ORIGINAL

IN THE UNITED STATES DISTRICT COURT
NORTHERN DISTRICT OF GEORGIA
ATLANTA DIVISION

DATASCAPE, INC.,
a Georgia Corporation,

Plaintiff,

v.

SAMSUNG ELECTRONICS AMERICA, INC.,
a New York Corporation,
AMAZON.COM, INC.,
a Delaware Corporation,
BEST BUY COMPANY, INC.,
a Minnesota Corporation
CIRCUIT CITY STORES, INC.,
a Virginia Corporation,
OFFICE DEPOT, INC.,
a Delaware Corporation,
SPRINT CORPORATION,
a Delaware Corporation, and
STAPLES, INC.,
a Delaware Corporation,

Defendants.

Civil Action File No.:

1:04-CV-1642

CAP

JURY TRIAL DEMANDED

COMPLAINT

The Plaintiff, Datascape, Inc., states its Complaint as follows:

JURISDICTION AND VENUE

1. This is an action for patent infringement arising under the patent laws of the United States, Title 35, United States Code.

2. This Court has subject matter jurisdiction over all causes of action set forth herein pursuant to 28 U.S.C. §§ 1331 and 1338(a).

3. Venue is proper in this judicial district and division pursuant to 28 U.S.C. §§ 1391(b) and (c).

THE PARTIES

4. Plaintiff, Datascape, Inc. ("Datascape") is a Georgia Corporation having a principal place of business at 8613 Roswell Road, Suite 202, Atlanta, Georgia 30350.

5. Datascape is the owner, by assignment, of all right, title, and interest in and to United States Patent No. 5,742,845 (Exhibit "A"), United States Patent No. 5,905,908 (Exhibit "B"), United States Patent No. 6,366,967 (Exhibit "C"), United States Patent No. 6,684,269 (Exhibit "D"), United States Patent No. 6,694,387 (Exhibit "E"), and United States Patent No. 6,745,259 (Exhibit "F"), including the right to bring suit for patent infringement. The above-listed Datascape patents are hereinafter referred to as the "Datascape patents-in-suit."

6. Upon information and belief, Defendant, Samsung Electronics America, Inc., is a New York Corporation having a principal place of business at 105 Challenger Road, Ridgefield Park, New Jersey 07660-0511. Upon information and belief, the Court has personal jurisdiction over this Defendant.

7. Upon information and belief, Defendant, Amazon.com, Inc., is a Delaware Corporation having a principal place of business in Seattle, Washington. Upon information and belief, the Court has personal jurisdiction over this Defendant.

8. Upon information and belief, Defendant, Best Buy Company, Inc., is a Minnesota Corporation with a principal place of business at 7601 Penn Avenue South, Richfield, Minnesota 55423. Upon information and belief, the Court has personal jurisdiction over this Defendant.

9. Upon information and belief, Defendant, Circuit City Stores, Inc., is a Virginia Corporation having a principal place of business at 9950 Mayland Drive, Richmond, Virginia 23233. Upon information and belief, the Court has personal jurisdiction over this Defendant.

10. Upon information and belief, Defendant, Office Depot, Inc., is a Delaware Corporation having a principal place of business at 2200 Old Germantown Road, Delray Beach, Florida 33445. Upon information and belief, the Court has personal jurisdiction over this Defendant.

11. Upon information and belief, Defendant, Sprint Corporation, is a Delaware Corporation having a principal place of business at 6200 Spring Parkway, Overland Park, Kansas 66251. Upon information and belief, the Court has personal jurisdiction over this Defendant.

12. Upon information and belief, Defendant, Staples, Inc., is a Delaware Corporation having a principal place of business at 500 Staples Drive, Framingham, Massachusetts 01702. Upon information and belief, the Court has personal jurisdiction over this Defendant.

13. All Defendants are hereinafter collectively referred to as the "Defendants."

14. Upon information and belief, the Defendants offer for sale, sell, and/or use Internet-enabled wireless handsets manufactured by Samsung Electronics Company, Limited and/or Samsung Electronics America, Inc. (hereinafter collectively referred to as "Samsung") in the State of Georgia, within this judicial district, and elsewhere throughout the United States.

THE CONTROVERSY

15. The Datascape patents-in-suit are valid and enforceable.

16. The Datascape patents-in-suit have been licensed by Nokia, Inc. for Internet-enabled wireless handsets.

17. The Datascape patents-in-suit have been licensed by the American Express Company for Internet-enabled smart cards, such as the "Blue" smart card.

18. The Datascape patents-in-suit have also been licensed by the Rand McNally & Co. for Internet-enabled personal digital assistants ("PDAs"), such as the TripLink device.

19. The Defendants have in the past and continue to offer for sale, sell, and/or use one or more products and/or processes that infringe one or more claims of each of the Datascape patents-in-suit. For example, the Defendants' offers for sale and/or sales of Samsung Internet-enabled wireless handsets infringe the Datascape patents-in-suit.

20. In particular, the Defendants' offers for sale and sales of one or more of Samsung's Internet-enabled wireless handsets, such as for example model numbers SGH-n625, SGH-s307, SGH-x427, SCH-3500, SCH-6100, SCH-8500, SPH-a400, SPH-a460, SPH-a500, SPH-i300, SPH-i330, SPH-n200, SPH-n240, SPH-n300, SPH-n400, UpRoar, a600, a620/VGA1000, VM-A680, SPH-i500, a660/VI660, SGH-n105, SGH-q105, SGH-r225, SGH-r225m, SGH-e105, SGH-x105, SGH-e715, SCH-a310, SCH-a530, SCH-n150, SCH-n300, SCH-t300, SPH-i700, SCH-a610, and/or SCH-i600 infringe one or more claims of each of the Datascape patents-in-suit.

21. Numerous issues relating to the infringement of the Datascape patents-in-suit are common as to each of the Defendants. For example, such issues include the Court's construction of the patent claims, the application of the construed claims to the accused infringing Samsung products/processes, and the validity and enforceability of the Datascape patents-in-suit. These common issues may be efficiently resolved in a single case.

22. Datascape placed Samsung on notice of its past, present, and future infringement of the Datascape patents-in-suit on August 1, 2002. See Exhibit "G."

COUNT ONE: PATENT INFRINGEMENT

23. Datascape realleges and incorporates herein the allegations of paragraphs 1 through 22 of this Complaint as if fully set forth herein.

24. Upon information and belief, the Defendants have engaged in the offer for sale, sale, and/or use of products and/or processes that constitute direct infringement, contributory infringement, and/or inducement to infringe one or more claims of the Datascape patents-in-suit in violation of 35 U.S.C. §271. The infringing products and/or processes offered for sale, sold, and/or used by the Defendants include, but are not necessarily limited to, Samsung's Internet-enabled wireless handsets.

25. The Defendants' infringement of the Datascape patents-in-suit has been, and continues to be, willful.

26. Datascape has and continues to suffer damages as a direct and proximate result of the Defendants' infringement of the Datascape patents-in-suit and will suffer additional and irreparable damages unless the Defendants are permanently enjoined by this Court from continuing their infringement. Datascape has no adequate remedy at law.

27. Datascape is entitled to: (1) damages adequate to compensate it for the Defendants' infringement, which amounts to, at a minimum, a reasonable royalty; (2) treble damages; (3) its attorneys' fees and costs; and (4) a preliminary and permanent injunction.

PRAYER FOR RELIEF

WHEREFORE, Datascape, Inc. seeks the following relief:

a. That Defendants be ordered to pay damages adequate to compensate Datascape for Defendants' infringement of the Datascape patents-in-suit pursuant to 35 U.S.C. §284;

b. That Defendants be ordered to pay treble damages and attorneys' fees pursuant to 35 U.S.C. §§284 and 285;

c. That Defendants be enjoined from further infringement of the Datascape patents-in-suit pursuant to 35 U.S.C. §283;

d. That Defendants be ordered to pay prejudgment interest;

e. That Defendants be ordered to pay all costs associated with this action; and

f. That Datascape be granted such other and additional relief as the Court deems just and proper.

DEMAND FOR JURY TRIAL

Pursuant to Fed. R. Civ. P. 38(b), Datascape demands a trial by jury of all issues triable of right by a jury.

This 8th day of June, 2004.

Stephen R. Risley
Georgia State Bar No. 606545
J. Scott Culpepper
Georgia State Bar No. 200950
N. Andrew Crain
Georgia State Bar No. 193081
THOMAS, KAYDEN, HORSTEMEYER & RISLEY, L.L.P.
100 Galleria Parkway
Suite 1750
Atlanta, Georgia 30339
Telephone: (770) 933-9500
Facsimile: (770) 951-0933

Attorneys for Plaintiff,
Datascape, Inc.

(00138101)

EXHIBIT / ATTACHMENT

A

(To be scanned in place of tab)

US005742845A

United States Patent [19]

Wagner

[11] Patent Number: 5,742,845

[45] Date of Patent: Apr. 21, 1998

[54] **SYSTEM FOR EXTENDING PRESENT OPEN NETWORK COMMUNICATION PROTOCOLS TO COMMUNICATE WITH NON-STANDARD I/O DEVICES DIRECTLY COUPLED TO AN OPEN NETWORK**

[75] Inventor: **Richard Hiers Wagner**, Dunwoody, Ga.

[73] Assignee: **Datascape, Inc.**, Atlanta, Ga.

[21] Appl. No.: **493,772**

[22] Filed: **Jun. 22, 1995**

[51] Int. Cl.[6] **G06F 13/14**; G06F 13/42

[52] U.S. Cl. **395/831**; 395/500; 395/226

[58] Field of Search 395/831, 500, 395/216, 217, 218, 221, 226, 242, 187.01

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,410,962	10/1983	Daniels et al.	364/900
4,438,511	3/1984	Barna	370/19
4,774,655	9/1988	Kollin et al.	364/200
4,942,552	7/1990	Merrill et al.	364/900
5,036,484	7/1991	McCoy et al.	364/900
5,083,262	1/1992	Haff, Jr.	395/500
5,113,517	5/1992	Beard et al.	364/200
5,119,465	6/1992	Jack et al.	395/500
5,149,945	9/1992	Johnson et al.	235/380
5,161,222	11/1992	Montejo et al.	395/500
5,179,690	1/1993	Ishikawa	395/500
5,212,369	5/1993	Karlisch et al.	235/380
5,239,662	8/1993	Danielson et al.	395/800
5,278,955	1/1994	Forte et al.	395/200
5,278,972	1/1994	Baker et al.	395/500
5,294,782	3/1994	Kumar	235/462
5,297,249	3/1994	Bernstein et al.	395/156
5,367,572	11/1994	Weiss	380/23
5,386,517	1/1995	Sheth et al.	395/275
5,530,844	6/1996	Phillips et al.	395/500
5,548,721	8/1996	Deaslow	395/187.01
5,550,984	8/1996	Gelb	395/200.17
5,577,233	11/1996	Goettelmann et al.	395/500
5,581,708	12/1996	Iijima	395/200.14

OTHER PUBLICATIONS

HTML file entitled: "The Common Gateway Interface", available at httpd@ncsa.uiuc.edu, circa Jun. 1995.

(List continued on next page.)

Primary Examiner—Dinh C. Dung
Assistant Examiner—Rehana Perveen
Attorney, Agent, or Firm—Morris, Manning & Martin, L.L.P.

[57] **ABSTRACT**

An open network system for supporting input/output (I/O) operations for non-standard I/O devices are disclosed. The system includes a server coupled to a plurality of I/O devices through an open network and an extended open system protocol that supports communication with devices that are not personal computers (PCs). These devices include magnetic stripe readers, check readers, smart card readers, credit card terminals, screen phone terminals, PIN pads, printers, and the like. The extended open network protocol includes tags which identify device and input operations and attributes which identify the location, data exchange method, and data variable names for the retrieval, acquisition, and submission of data between the server and I/O devices. Preferably, the open network protocol is implemented in a Hyper Text Transport Protocol (HTTP). Preferably, the system includes a common gateway interface (CGI) at the server which converts protocol statements communicated between the server and I/O devices to application language statements for providing data to an application program coupled to the server. Most preferably, the application statements and protocol statements are constructed in integrated statements with an editor. The editor ensures that data identifiers in the application and protocol statements are compatible. The integrated statements are then parsed by the editor to segregate the protocol statements from the application statements. The protocol statements are downloaded in a file to a client program at an I/O device for processing. The application statements are stored in a file for use by the application. In this manner, generation of the files for client and application processing are automatically done without the user ensuring the correlation of the data fields in the two files.

33 Claims, 25 Drawing Sheets

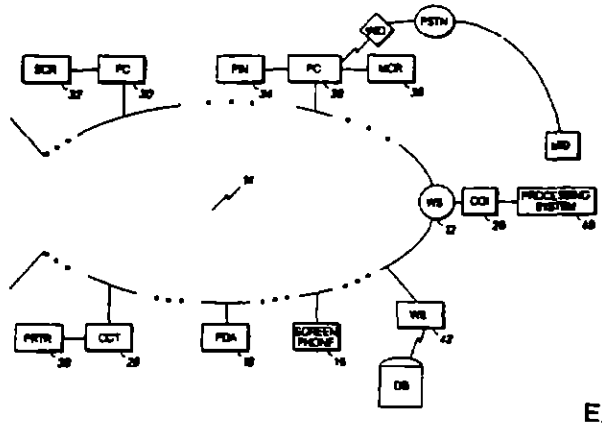

EXHIBIT A
PAGE 1 OF 39

OTHER PUBLICATIONS

HTML file entitiled: "Critique of Secure–HTTP", available at shttp://www.spyglass.com/secure, Apr. 19, 1995.
Directory of abstracts of HTML files available from IEFT Secretariat—Corporation for National Research Initiatives; circa Jun. 1995.
HTML file entitled: "A Beginner's Guide to HTML" available from National Center for Supercomputing Applications/pubs@ncsa.uiuc.edu; circa Jun. 1995.
Memo entitled: Uniform Resource Locators (URL), by URI working group of the Internal Engineering Task Force; Dec. 1994.
Press release entitled: "VeriFone to acquire Enterprise Integration Technologies (EIT), Internet Commerce Software and Services Leader", VeriFone, Inc. and Enterprise Intergration Technologies; Redwood, California; Aug. 21, 1995.
Spyglass white paper entitled: "Electronic Commerce Standards for the WWW", Spyglass, Inc.http://www.spyglass.com; Apr. 1995.
Memo entitled: "The Secure Hypertext Transfer Protocol"; E. Rescorla and A. Schiffman; Jul. 1995.
Article entitled "Cash on the Wirehead" by Andrew Singleton, in the Jun. 1995 issue of BYTE Magazine.
"Beyond the Web: Excavating the Real World Via Mosaic", Goldberg et al. 2nd international WWW, Oct. 1994.

EXHIBIT A
PAGE 2 OF 59

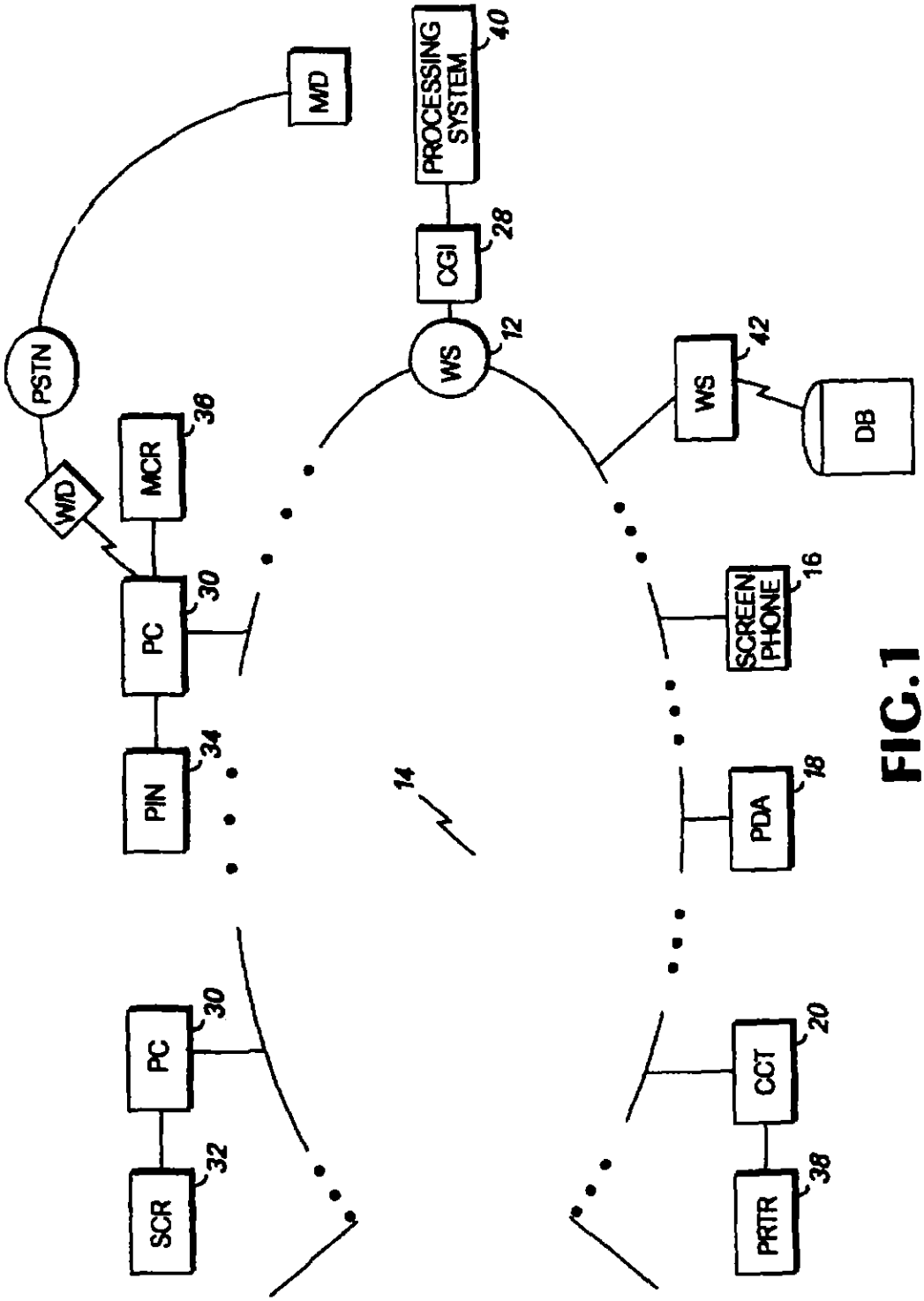

FIG. 1

EXHIBIT A
PAGE 3 OF 39

HTML+D Attributes		Description
<FORM ACTION = "url"		To/From Web Server URL
	FROM "file name"	From Terminal Local File
	TO PRINTER	To Local Printer
	TO "file name"	To Terminal Local File
	FROM SCR	From Smart Card Reader
	TO SCR	To Smart Card Reader
METHOD = "GET"		Retrieve Data
	"POST"	Store Data
	"PAYMENT"	Directive to deliver INPUT data to a private Payment Network for authorization and settlement.
	SQL <database name>	SQL statement database table

Attribute	Value	Description
<INPUT TYPE =	"text"	
	"password"	
	"checkbox"	
	"radio"	
	"submit"	
	"reset"	
NAME =	<field name>	
VALUE =	<initial value>	
CHECKED =		
SIZE =		
MAXLENGTH =		
>		

Attribute	HTML+D Value	Terminal Device
TYPE =	*"MSRT1"*	Mag Stripe Reader - Track 1
	"MSRT2"	Mag Stripe Reader - Track 2
	"KEY"	Terminal Command Keypad
	"PIN"	PIN Pad
	"BCW"	Bar Code Wand
	"MICR"	Check MICR Reader
	"AMT"	Dollar amount key input mask
	"INT"	Integer key input mask
	"LOCAL"	Input from Local Variable
	"AUTOSUBMIT"	Submit FORM to ACTION URL
NAME =	*ip_address*	Local Variable - Terminal's IP Address
	host_phone	Local Variable - Local Internet Access Phone Number
	tid	Local Variable - Terminal ID
	work_key	Local Variable - PIN encryption working key
	datetime	Local Variable - Date and time
	deposit_acct	Local Variable - Merchant Deposit Account

FIG.2

EXHIBIT A
PAGE 4 OF 39

SQL Statements

The following SQL commands represent a subset of the entire command set that varies by database vendor.

HTML+D Attributes	Description
SELECT, field_name,..*	Request field_name (one or many) from a database table
FROM=<table name>,..	Database table name
WHERE=<condition>	Conditional selection of data
name = "constant"	
name LIKE "constant"	
name IN "constant"	
AND	
OR	
ORDER=ASC	Request in ascending order
DESC	..descending
2	..by2's
GROUP=<name>	
INSERT TABLE=<table name>	Insert new data in database table
VALUES="constants"	
UPDATE FROM	
<table name>	Update field_name in database table
SET=field_name="constant"	
[WHERE=<condition>]	Update if WHERE clause is satisfied
DELETE FROM <table name>	Delete all columns that satisfy
[WHERE=<condition>]	WHERE clause
CREATE TABLE<table_name>	Create database table
PRIMARY KEY<name>	

