X17691*—boxed sets (A) 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:

- - -

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:
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., “browser/Type”, and E-mail address, i.e., “urlInfo”, is sent from the E-mail client to the E-mail sensor server. An example applet follows:

```java
import java.applet.*;
import java.awt.*;
import java.awt.geom.*;
import java.util.*;
import java.io.*;

public class MailSensor extends Applet {
    static public void main(String argv[]) {
        new MailSensor();
    }

    public void MailSensor() {
        // ...
    }

    try {
        URL url = new URL(urlInfo);
        try {
            url.getContent();
        }
        catch (IOException e) { throw e; }

        catch (MalformedURLException e2) { throw 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 ESMT)
  id SAA06538 for <kk@townsend.com>, Wed, Jan. 27, 1999 18:23:32-0800
```

The E-mail address of each recipient, i.e., “email”, is included as a parameter in the above HTML “src” statement 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=geapie1@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
Received: from appl1.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:

<!--
**
**
**
-->

This HTML comment 514 is not displayed if the E-mail client can
process and display HTML. If the first E-mail sensor was
incomplete and the E-mail sensor 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 alter-
native 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.

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 deter-
mines 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 mes-
 sage 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 in-
crements 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
11
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_global101.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"))
End Sub

25
oConn.open Session ("D1DB_ConnectionString")
browserType=Request.ServerVariables("HTTP__
User_Agent")
If browserType="" Then
End If
BrowserType="HTML"
End If
Set gClientldCmd=Server.CreateObject
("ADODB.Command")
GetClientldCmd.ActiveConnection=oConn
GetClientldCmd.CommandType=1
GetClientldCmd.CommandText="select em_client_id
from E-mail_clients where name="
Set E-mailNameParm=GetClientldCmd.CreateParameter
("E-mailparm", 8, 1)
GetClientldCmd.Parameters.Append E-mailNameParm
GetClientldCmdCmd()=browserType
Set ors=GetClientldCmd.Executes
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-mailld=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 E-mailParaEx=updateE-mailMetaCmd.CreateParameter("clientldparm", 2, 1)
updateE-mailMetaCmd.Parameters.Append E-mailParaEx
Set E-mailMetaCmd=updateE-mailMetaCmd.CreateParameter("E-mailparm", 8, 1)
updateE-mailMetaCmd.Parameters.Append E-mailParaEx
updateE-mailMetaCmdCmd()=E-mailld
updateE-mailMetaCmd.Executes
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.Executes(SQLQuery)
If Not ors.EOF Then
If ors.Fields("opened")=opened 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="+CSrr(newCount)+" where catalog__id="+Request.QueryString("catid")
End If
ors.Close
On Error Resume Next
Set ors=oConn.Executes(SQLQuery)
If Err.Number <> 0 Then
Call LogError("ERROR 532: Could not updated
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 "sensorsensor.domain.com" to run program
"sensor server program" with parameters "E-mail address"
and "unique mail code". In this specific example, the internet
address "sensorsensor.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_EMAILS” 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”.

### TABLE 1

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Type</th>
<th>Length</th>
<th>Column Name</th>
<th>Type</th>
<th>Length</th>
<th>Column Name</th>
<th>Type</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>catalog_id</td>
<td>int</td>
<td>4</td>
<td>member_id</td>
<td>int</td>
<td>4</td>
<td>err_client_id</td>
<td>smallint</td>
<td>2</td>
</tr>
<tr>
<td>campaign_id</td>
<td>int</td>
<td>4</td>
<td>E-mail</td>
<td>varchar</td>
<td>250</td>
<td>name</td>
<td>char</td>
<td>250</td>
</tr>
<tr>
<td>member_id</td>
<td>int</td>
<td>4</td>
<td>low_E-mail</td>
<td>varchar</td>
<td>250</td>
<td>ms_dhtml</td>
<td>bit</td>
<td>1</td>
</tr>
<tr>
<td>catalog_type</td>
<td>char</td>
<td>1</td>
<td>ent_client_id</td>
<td>smallint</td>
<td>2</td>
<td>ns_dhtml</td>
<td>bit</td>
<td>1</td>
</tr>
<tr>
<td>catalog_id</td>
<td>char</td>
<td>64</td>
<td>status</td>
<td>tinyint</td>
<td>1</td>
<td>inline_images</td>
<td>bit</td>
<td>1</td>
</tr>
<tr>
<td>mailed</td>
<td>datetime</td>
<td>8</td>
<td>valid</td>
<td>bit</td>
<td>1</td>
<td>browser_based</td>
<td>bit</td>
<td>1</td>
</tr>
<tr>
<td>opened</td>
<td>datetime</td>
<td>8</td>
<td>bounced_ent</td>
<td>tinyint</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>opened_count</td>
<td>tinyint</td>
<td>1</td>
<td>last_bounced</td>
<td>smalldatetime</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bounced_ent</td>
<td>tinyint</td>
<td>1</td>
<td>mp_override</td>
<td>bit</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>domain</td>
<td>varchar</td>
<td>250</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 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 HTML rather than HTML, the E-mail 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 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.

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 1x1 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 1x1 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.