FIG.3

EXHIBIT A
PAGE 5 OF 39

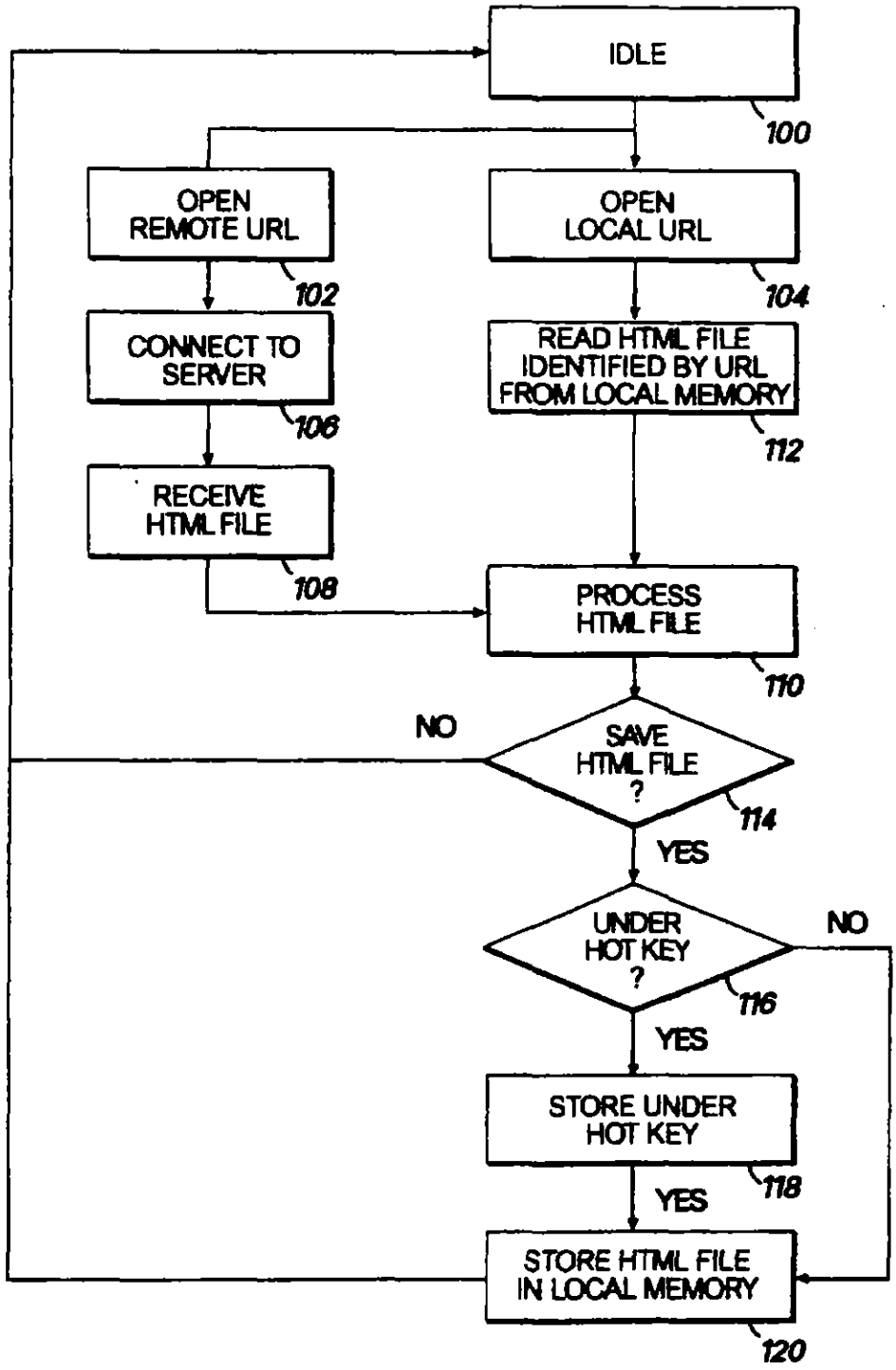

FIG.4

EXHIBIT A
PAGE 6 OF 39

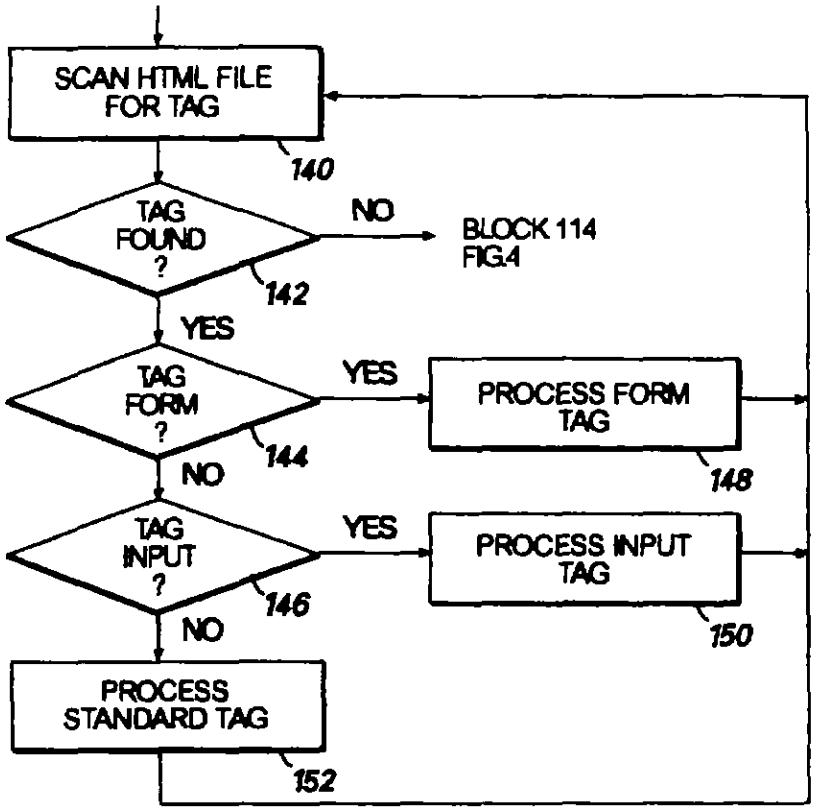

FIG.5

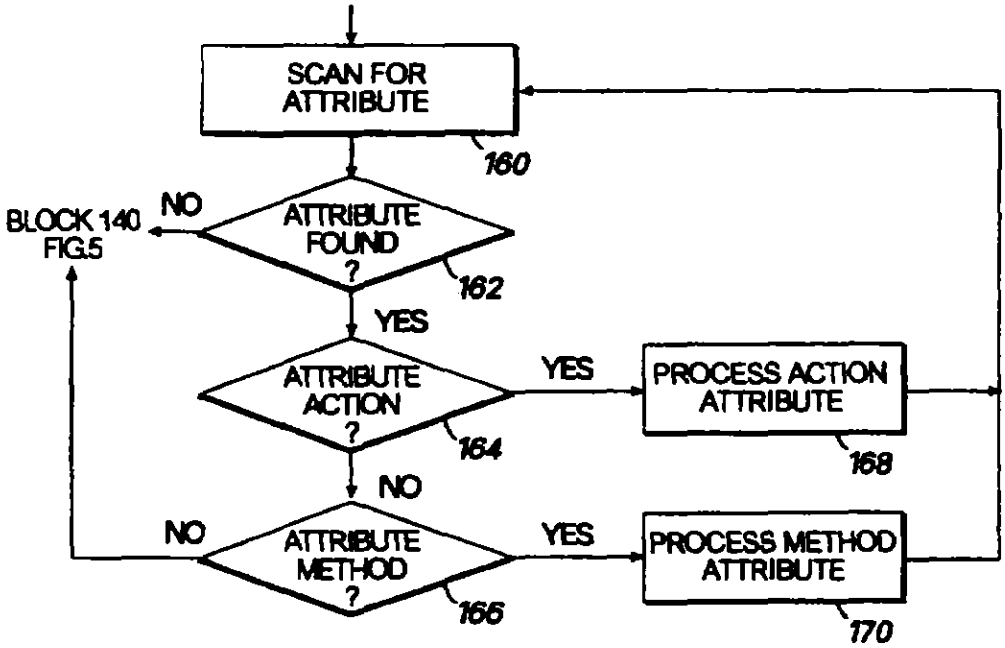

FIG.6

EXHIBIT A
PAGE 7 OF 39

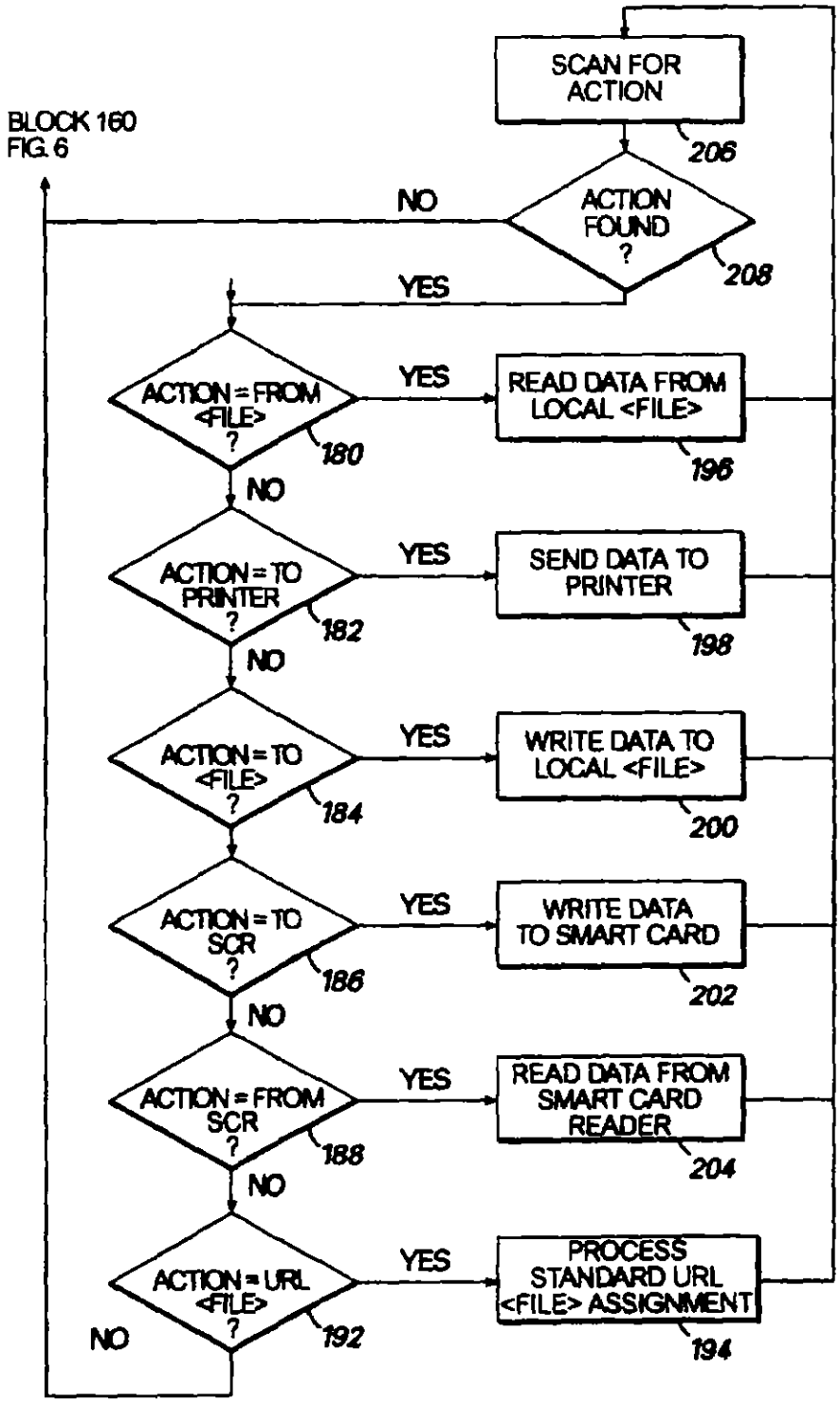

FIG. 7

EXHIBIT A
PAGE 8 OF 39

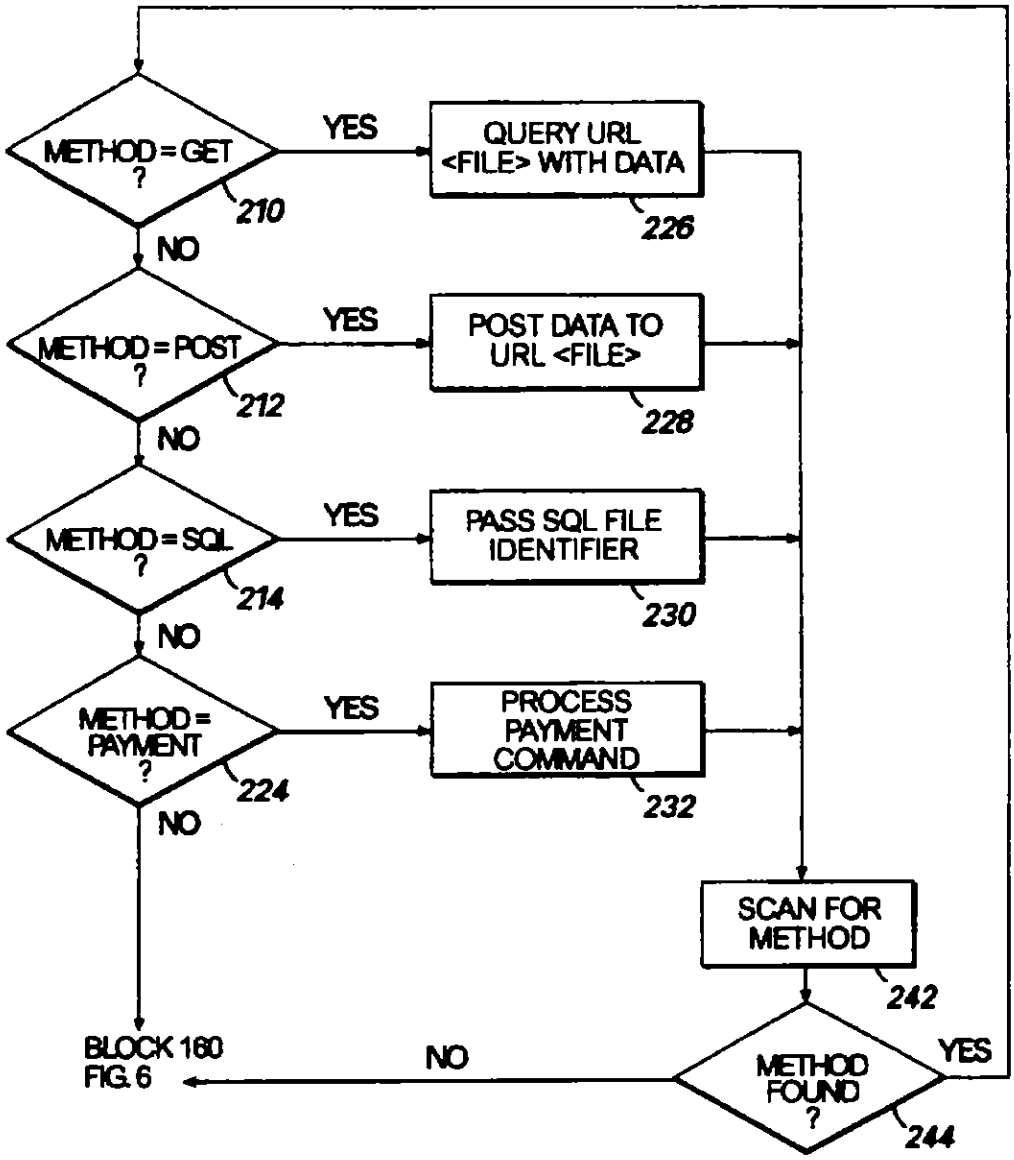

FIG.8

EXHIBIT A
PAGE 9 OF 39

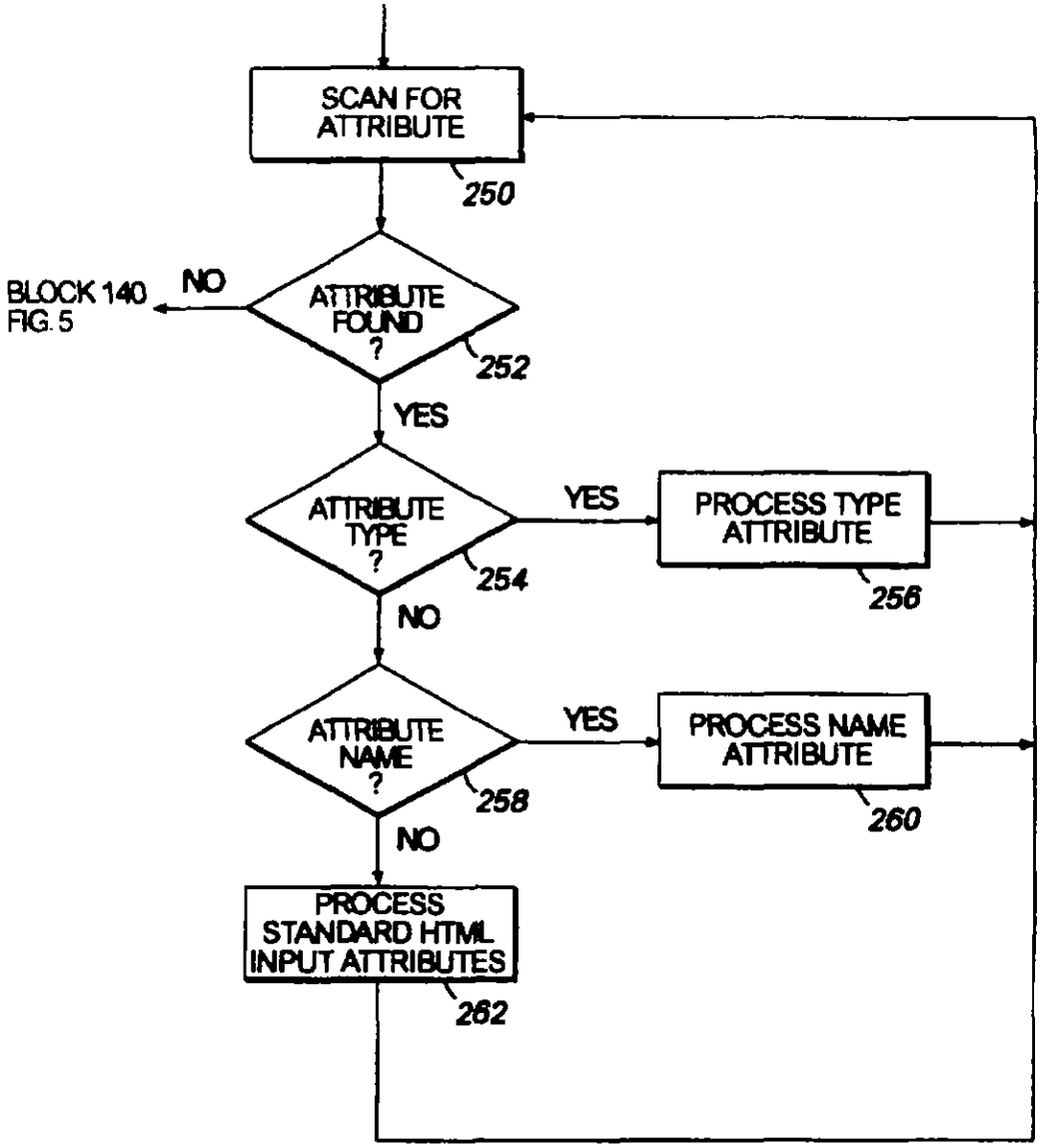

FIG. 9

EXHIBIT A
PAGE 10 OF 39

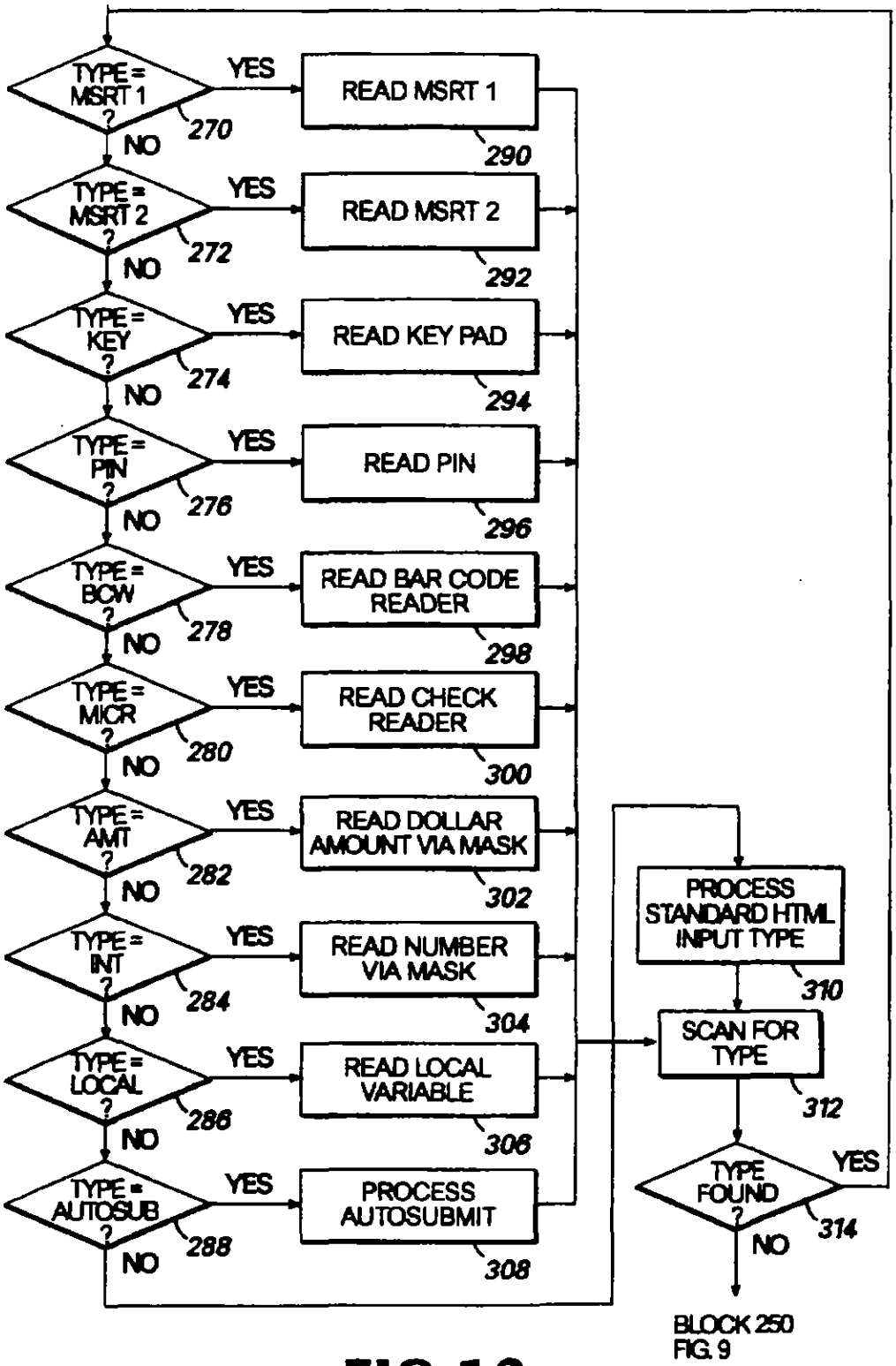

FIG.10

EXHIBIT A
PAGE 11 OF 39

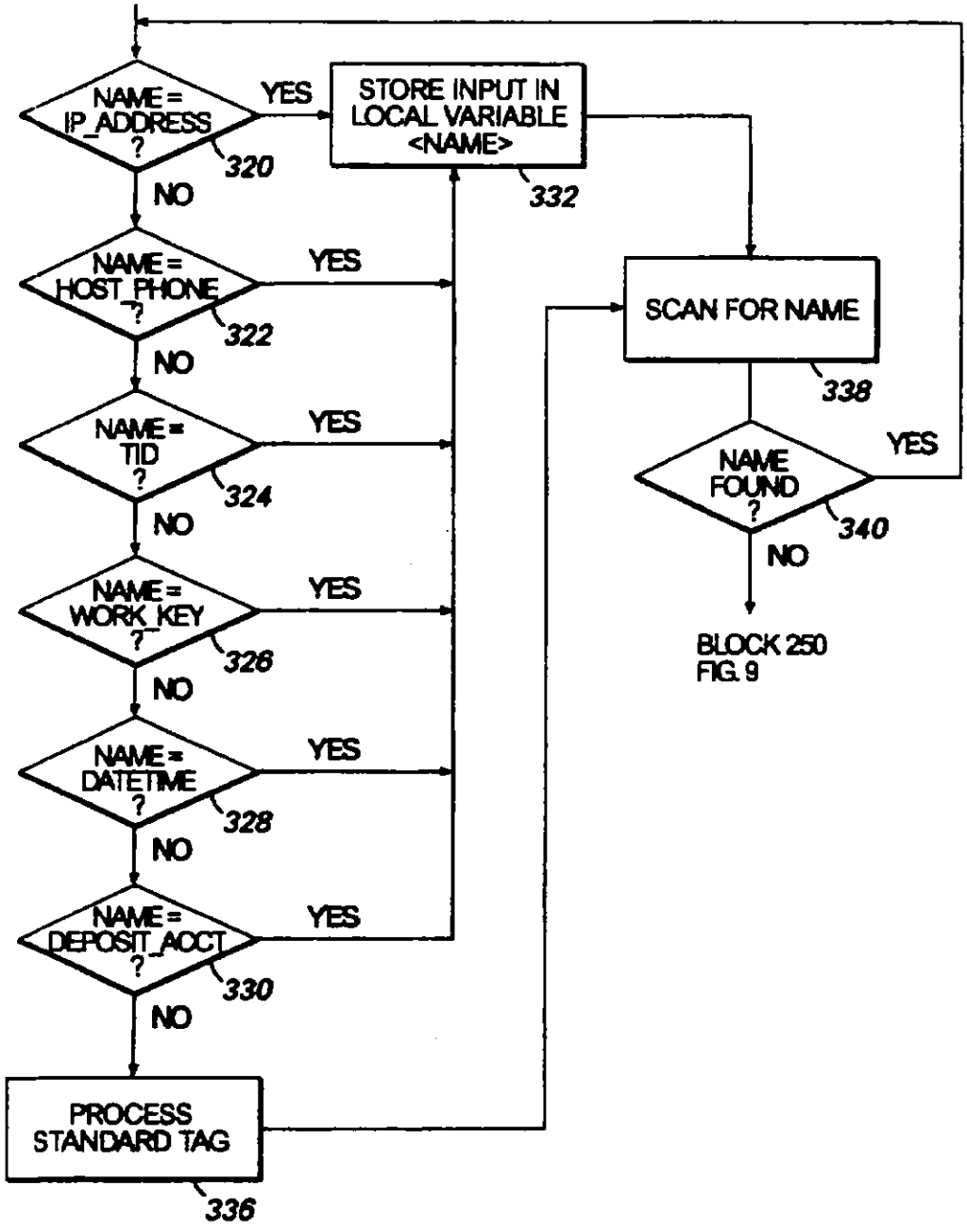

FIG.11

EXHIBIT A
PAGE 12 OF 39

```
1.  <FORM ACTION=URL      METHOD= GET>
2.  <FORM ACTION=URL      METHOD= POST>
3.  <FORM ACTION=URL      METHOD= SQL <database_name>
```

FIG.12

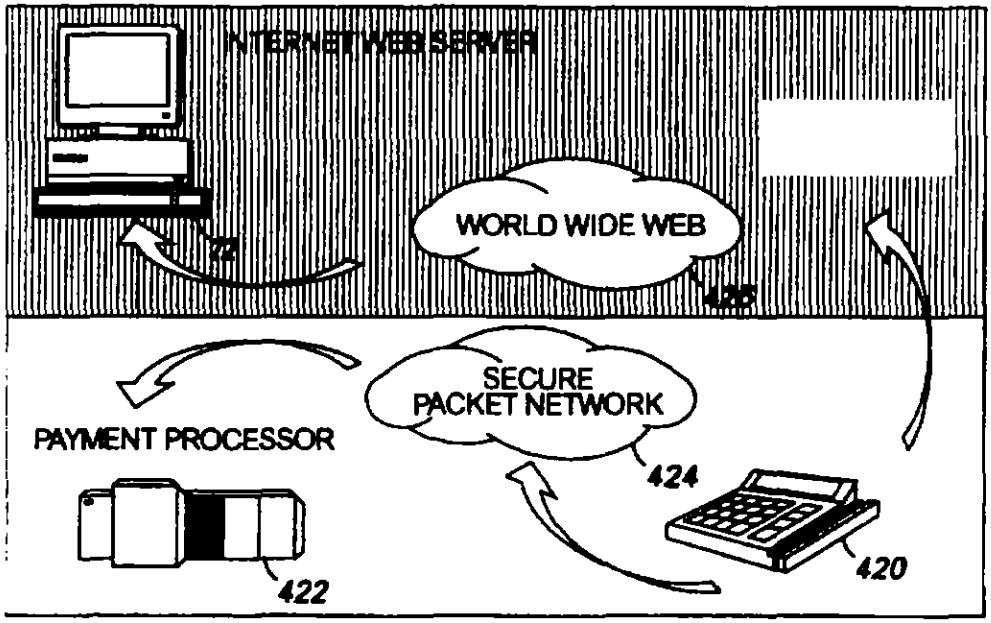

FIG.13A

```
<FORM ACTION=<filename>      METHOD=PAYMENT>
<INPUT TYPE=AUTOSUBMIT>
</FORM>
```

FIG.13B

```
<FORM ACTION=dsinet METHOD=PAYMENT>
<INPUT TYPE=LOCAL NAME=DEPOSIT_ACCT VALUE=1234568902345678 90>
<INPUT TYPE=AUTOSUBMIT>
</FORM>
```

FIG.13C

EXHIBIT A
PAGE 13 OF 39

1.a. Transaction Request HTML+D

500

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
     METHOD=SQL
          "BEGIN TRAN
          IF NOT EXISTS ( SELECT substring(account, 1, 20) FROM auth_table)
               BEGIN
               INSERT TABLE=log_table VALUES=(getdate(),tid, substring (account, 1,20) ,
               substring( account, 22, 4), amount)
               SELECT * FROM log_table WHERE trandate = getdate()
               END
          ELSE SELECT * FROM error_table WHERE error_no=1
          COMMIT TRAN">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER ACCOUNT NUMBER:
<INPUT TYPE="MSRT2" SIZE=40 NAME=account>
ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

1.b. Transaction Accepted Response

510

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
JUNE 1 1995              10:30AM          PURCHASE</P>
TERMINAL ID:                            999999999</P>
ACCOUNT NUMBER          99999999999999999999</P>
EXP DATE:                                   99/99</P>
AMOUNT                                 $9999.99</P>
AUTH NUMBER                             999999999</P>
</p>
- - - - - - - - - - - - - - - - - - - - - - - - </p>
          CUSTOMER SIGNATURE </p>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

1.c. Transaction Declined or Submit Error Response

520

```
<HTML>
<BODY>
DECLINED<MESSAGE>
</BODY>
</HTML>
```

FIG.14

EXHIBIT A

PAGE 14 OF 39

2.a. Transaction Request HTML+D

550

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
METHOD=SQL"BEGIN TRAN
        IF NOT EXISTS ( SELECT substring(account, 1, 20) FROM auth_table)
        BEGIN
                INSERT TABLE=log_table VALUES=(getdate(),tid, substring( account, 1,20),
                substring( account, 22, 4), amount)
        END
        ELSE  BEGIN
                        SELECT * FROM error_table WHERE error_no=1
                        RETURN
                END
        INSERT TABLE=order_table VALUES=( getdate(), cust_name,address,city, state, zip,
                part_code, unit_price, tax, ship_method, ship_chrg, unit_price + tax +
                ship_chrg, substring( account, 1, 20) ,substring( account, 22, 4))
        SELECT * FROM order_table WHERE trandate = getdate()
        COMMIT TRAN">
CUSTOMER NAME:
<INPUT TYPE="TEXT"SIZE=30 NAME=cust_name></p>
ADDRESS:
<INPUT TYPE="TEXT" SIZE=40 NAME=address></p>
CITY:
<INPUT TYPE="TEXT" SIZE=20 NAME=city><p>
STATE:
<INPUT TYPE="TEXT" SIZE=2 NAME=state>
ZIP:
<INPUT TYPE="TEXT" SIZE=10 NAME=address></p>
SCAN PART CODE:
<INPUT TYPE="BCW" SIZE=9 NAME=part_code></p>
ENTER UNIT PRICE:
<INPUT TYPE="AMT" SIZE=8 NAME=unit_price></p>
TAX:
<INPUT TYPE="AMT" SIZE=6 NAME=tax></p>
SHIPPING METHOD:
<INPUT TYPE="TEXT" SIZE=10 NAME=ship_method></p>
SHIPPING AMOUNT:
<INPUT TYPE="AMT" SIZE=6 NAME=ship_chrg></p>
SLIDE CARD:
<INPUT TYPE="MSRTZ" SIZE=40 NAME=account></p>
<INPUT TYPE="SUBMIT" >
</FORM>
</BODY>
</HTML>
```

FIG.15A

EXHIBIT A
PAGE 15 OF 39

2.b. Transaction Accepted Response

555

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
ORDER # 9999999999 APPROVED</p>
JUNE 1 1995          10:30AM         PURCHASE</P>
TERMINAL ID:                         999999999</P>
NAME: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
ADDRESS:</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
CITY: XXXXXXXXXXXXXXXXXXXXXXXXX</p>
STATE: XX                  ZIP:XXXXXXXXXX</p>
ACCOUNT NUMBER:            99999999999999999999</p>
EXP DATE:                                  99/99</p>
PART CODE:                             999999999</p>
UNIT PRICE:                             $9999.99</p>
SHIP METHOD:XXXXXXX CHARGE              $9999.99</p>
TOTAL AMOUNT:                           $9999.99</p>
AUTH NUMBER:                           999999999</p>
</p>

- - - - - - - - - - - - - - - - - - - - - - - - </p>
          CUSTOMER SIGNATURE</p>

</FORM>
</BODY>
</HTML>
```

2.c. Transaction Declined or Submit Error Response

560

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.15B

EXHIBIT A
PAGE 16 OF 39

3.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
METHOD SQL
    "INSERT TABLE=order_table VALUES=( getdate(), cust_name,address,city, state, zip,
        part_code, unit_price, tax, ship_method, ship_chrg,unit_price + tax +
        ship_chrg, substring( account, 1, 20) ,substring( account, 22, 4))
    SELECT * FROM order_table WHERE trandate = getdate()">
<INPUT TYPE="LOCAL" NAME=tid>
CUSTOMER NAME:
<INPUT TYPE="TEXT" SIZE=30 NAME=cust_name></p>
ADDRESS:
<INPUT TYPE="TEXT" SIZE=40 NAME=address></p>
CITY:
<INPUT TYPE="TEXT" SIZE=20 NAME=city><p>
STATE:
<INPUT TYPE="TEXT" SIZE=2 NAME=state>
ZIP:
<INPUT TYPE="TEXT" SIZE=10 NAME=address></p>
ENTER PART CODE:
<INPUT TYPE="TEXT" SIZE=10 NAME=part_code></p>
ENTER UNIT PRICE:
<INPUT TYPE="AMT" SIZE=8 NAME=unit_price></p>
TAX:
<INPUT TYPE="AMT" SIZE=5 NAME=tax></p>
SHIPPING METHOD:
<INPUT TYPE="TEXT" SIZE=10 NAME=ship_method></p>
SHIPPING AMOUNT:
<INPUT TYPE="AMT" SIZE=5 NAME=ship_chrg></p>
<INPUT TYPE="SUBMIT" >
</FORM>
</BODY>
</HTML>
```

FIG.16A

EXHIBIT A
PAGE 17 OF 39

3.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
ORDER # 9999999999 APPROVED</p>
JUNE 1 1995 10:30AM PURCHASE</P>
TERMINAL ID: 99999999</P>
NAME: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
ADDRESS:</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
CITY: XXXXXXXXXXXXXXXXXXXXXXXX</p>
STATE: XX ZIP:XXXXXXXXX</p>
PART CODE: 999999999</p>
UNIT PRICE: $9999.99</p>
SHIP METHOD:XXXXXXX CHARGE: $9999.99</p>
TOTAL AMOUNT: $9999.99</p>
</FORM>
<FORM ACTION=<file_name> METHOD=PAYMENT>
<INPUT TYPE=AUTOSUBMIT>
</FORM>
</BODY>
</HTML>
```

3.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.16B

EXHIBIT A
PAGE 18 OF 39

4.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=SCR1 METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
SLIDE CARD:
<INPUT TYPE="MSRT2" SIZE=40 NAME=track2>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=6 NAME=amount>
<INPUT TYPE="LOCAL" NAME=work_key>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

4.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99                    TIME:9999A</P>
ACCOUNT NUMBER  9999999999999999999</p>
EXP DATE:                              99/99</p>
AMOUNT:                              $9999.99</p>
AUTH NUMBER                        999999999</p>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

4.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.17A

EXHIBIT A
PAGE 19 OF 39

5.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=SCR2 METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
ENTER PIN:
<INPUT TYPE="PASSWORD" SIZE=4 NAME=pin>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="LOCAL" NAME=work_key>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

5.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99                              TIME:9999A</P>
ACCOUNT NUMBER    9999999999999999999</P>
EXP DATE:                                  99/99</P>
AMOUNT                                   $9999.99</P>
AUTH NUMBER                             999999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

5.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.17B

EXHIBIT A
PAGE 20 OF 39

6.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=host_URL METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
SLIDE CARD
<INPUT TYPE="MSRT2" SIZE=40 NAME=track2>
<INPUT TYPE="PIN" SIZE=4 NAME=pin>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

6.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99 TIME:9999A</P>
TERMINAL ID: 999999999</P>
ACCOUNT NUMBER 9999999999999999999</P>
EXP DATE: 99/99</P>
AMOUNT $9999.99</P>
AUTH NUMBER 999999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

6.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
<INPUT TYPE="LOCAL" NAME=work_key VALUE="9999999999999999">
</BODY>
</HTML>
```

FIG.18

EXHIBIT A
PAGE 21 OF 39

7.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
      METHOD=SQL
            "IF EXISTS ( SELECT account FROM check_table)
                  SELECT * FROM check_table WHERE account = DDAaccount
             ELSE SELECT * FROM error_table WHERE error_no=1">
<INPUT TYPE="LOCAL" NAME=tid>
SCAN CHECK:
<INPUT TYPE="MICR" SIZE=20 NAME=account>
ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

7.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99                          TIME:9999</P>
TERMINAL ID:                           99999999</P>
ACCOUNT NUMBER   99999999999999999999</P>
AMOUNT                                 $9999.99</P>
AUTH NUMBER                            99999999</P>
</FORM>
APPROVED:99999999</P>
</BODY>
</HTML>
```

7.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.19

EXHIBIT A
PAGE 22 OF 39

8.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
      METHOD=SQL
            * BEGIN TRAN
              IF NOT EXISTS (SELECT substring(account,51,20) FROM auth_table)
                    BEGIN
                    SELECT cur_bal FROM cust_tbl WHERE substring(account,51,20)=account
                    SELECT amount = amount - (points / .01)
                    SELECT cur_bal = cur_bal + (amount * .01)
                    UPDATE TABLE=cust_Tbl VALUES=(getdate(), account, cur_bal - points)
                    SELECT * FROM log_table WHERE trandate = getdate()
                    INSERT TABLE=log_table VALUES=(getdate(), tid, substring(account,51,20),
                    substring(account,72,4), amount)
                    END
      ELSE SELECT * FROM error_table WHERE error_no="1"
      COMMIT TRAN">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER ACCOUNT NUMBER:
<INPUT TYPE="MSRT1" SIZE=90 NAME=account>
ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
REDEEM POINTS?
<INPUT TYPE="INT" SIZE=6 NAME=points>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

8.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
JUNE 1 1995        10:30AM          PURCHASE<P>
TERMINAL ID:                        999999999<P>
ACCOUNT NUMBER    9999999999999999999<P>
EXP DATE                                99/99<P>
AMOUNT                              $9999.99<P>
AUTH NUMBER                        999999999<P>
</p>
- - - - - - - - - - - - - - - - - - - - - - - </p>
          CUSTOMER SIGNATURE</p>
</p>
THANK YOU!</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
POINTS REDEEMED:                    999999</p>
POINTS EARNED:                      999999</p>
CURRENT POINT BALANCE:              999999</p>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

8.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <MESSAGE>
</BODY>
</HTML>
```

FIG.20

EXHIBIT A
PAGE 23 OF 39

9.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
        METHOD=SQL
                "SELECT fields FROM table WHERE condition">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER SEARCH TABLE NAME:
<INPUT TYPE="TEXT" SIZE=10 NAME=table>
ENTER SEARCH FIELD NAMES:
<INPUT TYPE="TEXT" SIZE=100 NAME=fields>
ENTER SEARCH CONDITION:
<INPUT TYPE="TEXT" SIZE=50 NAME=condition>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

9.b. Transaction Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
FIELD1  FIELD2  FIELD3  ------  FIELDN  </p>
------  ------  ------          ------  </p>
XXXXX   XXXXX   XXXXX           XXXXX   </p>
XXXXX   XXXXX   XXXXX           XXXXX   </p>
.
.
XXXXX   XXXXX   XXXXX           XXXXX   </p>
</FORM>
</BODY>
</HTML>
```

FIG.21

EXHIBIT A
PAGE 24 OF 39

10.a Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
      METHOD=SQL
      * INSERT TABLE=log_table VALUES=( getdate(), tid, gross_sales, opn_chks, voids,
      emp_disc, mgr_disc, vip_card, man_over, coupons, sales_tax, c_dep1, c_dep2,
      c_dep3, c_dep4, chg_fund,cc_dep, batch_no, chrg_sales, paid_outs, co_sales,
      cc_sales, te_sales, gross_sales - opn_chks - voids - emp_disc - mgr_disc - vip_card-
      man_over - coupons - sales_tax, gross_sales - opn_chks - voids -
      emp_disc - mgr_disc - vip_card - man_over - coupons - c_dep1 - c_dep2 -
      c_dep3 - c_dep4 - chg_fund - cc_dep - batch_no - chrg_sales - paid_outs)
      SELECT * FROM log_table WHERE trandate = getdate()">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER GROSS SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=gross_sales>
ENTER OPEN CHECKS:
<INPUT TYPE="INT" SIZE=7 NAME=opn_chks>
ENTER VOIDS:
<INPUT TYPE="INT" SIZE=7 NAME=voids>
ENTER EMP DISCOUNTS:
<INPUT TYPE="INT" SIZE=7 NAME=emp_disc>
ENTER MGR DISCOUNT:
<INPUT TYPE="INT" SIZE=7 NAME=mgr_disc>
ENTER VIP CARD:
<INPUT TYPE="INT" SIZE=7 NAME=vip_card>
ENTER MANUAL OVERRINGS:
<INPUT TYPE="INT" SIZE=7 NAME=man_over>
ENTER COUPONS:
<INPUT TYPE="INT" SIZE=7 NAME=coupons>
ENTER SALES TAX:
<INPUT TYPE="AMT" SIZE=8 NAME=sales_tax>
ENTER CASH DEPOSIT 1:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep1>
ENTER CASH DEPOSIT 2:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep2>
ENTER CASH DEPOSIT 3:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep3>
ENTER CASH DEPOSIT 4:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep4>
ENTER CHANGE FUND:
<INPUT TYPE="AMT" SIZE=8 NAME=chg_fund>
```

FIG.22A

EXHIBIT A
PAGE 25 OF 39

```
ENTER CC DEPOSIT:
<INPUT TYPE="AMT" SIZE=8 NAME=cc_dep>
ENTER BATCH #:
<INPUT TYPE="INT" SIZE=3 NAME=batch_no>
ENTER CHARGE SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=chrg_sales>
ENTER PAID OUTS:
<INPUT TYPE="INT" SIZE=8 NAME=paid_outs>
ENTER CARRY OUT SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=co_sales>
ENTER CREDIT CARD SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=cc_sales>
ENTER TAX EXEMPT SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=te_sales>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

10.b. Transaction Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
JUNE 1 1995        10:30AM         DAILY REPORT</P>
TERMINAL ID:                          99999999</P>
GROSS SALES                          999999.99</P>
 VOIDS                        99      99999.99</P>
 EMP DISCOUNTS                99      99999.99</P>
 MANAGER DISCOUNTS            99      99999.99</P>
 VIP CARD                     99      99999.99</P>
 COUPONS                      99      99999.99</P>
 MANUAL OVERRINGS             99      99999.99</P>
 SALES TAX                           999999.99</P>
 CASH DEPOSIT 1                      999999.99</P>
 CASH DEPOSIT 2                      999999.99</P>
 CASH DEPOSIT 3                      999999.99</P>
 CASH DEPOSIT 4                      999999.99</P>
 CASH DEPOSIT 5                      999999.99</P>
 CHANGE FUND                         999999.99</P>
 CC DEPOSIT                  999     999999.99</P>
 CHARGE SALES                        999999.99</P>
 PAID OUTS                    99      99999.99</P>
 CARRY OUT SALES                     999999.99</P>
 CREDIT CARD SALES                   999999.99</P>
 TAX EXEMPT SALES                    999999.99</P>
 ------------------------------------------</P>

NET SALES                              9999999</P>
OVER/SHORT                             9999999</P>
</FORM>
</BODY>
</HTML>
```

FIG.22B

11.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM  ACTION=MAIL TO: mail_to>
 ENTER MAIL ADDRESS:
<INPUT TYPE="TEXT" SIZE=20 NAME=mail_to>
ENTER MESSAGE:
<INPUT TYPE="TEXT" SIZE=100>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

FIG.23

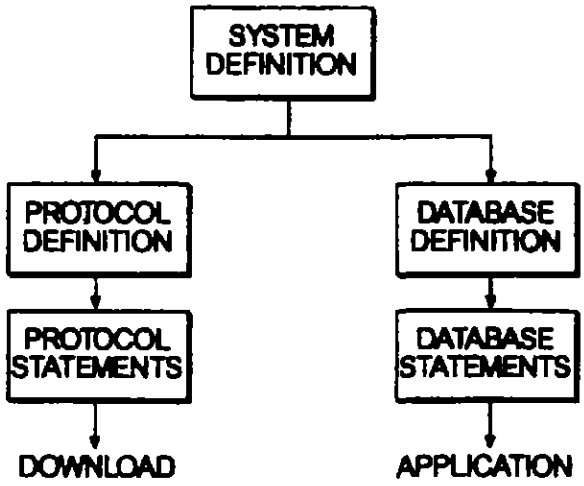

FIG.24A

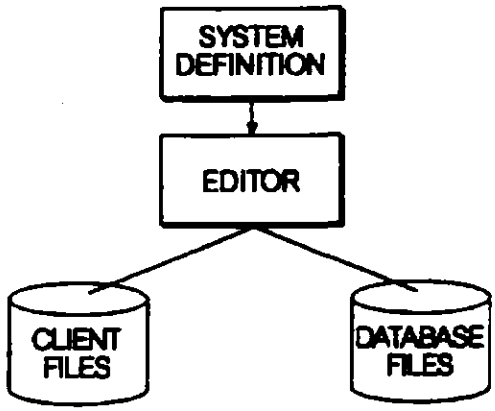

FIG.24B

EXHIBIT A
PAGE 27 OF 39

SYSTEM FOR EXTENDING PRESENT OPEN NETWORK COMMUNICATION PROTOCOLS TO COMMUNICATE WITH NON-STANDARD I/O DEVICES DIRECTLY COUPLED TO AN OPEN NETWORK

FIELD OF THE INVENTION

This invention relates to data transaction systems, and more particularly, to data transaction systems using non-standard input/output devices.

BACKGROUND OF THE INVENTION

Data transaction systems which communicate with a plurality of remote terminals to transfer information used to complete a transaction or compile a database are well known. Typically, such systems include a central transaction processing system which may maintain a database of information such as customer or consumer data. Exemplary information in such a database may include customer identification, customer account numbers, credit limits and/or account balances from which a customer may draw. The central transaction processing system is typically coupled to a plurality of remote transaction or data input terminals. Transaction computers may include special purpose devices such as automatic teller machines (ATMs), point of sale (POS) terminals, credit card terminals, and screen phone terminals. Screen phone terminals are devices which integrate a telephone with an ATM-like device and possibly a magnetic card swipe reader. Data input terminals may include personal computers (PCs) interfaced to data collection devices or special purpose data collection terminals or monitors.

In these known data transaction systems, a user usually initiates a transaction by requesting access to funds in an account or from a credit line maintained by the central processing system. The request is transmitted to the central processing system which performs a verification to determine whether the user is a valid user of the system, has an account within the system, and that the amount of the transaction is within the limits of the consumer's credit line or that the user has the requested funds available in an existing account monitored by the central processing system. The central processing system then transmits authorization for or denial of the transaction to the remote terminal. In response to the message from the central processing system, the remote terminal dispenses cash (for an ATM) or the merchant provides the goods being purchased to the user if the authorization message indicates that the consumer's funds will be transferred to the merchant's account. Similar communication exchanges occur in data systems where electronic documents and other information are provided to a central site for compilation or processing. Consequently, this background discussion applies to all such transaction and data systems. Though the remainder of the discussion is directed to transaction systems, the reader should appreciate that the comments also apply to data systems as well.

The remote terminals may be coupled to the central processing system in several ways. For example, in some ATM systems, the ATMs are coupled to the central processing system through dedicated telephone or other data communication lines. These systems are preferred because they provide a relatively high degree of security since the dedicated data line coupling the central processing system to the ATM is not generally accessible by members of the public. The physical security of the dedicated data line is, however, expensive because no other traffic may utilize the line. Thus, the cost of leasing the dedicated line to an ATM with relatively low volumes of transactions may yield a high communication cost per transaction.

In an effort to reduce the communication cost per transaction, some transaction or data systems utilize telephone lines through a publicly-switched telephone network (PSTN) which may be accessed by other members of the public. Specifically, devices such as credit card terminals and screen phone terminals typically include a modem which converts the digital messages of the remote terminal into frequency modulated analog signals which may be transmitted over telephone lines to a modem at the central processing system. In other systems, the terminal may communicate digital data directly over ISDN lines of the PSTN to the central processing system. This line of communication between a remote terminal and the central processing system is performed by having the remote terminal dial a telephone number associated with the central processing system to establish communication with the central processing system. This type of communication path is relatively secure because the switching networks for the communication traffic through the PSTN are not readily accessible by the public and during the course of the financial transaction, only the central processing system and remote terminal are on the line.

Regardless of the communication method used to couple the central processing system to the remote terminals, the protocol and data formats used between the devices is typically proprietary. That is, the operator of each financial transaction system designs its own protocol and data message format for communication with the processor at the central site or generates a variant within a standard such as those established by the ANSI committee or the like for such communication. As a result, the remote terminals must include software that supports each operator's protocol and message formats in order to be compatible with an operator's central site. For example, application software in a credit terminal such as the TRANZ330, TRANZ380, or OMNI390 manufactured by VeriFone implement one or more of the communication protocols and formats for National Data Corporation (NDC), VISANET, MASTERCARD, BUYPASS, and National Bancard Corporation (NaBANCO) system processors in order to support transactions with the most popular transaction centers. Thus, the communication software absorbs a significant amount of terminal resources which could be used to support other terminal operations.

A related problem arises from the expanding home banking market. A customer of home banking system typically uses a screen phone terminal or a personal computer (PC) having a modem to establish communication through a PSTN to a central transaction processing system. Again, the operator of the central processing system must provide information regarding the data message formats for communicating with the central processing system to a vendor of software for the home banking terminals or must provide that software to its customers. As a result, home banking customers must purchase software to communicate with each banking system of which the customer wants to be a member. This cost and the need to install additional communication programs may make some consumers reluctant to be a member of more than one banking system or to change banking systems.

A communication system becoming increasingly popular and which provides standardized communication is the Internet. The Internet is an open network of networks which communicate through a variety of physical communication

EXHIBIT A
PAGE 28 OF 39

devices such as telephone lines, direct communication lines, and the like. Each network is coupled to the main Internet network for communication through a host computer supporting a TCP/IP router or bridger. The host computer typically includes a program, frequently called a Web server, which acts as a gateway to resources at the host computer which may be resident on the host computer or a network coupled to the host computer. Each server has an address identifying the location of the resources available through the Web server. The router recognizes communication for the server and directs the message to the server or it recognizes that the communication should be forwarded to another server. As a result, communication within the Internet may be point-to-point, but more likely, the communication path is a somewhat circuitous one with the information passing through the routers of multiple servers before reaching its final destination.

A number of message protocols and formats have been developed for the Internet. The physical communication protocol and data message format is the Transport Control Protocol/Internet Protocol (TCP/IP). The TCP/IP protocol involves multiple layers of encapsulating headers containing communication information which are used to provide byte streams or datagram communications to computers on the networks coupled to the Internet. Encapsulated within TCP/IP headers are protocols which are used to format the data messages or transfer data from one computer to another computer coupled to the Internet. These protocols include File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), Telnet, and Hyper Text Transport Protocol (HTTP). The advantage of these protocols is that each provides a standardized communication format for transferring information between computers on the Internet. These protocols are typically called open system protocols as they are publicly known and may be utilized by any programmer to develop programs for communicating with another computer coupled to the Internet. These non-proprietary protocols have contributed to the acceptance of using the Internet as an open network for coupling computer networks together.

While the Internet provides an open network for computer communication with publicly accessible protocols and formats, the Internet suffers from a number of limitations which preclude its effective use as a transaction or data system which uses non-standard I/O terminals and devices. First, circuitous communication presents a number of security issues for such a system. For example, a Web server could incorporate a router which examines the address of each message coming through it and upon recognizing an address associated with a central transaction processing system, copy the data message for the unauthorized retrieval of customer-sensitive information such as account numbers and personal identification numbers (PINs) which may be contained in the message.

A second limitation of open networks such as the Internet is that communication on such networks is only supported for computers acting as servers or clients. Specifically, all of the protocols and formats are constructed for standard input/output (I/O) operations for a PC terminal. That is, text information is directed to a standard monitor screen, user input is expected from a standard keyboard, and files are transferred to standard peripherals such as a hard disk or diskette drive. Especially absent is the ability in open network protocols for communication with devices that only use communication interfaces such as RS-232C. As a result, communication over the Internet is primarily performed with standard PCs through network communication methods and interfaces.

This presents a number of problems for home banking or for interfacing non-standard I/O terminals such as credit card terminals or screen phones to open networks such as the Internet either directly or through a PC. Generally, non-standard I/O devices are devices which interface to a PC through a port not normally used for networks, such as a RS-232C port, or are devices which have limited input and output capabilities such as small screen displays or ten keypads. These devices are not supported on the Internet because servers use protocols that communicate with PCs supporting standard QWERTY keyboards and standard monitors. Consequently, users are limited to entering account numbers and the like through a keyboard of a PC-like device for processing at a central transaction processing system. To request a transaction, one need only have a person's credit card account number. If the credit card number had to be input through a magnetic card reader, unauthorized access to a customer's account would be less likely since physical possession of the credit card would be required to initiate the transaction.

Another limitation of the standard I/O devices currently supported by the open network protocols is the lack of encryption. For example, PIN pads, which are typically incorporated in ATMs, automatically encrypt in hardware a PIN entered by a user. Such devices typically encrypt the number by implementing a data encryption standard (DES) algorithm in hardware before the PIN is transmitted or stored. When a standard keyboard is used to input the PIN, no hardware encryption is performed and, as a result, an unencrypted copy of the PIN is provided to the memory of the PC. Storage of unencrypted PINs is in contravention of current banking regulations. If PIN pads could be read via Internet protocols, then such a lapse in PIN security would be less likely to occur.

Another I/O device not supported on open networks are smart cards which are increasing in use. Smart cards include a processor and memory in which information regarding the amount of funds in a particular account, a transaction history, account numbers, and customer data may be stored. The card may be read through a smart card reader which is a computer having a processor and memory but usually provided with non-QWERTY keypads and limited displays. A transaction processor may validate a card owner through a PIN provided through a keypad, determine the mount of money remaining on the card and debit the card itself for a transaction amount by communicating with the smart card reader with one of the proprietary protocols discussed above. Such information is not readily obtainable by the owner of the card and so cannot be entered through a keyboard or the like. Smart card readers are non-standard devices which may be coupled to a PC through a COMM1 or COMM2 port. However, none of the standard protocols and message formats for open network communications currently provide I/O operations for such devices.

All systems which attempt to provide three party communication to execute an electronic transaction suffer from a number of limitations which present risks greater than those in a normal transaction performed at the point of sale. In a typical point of sale (POS) transaction, the consumer hands a debit or credit card to a merchant's agent who may examine the card for security markings such as holograms, watermarks, or a cardholder signature. The agent then places the card into a reader for acquiring information from the card and, in some cases, have the consumer enter a PIN into a PIN entry device which encrypts the PIN in a hardware implemented scheme. If the PIN is entered, it is transmitted with the information from the card to a processing center, typi-

cally in one of the formats discussed above, under a X0.25 protocol or the like. The processing center returns an authorization granted or denied message. The reader typically has a printer coupled to it through an RS-232C port or the like and a purchase agreement is printed. The consumer signs the agreement, the merchant's agent may verify the signature, and the merchant retains an original of the agreement and the consumer a copy. In this scenario, the merchant has initiated the communication to the processing center. The safeguards noted above permit the processing center to charge a merchant a lower processing fee than when a consumer initiates a transaction. Consumer initiated transactions present a greater risk because the consumer provides an agent an account number in a telephone conversation or non-encrypted DTMF transmission. Thus, there is no card inspection, signature verification, or PIN verification. As a result, such transactions are limited to credit cards because debit cards require that the cardholder be present to enter a PIN into an appropriate PIN entry device.

What is needed is a system that permits consumers remote from a merchant to order goods and present payment in a secured manner so the merchant's risk and processing costs, as well as a cardholder's exposure to fraud, is reduced. What is needed is a way for a processing center to communicate through an open network with non-standard I/O devices such as credit card terminals, personal digital assistants, and screen phone terminals or with non-standard I/O devices coupled to the open network through a PC or the like. What is needed is a transaction or data system which utilizes an open network such as the Internet to support electronic transactions or data compilation in a secure manner without undue limitation as to the devices with which communication may be made.

SUMMARY OF THE INVENTION

The present invention provides transaction and data systems which may be implemented on an open network such as the Internet. The system comprises a server for communicating in an open network protocol and a plurality of input/output (I/O) devices coupled to the server through an open network, the I/O devices communicating with the server in the extended open network protocol that supports communication with non-standard I/O devices over the open network. The system of the present invention provides a server with the capability of communicating with a number of I/O devices useful in transaction and data systems which heretofore have been unsupported on an open network system such as the Internet.

The system of the present invention is implemented by extending present open network communication protocols and data message formats to communicate with non-standard I/O devices either coupled to an open network as a client or coupled to an open network through a client, such as a PC, credit card terminal, screen phone, or PDA. That is, commands which are compatible with the communication scheme of a presently-implemented protocol for the Internet are used and additions are made to commands implemented within the control structure of that existing protocol to support non-standard I/O device communication. At the server, the extended protocol is further supported by a common gateway interface (CGI) which converts the communication from a non-standard I/O device to a format which is compatible with a transaction or data application program which may be executed on the server or a computer coupled to the server. In this manner, the CGI permits the processing of the extended capability commands to be segregated from the communication functions performed by the server.

Preferably, the server and the I/O devices communicate through an Internet protocol and most preferably, the Hyper Text Transport Protocol (HTTP), to exchange data between an application program and non-standard I/O devices over an open network. Although HTTP is the preferred protocol used to implement the present invention, other protocols such as Telnet or SMTP, for example, may also be extended in a similar manner. Specifically, the HTTP protocol is expanded to communicate with printers, magnetic card readers, credit card terminals, smart card readers, check readers, PIN pads, bar-code readers, PDAs, or the like, and includes a command which instructs a non-standard I/O device to disconnect from the open network and re-couple to a transaction processing system to transfer funds from a consumer account to a merchant account through a PSTN or dedicated data line. By using these extended capability commands within HTTP, a processing system may operate on an open network such as the Internet and communicate with transaction or other data I/O devices which have not previously been able to couple to such open networks. Such a system may be used to execute a transaction between a consumer and a merchant so the merchant receives remittance information in a timely manner. The system permits the consumer to initiate a transaction and order from a merchant and then use a more secure link supported by PIN entry devices or the like to reduce the risk of fraud for the transaction.

Because the server may communicate through such open networks with non-standard I/O devices, the transaction or data processing system is available for the ever-expanding market available through the Internet. Such a system is able to communicate with non-standard I/O devices in myriad locations such as retail establishments or in consumers' homes. For example, a consumer may utilize the standard capability of an Internet protocol to communicate with a server that provides information regarding services or goods for sale over the Internet and then consummate a sales transaction by using the extended capability of the Internet protocol. Such a home consumer could provide transaction data through a smart card reader coupled to a COMM1 or COMM2 port of a PC. A database program executing at the server for the central processing site may accept product ordering information from a non-standard keypad or touch screen associated with a screen phone terminal at the remote site and then communicate with the smart card reader to consummate the transaction. Such a transaction system requires that the consumer have physical possession of the smart or credit card and not simply knowledge of the account number. Likewise, the server would be able to communicate with a PIN pad or the like to ensure the hardware encryption of PINs and other data before it is transmitted to the server site. Such a system is less susceptible to consumer fraud.

Another feature of the present invention is a PAYMENT command implemented in the extended Internet protocol that directs a non-standard I/O device or a PC interfaced with such devices to communicate with a transaction processor through an alternative communication link. In one form, the PAYMENT command is used by a merchant terminal to submit a consumer's account number with a merchant deposit account number through a PSTN network or the like to the processing center. In another form of the PAYMENT command, a client program in a consumer's terminal receives an account number for a merchant account from a merchant's server with the PAYMENT command. On receipt of this command, the client program suspends its operation and passes the account number to a conventional

bank processing program co-resident in memory. The bank processing program establishes a standard communication link with a transaction processing system through a dedicated data line or a PSTN network. Using that communication link, the bank processing program executes a commercial transaction using a standard VISA protocol or the like. The consumer may use a magnetic stripe reader and a PIN entry device to improve the security of the data transmission. The transaction center may transmit remittance data over the open network to the merchant so the merchant is apprised of payment and ships the ordered product. Once this consumer initiated transaction is complete, the bank processing program terminates and returns control to the client program which may terminate communication with the open network or retrieve information from another server on the open network for another transaction. In this way, the user may use the open network for non-confidential communication such as collecting product information, pricing, and product availability. This information may be collected quickly and efficiently using the extended Internet protocol. The conventional bank processing program and more secure communication links may then be used for the confidential information required for the transaction. Thus, the present invention is able to combine the features and advantages of the Internet with the more secure communication link and data security enhancing devices of systems presently known.

Preferably, an editor is provided which permits a user to define an application database table with data fields, define client application data fields, and define the integrated forms for communicating data between the defined database tables and a client application. The editor verifies the syntax of the user generated integrated forms containing extended Internet protocol statements and client application statements. The editor ensures that the variable names for the client application and the data fields for the database application correspond. Following the generation of the integrated form, the editor parses the integrated form to segregate the database language statements from the extended Internet protocol statements. A database language identifier is substituted in the Internet protocol statements for the database statements contained in the integrated form. The Internet protocol statements are downloaded as a file which is interpreted by the client program for the collection and submission of data from non-standard I/O devices to the database application. The database language statements segregated from the extended Internet protocol statements are placed in a second file which is named to correspond to the database table defined by the user. The CGI application recognizes the database language identifier contained in the returned forms of the Internet protocol statements. The CGI application correlates the database identifier with the file previously generated by the editor which contains the database command statements. The application then inserts the data from the returned form into the database command statements and provides the re-integrated database command statements to the database application. In this manner, the database may be queried by or retrieve data from the non-standard I/O device. In the most preferred embodiment, the editor permits a user to develop integrated forms comprised of the extended HTML language and standard query language (SQL) database application statements. In this manner, the user does not have to manually generate the SQL commands, the HTML commands, and carefully correlate the data fields of the two commands in order to implement a transaction between a client and a database.

These and other advantages and features of the present invention may be discerned from reviewing the accompanying drawings and the detailed description of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention may take form in various components and arrangement of components and in various steps and arrangement of steps. The drawings are only for purposes of illustrating a preferred embodiment and are not to be construed as limiting the invention.

FIG. 1 is a diagram of an open network system in which the present invention is utilized;

FIG. 2 is a diagram of the format of the FORM and INPUT tags implemented in the preferred embodiment of the present invention;

FIG. 3 is a diagram of the preferred SQL commands supported in the preferred embodiment of the present invention;

FIG. 4 is a flowchart of the high level processing of the client program which interprets the HTML files of the preferred embodiment of the present invention;

FIG. 5 is a flowchart of the HTML file processing performed by the client program of the preferred embodiment of the present invention;

FIG. 6 is a flowchart of the attribute processing for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. 7 is a flowchart of the processing of the ACTION attribute for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. 8 is a flowchart of the processing for the METHOD attribute for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. 9 is a flowchart of the attribute processing for the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. 10 is a flowchart of the processing for the TYPE attribute for the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. 11 is a flowchart of the processing for the NAME attribute of the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. 12 is a diagram of the format for the ACTION attribute for the FORM tag performed by the common gateway interface between the Web server and an application program;

FIG. 13A is a diagram of the possible communication paths which may be used by an I/O device according to the principles of the present invention;

FIG. 13B shows an exemplary FORM tag and INPUT tag for the PAYMENT method implemented in a merchant's terminal according to the principles of the present invention;

FIG. 13C shows an exemplary FORM tag and INPUT tag for the PAYMENT method implemented in a consumer's terminal according to the principles of the present invention;

FIG. 14 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a card initiated payment authorization and capture transaction;

FIG. 15 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a bar code reader input with card-initiated payment authorization transaction;

FIG. 16 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to

generate the HTML files for the client program and the SQL files for the application program for a key input order with secure payment transaction;

FIG. 17A shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a smart card debit (Type 1) transaction;

FIG. 17B shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a smart card debit (Type 2) transaction;

FIG. 18 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a debit card transaction;

FIG. 19 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a check verification transaction;

FIG. 20 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a customer frequency transaction;

FIG. 21 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for an item search transaction;

FIG. 22 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for retail store end of day reporting;

FIG. 23 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a store reporting an e-mail transaction;

FIG. 24A is a diagram of a manual development process for the files interpreted by the client program and the files interpreted by the application program in accordance with the principles of the present invention; and

FIG. 24B is a diagram of the generation of the files interpreted by the client program and the files interpreted by application program performed by an editor constructed in accordance with the principles of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

A transaction or data system constructed in accordance with the principles of the present invention is shown in FIG. 1. The system 10 includes a Web server 12 which is coupled to an open network 14 such as the Internet for communication with various I/O devices and terminals. For example, the I/O devices which may be coupled directly to network 14 include standard I/O devices already supported by Internet protocols such as PCs 30 and non-standard I/O devices such as a screen phone terminal 16, a personal digital assistant (PDA) 18, and a credit card terminal 20. Other exemplary non-standard I/O devices such as smart card reader 32, personal identification number (PIN) pad 34, magnetic card swipe reader 36, printer 38, or the like, may be coupled to PCs through non-standard I/O ports such as COMM1 and COMM2 ports or to other non-standard I/O devices such as phone terminal 16, PDA 18, or credit card terminal 20. Typically, these devices are coupled to PCs or devices 16,18, or 20 through an interface such as a RS-232C interface. Merchants or other vendors may use a Web server 2 to couple to network 14 to communicate with the devices and processing system 40.

The Web server 12 is preferably coupled to a Common Gateway Interface (CGI) application 28 which converts and communicates the data and commands between the devices on network 14 and the processing system 40 so the I/O devices do not have to use the database command language to interact with the database. System 40 and the devices may communicate directly if they are implemented in the same language or if a user implements a communication interface such as CGI 28 that correlates data fields in the client with those in system 40. Server 12, CGI 28, and the applications supporting system 40 may all reside on a single host computer or they may reside on separate computers coupled together by a local area network (LAN) or a wide area network (WAN). Preferably, the application interfaces with a database which supports Open Data Base Connectivity (ODBC) and Structured Query Language (SQL).

The communication sessions between the I/O devices coupled to the open network 14 and the Web server 12 are generally conducted in the same fashion as Internet protocol communication sessions are currently performed. That is, the I/O device establishes a communication connection with Web server 12, sends a request to the Web server, the Web server responds to the request and the I/O device or server closes the connection. Preferably, the non-standard I/O devices or PCs interfaced to such devices selectively couple to a local access port on the open network 14 through a local modem/ISDN connection. In this manner, the device is only coupled to the open network 14 when a transaction or a data operation is to be performed. While connected to the open network 14, a device may access a number of servers to accomplish a purpose. For example, a device may couple to a local access port and communicate with a first server to check inventory levels at a site, communicate with a second server to order stock for the inventory, and communicate with a third server to settle payment for the ordered goods. When all aspects of the transaction are complete, the connection with the local access port is terminated. In the preferred embodiment of the present invention, the protocol used to transport data messages between Web server 12 and the I/O devices coupled to the open network 14 is the Hyper Text Transport Protocol (HTTP), although other open system protocols utilized on the Internet may be used.

In standard HTTP protocol, a client program executing in one of the I/O devices may initiate communication with a server by sending a query message of the format:

http://<host>:<port>/<path>?<search part>

The message identifies the client as seeking communication with a HTTP server at the host address on the specified port. In the HTTP protocol, the default value for the port is 80 and the host address is the Internet protocol (IP) address of the type well-known in the art. The path value selects the file in the HTTP server which is activated in response to the message and the search part specifies a query for the selected file. In the initial communication, the query may be omitted so that the selected host file responds to the client program before a query is processed.

In the present invention, the client program uses a similar message to initiate a transaction or data operation, except

EXHIBIT A
PAGE 32 OF 39

that database commands are preferably embedded in a file at the server 12 and not in the "search part" of the command. although search parts may be constructed in accordance with the principles of the present invention that support non-standard I/O devices. Preferably, the client program interprets Hyper Text Markup Language (HTML) files containing HTML commands for communicating data between non-standard I/O devices and server 12. Most preferably, the HTML commands contain identifiers which are used by the CGI to place data returned in the forms of the HTML commands into database commands for queries or data insertions for the database. HTML is a command language well known for the retrieval and display of electronic documents for standard I/O devices such as PCs supported by full screen monitors, QWERTY keyboards, and standard peripherals such as hard disk drives and diskette drives. Standard HTML commands use text and previously known commands that reference Universal Resource Locators (URLs) to support the communication of electronic documents. These documents are files which may contain HTML commands, text, audio, video, or image data. The present invention extends HTML with commands that support communication between the server and the non-standard I/O devices.

In the HTTP protocol, data may be obtained during a communication session by using a tag called a FORM as part of the file defined by <path> in the command discussed above. The FORM format for standard HTTP is:

<FORM ACTION="URL"
METHOD=GET|POST>
Command
</FORM>

where "|" is an "OR" operator. The commands supported by standard HTTP are INPUT, SELECT, and TEXTAREA. Additionally, standard HTTP permits the inclusion of text data in the command area. In the present invention, HTML has been extended to support new ACTIONs, METHODs, and INPUTs.

In accordance with the principles of the present invention, tags are preferably used to identify device transfers and input operations. Preferably, the FORM tag is used to identify device transfers and ACTION and METHOD attributes further identify the device operation. As shown in FIG. 2, the extended ACTION field may include a FROM and TO attribute for accessing a local terminal file or smart card reader or a TO PRINTER attribute for directing output data to a printer local to the I/O device. The FROM and TO attributes for accessing local files and smart card readers and for directing output data to a local printer have previously been unsupported in any Internet protocol. As a result, the server 12 may access non-standard I/O peripherals for any of the I/O devices used in the transaction or data system 10. The ACTION="URL" is a part of standard HTTP and is well known.

The METHOD attributes may include the GET, POST, PAYMENT, or SQL methods. The GET and POST methods are currently supported in standard HTTP and are well known. The PAYMENT attribute is a directive to deliver data retrieved by an INPUT command to a private payment network for authorization and settlement and is not available in current Internet protocols. This directive is used by the client program to activate a conventional financial transaction application which communicates with the transaction system over a dedicated data line or PSTN in a known protocol such as VISA. Such an attribute is used where the more secure physical connection between remote site and transaction system and data encryption devices or the like are preferred. The SQL method preferably identifies a database language file which CGI 28 uses to correlate data in the HTML FORM to an insertion or query command contained in the file.

The preferred format for the INPUT tag which is used to identify input operations is also shown in FIG. 2. The TYPE and NAME attributes are used to define a non-standard I/O device or local storage variable for the input of data. The TYPE field values "text," "password," "checkbox," "radio," "submit," and "reset" are previously known, as are the attributes NAME, VALUE, CHECKED, SIZE, and MAXLENGTH. To support the extended capability of the present invention, the TYPE attribute preferably includes attributes MSRT1 for reading track 1 of a magnetic swipe reader, MSRT2 for reading a magnetic swipe reader track 2, KEY for reading input from a terminal command keypad, PIN for reading a personal identification number pad, BCW for reading a bar code wand, MICR for reading a check magnetic code reader, ATM for reading a dollar amount via a key input mask, INT for reading an integer via a key input mask, LOCAL for reading input from a variable in the local storage of an I/O device, and AUTOSUBMIT for returning a FORM with information to the server.

The NAME attribute used with the INPUT tag identifies reserved word names for local storage in the device executing the client program. Preferably, the NAME attribute identifies ip_, host_phone, tid, work_key, datetime, and deposit_acct as local storage areas in the local device for the terminal's Internet Protocol (IP) address, Internet access phone number, terminal ID, PIN encryption working key, date/time, and merchant account number, respectively. These attributes are used with the INPUT tag to read non-standard I/O devices which may be coupled to open network 14. For example, an INPUT TYPE=MSRT1 attribute causes the client program residing within a magnetic stripe reader to input data from track 1 of a stripe reader and insert that data into a FORM which is returned to Web server 12 for processing by an INPUT TYPE=AUTOSUBMIT statement.

Preferably, the database language commands which may be embedded in the extended HTML are SQL commands such as those shown in FIG. 3, although other database languages may be used. The SELECT command may include the names of data fields in a database so the device on network 14 may request a data item from a database at the central processing system. The database table is identified by the FROM attribute and the conditional selection of data from an identified database table may be defined by a WHERE attribute. Additionally, records may be requested from an identified database in ascending or descending order or in groups of two records at a time using the ORDER attribute. Additionally, the SELECT field command with the GROUP attribute provides I/O devices with the capability of retrieving records grouped under an identified name. Additionally, the I/O devices may either insert new data into an identified database with the INSERT attribute or update data already existing in a database with the UPDATE attribute. The values for the INSERT attribute may be identified with the VALUES attribute, and the SET and WHERE attributes may be used to define and conditionally update values in the identified database. Preferably, the present invention implements two DELETE and CREATE attributes. The DELETE attribute deletes all items in an identified column of a database table which may satisfy a condition defined by a WHERE attribute. The CREATE attribute creates a database table having a primary key identified by the PRIMARY KEY attribute.

Preferably, the server program executes on a computer system having at least an Intel 80386 or better processor with at least 4 megabytes of RAM and at least 3 megabytes of hard disk space available. The computer system running the server may operate any known server platform operating system such as WINDOWS 3.1, WINDOWS 95, or WINDOWS NT, UNIX, AIX, and others. The non-standard I/O devices require a processor of a Z80A type or better, at least 32K bytes of RAM, and at least 32K bytes of ROM. The device includes a modem capable of at least 1200 bits-per-second (bps) but other modem speeds may be used for communication between client and server. Alternatively, the device may be coupled to a LAN which in turn is coupled to the Internet for communication with server 12. A typical non-standard device which executes the client program is a VeriFone OMNI390, OMNI395, or VuFone terminal. OMNI390, OMNI395, and VuFone are trademarks of VeriFone, Inc., of Redwood City, Calif. Other exemplary devices include Phillips Screen phone, Hypercomm T7 terminal, and Apple Computer Newton MessagePad.

To build the preferred HTML files which CGI 28 preferably uses to implement the client program and database application, the user preferably uses an off-line editor. The files generated by the editor are preferably comprised of an integrated statements formed from HTML statements and database statements for retrieving and writing data with the database. Exemplary files showing such integrated statements for performing transactions are depicted in FIGS. 14–23B. After such a file is generated, the editor parses the integrated statements into HTML statements and into database statements such as SQL commands. The HTML files required by the client program to support communication with a transaction or data processing center may be downloaded to a device or PC for execution. The files containing the database application statements used by the CGI interface to communicate data with the database application program preferably reside on server 12. Preferably, the database files used by the CGI interface include SQL commands for the application program interfaced to an ODBC compliant database.

The general format of the HTML commands in the HTML files used for communication with a client program and server are of the general format: TAG ATTRIBUTE. Preferably, the TAG field may be one of FORM, INPUT, SQL, or TEXTAREA. The ATTRIBUTE field value depends upon the TAG value. Preferably, the FORM tag may include the ACTION or METHOD attributes where the ACTION attributes include the FROM<file>, TO PRINTER, TO<file>, and TO SCR values noted above, as well as the standard HTML ACTION value of URL=<file>. The METHOD attributes include the PAYMENT and SQL attributes noted above, as well as the standard HTML METHOD values of GET and POST. Also in accordance with the principles of the present invention, the INPUT tag may include TYPE, NAME, VALUE, CHECKED, SIZE, and MAXLENGTH attributes. These attributes are previously supported for the INPUT tag in HTML, however, the present invention further includes TYPE values of MSRT1, MSRT2, KEY, PIN, BCW, MICR, AMT, INT, LOCAL, and AUTOSUB, as well as the standard HTML TYPE values of TEXT, PASSWORD, CHECKBOX, RADIO BUTTON, SUBMIT, and RESET. The present invention also supports NAME attributes of IP_ADDRESS, HOST_PHONE, TID, WORK_KEY, DATETIME, and DEPOSIT_ACCT to identify local storage areas as well as standard HTML NAME attribute <Field NM> to identify a FORM variable.

The preferred high level processing of the client program is shown in FIG. 4. That processing includes an idle step (Block 100) in which the program performs general housekeeping tasks such as maintaining internal time, scanning for input which may activate the device, or other known functions. Further processing is activated by some operator action at the device or PC which causes the device to either open a remote URL (Block 102) or open a local URL (Block 104). If a remote URL is required, the device transmits a message of the format discussed previously which is routed through the open network and delivered to a server 12 for a transaction or data processing system (Block 106). The HTML file selected at the server 12 is identified by the remote URL in the initial communication between the device and server 12 and that URL is used to return the selected HTML file to the device for processing (Blocks 108, 110).

FIG. 4 also shows that an operator may initiate an open local URL function by typing in a command or by pushing a hot key which is associated with a local URL. The I/O device reads the HTML file identified by the URL from local memory (Block 112) and passes the HTML file to the function for processing HTML files (Block 110). After a file is processed (Block 110), the client program determines whether the HTML file is to be stored (Block 114). If it is not, the process returns to the idle processing (Block 100). Otherwise, the process determines whether the HTML file is to be associated with a hot key (Block 116) and, if it is, it stores the file and generates the link between a hot key and the stored file (Blocks 118, 120). If the HTML file is only to be stored, no association is made with a hot key and the file is simply stored in local memory (Block 20). The client program then returns to idle processing (Block 100).

The high-level processing for the HTML file (Block 110, FIG. 4) is shown in further detail in FIG. 5. The process begins by scanning the HTML file for a TAG (Block 140). If no TAG is found, the file is not in proper format for processing and processing returns to Block 114 discussed in FIG. 4 above. If a TAG is found (Block 142), the process determines whether the TAG is a FORM TAG (Block 144) or an INPUT TAG (Block 146). If it is a PORM TAG, then the FORM TAG is processed and the program continues by looking for other TAGS to process (Block 140). If the TAG is an INPUT TAG, the INPUT TAG is processed (Block 150) and the program continues by looking for other TAGS to process (Block 140). If the TAG is one of the standard HTML TAGS, the program implements the TAG in standard known ways (Block 152) and then scans for other TAGs to process (Block 140).

Processing the ATTRIBUTES used to implement a FORM TAG is shown in FIG. 6. That process continues by scanning the HTML file for an attribute (Block 160). If an attribute is not found (Block 162), the program returns to scan for other TAGS (Block 140, FIG. 5). If an attribute is found, the program determines whether it is an ACTION attribute (Block 164) or a METHOD attribute (Block 166). Depending on the type of attribute, the appropriate function for processing the attribute is executed (Blocks 168 or 170) and scanning for additional attributes continues (Block 160). If the attribute is not an ACTION or METHOD attribute, there is an error in the file and processing returns to scan for other TAGs.

The processing for the ACTION attribute is shown in FIG. 7. There, the ACTION attribute is examined to determine whether it is a FROM<file> (Block 180), TO PRINTER (Block 182), TO<file> (Block 184), TO SCR (Block 186), FROM SCR (Block 188) or a URL=<file> (Block 192). The URL=<file> ACTION is a standard HTML action which is processed in a known way (Block 194). The FROM <file>

EXHIBIT A
PAGE 34 OF 39

action is processed by reading data from a file associated with the I/O device or PC interfaced to the I/O device (Block 196). The TO PRINTER action results in data in the FORM being sent to the printer (Block 198) while the TO <file> action results in data in the FORM being written to a local file (Block 200). The TO SCR action causes data to be written to the smart card via a smart card reader (Block 202) and the FROM SCR reads data from a smart card through a smart card reader (Block 204). After the appropriate action processing takes place, the HTML file is scanned for additional ACTION values to perform (Block 206), and if one is found, the process continues. If no attribute is located (Block 208), the process returns to scan for other attributes (Block 160, FIG. 6).

The processing for the METHOD attributes for FORM tags are shown in FIG. 8. The process determines which type of METHOD is present in the FORM and then properly processes the attribute. For the GET and POST methods (Blocks 210, 212) the processing is the same as that performed in standard HTML (Blocks 226, 228). That is, for the GET method, the identified URL<file> is queried for data while the POST attribute causes data to be transferred to the URL<file>. The preferred METHOD attributes extending the HTML implementation of the present invention are SQL (Block 214), and PAYMENT (Block 224) attributes. The SQL attribute is preferably not expanded into a SQL command at the client, but rather is expanded by the CGI 28 at server 12 by correlating the data or variable field names in a returned form with the SQL commands stored at the server. This processing is done in a manner described in more detail below. The client program passes the SQL file identifer to the server 12 (Block 230). The processing of the PAYMENT command (Block 232) is discussed in more detail below. The HTML file is scanned for other METHODs (Block 242, 244), and, if one is found, the processing continues by identifying the METHOD (Blocks 210–224). Otherwise (Block 244), the process returns to scan the HTML file for other ACTION or METHOD attributes (Block 160, FIG. 6).

Processing for the INPUT tag is shown in FIG. 9. The process scans the HTML file following the INPUT tag for attributes (Block 250). If no attributes are found (Block 252), the process continues by scanning the HTML file for other tags to process (Block 140, FIG. 5). If an attribute is found and it is a TYPE attribute (Block 254), it is processed (Block 256), and if the attribute is a NAME attribute (Block 258), it is processed (Block 260). Both the TYPE and NAME processing is shown in more detail in FIGS. 10 and 11, respectively. If the attribute is neither a NAME or TYPE attribute, it is a standard attribute for an INPUT tag supported by standard HTML and is processed in a known manner (Block 262). Following processing of the INPUT attribute, the HTML file is scanned for other attributes to process (Block 250).

Processing for the TYPE attribute is shown in FIG. 10. The process first identifies the TYPE attribute for the INPUT tag and then performs the appropriate processing. The new TYPE attributes of the preferred embodiment of the present invention are MSRT1 (Block 270), MSRT2 (Block 272), KEY (Block 274), PIN (Block 276), BCW (Block 278), MICR (Block 280), AMT (Block 282), INT (Block 284), LOCAL (Block 286), and AUTOSUB (Block 288). If the TYPE attribute is not one of these, it is a standard HTML type attribute that is processed in a known mariner (Block 310). Each of the new HTML TYPES supported by the present invention causes an I/O operation with a non-standard device. Specifically, these operations are the reading of Track 1 of the magnetic stripe reader (Block 290), the reading of the second track of the magnetic stripe reader (Block 292), the reading of a keypad (Block 294), the reading of an encrypted PIN through a PIN entry device (Block 296), the reading of a bar code through a bar code reader (Block 298), the reading of encoded data on a check through a magnetic check reader (Block 300), the reading of a dollar amount from a keypad through a key input mask (Block 302), the reading of a number from a keypad through a key input mask (Block 304), the reading of data from a local variable (Block 306), and the submission of the data read from one of these devices in a FORM returned to the server 12 (Block 308). The data mask for AMT constrains the dollar amount read to a predetermined number of characters with only two characters following the decimal point. The data mask for INT ensures the number is an integer value within a predetermined range. Processing continues by scanning the HTML file for other TYPE attributes (Block 312) and, if another TYPE attribute is found (Block 314), processing continues by determining the TYPE attribute and performing the appropriate processing. Otherwise, the process returns to scan the HTML file for other attributes (Block 250, FIG. 9).

The NAME attribute processing is performed in accordance with the process shown in FIG. 11. That process examines the NAME attribute to determine if the variable name identified by the attribute is IP_ADDRESS, HOST_PHONE, TID, WORK_KEY, DATETIME, or DEPOSIT_ACCT (Blocks 320, 322, 324, 326, 328, 330). If they are, the INPUT value resulting from one of the INPUTS in a FORM of the HTML file is stored in a local variable identified by the NAME attribute. Following storage (Block 332), the file is scanned for other NAME attributes (Block 328) and, if there are none (Block 332), processing continues by scanning for other attributes for the INPUT tag (Block 250, FIG. 9). If the NAME attribute is a standard HTML INPUT NAME, it is processed by known methods (Block 336). Processing then continues by scanning for other NAME attributes to process (Block 338, 340). Otherwise, the process returns to scan the HTML file for other attributes (Block 250, FIG. 9).

CGI 28 receives Internet protocol statements in a file transmitted from a client program and provides data from those statements to the application(s) implemeating system 40 and receives the output of system 40 and provides them to the client program in a file. CGI 28 may be implemented by a program developed by a user using a manual development method as shown in FIG. 24A. That method requires a user to generate a system definition from which a file statement definition for the client and application are developed to implement the transactional or data system. Using the file statement definitions, the user generates the files for the client and database programs which are interpreted by the respective programs to implement transactions or data processing. This process requires the user to not only have knowledge regarding the transaction or data process but specific details of the interaction between the client and database. The user is further required to resolve and correlate all data identifiers in the statements for the client and database environments.

Preferably, CGI 28 is developed with an editor that only requires the user to define the system with statements which are an integration of the protocol statements and the database language. The process implemented by this editor is shown in FIG. 24B. Examples of such integrated statements for files which implement a specific transaction are shown in FIGS. 14 to 23B. The editor verifies the syntax of the integrated statements and correlates the data variables of the

protocol statements with the data fields of the database. Following the generation of the integrated statements, the editor segregates the protocol statements from the database language statements. The protocol statements are stored in files which are identified as being for a particular transaction or data process and the database statements are stored in files which are identified as being for a particular transaction or data process on an identified database table. The editor places a database file identifier in the protocol statements which contained embedded database statements. The database file identifiers are used by CGI 28 to select the file for the appropriate transaction so CGI 28 may correlate data variables in the protocol statements with data fields in the database files. The files containing statements to be interpreted by the client program are then downloaded to the appropriate terminals, and the database files containing database language statements are stored on the system executing the CGI 28.

Alternatively, the editor of the present invention may parse integrated statements which are segregated into source code statements for first and second processors, such as editor further includes a compiler to generate executable code for each processor and, if the processors execute differing source code, a compiler for each source code language. The executable code may then be downloaded to the respective processors for execution.

More specifically, the editor preferably places the database statements for one of the transactions of the preferred embodiment in a file identified by the database name following SQL in FIG. 12. The attributes and tags forming the HTML statements for one of the transactions of the preferred embodiment are placed in a file generally denoted as <html_file>.HTM. The name <html_file> is a name which identifies one of the transactions. Where SQL statements are in the fields of the integrated statements shown in FIGS. 14 to 23B, the string "<html_file>.SQL" is substituted as the database name in the statements of the <html_file>.HTM file. When the CGI executable file is initiated and parses the returning forms, the returned data is placed in the corresponding "<html_file>.SQL" file which is passed to the application program as a command line argument. In this manner, an abbreviated form for the SQL commands may be communicated over the open network between the client and CGI and the CGI may be able to expand those abbreviated SQL commands into the appropriate SQL commands which the application program requires to manipulate the ODBC database.

To effectuate a transaction, for example, an operation at a terminal with non-standard I/O devices may activate a terminal file with a hot key or other action. In processing the activated file, the client program may acquire data which is stored in a local variable or accessible through a non-standard I/O device. This data may then be stored in a FORM and submitted to a server file at a processing system address. The server file activates CGI 28 which retrieves data from the FORM and incorporates it into database statements in the database file for the appropriate transaction and database. If the database statement is a query, the requested data is returned to the CGI in the database file and the CGI places it in the corresponding FORM variables so the server may return the data to the terminal. If the database statement provides data to a database to obtain an authorization, for example, the action performed by the database application in response to the data is placed in the corresponding FORM and returned to the terminal. In this way, data is exchanged between the terminal and the database application. This exchange is supported by CGI 28 even though the server/client communication is performed in an open system protocol, such as HTTP, and the database application is performed in another language, such as SQL. CGI 28 is able to convert and exchange the data between the client and database without the user having to specifically design and implement a conversion program.

The communication paths available for a device implementing the present invention are shown in FIG. 13A. As shown there, an I/O device **420** is coupled through the WorldWide Web open network **426** to an Internet Web server **12**. This connection may be implemented with the preferred extended capability HTML described above. Although HTML files may be encrypted to enhance the security of the document as it is communicated across the Internet, the operator of the system may choose to utilize a more secure physical connection between the device **420** and the Web server **12**. To obtain this alternative connection, the PAYMENT command for the METHOD attribute is preferably used. One form of the PAYMENT command is for a merchant's terminal and the other is for a consumer's terminal. In either terminal, the client program which supports the extended capability HTML operates independently but co-resident in memory with a certified bank card authorization and capture application, which may be provided by a financial institution or a bank card processor.

For the form of the command shown in FIG. 13B, the client program in the merchant terminal suspends its execution and passes the terminal identifier, stored locally, which identifies the merchant's account and the consumer account information read via a magnetic stripe reader or the like, to the bank card application. The bank card application communicates this information via a PSTN 424 or the like to a transaction processor **422**. The processor **422** authorizes or denies the transaction and, if authorized, a printer at the merchant terminal prints a purchase agreement which the consumer may execute to complete the transaction.

In response to a HTML file having a FORM with an ACTION attribute equal to an executable file name for a bank card application program or the like, a METHOD attribute with a field value of PAYMENT, and an INPUT tag with a TYPE attribute of LOCAL_NAME which identifies a deposit only account supplied by a merchant (as shown in FIG. 13C), the client program is suspended and control is transferred to the bank processing application. The bank processing application then uses a modem or ISDN D channel using T3 POS protocol or the like to connect to a secure packet network **424** to connect in a virtual point-to-point manner with a payment processor through a PSTN network or the like. This physical connection provides an additional security element to the encrypted data for the transaction of account information. PIN numbers encrypted by PIN pads provided at the consumer site, and other sensitive information. The bank processor **422** may submit remittance data to the merchant, via the Web or otherwise. After receiving the remittance data, the merchant may ship the product to the consumer. Thus, in this manner, the I/O device may communicate with a plurality of Web servers to "shop" for a best price, delivery date, or other relevant information for selecting a preferred transaction, and then execute the PAYMENT method to utilize a more secure physical communication connection and data security devices to consummate the financial elements of the transaction with less risk and costs for the merchant, consumer, and bank processor.

The preferred integrated HTML/SQL statements which support a card initiated payment authorization and capture transaction are shown in FIG. **14**. A first file **500** includes

EXHIBIT A
PAGE 36 OF 39

statements which identify the URL database from which the non-standard I/O device seeks authorization for a transaction. The prompts to the operator to enter the account number and amount of the transaction are supported by the INPUT tags which read the second track of the magnetic stripe reader to accept a number of up to 40 characters and assign that information from that track to a variable, and to input the up to 8 characters from the keyboard or the like into a variable called AMOUNT. The INPUT tag with the TYPE attribute of AUTOSUBMIT returns the form to the server for processing in accordance with the method defined in the returned form. As shown in FIG. 14, that METHOD statement causes CGI 28 to incorporate returned data into SQL commands which query the database as to whether the subfield of the track 2 data representing the account number is present in the authorization table of the database. If the data is not present, then a new record is inserted into a table labeled "log_table". The new record consists of the account number and the amount returned in the FORM. Based upon the results of this processing, the application program supplies the data fields to the FORM which will be returned to the client program for printing the transaction record. That file **510** is shown in FIG. 14. The ACTION attribute TO PRINTER and the POST METHOD causes the data in the next eight lines to be directed to the printer coupled to the non-standard I/O device for printing the transaction form. The customer may then execute the printed form to complete the transaction. If the transaction is declined or an error is otherwise encountered, the file **520** is used to return a denial to the client program.

In a similar manner, the preferred integrated statements for a bar code order input with card-initiated payment authorization is shown in FIG. 15. The file **550**, supported by the present invention which implements the transaction request, is again directed to the proper database by the ACTION attribute. The necessary customer information such as name and address may be input through a standard keyboard. The HTML command in the present invention also permits the form to receive the bar code, unit price, and credit card information in a manner similar to that discussed above for the magnetic card reader. Once this information is returned to the server and CGI interface, it is processed by the application program in accordance with the METHOD identified in the returned form. The method of HTML file **550** also creates a database order_table having the information shown in the method. Again, if the transaction is approved, the data for the order and customer acceptance of the order is provided in HTML file **555**, which is directed by the ACTION attribute to the printer at the non-standard I/O device. If the account number is not in the authorization database, the authorization declined or error response is provided in correspondence with the statements in file **560**.

In a similar manner, FIGS. 16–22 show the integrated statements for a transaction request, authorization response, or authorization declined response files for key input order with secure payment transaction (FIG. 16), a smart card-debit (Type 1) transaction (FIG. 17A), a smart card debit (Type 2) transaction (FIG. 17B), a debit card transaction (FIG. 18), a check verification transaction (FIG. 19), a customer frequency transaction (FIG. 20), an item search transaction for which there is no denial (FIG. 21), retail store end of day reporting (FIG. 22) and a store reporting an e-mail transaction (FIG. 23).

While the present invention has been illustrated by the description of a preferred and alternative embodiments and processes, and while the preferred and alternative embodiments and processes have been described in considerable detail, it is not the intention of the applicant to restrict or in any way limit the scope of the appended claims to such detail. Additional advantages and modifications will readily appear to those skilled in the art. For example, rather than expanding HTTP to support non-standard I/O devices, the FTP, POP, SMTP, TELNET or other protocols may be expanded in like manner to couple non-standard I/O devices to the Internet. Similarly, the preferred implementation of the present invention supports a variety of non-standard I/O devices and I/O operations. An Internet protocol may be constructed in accordance with the principles of the present invention to support only selected I/O devices or operations disclosed in the present application. The invention in its broadest aspects is therefore not limited to the specific details, preferred embodiment, and illustrative examples shown and described. Accordingly, departures may be made from such details without departing from the spirit or scope of applicant's general inventive concept.

What is claimed is:

1. An open network processing system comprising:

a server for communicating in an extended open network protocol; and

a plurality of input/output (I/O) devices coupled to said server through an open network, said I/O devices communicating with said server in said extended open network protocol that supports communication with non-standard I/O devices;

wherein said extended open network protocol further comprising:

tags for identifying one of said I/O devices and input operation to be performed with said one of said I/O devices;

action attributes for defining said identified device operation to be performed with a local resource for one of said I/O devices; and

method attributes for defining a data transfer method for providing data between said server and said I/O devices.

2. The system of claim 1, wherein said tags, action attributes, and methods are implemented in an Hyper Text Transport Protocol.

3. The system of claim 1, said tags further comprising:

form tags for identifying device operations; and

input tags for identifying input for one of said I/O devices.

4. The system of claim 1, said action attributes further comprising:

file operation identifiers for a non-standard I/O device;

printer operation identifier for a non-standard I/O device; and

data operation identifiers for a non-standard I/O device.

5. The system of claim 1, said method attributes further comprising:

universal resource file operation identifiers;

database identifier for identifying a database associated with said server; and

a method command for identifying an alternative communication network for communication between one of said I/O devices and another processor.

6. The system of claim 1, said extended open network protocol further comprising:

types for identifying non-standard I/O devices for data input; and

names for identifying data storage at one of said I/O devices.

7. The system of claim 1 further comprising:

EXHIBIT A
PAGE 37 OF 39

protocol statements for said extended open network protocol which support communication between said server and said I/O devices;

application statements for an application on said server; and

a common gateway interface (CGI) for providing data between said protocol statements and said application statements.

8. The system of claim 7 wherein said protocol statements are HTTP statements and said application processing statements are SQL statements.

9. The system of claim 1 wherein at least one of said I/O devices is a magnetic stripe reader so that data read from a magnetic strip may be communicated with said server by said extended open network protocol over said open network whereby possession of a card bearing said stripe may be used for a transaction over said open network.

10. The system of claim 9 wherein said extended open network protocol includes input tags and types for reading a first and a second track from said magnetic stripe reader.

11. The system of claim 1 wherein at least one of said I/O devices is a Personal Identification Number (PIN) entry device so that a PIN automatically encrypted in hardware by said PIN entry device may be communicated to said server by said extended open network protocol.

12. The system of claim 1 wherein at least one of said I/O devices is a personal digital assistant so that data may be communicated between said personal digital assistant and said server by said extended open network protocol.

13. The system of claim 1 wherein at least one of said I/O devices is a check magnetic reader so that data encoded on a check and read by said check magnetic reader may be communicated between said check reader and said server by said extended open network protocol.

14. The system of claim 1 wherein at least one of said I/O devices is a bar code reader so that data encoded with bar codes and read by said bar code reader may be communicated between said bar code reader and said server by said extended open network protocol.

15. The system of claim 1 wherein at least one of said I/O devices is one of a credit card terminal and screen phone terminal so that data may be communicated between said one terminal and said server by said extended open network protocol.

16. The system of claim 15 wherein said extended open network protocol communicates data from one of a keypad, local variable, and a magnetic stripe reader of said one terminal with said server by said extended open network protocol.

17. The system of claim 1 wherein at least one of said I/O devices is a personal computer (PC) interfaced to at least one non-standard I/O device so that said non-standard I/O device and said server may communicate data over said open network through said PC by said extended open network protocol.

18. The system of claim 17 wherein said non-standard I/O device interfaced to said PC is one of a magnetic stripe reader, a PIN pad, a printer, or a smart card reader.

19. An open network transaction system comprising:

a server for communicating files in an extended open network protocol; and

a plurality of I/O devices coupled to said server through an open network, each said I/O device includes a client program for communicating with said server in an extended open network protocol for communicating data between non-standard I/O devices and said server;

wherein said extended open network protocol further comprising:

tags for identifying one of said I/O devices and input operation to be performed with said one of said I/O devices;

action attributes for defining said identified device operation to be performed with a local resource for one of said I/O devices; and

method attributes for defining a data transfer method for providing data between said server and said I/O devices.

20. The system of claim 19, further comprising:

an application at said server; and

a common gateway interface between said server and said client programs, said common gateway interface for communicating data between said client programs and said application.

21. The system of claim 20 wherein said application is a database application.

22. The system of claim 21 wherein said database application is a Structured Query Language application.

23. The system of claim 20 wherein said extended open network protocol is an extended Hyper Text Transport Protocol.

24. The system of claim 20 wherein said server communicates files to said client program for processing; and

said client program returns data from a non-standard I/O device to said server in a form communicated in said processed file.

25. The system of claim 24 wherein said non-standard I/O device is one of a magnetic stripe reader, a PIN pad, a screen phone terminal, a credit card terminal, and a smart card reader.

26. The system of claim 24 wherein said common gateway interface incorporates said data from said non-standard I/O device in an application statement for said application.

27. The system of claim 26 wherein said common gateway interface incorporates data received from said application into a form communicated by said server to said one of said I/O devices.

28. The system of claim 27 wherein said form communicated by said server to said one of said I/O devices includes a command for an alternative communication link so that a program co-resident in said one of said I/O devices is activated for said alternative communication link.

29. The system of claim 28 wherein said I/O device is a merchant terminal, said merchant terminal communicates a locally stored terminal id and consumer account data input via a non-standard I/O device to a processor through said alternative communication link by said co-resident program to complete a transaction.

30. The system of claim 28 wherein said I/O device is a consumer terminal and said form communicated by said server includes a merchant account number, said consumer terminal communicates said merchant account number and a consumer number to a processor through said alternative communication link by said co-resident program to complete a transaction.

31. The system of claim 30 wherein said consumer account number is verified through a non-standard I/O device.

32. The system of claim 31 wherein said non-standard I/O device is one of a PIN entry device and a magnetic card reader.

33. The system of claim 30 wherein said processor communicates a remittance data to a merchant server corresponding to said merchant account number via said open network.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF CORRECTION

PATENT NO. : 5,742,845

DATED : April 21, 1998

INVENTOR(S) : Wagner

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

column 4, line 44 should read "...a PIN provided through a keypad, determine the amount of ..."

column 5, line 1 should read "...cally in one of the formats discussed above under X.25..."

column 12, line 27 should read "...identifies ip_address, host_phone, tid, work_key, datetime, and..."

Signed and Sealed this

Twenty-fourth Day of November, 1998

Attest:

BRUCE LEHMAN

Attesting Officer

Commissioner of Patents and Trademarks

EXHIBIT A

PAGE 39 OF 39

EXHIBIT / ATTACHMENT

B

(To be scanned in place of tab)

US005905908A

United States Patent [19]

Wagner

[11] Patent Number: 5,905,908

[45] Date of Patent: *May 18, 1999

[54] **OPEN NETWORK SYSTEM FOR I/O OPERATIONS WITH NON-STANDARD I/O DEVICES UTILIZING EXTENDED PROTOCOL INCLUDING DEVICE IDENTIFIER AND IDENTIFIER FOR OPERATION TO BE PERFORMED WITH DEVICE**

[75] Inventor: **Richard Hlers Wagner**, Dunwoody, Ga.

[73] Assignee: **Datascape, Inc.**, Atlanta, Ga.

[*] Notice: This patent is subject to a terminal disclaimer.

[21] Appl. No.: **08/995,123**

[22] Filed: **Dec. 19, 1997**

Related U.S. Application Data

[63] Continuation of application No. 08/493,772, Jun. 22, 1995, Pat. No. 5,742,845.

[51] **Int. Cl.**[6] **G06F 13/14**; G06F 13/42; G06F 15/16

[52] **U.S. Cl.** **395/831**; 395/200.33; 395/200.58; 395/500; 370/401

[58] **Field of Search** 395/200.57, 183.01, 395/500, 821, 822, 823, 831, 200.58, 200.33, 200.32; 705/26; 370/401

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,499,343	3/1996	Pettus	395/200.2
5,530,844	6/1996	Phillips et al.	395/500
5,581,478	12/1996	Cruse et al.	364/505
5,678,002	10/1997	Fawcett et al.	395/183.01
5,732,219	3/1998	Blumer et al.	395/200.57

Primary Examiner—Thomas C. Lee
Assistant Examiner—Rehana Perveen
Attorney, Agent, or Firm—Morris, Manning & Martin LLP

[57] **ABSTRACT**

An open network system for supporting input/output (I/O) operations for non-standard I/O devices are disclosed. The system includes a server coupled to a plurality of I/O devices through an open network and an extended open system protocol that supports communication with devices that are not personal computers (PCs). These devices include magnetic stripe readers, check readers, smart card readers, credit card terminals, screen phone terminals, PIN pads, printers, and the like. The extended open network protocol includes tags which identify device and input operations and attributes which identify the location, data exchange method, and data variable names for the retrieval, acquisition, and submission of data between the server and I/O devices. Preferably, the open network protocol is implemented in a Hyper Text Transport Protocol (HTTP). Preferably, the system includes a common gateway interface (CGI) at the server which converts protocol statements communicated between the server and I/O devices to application language statements for providing data to an application program coupled to the server. Most preferably, the application statements and protocol statements are constructed in integrated statements with an editor. The editor ensures that data identifiers in the application and protocol statements are compatible. The integrated statements are then parsed by the editor to segregate the protocol statements from the application statements. The protocol statements are downloaded in a file to a client program at an I/O device for processing. The application statements are stored in a file for use by the application. In this manner, generation of the files for client and application processing are automatically done without the user ensuring the correlation of the data fields in the two files.

20 Claims, 25 Drawing Sheets

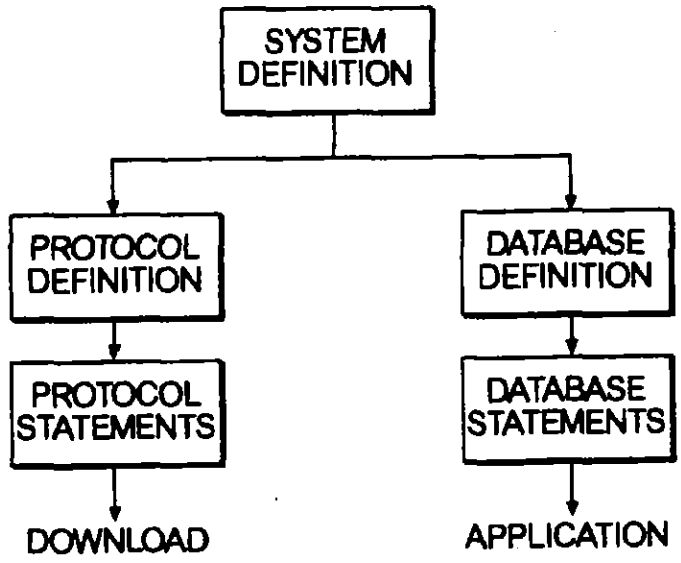

EXHIBIT B
PAGE 1 OF 37

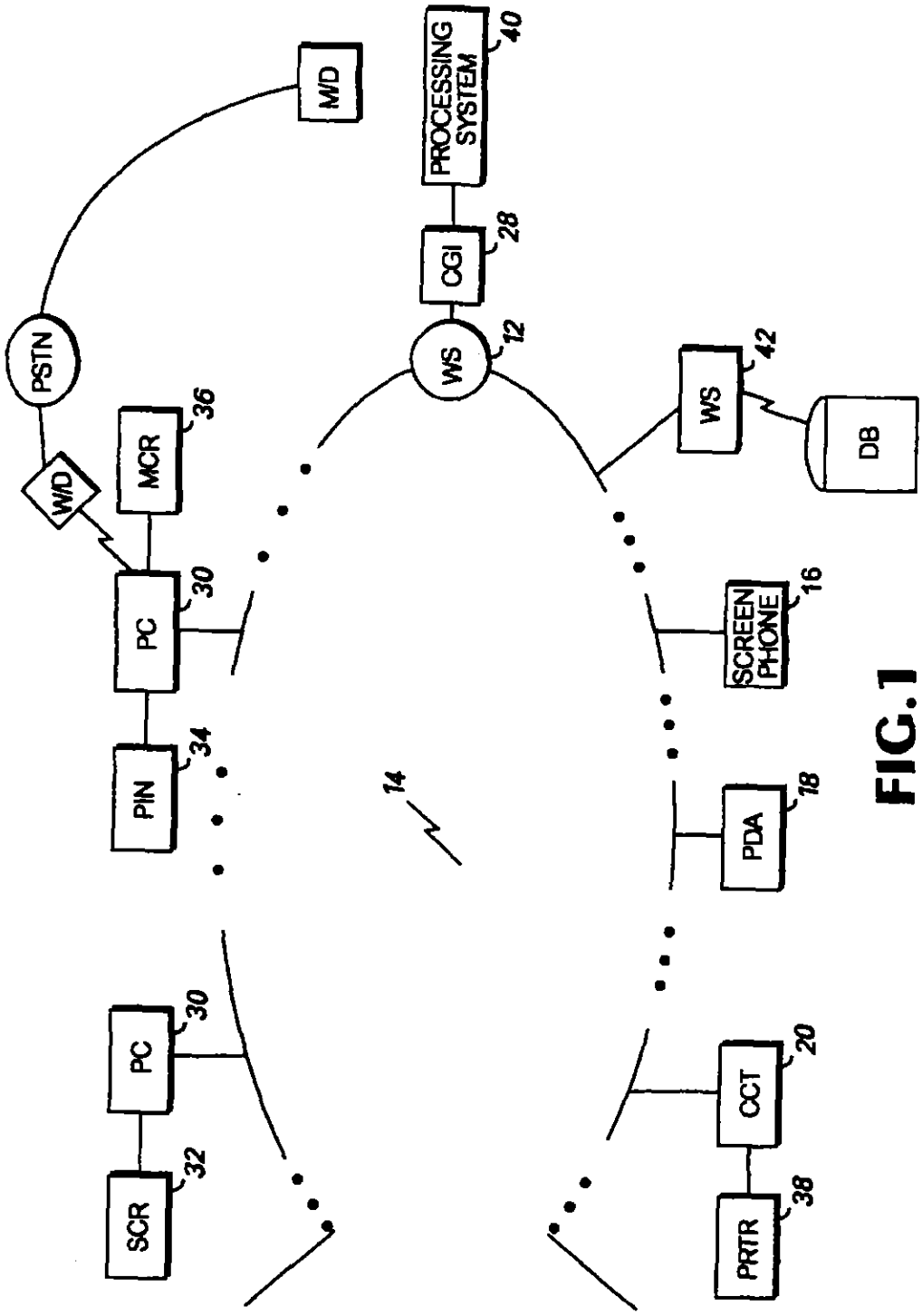

FIG. 1

EXHIBIT B
PAGE 2 OF 37

HTML+D Attributes		Description
<FORM ACTION =	"url"	To/From Web Server URL
	FROM "file name"	From Terminal Local File
	TO PRINTER	To Local Printer
	TO "file name"	To Terminal Local File
	FROM SCR	From Smart Card Reader
	TO SCR	To Smart Card Reader
METHOD =	"GET"	Retrieve Data
	"POST"	Store Data
	"PAYMENT"	Directive to deliver INPUT data to a private Payment Network for authorization and settlement.
	SQL <database name>	SQL statement database table

Attribute	Value	Description
<INPUT TYPE =	"text"	
	"password"	
	"checkbox"	
	"radio"	
	"submit"	
	"reset"	
NAME =	<field name>	
VALUE =	<initial value>	
CHECKED =		
SIZE =		
MAXLENGTH =		
>		

Attribute	HTML+DValue	Terminal Device
TYPE =	*"MSRT1"*	Mag Stripe Reader - Track 1
	"MSRT2"	Mag Stripe Reader - Track 2
	"KEY"	Terminal Command Keypad
	"PIN"	PIN Pad
	"BCW"	Bar Code Wand
	"MICR"	Check MICR Reader
	"AMT"	Dollar amount key input mask
	"INT"	Integer key input mask
	"LOCAL"	Input from Local Variable
	"AUTOSUBMIT"	Submit FORM to ACTION URL
NAME =	*ip_address*	Local Variable - Terminal's IP Address
	host_phone	Local Variable - Local Internet Access Phone Number
	tid	Local Variable - Terminal ID
	work_key	Local Variable - PIN encryption working key
	datetime	Local Variable - Date and time
	deposit_acct	Local Variable - Merchant Deposit Account

FIG.2

EXHIBIT B
PAGE 3 OF 37

SQL Statements

The following SQL commands represent a subset of the entire command set that varies by database vendor.

HTML+D Attributes	Description
SELECT, field_name,..*	Request field_name (one or many) from a database table
FROM=<table name>,..	Database table name
WHERE=<condition>	Conditional selection of data
name = "constant"	
name LIKE "constant"	
name IN "constant"	
AND	
OR	
ORDER=ASC	Request in ascending order
DESC	..descending
2	..by2's
GROUP=<name>	
INSERT TABLE=<table name>	Insert new data in database table
VALUES="constants"	
UPDATE FROM	
<table name>	Update field_name in database table
SET=field_name="constant"	
[WHERE=<condition>]	Update if WHERE clause is satisfied
DELETE FROM <table name>	Delete all columns that satisfy
[WHERE=<condition>]	WHERE clause
CREATE TABLE<table_name>	Create database table
PRIMARY KEY<name>	

FIG. 3

EXHIBIT B
PAGE 4 OF 37

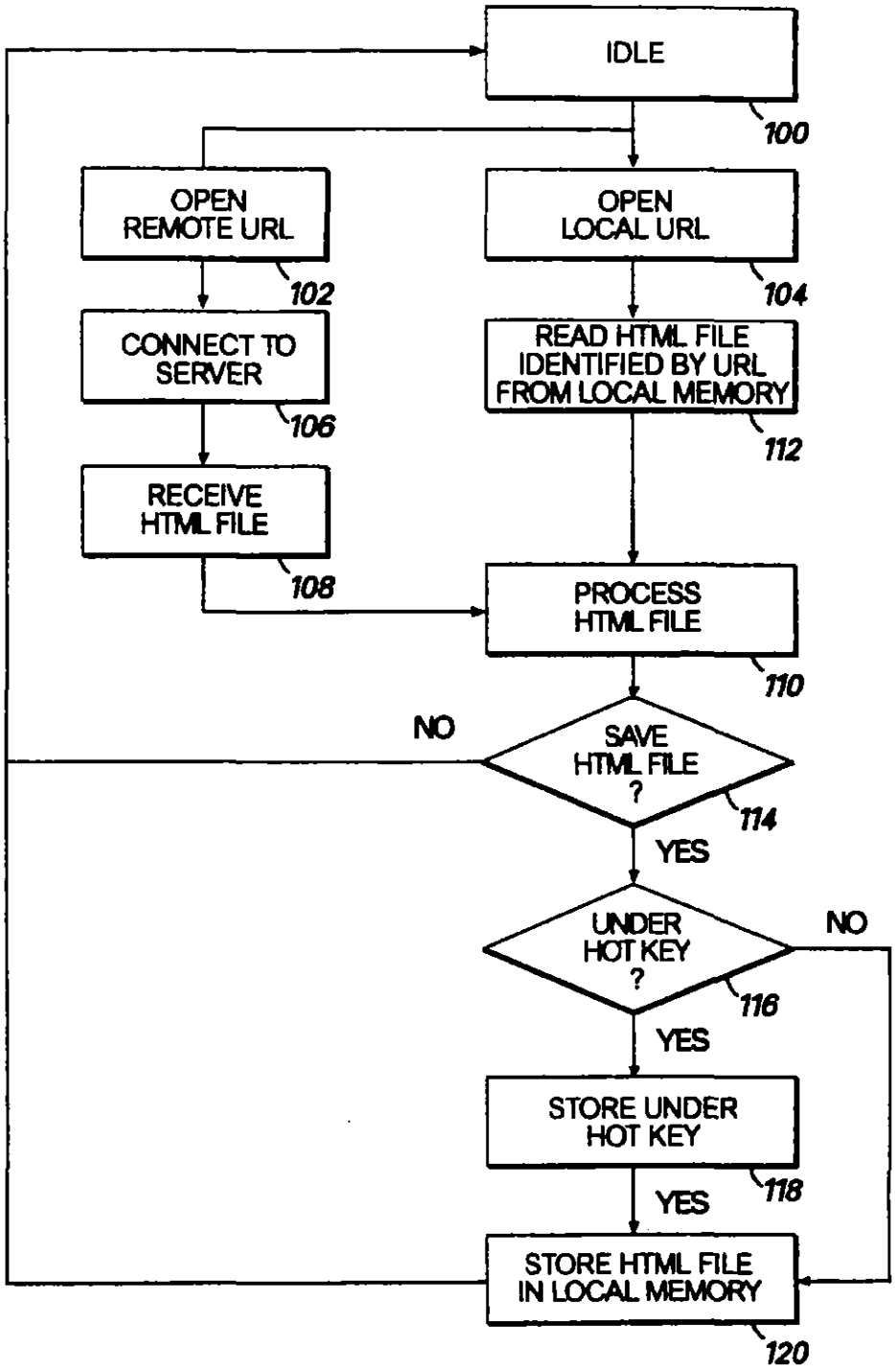

FIG.4

EXHIBIT 8
PAGE 5 OF 37

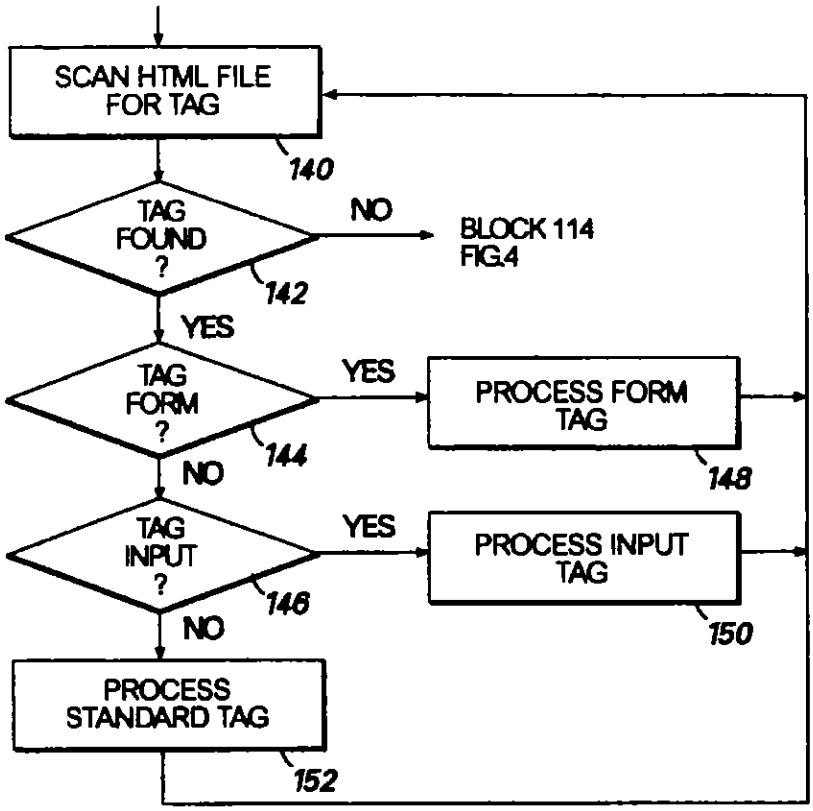

FIG.5

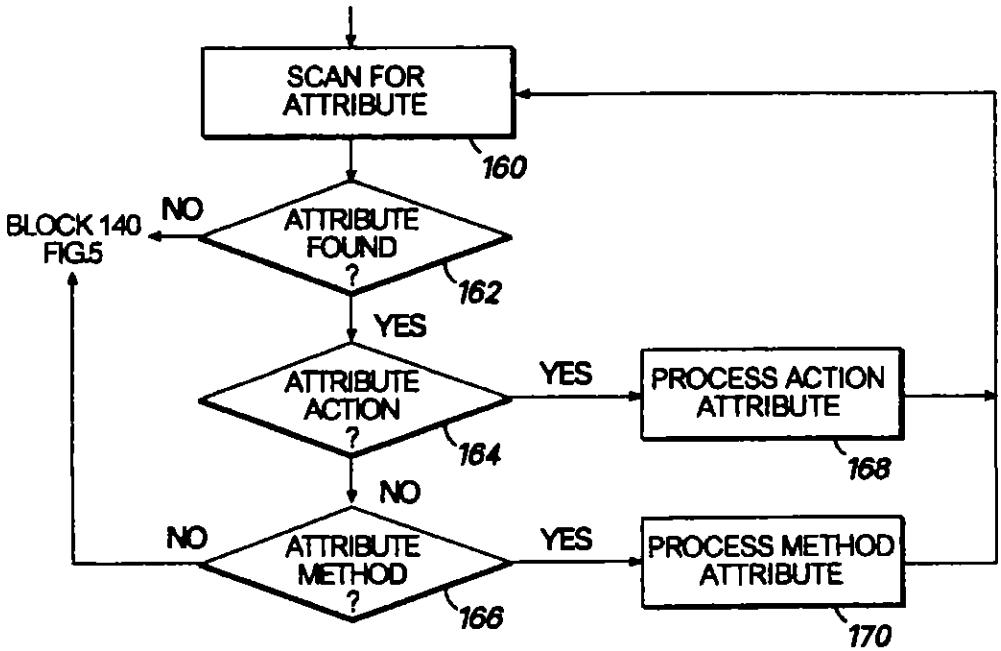

FIG.6

EXHIBIT B
PAGE 6 OF 37

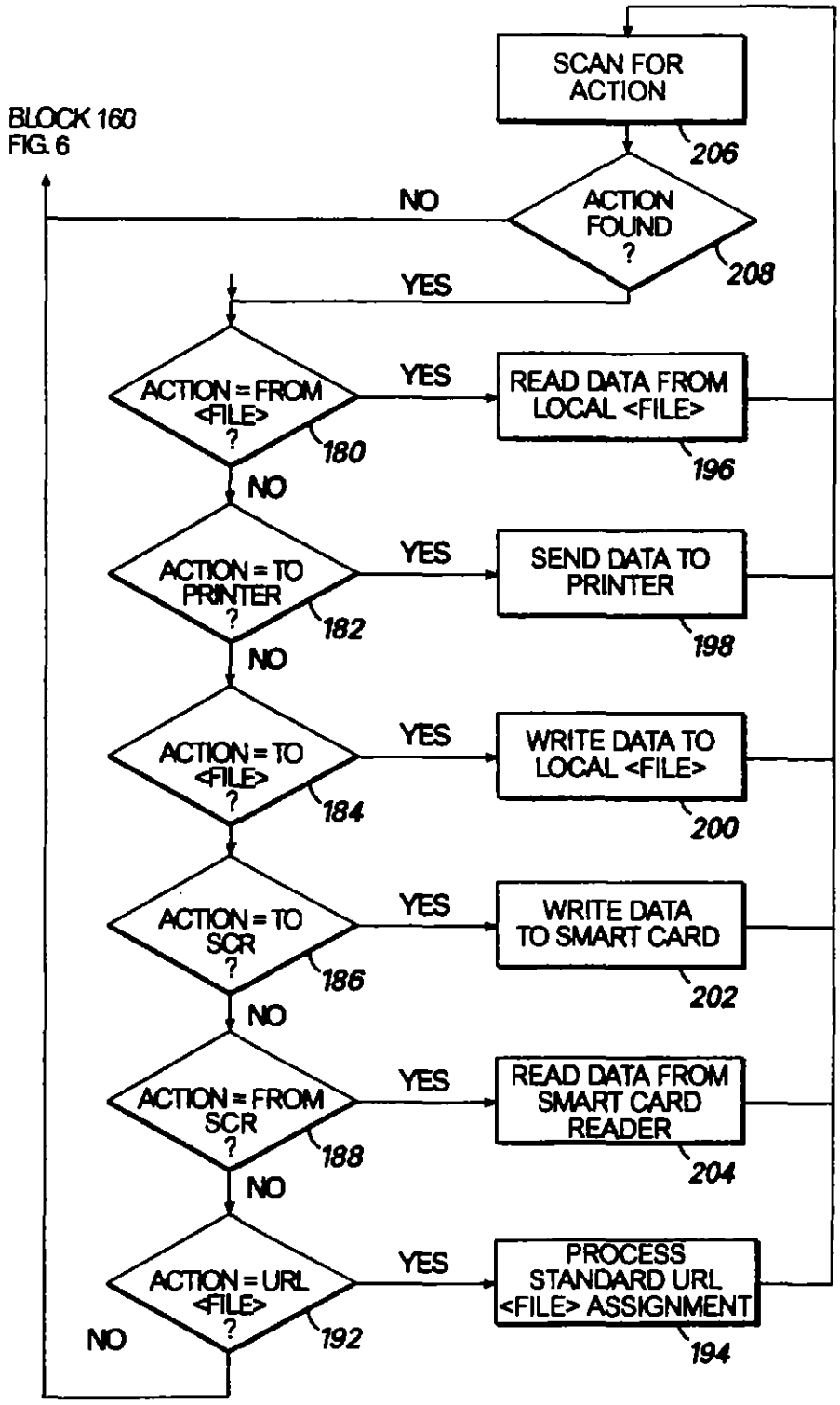

FIG.7

EXHIBIT B
PAGE 7 OF 37

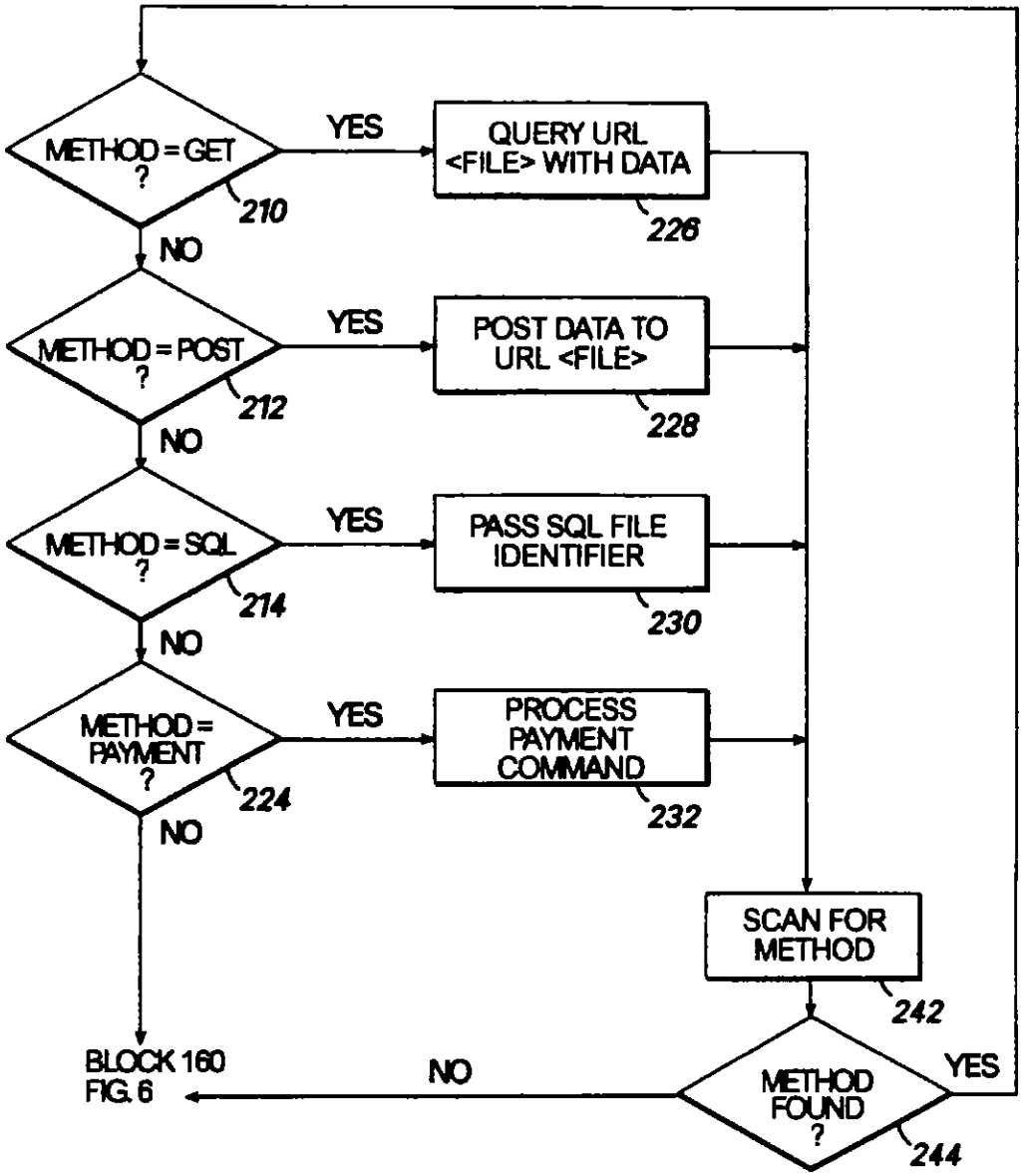

FIG.8

EXHIBIT B
PAGE 8 OF 37

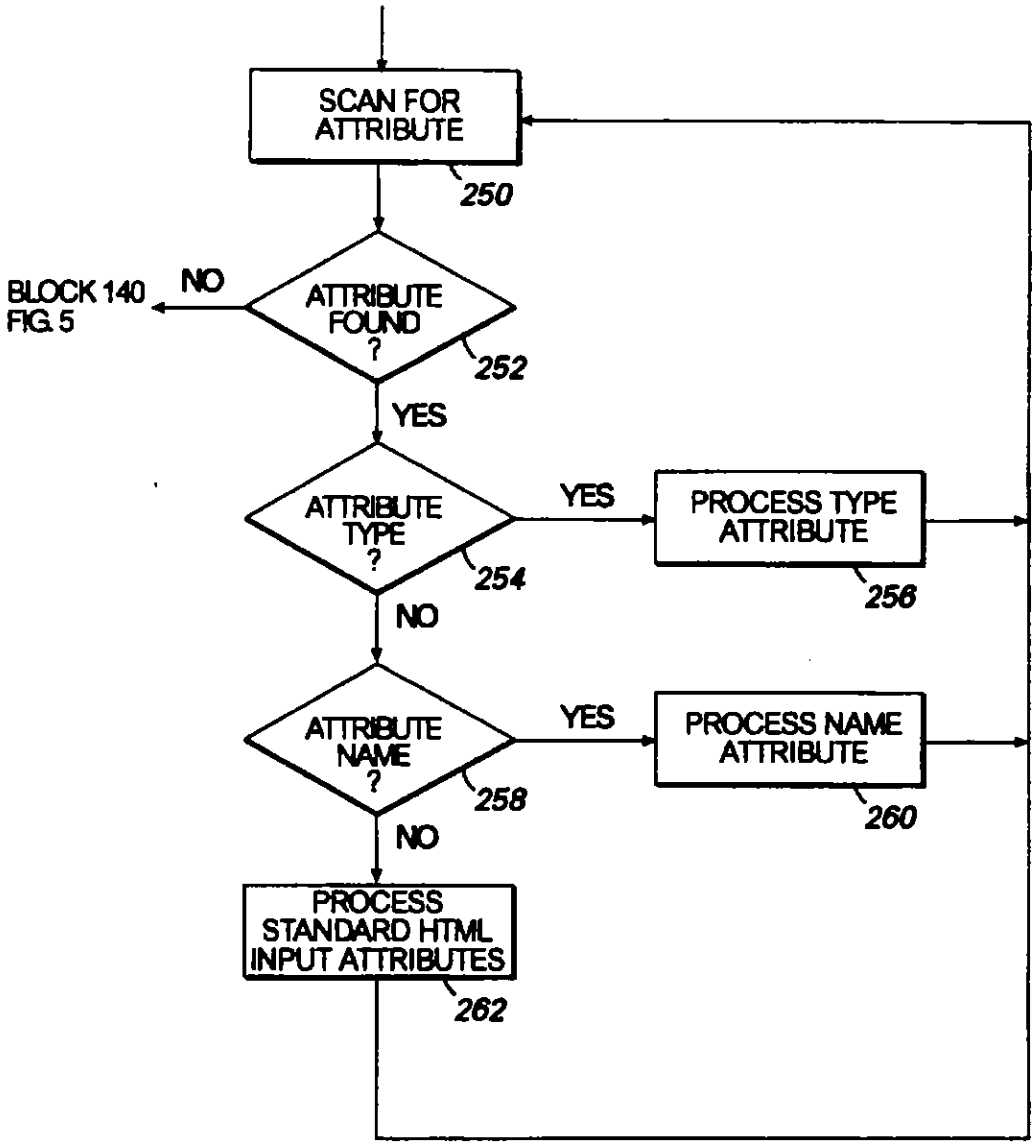

FIG. 9

EXHIBIT B
PAGE 9 OF 37

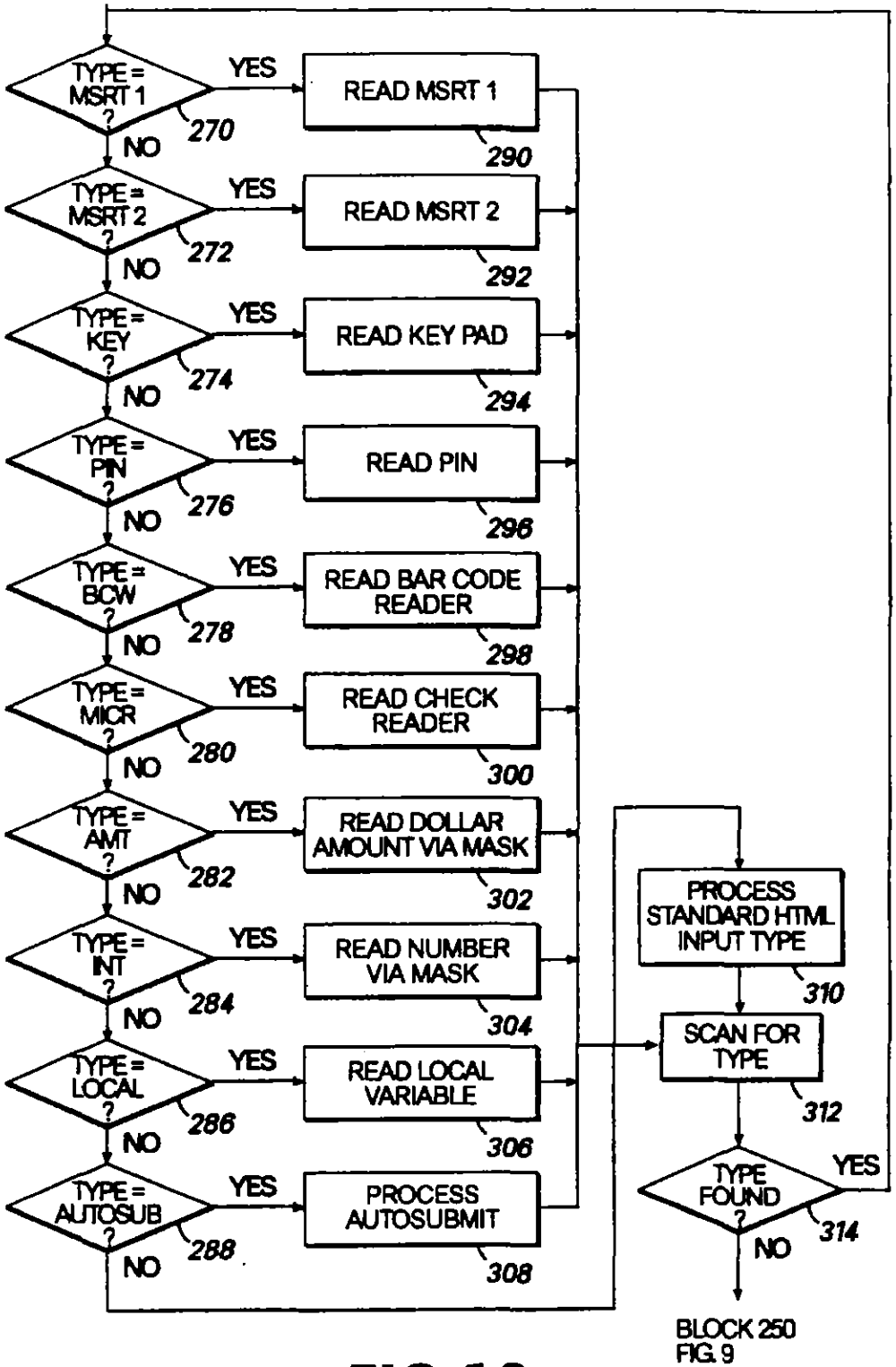

FIG. 10

EXHIBIT B
PAGE 10 OF 37

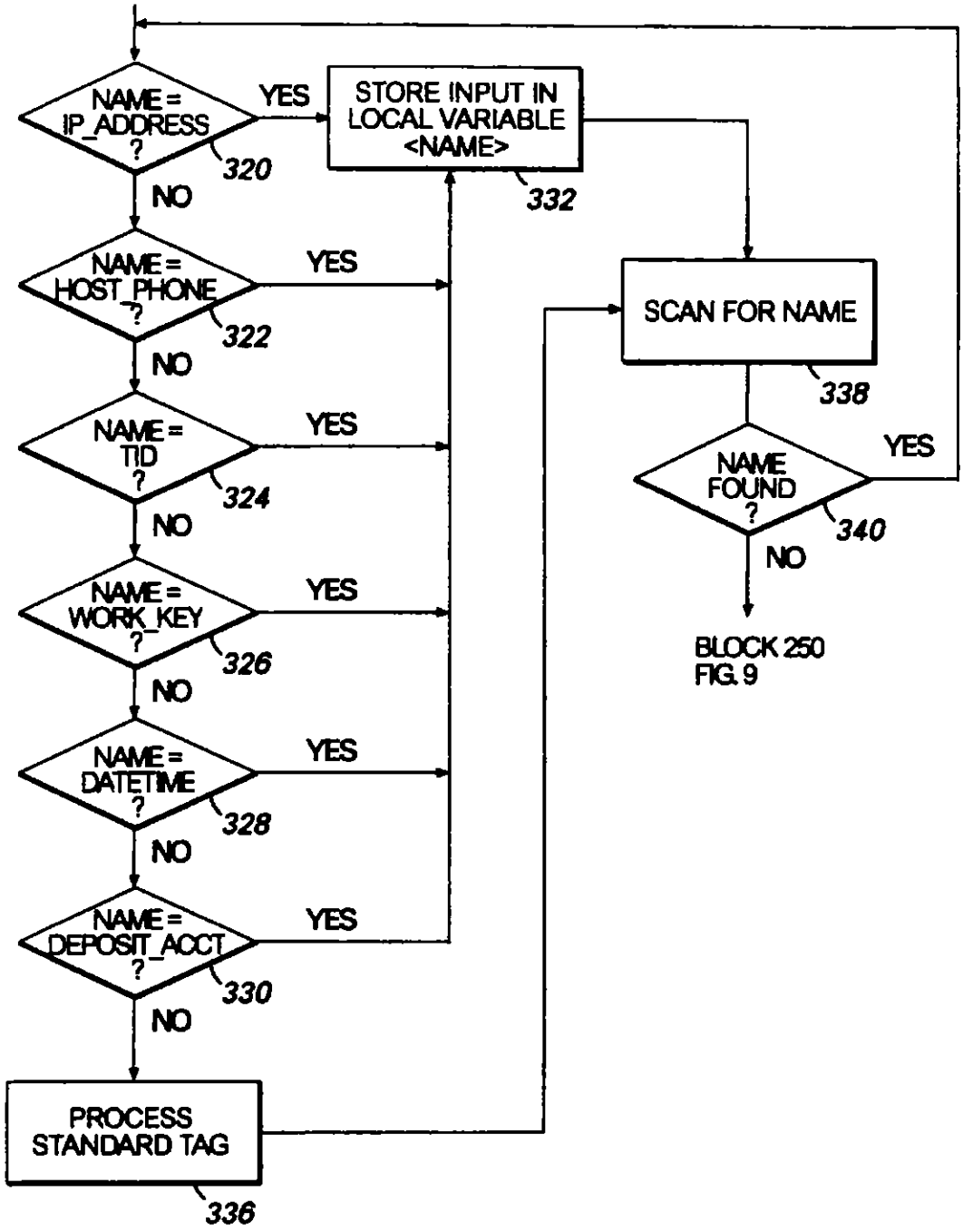

FIG.11

EXHIBIT B
PAGE 11 OF 37

1. <FORM ACTION=URL METHOD= GET>
2. <FORM ACTION=URL METHOD= POST>
3. <FORM ACTION=URL METHOD= SQL <database_name>

FIG.12

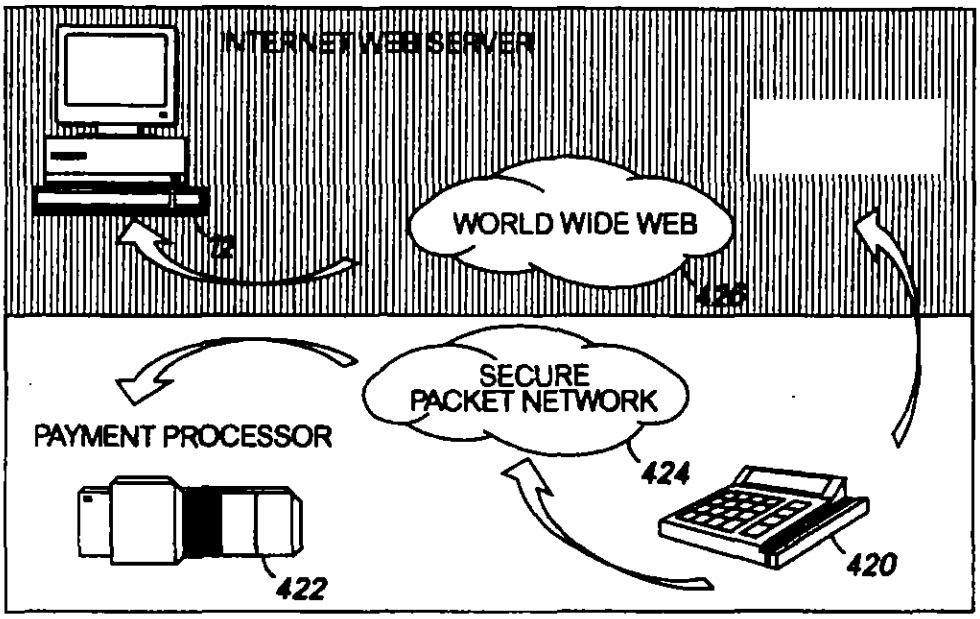

FIG.13A

<FORM ACTION=<filename> METHOD=PAYMENT>
<INPUT TYPE=AUTOSUBMIT>
</FORM>

FIG.13B

<FORM ACTION=dsinet METHOD=PAYMENT>
<INPUT TYPE=LOCAL NAME=DEPOSIT_ACCT VALUE=1234568902345678 90>
<INPUT TYPE=AUTOSUBMIT>
</FORM>

FIG.13C

EXHIBIT B
PAGE 12 OF 37

1.a. Transaction Request HTML+D

500

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
    METHOD=SQL
        "BEGIN TRAN
        IF NOT EXISTS ( SELECT substring(account, 1, 20) FROM auth_table)
            BEGIN
            INSERT TABLE=log_table VALUES=(getdate(),tid, substring (account, 1,20),
            substring( account, 22, 4), amount)
            SELECT * FROM log_table WHERE trandate = getdate()
            END
        ELSE  SELECT * FROM error_table WHERE error_no=1
        COMMIT TRAN">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER ACCOUNT NUMBER:
<INPUT TYPE="MSRT2" SIZE=40 NAME=account>
ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

1.b. Transaction Accepted Response

510

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
JUNE 1 1995          10:30AM      PURCHASE</P>
TERMINAL ID:                       999999999</P>
ACCOUNT NUMBER          99999999999999999999</P>
EXP DATE:                              99/99</P>
AMOUNT                              $9999.99</P>
AUTH NUMBER                        999999999</P>
</p>
-------------------------------------------</p>
          CUSTOMER SIGNATURE </p>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

1.c. Transaction Declined or Submit Error Response

520

```
<HTML>
<BODY>
DECLINED<MESSAGE>
</BODY>
</HTML>
```

FIG.14

EXHIBIT B
PAGE 13 OF 37

2.a. Transaction Request HTML+D

550

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
METHOD=SQL"BEGIN TRAN
        IF NOT EXISTS ( SELECT substring(account, 1, 20) FROM auth_table)
        BEGIN
                INSERT TABLE=log_table VALUES=(getdate(),tid, substring( account, 1,20),
                substring( account, 22, 4), amount)
        END
        ELSE  BEGIN
                        SELECT * FROM error_table WHERE error_no=1
                        RETURN
                END
        INSERT TABLE=order_table VALUES=( getdate(), cust_name,address,city, state, zip,
                part_code, unit_price, tax, ship_method, ship_chrg, unit_price + tax +
                ship_chrg, substring( account, 1, 20) ,substring( account, 22, 4))
        SELECT * FROM order_table WHERE trandate = getdate()
        COMMIT TRAN">
CUSTOMER NAME:
<INPUT TYPE="TEXT"SIZE=30 NAME=cust_name></p>
ADDRESS:
<INPUT TYPE="TEXT" SIZE=40 NAME=address></p>
CITY:
<INPUT TYPE="TEXT" SIZE=20 NAME=city><p>
STATE:
<INPUT TYPE="TEXT" SIZE=2 NAME=state>
ZIP:
<INPUT TYPE="TEXT" SIZE=10 NAME=address></p>
SCAN PART CODE:
<INPUT TYPE="BCW" SIZE=9 NAME=part_code></p>
ENTER UNIT PRICE:
<INPUT TYPE="AMT" SIZE=8 NAME=unit_price></p>
TAX:
<INPUT TYPE="AMT" SIZE=5 NAME=tax></p>
SHIPPING METHOD:
<INPUT TYPE="TEXT" SIZE=10 NAME=ship_method></p>
SHIPPING AMOUNT:
<INPUT TYPE="AMT" SIZE=5 NAME=ship_chrg></p>
SLIDE CARD:
<INPUT TYPE="MSRTZ" SIZE=40 NAME=account></p>
<INPUT TYPE="SUBMIT">
</FORM>
</BODY>
</HTML>
```

FIG.15A

EXHIBIT B
PAGE 14 OF 37

2.b. Transaction Accepted Response

555

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
ORDER # 9999999999 APPROVED</p>
JUNE 1 1995        10:30AM        PURCHASE</P>
TERMINAL ID:                      999999999</P>
NAME: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
ADDRESS:</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
CITY: XXXXXXXXXXXXXXXXXXXXXXXXXX</p>
STATE: XX                  ZIP:XXXXXXXXXX</p>
ACCOUNT NUMBER:            99999999999999999999</p>
EXP DATE:                                 99/99</p>
PART CODE:                            999999999</p>
UNIT PRICE:                            $9999.99</p>
SHIP METHOD:XXXXXXX CHARGE             $9999.99</p>
TOTAL AMOUNT:                          $9999.99</p>
AUTH NUMBER:                          999999999</p>
</p>

-------------------------------------------</p>
          CUSTOMER SIGNATURE</p>

</FORM>
</BODY>
</HTML>
```

2.c. Transaction Declined or Submit Error Response

560

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.15B

3.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
METHOD SQL
    "INSERT TABLE=order_table VALUES=( getdate(), cust_name,address,city, state, zip,
        part_code, unit_price, tax, ship_method, ship_chrg,unit_price + tax +
        ship_chrg, substring( account, 1, 20) ,substring( account, 22, 4))
    SELECT * FROM order_table WHERE trandate = getdate()">
<INPUT TYPE="LOCAL" NAME=tid>
CUSTOMER NAME:
<INPUT TYPE="TEXT" SIZE=30 NAME=cust_name></p>
ADDRESS:
<INPUT TYPE="TEXT" SIZE=40 NAME=address></p>
CITY:
<INPUT TYPE="TEXT" SIZE=20 NAME=city><p>
STATE:
<INPUT TYPE="TEXT" SIZE=2 NAME=state>
ZIP:
<INPUT TYPE="TEXT" SIZE=10 NAME=address></p>
ENTER PART CODE:
<INPUT TYPE="TEXT" SIZE=10 NAME=part_code></p>
ENTER UNIT PRICE:
<INPUT TYPE="AMT" SIZE=8 NAME=unit_price></p>
TAX:
<INPUT TYPE="AMT" SIZE=5 NAME=tax></p>
SHIPPING METHOD:
<INPUT TYPE="TEXT" SIZE=10 NAME=ship_method></p>
SHIPPING AMOUNT:
<INPUT TYPE="AMT" SIZE=5 NAME=ship_chrg></p>
<INPUT TYPE="SUBMIT" >
</FORM>
</BODY>
</HTML>
```

FIG.16A

EXHIBIT B
PAGE 16 OF 37

3.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
ORDER # 9999999999 APPROVED</p>
JUNE 1 1995 10:30AM PURCHASE</P>
TERMINAL ID: 99999999</P>
NAME: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
ADDRESS:</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
CITY: XXXXXXXXXXXXXXXXXXXXXXX</p>
STATE: XX ZIP:XXXXXXXXX</p>
PART CODE: 999999999</p>
UNIT PRICE: $9999.99</p>
SHIP METHOD:XXXXXXX CHARGE: $9999.99</p>
TOTAL AMOUNT: $9999.99</p>
</FORM>
<FORM ACTION=<file_name> METHOD=PAYMENT>
<INPUT TYPE=AUTOSUBMIT>
</FORM>
</BODY>
</HTML>
```

3.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.16B

EXHIBIT B
PAGE 17 OF 37

4.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=SCR1 METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
SLIDE CARD:
<INPUT TYPE="MSRT2" SIZE=40 NAME=track2>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="LOCAL" NAME=work_key>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

4.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99 TIME:9999A</P>
ACCOUNT NUMBER 9999999999999999999</p>
EXP DATE: 99/99</p>
AMOUNT: $9999.99</p>
AUTH NUMBER 999999999</p>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

4.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.17A

EXHIBIT B
PAGE 18 OF 37

5.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=SCR2 METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
ENTER PIN:
<INPUT TYPE="PASSWORD" SIZE=4 NAME=pin>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="LOCAL" NAME=work_key>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

5.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99 TIME:9999A</P>
ACCOUNT NUMBER 9999999999999999999</P>
EXP DATE: 99/99</P>
AMOUNT $9999.99</P>
AUTH NUMBER 999999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

5.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.17B

EXHIBIT B
PAGE 19 OF 37

6.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=host_URL METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
SLIDE CARD
<INPUT TYPE="MSRT2" SIZE=40 NAME=track2>
<INPUT TYPE="PIN" SIZE=4 NAME=pin>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

6.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99 TIME:9999A</P>
TERMINAL ID: 99999999</P>
ACCOUNT NUMBER 9999999999999999999</P>
EXP DATE: 99/99</P>
AMOUNT $9999.99</P>
AUTH NUMBER 999999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

6.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
<INPUT TYPE="LOCAL" NAME=work_key VALUE="99999999999999999">
</BODY>
</HTML>
```

FIG.18

EXHIBIT B
PAGE 20 OF 37

7.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
      METHOD=SQL
            "IF EXISTS ( SELECT account FROM check_table)
                  SELECT * FROM check_table WHERE account = DDAaccount
             ELSE SELECT * FROM error_table WHERE error_no=1">
<INPUT TYPE="LOCAL" NAME=tid>
SCAN CHECK:
<INPUT TYPE="MICR" SIZE=20 NAME=account>
 ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

7.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99                          TIME:9999</P>
TERMINAL ID:                           999999999</P>
ACCOUNT NUMBER   99999999999999999999</P>
AMOUNT                                   $9999.99</P>
AUTH NUMBER                            999999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

7.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.19

EXHIBIT B
PAGE 21 OF 37

8.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
        METHOD=SQL
                * BEGIN TRAN
                IF NOT EXISTS ( SELECT substring(account,51 ,20) FROM auth_table)
                        BEGIN
                        SELECT cur_bal FROM cust_tbl WHERE substring(account,51 ,20)=account
                        SELECT amount = amount - ( points / .01 )
                        SELECT cur_bal = cur_bal + ( amount * .01 )
                        UPDATE TABLE=cust_tbl VALUES=( getdate(), account, cur_bal - points )
                        SELECT * FROM log_table WHERE trandate = getdate()
                        INSERT TABLE=log_table VALUES=( getdate(), tid, substring(account,51 ,20),
                        substring( account, 72, 4), amount)
                        END
        ELSE SELECT * FROM error_table WHERE error_no=1"
        COMMIT TRAN">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER ACCOUNT NUMBER:
<INPUT TYPE="MSRT1" SIZE=90 NAME=account>
ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
REDEEM POINTS?
<INPUT TYPE="INT" SIZE=6 NAME=points>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

8.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
JUNE 1 1995        10:30AM              PURCHASE</P>
TERMINAL ID:                           999999999</P>
ACCOUNT NUMBER        99999999999999999999</P>
EXP DATE                                   99/99</P>
AMOUNT                                  $9999.99</P>
AUTH NUMBER                            999999999</P>
</p>
- - - - - - - - - - - - - - - - - - - - - - - </p>
          CUSTOMER SIGNATURE</p>
</p>
THANK YOU!</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
POINTS REDEEMED:                          999999</p>
POINTS EARNED:                            999999</p>
CURRENT POINT BALANCE:                    999999</p>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

8.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <MESSAGE>
</BODY>
</HTML>
```

FIG.20

EXHIBIT 8
PAGE 22 OF 37

9.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
      METHOD=SQL
            "SELECT fields FROM table WHERE condition">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER SEARCH TABLE NAME:
<INPUT TYPE="TEXT" SIZE=10 NAME=table>
ENTER SEARCH FIELD NAMES:
<INPUT TYPE="TEXT" SIZE=100 NAME=fields>
ENTER SEARCH CONDITION:
<INPUT TYPE="TEXT" SIZE=50 NAME=condition>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

9.b. Transaction Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
FIELD1  FIELD2  FIELD3  ***** FIELDN  </p>
------  ------  ------        ------  </p>
XXXXX   XXXXX   XXXXX         XXXXX   </p>
XXXXX   XXXXX   XXXXX         XXXXX   </p>
•
•
XXXXX   XXXXX   XXXXX         XXXXX   </p>
</FORM>
</BODY>
</HTML>
```

FIG.21

EXHIBIT B
PAGE 23 OF 37

10.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM  ACTION=dbase_URL
        METHOD=SQL
        * INSERT TABLE=log_table VALUES=( getdate(), tid, gross_sales, opn_chks, voids,
        emp_disc, mgr_disc, vip_card, man_over, coupons, sales_tax, c_dep1, c_dep2,
        c_dep3, c_dep4, chg_fund,cc_dep, batch_no, chrg_sales, paid_outs, co_sales,
        cc_sales, te_sales, gross_sales - opn_chks - voids - emp_disc - mgr_disc - vip_card-
        man_over - coupons - sales_tax, gross_sales - opn_chks - voids -
        emp_disc - mgr_disc - vip_card - man_over - coupons - c_dep1 - c_dep2 -
        c_dep3 - c_dep4 - chg_fund - cc_dep - batch_no - chrg_sales - paid_outs)
        SELECT * FROM log_table WHERE trandate = getdate()">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER GROSS SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=gross_sales>
ENTER  OPEN CHECKS:
<INPUT TYPE="INT" SIZE=7 NAME=opn_chks>
ENTER VOIDS:
<INPUT TYPE="INT" SIZE=7 NAME=voids>
ENTER EMP DISCOUNTS:
<INPUT TYPE="INT" SIZE=7 NAME=emp_disc>
ENTER MGR DISCOUNT:
<INPUT TYPE="INT" SIZE=7 NAME=mgr_disc>
ENTER VIP CARD:
<INPUT TYPE="INT" SIZE=7 NAME=vip_card>
ENTER MANUAL OVERRINGS:
<INPUT TYPE="INT" SIZE=7 NAME=man_over>
ENTER COUPONS:
<INPUT TYPE="INT" SIZE=7 NAME=coupons>
ENTER SALES TAX:
<INPUT TYPE="AMT" SIZE=8 NAME=sales_tax>
ENTER CASH DEPOSIT 1:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep1>
ENTER CASH DEPOSIT 2:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep2>
ENTER CASH DEPOSIT 3:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep3>
ENTER CASH DEPOSIT 4:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep4>
ENTER CHANGE FUND:
<INPUT TYPE="AMT" SIZE=8 NAME=chg_fund>
```

FIG.22A

EXHIBIT B
PAGE 24 OF 37

```
ENTER CC DEPOSIT:
<INPUT TYPE="AMT" SIZE=8 NAME=cc_dep>
ENTER BATCH #:
<INPUT TYPE="INT" SIZE=3 NAME=batch_no>
ENTER CHARGE SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=chrg_sales>
ENTER PAID OUTS:
<INPUT TYPE="INT" SIZE=8 NAME=paid_outs>
ENTER CARRY OUT SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=co_sales>
ENTER CREDIT CARD SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=cc_sales>
ENTER TAX EXEMPT SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=te_sales>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

10.b. Transaction Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
JUNE 1 1995        10:30AM        DAILY REPORT</P>
TERMINAL ID:                      999999999</P>
GROSS SALES                       999999.99</P>
 VOIDS                      99      99999.99</P>
 EMP DISCOUNTS              99      99999.99</P>
 MANAGER DISCOUNTS          99      99999.99</P>
 VIP CARD                   99      99999.99</P>
 COUPONS                    99      99999.99</P>
 MANUAL OVERRINGS           99      99999.99</P>
 SALES TAX                         999999.99</P>
 CASH DEPOSIT 1                    999999.99</P>
 CASH DEPOSIT 2                    999999.99</P>
 CASH DEPOSIT 3                    999999.99</P>
 CASH DEPOSIT 4                    999999.99</P>
 CASH DEPOSIT 5                    999999.99</P>
 CHANGE FUND                       999999.99</P>
 CC DEPOSIT                 999     999999.99</P>
 CHARGE SALES                      999999.99</P>
 PAID OUTS                   99      99999.99</P>
 CARRY OUT SALES                   999999.99</P>
 CREDIT CARD SALES                 999999.99</P>
 TAX EXEMPT SALES                  999999.99</P>
 ----------------------------------------</P>

 NET SALES                          9999999</P>
 OVER/SHORT                         9999999</P>
</FORM>
</BODY>
</HTML>
```

FIG.22B

EXHIBIT B
PAGE 25 OF 37

11.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM  ACTION=MAIL TO: mail_to>
 ENTER MAIL ADDRESS:
<INPUT TYPE="TEXT" SIZE=20 NAME=mail_to>
ENTER MESSAGE:
<INPUT TYPE="TEXT" SIZE=100>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

FIG.23

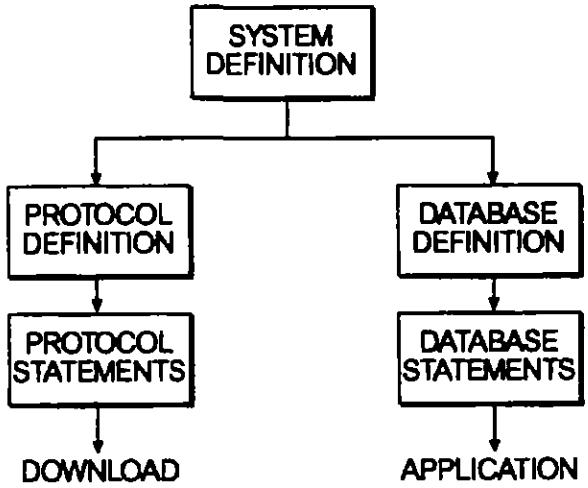

FIG.24A

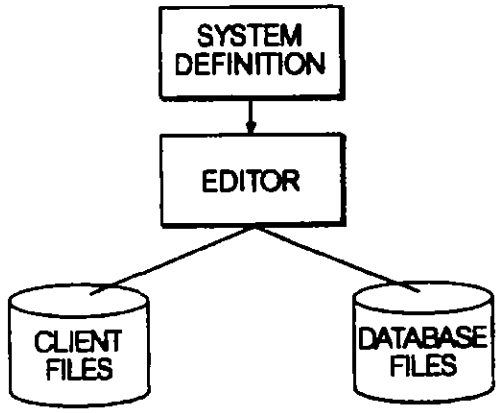

FIG.24B

OPEN NETWORK SYSTEM FOR I/O OPERATIONS WITH NON-STANDARD I/O DEVICES UTILIZING EXTENDED PROTOCOL INCLUDING DEVICE IDENTIFIER AND IDENTIFIER FOR OPERATION TO BE PERFORMED WITH DEVICE

CROSS REFERENCES TO RELATED APPLICATIONS

This application is a continuation of application Ser. 08/493,772 filed Jun. 22, 1995 which application is now U.S. Pat. No. 5,742,845.

FIELD OF THE INVENTION

This invention relates to data transaction systems, and more particularly, to data transaction systems using non-standard input/output devices.

BACKGROUND OF THE INVENTION

Data transaction systems which communicate with a plurality of remote terminals to transfer information used to complete a transaction or compile a database are well known. Typically, such systems include a central transaction processing system which may maintain a database of information such as customer or consumer data. Exemplary information in such a database may include customer identification, customer account numbers, credit limits and/or account balances from which a customer may draw. The central transaction processing system is typically coupled to a plurality of remote transaction or data input terminals. Transaction computers may include special purpose devices such as automatic teller machines (ATMs), point of sale (POS) terminals, credit card terminals, and screen phone terminals. Screen phone terminals are devices which integrate a telephone with an ATM-like device and possibly a magnetic card swipe reader. Data input terminals may include personal computers (PCs) interfaced to data collection devices or special purpose data collection terminals or monitors.

In these known data transaction systems, a user usually initiates a transaction by requesting access to funds in an account or from a credit line maintained by the central processing system. The request is transmitted to the central processing system which performs a verification to determine whether the user is a valid user of the system, has an account within the system, and that the amount of the transaction is within the limits of the consumer's credit line or that the user has the requested funds available in an existing account monitored by the central processing system. The central processing system then transmits authorization for or denial of the transaction to the remote terminal. In response to the message from the central processing system, the remote terminal dispenses cash (for an ATM) or the merchant provides the goods being purchased to the user if the authorization message indicates that the consumer's funds will be transferred to the merchant's account. Similar communication exchanges occur in data systems where electronic documents and other information are provided to a central site for compilation or processing. Consequently, this background discussion applies to all such transaction and data systems. Though the remainder of the discussion is directed to transaction systems, the reader should appreciate that the comments also apply to data systems as well.

The remote terminals may be coupled to the central processing system in several ways. For example, in some

ATM systems, the ATMs are coupled to the central processing system through dedicated telephone or other data communication lines. These systems are preferred because they provide a relatively high degree of security since the dedicated data line coupling the central processing system to the ATM is not generally accessible by members of the public. The physical security of the dedicated data line is, however, expensive because no other traffic may utilize the line. Thus, the cost of leasing the dedicated line to an ATM with relatively low volumes of transactions may yield a high communication cost per transaction.

In an effort to reduce the communication cost per transaction, some transaction or data systems utilize telephone lines through a publicly-switched telephone network (PSTN) which may be accessed by other members of the public. Specifically, devices such as credit card terminals and screen phone terminals typically include a modem which converts the digital messages of the remote terminal into frequency modulated analog signals which may be transmitted over telephone lines to a modem at the central processing system. In other systems, the terminal may communicate digital data directly over ISDN lines of the PSTN to the central processing system. This line of communication between a remote terminal and the central processing system is performed by having the remote terminal dial a telephone number associated with the central processing system to establish communication with the central processing system. This type of communication path is relatively secure because the switching networks for the communication traffic through the PSTN are not readily accessible by the public and during the course of the financial transaction, only the central processing system and remote terminal are on the line.

Regardless of the communication method used to couple the central processing system to the remote terminals, the protocol and data formats used between the devices is typically proprietary. That is, the operator of each financial transaction system designs its own protocol and data message format for communication with the processor at the central site or generates a variant within a standard such as those established by the ANSI committee or the like for such communication. As a result, the remote terminals must include software that supports each operator's protocol and message formats in order to be compatible with an operator's central site. For example, application software in a credit terminal such as the TRANZ330, TRANZ380, or OMN1390 manufactured by VeriFone implement one or more of the communication protocols and formats for National Data Corporation (NDC), VISANET, MASTERCARD, BUYPASS, and National Bancard Corporation (NaBANCO) system processors in order to support transactions with the most popular transaction centers. Thus, the communication software absorbs a significant amount of terminal resources which could be used to support other terminal operations.

A related problem arises from the expanding home banking market. A customer of home banking system typically uses a screen phone terminal or a personal computer (PC) having a modem to establish communication through a PSTN to a central transaction processing system. Again, the operator of the central processing system must provide information regarding the data message formats for communicating with the central processing system to a vendor of software for the home banking terminals or must provide that software to its customers. As a result, home banking customers must purchase software to communicate with each banking system of which the customer wants to be a

EXHIBIT B
PAGE 27 OF 37

member. This cost and the need to install additional communication programs may make some consumers reluctant to be a member of more than one banking system or to change banking systems.

A communication system becoming increasingly popular and which provides standardized communication is the Internet. The Internet is an open network of networks which communicate through a variety of physical communication devices such as telephone lines, direct communication lines, and the like. Each network is coupled to the main Internet network for communication through a host computer supporting a TCP/IP router or bridger. The host computer typically includes a program, frequently called a Web server, which acts as a gateway to resources at the host computer which may be resident on the host computer or a network coupled to the host computer. Each server has an address identifying the location of the resources available through the Web server. The router recognizes communication for the server and directs the message to the server or it recognizes that the communication should be forwarded to another server. As a result, communication within the Internet may be point-to-point, but more likely, the communication path is a somewhat circuitous one with the information passing through the routers of multiple servers before reaching its final destination.

A number of message protocols and formats have been developed for the Internet. The physical communication protocol and data message format is the Transport Control Protocol/Internet Protocol (TCP/IP). The TCP/IP protocol involves multiple layers of encapsulating headers containing communication information which are used to provide byte streams or datagram communications to computers on the networks coupled to the Internet. Encapsulated within TCP/IP headers are protocols which are used to format the data messages or transfer data from one computer to another computer coupled to the Internet. These protocols include File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), Telnet, and Hyper Text Transport Protocol (HTTP). The advantage of these protocols is that each provides a standardized communication format for transferring information between computers on the Internet. These protocols are typically called open system protocols as they are publicly known and may be utilized by any programmer to develop programs for communicating with another computer coupled to the Internet. These non-proprietary protocols have contributed to the acceptance of using the Internet as an open network for coupling computer networks together. While the Internet provides an open network for computer communication with publicly accessible protocols and formats, the Internet suffers from a number of limitations which reclude its effective use as a transaction or data system which uses non-standard I/O terminals and devices. First, circuitous communication presents a number of security issues for such a system. For example, a Web server could incorporate a router which examines the address of each message coming through it and upon recognizing an address associated with a central transaction processing system, copy the data message for the unauthorized retrieval of customer-sensitive information such as account numbers and personal identification numbers (PINs) which may be contained in the message.

A second limitation of open networks such as the Internet is that communication on such networks is only supported for computers acting as servers or clients. Specifically, all of the protocols and formats are constructed for standard input/output (I/O) operations for a PC terminal. That is, text information is directed to a standard monitor screen, user input is expected from a standard keyboard, and files are transferred to standard peripherals such as a hard disk or diskette drive. Especially absent is the ability in open network protocols for communication with devices that only use communication interfaces such as RS-232C. As a result, communication over the Internet is primarily performed with standard PCs through network communication methods and interfaces.

This presents a number of problems for home banking or for interfacing non-standard I/O terminals such as credit card terminals or screen phones to open networks such as the Internet either directly or through a PC. Generally, non-standard I/O devices are devices which interface to a PC through a port not normally used for networks, such as a RS-232C port, or are devices which have limited input and output capabilities such as small screen displays or ten keypads. These devices are not supported on the Internet because servers use protocols that communicate with PCs supporting standard QWERTY keyboards and standard monitors. Consequently, users are limited to entering account numbers and the like through a keyboard of a PC-like device for processing at a central transaction processing system. To request a transaction, one need only have a person's credit card account number. If the credit card number had to be input through a magnetic card reader, unauthorized access to a customer's account would be less likely since physical possession of the credit card would be required to initiate the transaction.

Another limitation of the standard I/O devices currently supported by the open network protocols is the lack of encryption. For example, PIN pads, which are typically incorporated in ATMs, automatically encrypt in hardware a PIN entered by a user. Such devices typically encrypt the number by implementing a data encryption standard (DES) algorithm in hardware before the PIN is transmitted or stored. When a standard keyboard is used to input the PIN, no hardware encryption is performed and, as a result, an unencrypted copy of the PIN is provided to the memory of the PC. Storage of unencrypted PINs is in contravention of current banking regulations. If PIN pads could be read via Internet protocols, then such a lapse in PIN security would be less likely to occur.

Another I/O device not supported on open networks are smart cards which are increasing in use. Smart cards include a processor and memory in which information regarding the amount of funds in a particular account, a transaction history, account numbers, and customer data may be stored. The card may be read through a smart card reader which is a computer having a processor and memory but usually provided with non-QWERTY keypads and limited displays. A transaction processor may validate a card owner through a PIN provided through a keypad, determine the amount of money remaining on the card and debit the card itself for a transaction amount by communicating with the smart card reader with one of the proprietary protocols discussed above. Such information is not readily obtainable by the owner of the card and so cannot be entered through a keyboard or the like. Smart card readers are non-standard devices which may be coupled to a PC through a COMM1 or COMM2 port. However, none of the standard protocols and message formats for open network communications currently provide I/O operations for such devices.

All systems which attempt to provide three party communication to execute an electronic transaction suffer from a number of limitations which present risks greater than those in a normal transaction performed at the point of sale. In a typical point of sale (POS) transaction, the consumer

hands a debit or credit card to a merchant's agent who may examine the card for security markings such as holograms, watermarks, or a cardholder signature. The agent then places the card into a reader for acquiring information from the card and, in some cases, have the consumer enter a PIN into a PIN entry device which encrypts the PIN in a hardware implemented scheme. If the PIN is entered, it is transmitted with the information from the card to a processing center, typically in one of the formats discussed above, under a X.25 protocol or the like. The processing center returns an authorization granted or denied message. The reader typically has a printer coupled to it through an RS-232C port or the like and a purchase agreement is printed. The consumer signs the agreement, the merchant's agent may verify the signature, and the merchant retains an original of the agreement and the consumer a copy. In this scenario, the merchant has initiated the communication to the processing center. The safeguards noted above permit the processing center to charge a merchant a lower processing fee than when a consumer initiates a transaction. Consumer initiated transactions present a greater risk because the consumer provides an agent an account number in a telephone conversation or non-encrypted DTMF transmission. Thus, there is no card inspection, signature verification, or PIN verification. As a result, such transactions are limited to credit cards because debit cards require that the cardholder be present to enter a PIN into an appropriate PIN entry device.

What is needed is a system that permits consumers remote from a merchant to order goods and present payment in a secured manner so the merchant's risk and processing costs, as well as a cardholder's exposure to fraud, is reduced. What is needed is a way for a processing center to communicate through an open network with non-standard I/O devices such as credit card terminals, personal digital assistants, and screen phone terminals or with non-standard I/O devices coupled to the open network through a PC or the like. What is needed is a transaction or data system which utilizes an open network such as the Internet to support electronic transactions or data compilation in a secure manner without undue limitation as to the devices with which communication may be made.

SUMMARY OF THE INVENTION

The present invention provides transaction and data systems which may be implemented on an open network such as the Internet. The system comprises a server for communicating in an open network protocol and a plurality of input/output (I/O) devices coupled to the server through an open network, the I/O devices communicating with the server in the extended open network protocol that supports communication with non-standard I/O devices over the open network. The system of the present invention provides a server with the capability of communicating with a number of I/O devices useful in transaction and data systems which heretofore have been unsupported on an open network system such as the Internet.

The system of the present invention is implemented by extending present open network communication protocols and data message formats to communicate with non-standard I/O devices either coupled to an open network as a client or coupled to an open network through a client, such as a PC, credit card terminal, screen phone, or PDA. That is, commands which are compatible with the communication schema of a presently-implemented protocol for the Internet are used and additions are made to commands implemented within the control structure of that existing protocol to support non-standard I/O device communication. At the server, the extended protocol is further supported by a common gateway interface (CGI) which converts the communication from a non-standard I/O device to a format which is compatible with a transaction or data application program which may be executed on the server or a computer coupled to the server. In this manner, the CGI permits the processing of the extended capability commands to be segregated from the communication functions performed by the server.

Preferably, the server and the I/O devices communicate through an Internet protocol and most preferably, the Hyper Text Transport Protocol (HTTP), to exchange data between an application program and non-standard I/O devices over an open network. Although HTTP is the preferred protocol used to implement the present invention, other protocols such as Telnet or SMTP, for example, may also be extended in a similar manner. Specifically, the HTTP protocol is expanded to communicate with printers, magnetic card readers, credit card terminals, smart card readers, check readers, PIN pads, bar-code readers, PDAs, or the like, and includes a command which instructs a non-standard I/O device to disconnect from the open network and recouple to a transaction processing system to transfer funds from a consumer account to a merchant account through a PSTN or dedicated data line. By using these extended capability commands within HTTP, a processing system may operate on an open network such as the Internet and communicate with transaction or other data I/O devices which have not previously been able to couple to such open networks. Such a system may be used to execute a transaction between a consumer and a merchant so the merchant receives remittance information in a timely manner. The system permits the consumer to initiate a transaction and order from a merchant and then use a more secure link supported by PIN entry devices or the like to reduce the risk of fraud for the transaction.

Because the server may communicate through such open networks with non-standard I/O devices, the transaction or data processing system is available for the ever-expanding market available through the Internet. Such a system is able to communicate with non-standard I/O devices in myriad locations such as retail establishments or in consumers' homes. For example, a consumer may utilize the standard capability of an Internet protocol to communicate with a server that provides information regarding services or goods for sale over the Internet and then consummate a sales transaction by using the extended capability of the Internet protocol. Such a home consumer could provide transaction data through a smart card reader coupled to a COMM1 or COMM2 port of a PC. A database program executing at the server for the central processing site may accept product ordering information from a non-standard keypad or touch screen associated with a screen phone terminal at the remote site and then communicate with the smart card reader to consummate the transaction. Such a transaction system requires that the consumer have physical possession of the smart or credit card and not simply knowledge of the account number. Likewise, the server would be able to communicate with a PIN pad or the like to ensure the hardware encryption of PINs and other data before it is transmitted to the server site. Such a system is less susceptible to consumer fraud.

Another feature of the present invention is a PAYMENT command implemented in the extended Internet protocol that directs a non-standard I/O device or a PC interfaced with such devices to communicate with a transaction processor through an alternative communication link. In one

form, the PAYMENT command is used by a merchant terminal to submit a consumer's account number with a merchant deposit account number through a PSTN network or the like to the processing center. In another form of the PAYMENT command, a client program in a consumer's terminal receives an account number for a merchant account from a merchant's server with the PAYMENT command. On receipt of this command, the client program suspends its operation and passes the account number to a conventional bank processing program co-resident in memory. The bank processing program establishes a standard communication link with a transaction processing system through a dedicated data line or a PSTN network. Using that communication link, the bank processing program executes a commercial transaction using a standard VISA protocol or the like. The consumer may use a magnetic stripe reader and a PIN entry device to improve the security of the data transmission. The transaction center may transmit remittance data over the open network to the merchant so the merchant is apprised of payment and ships the ordered product. Once this consumer initiated transaction is complete, the bank processing program terminates and returns control to the client program which may terminate communication with the open network or retrieve information from another server on the open network for another transaction. In this way, the user may use the open network for non-confidential communication such as collecting product information, pricing, and product availability. This information may be collected quickly and efficiently using the extended Internet protocol. The conventional bank processing program and more secure communication links may then be used for the confidential information required for the transaction. Thus, the present invention is able to combine the features and advantages of the Internet with the more secure communication link and data security enhancing devices of systems presently known.

Preferably, an editor is provided which permits a user to define an application database table with data fields, define client application data fields, and define the integrated forms for communicating data between the defined database tables and a client application. The editor verifies the syntax of the user generated integrated forms containing extended Internet protocol statements and client application statements. The editor ensures that the variable names for the client application and the data fields for the database application correspond. Following the generation of the integrated form, the editor parses the integrated form to segregate the database language statements from the extended Internet protocol statements. A database language identifier is substituted in the Internet protocol statements for the database statements contained in the integrated form. The Internet protocol statements are downloaded as a file which is interpreted by the client program for the collection and submission of data from non-standard I/O devices to the database application. The database language statements segregated from the extended Internet protocol statements are placed in a second file which is named to correspond to the database table defined by the user. The CGI application recognizes the database language identifier contained in the returned forms of the Internet protocol statements. The CGI application correlates the database identifier with the file previously generated by the editor which contains the database command statements. The application then inserts the data from the returned form into the database command statements and provides the re-integrated database command statements to the database application. In this manner, the database may be queried by or retrieve data from the non-standard I/O device. In the most preferred embodiment, the editor permits a user to develop integrated forms comprised of the extended HTML language and standard query language (SQL) database application statements. In this manner, the user does not have to manually generate the SQL commands, the HTML commands, and carefully correlate the data fields of the two commands in order to implement a transaction between a client and a database.

These and other advantages and features of the present invention may be discerned from reviewing the accompanying drawings and the detailed description of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention may take form in various components and arrangement of components and in various steps and arrangement of steps. The drawings are only for purposes of illustrating a preferred embodiment and are not to be construed as limiting the invention.

FIG. 1 is a diagram of an open network system in which the present invention is utilized;

FIG. 2 is a diagram of the format of the FORM and INPUT tags implemented in the preferred embodiment of the present invention;

FIG. 3 is a diagram of the preferred SQL commands supported in the preferred embodiment of the present invention;

FIG. 4 is a flowchart of the high level processing of the client program which interprets the HTML files of the preferred embodiment of the present invention;

FIG. 5 is a flowchart of the HTML file processing performed by the client program of the preferred embodiment of the present invention;

FIG. 6 is a flowchart of the attribute processing for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. 7 is a flowchart of the processing of the ACTION attribute for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. 8 is a flowchart of the processing for the METHOD attribute for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. 9 is a flowchart of the attribute processing for the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. 10 is a flowchart of the processing for the TYPE attribute for the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. 11 is a flowchart of the processing for the NAME attribute of the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. 12 is a diagram of the format for the ACTION attribute for the FORM tag performed by the common gateway interface between the Web server and an application program;

FIG. 13A is a diagram of the possible communication paths which may be used by an I/O device according to the principles of the present invention;

FIG. 13B shows an exemplary FORM tag and INPUT tag for the PAYMENT method implemented in a merchant's terminal according to the principles of the present invention;

FIG. 13C shows an exemplary FORM tag and INPUT tag for the PAYMENT method implemented in a consumer's terminal according to the principles of the present invention;

FIG. 14 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to

generate the HTML files for the client program and the SQL files for the application program for a card initiated payment authorization and capture transaction;

FIGS. **15A** and **15B** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a bar code reader input with card-initiated payment authorization transaction;

FIGS. **16A** and **16B** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a key input order with secure payment transaction;

FIG. **17A** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a smart card debit (Type 1) transaction;

FIG. **17B** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a smart card debit (Type 2) transaction;

FIG. **18** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a debit card transaction;

FIG. **19** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a check verification transaction;

FIG. **20** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a customer frequency transaction;

FIG. **21** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for an item search transaction;

FIGS. **22A** and **22B** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for retail store end of day reporting;

FIG. **23** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a store reporting an e-mail transaction;

FIG. **24A** is a diagram of a manual development process for the files interpreted by the client program and the files interpreted by the application program in accordance with the principles of the present invention; and

FIG. **24B** is a diagram of the generation of the files interpreted by the client program and the files interpreted by application program performed by an editor constructed in accordance with the principles of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

A transaction or data system constructed in accordance with the principles of the present invention is shown is FIG. **1**. The system **10** includes a Web server **12** which is coupled to an open network **14** such as the Internet for communication with various I/O devices and terminals. For example, the I/O devices which may be coupled directly to network **14** include standard I/O devices already supported by Internet protocols such as PCs **30** and non-standard I/O devices such as a screen phone terminal **16**, a personal digital assistant (PDA) **18**, and a credit card terminal **20**. Other exemplary non-standard I/O devices such as smart card reader **32**, personal identification number (PIN) pad **34**, magnetic card swipe reader **36**, printer **38**, or the like, may be coupled to PCs through non-standard I/O ports such as COMM1 and COMM2 ports or to other non-standard I/O devices such as phone terminal **16**, PDA **18**, or credit card terminal **20**. Typically, these devices are coupled to PCs or devices **16**,**18**, or **20** through an interface such as a RS-232C interface. Merchants or other vendors may use a Web server **2** to couple to network **14** to communicate with the devices and processing system **40**.

The Web server **12** is preferably coupled to a Common Gateway Interface (CGI) application **28** which converts and communicates the data and commands between the devices on network **14** and the processing system **40** so the I/O devices do not have to use the database command language to interact with the database. System **40** and the devices may communicate directly if they are implemented in the same language or if a user implements a communication interface such as CGI **28** that correlates data fields in the client with those in system **40**. Server **12**, CGI **28**, and the applications supporting system **40** may all reside on a single host computer or they may reside on separate computers coupled together by a local area network (LAN) or a wide area network (WAN). Preferably, the application interfaces with a database which supports Open Data Base Connectivity (ODBC) and Structured Query Language (SQL).

The communication sessions between the I/O devices coupled to the open network **14** and the Web server **12** are generally conducted in the same fashion as Internet protocol communication sessions are currently performed. That is, the I/O device establishes a communication connection with Web server **12**, sends a request to the Web server, the Web server responds to the request and the I/O device or server closes the connection. Preferably, the non-standard I/O devices or PCs interfaced to such devices selectively couple to a local access port on the open network **14** through a local modem/ISDN connection. In this manner, the device is only coupled to the open network **14** when a transaction or a data operation is to be performed. While connected to the open network **14**, a device may access a number of servers to accomplish a purpose. For example, a device may couple to a local access port and communicate with a first server to check inventory levels at a site, communicate with a second server to order stock for the inventory, and communicate with a third server to settle payment for the ordered goods. When all aspects of the transaction are complete, the connection with the local access port is terminated. In the preferred embodiment of the present invention, the protocol used to transport data messages between Web server **12** and the I/O devices coupled to the open network **14** is the Hyper Text Transport Protocol (HTTP), although other open system protocols utilized on the Internet may be used.

In standard HTTP protocol, a client program executing in one of the I/O devices may initiate communication with a server by sending a query message of the format:

http://<host>:<port>/<path>?<search part>

The message identifies the client as seeking communication with a HTTP server at the host address on the specified

port. In the HTTP protocol, the default value for the port is 80 and the host address is the Internet protocol (IP) address of the type well-known in the art. The path value selects the file in the HTTP server which is activated in response to the message and the search part specifies a query for the selected file. In the initial communication, the query may be omitted so that the selected host file responds to the client program before a query is processed.

In the present invention, the client program uses a similar message to initiate a transaction or data operation, except that database commands are preferably embedded in a file at the server 12 and not in the "search part" of the command, although search parts may be constructed in accordance with the principles of the present invention that support non-standard I/O devices. Preferably, the client program interprets Hyper Text Markup Language (HTML) files containing HTML commands for communicating data between non-standard I/O devices and server 12.

Most preferably, the HTML commands contain identifiers which are used by the CGI to place data returned in the forms of the HTML commands into database commands for queries or data insertions for the database. HTML is a command language well known for the retrieval and display of electronic documents for standard I/O devices such as PCs supported by full screen monitors, QWERTY keyboards, and standard peripherals such as hard disk drives and diskette drives. Standard HTML commands use text and previously known commands that reference Universal Resource Locators (URLs) to support the communication of electronic documents. These documents are files which may contain HTML commands, text, audio, video, or image data. The present invention extends HTML with commands that support communication between the server and the non-standard I/O devices.

In the HTTP protocol, data may be obtained during a communication session by using a tag called a FORM as part of the file defined by <path> in the command discussed above. The FORM format for standard HTTP is:

```
<FORM ACTION="URL"
  METHOD=GET|POST
>
Command
</FORM>
```

where "|" is an "OR" operator. The commands supported by standard HTTP are INPUT, SELECT, and TEXTAREA. Additionally, standard HTTP permits the inclusion of text data in the command area. In the present invention, HTML has been extended to support new ACTIONs, METHODs, and INPUTs.

In accordance with the principles of the present invention, tags are preferably used to identify device transfers and input operations. Preferably, the FORM tag is used to identify device transfers and ACTION and METHOD attributes further identify the device operation. As shown in FIG. 2, the extended ACTION field may include a FROM and TO attribute for accessing a local terminal file or smart card reader or a TO PRINTER attribute for directing output data to a printer local to the I/O device. The FROM and TO attributes for accessing local files and smart card readers and for directing output data to a local printer have previously been unsupported in any Internet protocol. As a result, the server 12 may access non-standard I/O peripherals for any of the I/O devices used in the transaction or data system 10. The ACTION="URL" is a part of standard HTTP and is well known.

The METHOD attributes may include the GET, POST, PAYMENT, or SQL methods. The GET and POST methods are currently supported in standard HTTP and are well known. The PAYMENT attribute is a directive to deliver data retrieved by an INPUT command to a private payment network for authorization and settlement and is not available in current Internet protocols. This directive is used by the client program to activate a conventional financial transaction application which communicates with the transaction system over a dedicated data line or PSTN in a known protocol such as VISA. Such an attribute is used where the more secure physical connection between remote site and transaction system and data encryption devices or the like are preferred. The SQL method preferably identifies a database language file which CGI 28 uses to correlate data in the HTML FORM to an insertion or query command contained in the file.

The preferred format for the INPUT tag which is used to identify input operations is also shown in FIG. 2. The TYPE and NAME attributes are used to define a non-standard I/O device or local storage variable for the input of data. The TYPE field values "text," "password," "checkbox," "radio," "submit," and "reset" are previously known, as are the attributes NAME, VALUE, CHECKED, SIZE, and MAXLENGTH. To support the extended capability of the present invention, the TYPE attribute preferably includes attributes MSRT1 for reading track 1 of a magnetic swipe reader, MSRT2 for reading a magnetic swipe reader track 2, KEY for reading input from a terminal command keypad, PIN for reading a personal identification number pad, BCW for reading a bar code wand, MICR for reading a check magnetic code reader, ATM for reading a dollar amount via a key input mask, INT for reading an integer via a key input mask, LOCAL for reading input from a variable in the local storage of an I/O device, and AUTOSUBMIT for returning a FORM with information to the server.

The NAME attribute used with the INPUT tag identifies reserved word names for local storage in the device executing the client program. Preferably, the NAME attribute identifies ip_address, host_phone, tid, work_key, datetime, and deposit_acct as local storage areas in the local device for the terminal's Internet Protocol (IP) address, Internet access phone number, terminal ID, PIN encryption working key, date/time, and merchant account number, respectively. These attributes are used with the INPUT tag to read non-standard I/O devices which may be coupled to open network 14. For example, an INPUT TYPE=MSRT1 attribute causes the client program residing within a magnetic stripe reader to input data from track 1 of a stripe reader and insert that data into a FORM which is returned to Web server 12 for processing by an INPUT TYPE=AUTOSUBMIT statement.

Preferably, the database language commands which may be embedded in the extended HTML are SQL commands such as those shown in FIG. 3, although other database languages may be used. The SELECT command may include the names of data fields in a database so the device on network 14 may request a data item from a database at the central processing system. The database table is identified by the FROM attribute and the conditional selection of data from an identified database table may be defined by a WHERE attribute. Additionally, records may be requested from an identified database in ascending or descending order or in groups of two records at a time using the ORDER attribute. Additionally, the SELECT field command with the GROUP attribute provides I/O devices with the capability of retrieving records grouped under an identified name. Additionally, the I/O devices may either insert new data into an identified database with the INSERT attribute or update

EXHIBIT B
PAGE 32 OF 37

data already existing in a database with the UPDATE attribute. The values for the INSERT attribute may be identified with the VALUES attribute, and the SET and WHERE attributes may be used to define and conditionally update values in the identified database. Preferably, the present invention implements two DELETE and CREATE attributes. The DELETE attribute deletes all items in an identified column of a database table which may satisfy a condition defined by a WHERE attribute. The CREATE attribute creates a database table having a primary key identified by the PRIMARY KEY attribute.

Preferably, the server program executes on a computer system having at least an Intel 80386 or better processor with at least 4 megabytes of RAM and at least 3 megabytes of hard disk space available. The computer system running the server may operate any known server platform operating system such as WINDOWS 3.1, WINDOWS 95, or WINDOWS NT, UNIX, AIX, and others. The non-standard I/O devices require a processor of a Z80A type or better, at least 32K bytes of RAM, and at least 32K bytes of ROM. The device includes a modem capable of at least 1200 bits-per-second (bps) but other modem speeds may be used for communication between client and server. Alternatively, the device may be coupled to a LAN which in turn is coupled to the Internet for communication with server 12. A typical non-standard device which executes the client program is a VeriFone OMNI390, OMNI395, or VuFone terminal. OMNI390, OMNI395, and VuFone are trademarks of VeriFone, Inc., of Redwood City, Calif. Other exemplary devices include Phillips Screen phone, Hypercomm T7 terminal, and Apple Computer Newton MessagePad.

To build the preferred HTML files which CGI **28** preferably uses to implement the client program and database application, the user preferably uses an off-line editor. The files generated by the editor are preferably comprised of an integrated statements formed from HTML statements and database statements for retrieving and writing data with the database. Exemplary files showing such integrated statements for performing transactions are depicted in FIGS. 14–23B. After such a file is generated, the editor parses the integrated statements into HTML statements and into database statements such as SQL commands. The HTML files required by the client program to support communication with a transaction or data processing center may be downloaded to a device or PC for execution. The files containing the database application statements used by the CGI interface to communicate data with the database application program preferably reside on server 12. Preferably, the database files used by the CGI interface include SQL commands for the application program interfaced to an ODBC compliant database.

The general format of the HTML commands in the HTML files used for communication with a client program and server are of the general format: TAG ATTRIBUTE. Preferably, the TAG field may be one of FORM, INPUT, SQL, or TEXTAREA. The ATTRIBUTE field value depends upon the TAG value. Preferably, the FORM tag may include the ACTION or METHOD attributes where the ACTION attributes include the FROM<file>, TO PRINTER, TO<file>, and TO SCR values noted above, as well as the standard HTML ACTION value of URL=<file>. The METHOD attributes include the PAYMENT and SQL attributes noted above, as well as the standard HTML METHOD values of GET and POST. Also in accordance with the principles of the present invention, the INPUT tag may include TYPE, NAME, VALUE, CHECKED, SIZE, and MAXLENGTH attributes. These attributes are previously supported for the INPUT tag in HTML, however, the present invention further includes TYPE values of MSRT1, MSRT2, KEY, PIN, BCW, MICR, AMT, INT, LOCAL, and AUTOSUB, as well as the standard HTML TYPE values of TEXT, PASSWORD, CHECKBOX, RADIO BUTTON, SUBMIT, and RESET. The present invention also supports NAME attributes of IP_ADDRESS, HOST_PHONE, TID, WORK_KEY, DATETIME, and DEPOSIT_ACCT to identify local storage areas as well as standard HTML NAME attribute <Field_NM> to identify a FORM variable.

The preferred high level processing of the client program is shown in FIG. **4**. That processing includes an idle step (Block **100**) in which the program performs general housekeeping tasks such as maintaining internal time, scanning for input which may activate the device, or other known functions. Further processing is activated by some operator action at the device or PC which causes the device to either open a remote URL (Block **102**) or open a local URL (Block **104**). If a remote URL is required, the device transmits a message of the format discussed previously which is routed through the open network and delivered to a server **12** for a transaction or data processing system (Block **106**). The HTML file selected at the server **12** is identified by the remote URL in the initial communication between the device and server **12** and that URL is used to return the selected HTML file to the device for processing (Blocks **108**, **110**).

FIG. **4** also shows that an operator may initiate an open local URL function by typing in a command or by pushing a hot key which is associated with a local URL. The I/O device reads the HTML file identified by the URL from local memory (Block **112**) and passes the HTML file to the function for processing HTML files (Block **110**). After a file is processed (Block **110**), the client program determines whether the HTML file is to be stored (Block **114**). If it is not, the process returns to the idle processing (Block **100**). Otherwise, the process determines whether the HTML file is to be associated with a hot key (Block **116**) and, if it is, it stores the file and generates the link between a hot key and the stored file (Blocks **118**, **120**). If the HTML file is only to be stored, no association is made with a hot key and the file is simply stored in local memory (Block **20**). The client program then returns to idle processing (Block **100**).

The high-level processing for the HTML file (Block **110**, FIG. **4**) is shown in further detail in FIG. **5**. The process begins by scanning the HTML file for a TAG (Block **140**). If no TAG is found, the file is not in proper format for processing and processing returns to Block **114** discussed in FIG. **4** above. If a TAG is found (Block **142**), the process determines whether the TAG is a FORM TAG (Block **144**) or an INPUT TAG (Block **146**). If it is a FORM TAG, then the FORM TAG is processed and the program continues by looking for other TAGS to process (Block **140**). If the TAG is an INPUT TAG, the INPUT TAG is processed (Block **150**) and the program continues by looking for other TAGS to process (Block **140**). If the TAG is one of the standard HTML TAGS, the program implements the TAG in standard known ways (Block **152**) and then scans for other TAGs to process (Block **140**).

Processing the ATTRIBUTES used to implement a FORM TAG is shown in FIG. **6**. That process continues by scanning the HTML file for an attribute (Block **160**). If an attribute is not found (Block **162**), the program returns to scan for other TAGS (Block **140**, FIG. **5**). If an attribute is found, the program determines whether it is an ACTION attribute (Block **164**) or a METHOD attribute (Block **166**). Depending on the type of attribute, the appropriate function

EXHIBIT B
PAGE 33 OF 37

for processing the attribute is executed (Blocks 168 or 170) and scanning for additional attributes continues (Block 160). If the attribute is not an ACTION or METHOD attribute, there is an error in the file and processing returns to scan for other TAGs.

The processing for the ACTION attribute is shown in FIG. 7. There, the ACTION attribute is examined to determine whether it is a FROM<file> (Block 180), TO PRINTER (Block 182), TO<file> (Block 184), TO SCR (Block 186), FROM SCR (Block 188) or a URL=<file> (Block 192). The URL=<file> ACTION is a standard HTML action which is processed in a known way (Block 194). The FROM <file> action is processed by reading data from a file associated with the I/O device or PC interfaced to the I/O device (Block 196). The TO PRINTER action results in data in the FORM being sent to the printer (Block 198) while the TO <file> action results in data in the FORM being written to a local file (Block 200). The TO SCR action causes data to be written to the smart card via a smart card reader (Block 202) and the FROM SCR reads data from a smart card through a smart card reader (Block 204). After the appropriate action processing takes place, the HTML file is scanned for additional ACTION values to perform (Block 206), and if one is found, the process continues. If no attribute is located (Block 208), the process returns to scan for other attributes (Block 160, FIG. 6).

The processing for the METHOD attributes for FORM tags are shown in FIG. 8. The process determines which type of METHOD is present in the FORM and then properly processes the attribute. For the GET and POST methods (Blocks 210, 212) the processing is the same as that performed in standard HTML (Blocks 226, 228). That is, for the GET method, the identified URL<file> is queried for data while the POST attribute causes data to be transferred to the URL<file>. The preferred METHOD attributes extending the HTML implementation of the present invention are SQL (Block 214), and PAYMENT (Block 224) attributes. The SQL attribute is preferably not expanded into a SQL command at the client, but rather is expanded by the CGI 28 at server 12 by correlating the data or variable field names in a returned form with the SQL commands stored at the server. This processing is done in a manner described in more detail below. The client program passes the SQL file identifier to the server 12 (Block 230). The processing of the PAYMENT command (Block 232) is discussed in more detail below. The HTML file is scanned for other METHODs (Block 242, 244), and, if one is found, the processing continues by identifying the METHOD (Blocks 210-224). Otherwise (Block 244), the process returns to scan the HTML file for other ACTION or METHOD attributes (Block 160, FIG. 6).

Processing for the INPUT tag is shown in FIG. 9. The process scans the HTML file following the INPUT tag for attributes (Block 250). If no attributes are found (Block 252), the process continues by scanning the HTML file for other tags to process (Block 140, FIG. 5). If an attribute is found and it is a TYPE attribute (Block 254), it is processed (Block 256), and if the attribute is a NAME attribute (Block 258), it is processed (Block 260). Both the TYPE and NAME processing is shown in more detail in FIGS. 10 and 11, respectively. If the attribute is neither a NAME or TYPE attribute, it is a standard attribute for an INPUT tag supported by standard HTML and is processed in a known manner (Block 262). Following processing of the INPUT attribute, the HTML file is scanned for other attributes to process (Block 250).

Processing for the TYPE attribute is shown in FIG. 10. The process first identifies the TYPE attribute for the INPUT tag and then performs the appropriate processing. The new TYPE attributes of the preferred embodiment of the present invention are MSRT1 (Block 270), MSRT2 (Block 272), KEY (Block 274), PIN (Block 276), BCW (Block 278), MICR (Block 280), AMT (Block 282), INT (Block 284), LOCAL (Block 286), and AUTOSUB (Block 288). If the TYPE attribute is not one of these, it is a standard HTML type attribute that is processed in a known manner (Block 310). Each of the new HTML TYPES supported by the present invention causes an I/O operation with a non-standard device. Specifically, these operations are the reading of Track 1 of the magnetic stripe reader (Block 290), the reading of the second track of the magnetic stripe reader (Block 292), the reading of a keypad (Block 294), the reading of an encrypted PIN through a PIN entry device (Block 296), the reading of a bar code through a bar code reader (Block 298), the reading of encoded data on a check through a magnetic check reader (Block 300), the reading of a dollar amount from a keypad through a key input mask (Block 302), the reading of a number from a keypad through a key input mask (Block 304), the reading of data from a local variable (Block 306), and the submission of the data read from one of these devices in a FORM returned to the server 12 (Block 308). The data mask for AMT constrains the dollar amount read to a predetermined number of characters with only two characters following the decimal point. The data mask for INT ensures the number is an integer value within a predetermined range. Processing continues by scanning the HTML file for other TYPE attributes (Block 312) and, if another TYPE attribute is found (Block 314), processing continues by determining the TYPE attribute and performing the appropriate processing. Otherwise, the process returns to scan the HTML file for other attributes (Block 250, FIG. 9).

The NAME attribute processing is performed in accordance with the process shown in FIG. 11. That process examines the NAME attribute to determine if the variable name identified by the attribute is IP_ADDRESS, HOST_PHONE, TID, WORK_KEY, DATETIME, or DEPOSIT_ACCT (Blocks 320, 322, 324, 326, 328, 330). If they are, the INPUT value resulting from one of the INPUTS in a FORM of the HTML file is stored in a local variable identified by the NAME attribute. Following storage (Block 332), the file is scanned for other NAME attributes (Block 328) and, if there are none (Block 332), processing continues by scanning for other attributes for the INPUT tag (Block 250, FIG. 9). If the NAME attribute is a standard HTML INPUT NAME, it is processed by known methods (Block 336). Processing then continues by scanning for other NAME attributes to process (Block 338, 340). Otherwise, the process returns to scan the HTML file for other attributes (Block 250, FIG. 9).

CGI 28 receives Internet protocol statements in a file transmitted from a client program and provides data from those statements to the application(s) implementing system 40 and receives the output of system 40 and provides them to the client program in a file. CGI 28 may be implemented by a program developed by a user using a manual development method as shown in FIG. 24A. That method requires a user to generate a system definition from which a file statement definition for the client and application are developed to implement the transactional or data system. Using the file statement definitions, the user generates the files for the client and database programs which are interpreted by the respective programs to implement transactions or data processing. This process requires the user to not only have knowledge regarding the transaction or data process but

EXHIBIT B
PAGE 34 OF 37

specific details of the interaction between the client and database. The user is further required to resolve and correlate all data identifiers in the statements for the client and database environments.

Preferably, CGI **28** is developed with an editor that only requires the user to define the system with statements which are an integration of the protocol statements and the database language. The process implemented by this editor is shown in FIG. **24**B. Examples of such integrated statements for files which implement a specific transaction are shown in FIGS. **14** to **23**B. The editor verifies the syntax of the integrated statements and correlates the data variables of the protocol statements with the data fields of the database. Following the generation of the integrated statements, the editor segregates the protocol statements from the database language statements. The protocol statements are stored in files which are identified as being for a particular transaction or data process and the database statements are stored in files which are identified as being for a particular transaction or data process on an identified database table. The editor places a database file identifier in the protocol statements which contained embedded database statements. The database file identifiers are used by CGI **28**to select the file for the appropriate transaction so CGI **28** may correlate data variables in the protocol statements with data fields in the database files. The files containing statements to be interpreted by the client program are then downloaded to the appropriate terminals, and the database files containing database language statements are stored on the system executing the CGI **28**.

Alternatively, the editor of the present invention may parse integrated statements which are segregated into source code statements for first and second processors, such an editor further includes a compiler to generate executable code for each processor and, if the processors execute differing source code, a compiler for each source code language. The executable code may then be downloaded to the respective processors for execution.

More specifically, the editor preferably places the database statements for one of the transactions of the preferred embodiment in a file identified by the database name following SQL in FIG. **12**. The attributes and tags forming the HTML statements for one of the transactions of the preferred embodiment are placed in a file generally denoted as <html_file>.HTM. The name <html_file> is a name which identifies one of the transactions. Where SQL statements are in the fields of the integrated statements shown in FIGS. **14** to **23**B, the string "<html_file>.SQL" is substituted as the database name in the statements of the <html_file>.HTM file. When the CGI executable file is initiated and parses the returning forms, the returned data is placed in the corresponding "<html_file>.SQL" file which is passed to the application program as a command line argument. In this manner, an abbreviated form for the SQL commands may be communicated over the open network between the client and CGI and the CGI may be able to expand those abbreviated SQL commands into the appropriate SQL commands which the application program requires to manipulate the ODBC database.

To effectuate a transaction, for example, an operation at a terminal with non-standard I/O devices may activate a terminal file with a hot key or other action. In processing the activated file, the client program may acquire data which is stored in a local variable or accessible through a non-standard I/O device. This data may then be stored in a FORM and submitted to a server file at a processing system address. The server file activates CGI **28** which retrieves data from the FORM and incorporates it into database statements in the database file for the appropriate transaction and database. If the database statement is a query, the requested data is returned to the CGI in the database file and the CGI places it in the corresponding FORM variables so the server may return the data to the terminal. If the database statement provides data to a database to obtain an authorization, for example, the action performed by the database application in response to the data is placed in the corresponding FORM and returned to the terminal. In this way, data is exchanged between the terminal and the database application. This exchange is supported by CGI **28** even though the server/client communication is performed in an open system protocol, such as HTTP, and the database application is performed in another language, such as SQL. CGI **28** is able to convert and exchange the data between the client and database without the user having to specifically design and implement a conversion program.

The communication paths available for a device implementing the present invention are shown in FIG. **13**A. As shown there, an I/O device **420** is coupled through the WorldWide Web open network **426** to an Internet Web server **12**. This connection may be implemented with the preferred extended capability HTML described above. Although HTML files may be encrypted to enhance the security of the document as it is communicated across the Internet, the operator of the system may choose to utilize a more secure physical connection between the device **420** and the Web server **12**. To obtain this alternative connection, the PAYMENT command for the METHOD attribute is preferably used. One form of the PAYMENT command is for a merchant's terminal and the other is for a consumer's terminal. In either terminal, the client program which supports the extended capability HTML operates independently but co-resident in memory with a certified bank card authorization and capture application, which may be provided by a financial institution or a bank card processor.

For the form of the command shown in FIG. **13**B, the client program in the merchant terminal suspends its execution and passes the terminal identifier, stored locally, which identifies the merchant's account and the consumer account information read via a magnetic stripe reader or the like, to the bank card application. The bank card application communicates this information via a PSTN **424** or the like to a transaction processor **422**. The processor **422** authorizes or denies the transaction and, if authorized, a printer at the merchant terminal prints a purchase agreement which the consumer may execute to complete the transaction.

In response to a HTML file having a FORM with an ACTION attribute equal to an executable file name for a bank card application program or the like, a METHOD attribute with a field value of PAYMENT, and an INPUT tag with a TYPE attribute of LOCAL_NAME which identifies a deposit only account supplied by a merchant (as shown in FIG. **13**C), the client program is suspended and control is transferred to the bank processing application. The bank processing application then uses a modem or ISDN D channel using T3 POS protocol or the like to connect to a secure packet network **424** to connect in a virtual point-to-point manner with a payment processor through a PSTN network or the like. This physical connection provides an additional security element to the encrypted data for the transaction of account information, PIN numbers encrypted by PIN pads provided at the consumer site, and other sensitive information. The bank processor **422** may submit remittance data to the merchant, via the Web or otherwise. After receiving the remittance data, the merchant may ship

the product to the consumer. Thus, in this manner, the I/O device may communicate with a plurality of Web servers to "shop" for a best price, delivery date, or other relevant information for selecting a preferred transaction, and then execute the PAYMENT method to utilize a more secure physical communication connection and data security devices to consummate the financial elements of the transaction with less risk and costs for the merchant, consumer, and bank processor.

The preferred integrated HTML/SQL statements which support a card initiated payment authorization and capture transaction are shown in FIG. **14**. A first file **500** includes statements which identify the URL database from which the non-standard I/O device seeks authorization for a transaction. The prompts to the operator to enter the account number and amount of the transaction are supported by the INPUT tags which read the second track of the magnetic stripe reader to accept a number of up to 40 characters and assign that information from that track to a variable, and to input the up to 8 characters from the keyboard or the like into a variable called AMOUNT. The INPUT tag with the TYPE attribute of AUTOSUBMIT returns the form to the server for processing in accordance with the method defined in the returned form. As shown in FIG. **14**, that METHOD statement causes CGI **28** to incorporate returned data into SQL commands which query the database as to whether the subfield of the track **2** data representing the account number is present in the authorization table of the database. If the data is not present, then a new record is inserted into a table labeled "log-table". The new record consists of the account number and the amount returned in the FORM. Based upon the results of this processing, the application program supplies the data fields to the FORM which will be returned to the client program for printing the transaction record. That file **510** is shown in FIG. **14**. The ACTION attribute TO PRINTER and the POST METHOD causes the data in the next eight lines to be directed to the printer coupled to the non-standard I/O device for printing the transaction form. The customer may then execute the printed form to complete the transaction. If the transaction is declined or an error is otherwise encountered, the file **520** is used to return a denial to the client program.

In a similar manner, the preferred integrated statements for a bar code order input with card-initiated payment authorization is shown in FIG. **15**. The file **550**, supported by the present invention which implements the transaction request, is again directed to the proper database by the ACTION attribute. The necessary customer information such as name and address may be input through a standard keyboard. The HTML command in the present invention also permits the form to receive the bar code, unit price, and credit card information in a manner similar to that discussed above for the magnetic card reader. Once this information is returned to the server and CGI interface, it is processed by the application program in accordance with the METHOD identified in the returned form. The method of HTML file **550** also creates a database order_table having the information shown in the method. Again, if the transaction is approved, the data for the order and customer acceptance of the order is provided in HTML file **555**, which is directed by the ACTION attribute to the printer at the non-standard I/O device. If the account number is not in the authorization database, the authorization declined or error response is provided in correspondence with the statements in file **560**.

In a similar manner, FIGS. **16–22** show the integrated statements for a transaction request, authorization response, or authorization declined response files for key input order with secure payment transaction (FIG. **16**), a smart card-debit (Type 1) transaction (FIG. **17A**), a smart card debit (Type 2) transaction (FIG. **17B**), a debit card transaction (FIG. **18**), a check verification transaction (FIG. **19**), a customer frequency transaction (FIG. **20**), an item search transaction for which there is no denial (FIG. **21**), retail store end of day reporting (FIG. **22**) and a store reporting an e-mail transaction (FIG. **23**).

While the present invention has been illustrated by the description of a preferred and alternative embodiments and processes, and while the preferred and alternative embodiments and processes have been described in considerable detail, it is not the intention of the applicant to restrict or in any way limit the scope of the appended claims to such detail. Additional advantages and modifications will readily appear to those skilled in the art. For example, rather than expanding HTTP to support non-standard I/O devices, the FTP, POP, SMTP, TELNET or other protocols may be expanded in like manner to couple non-standard I/O devices to the Internet. Similarly, the preferred implementation of the present invention supports a variety of non-standard I/O devices and I/O operations. An Internet protocol may be constructed in accordance with the principles of the present invention to support only selected I/O devices or operations disclosed in the present application. The invention in its broadest aspects is therefore not limited to the specific details, preferred embodiment, and illustrative examples shown and described. Accordingly, departures may be made from such details without departing from the spirit or scope of applicant's general inventive concept.

What is claimed is:

1. An open network processing system comprising:

a server program coupled to an open network;

a non-standard input/output (I/O) device coupled to said open network; and

a client program for communicating data in an extended open network protocol between said server program and said non-standard I/O device, said extended network protocol including one identifier for said non-standard I/O device for a transaction and an identifier for an operation to be performed with said identified non-standard I/O device.

2. The system of claim 1, wherein said client program communicates said data in files having protocol statements conforming to said extended open network protocol.

3. The system of claim 2, wherein said files are identified by universal resource locators.

4. The system of claim 2, said server program further comprising a common gateway interface.

5. The system of claim 4, said common gateway interface providing data from said protocol statements conforming to said extended open network protocol to a transaction system.

6. The system of claim 5, said common gateway interface receiving data from said transaction system and providing said data in a file to said client program.

7. The system of claim 5, said common gateway interface correlating data in said protocol statements conforming to said extended open network protocol with data fields in database files for a database coupled to said server.

8. The system of claim 7, wherein said database files are passed to an application program for accessing said database coupled to said server.

9. A method for processing data over an open network for non-standard input/output (I/O) devices comprising the steps of:

coupling a server program to an open network;

coupling a non-standard I/O device to said open network;

communicating data conforming to an extended open network protocol between said server program and said non-standard I/O device;

identifying said non-standard I/O device in a protocol statement conforming to said extended network protocol; and

identifying an operation to be performed with said identified non-standard I/O device.

10. The method of claim **9**, said communicating step further comprising the step of:

using a client program to communicate said data between said server program and said non-standard I/O device in protocol statements that conform to said extended open network protocol.

11. The method of claim **10** further comprising the steps of:

grouping said protocol statements in files for said communication between said server program and said non-standard I/O device; and

identifying said files with universal resource locators.

12. The method of claim **9** further comprising the step of:

coupling a common gateway interface to said server program, said common gateway for communicating data between a database and said server program.

13. The method of claim **12**, said common gateway interface further performing the step of:

providing data from protocol statements conforming to said extended network protocol to a transaction system, said protocol statements being received in a file from said client program.

14. The method of claim **13**, said common gateway interface further performing the step of:

receiving data from said transaction system and providing said data to said client program in a file.

15. The method of claim **13** further comprising the step of:

correlating data in extended open network protocol statements with data fields in database files.

16. The method of claim **15** further comprising the step of passing database files to an application program for accessing said database.

17. An open network processing system comprising:

a server program coupled to an open network, said server program including a common gateway interface;

a non-standard input/output (I/O) device coupled to said open network;

a client program for communicating data in an extended open network protocol between said server program and said non-standard I/O device, said client program communicating said data in files having protocol statements conforming to said extended open network protocol; and

said common gateway interface provides data from said protocol statements conforming to said extended open network protocol to a transaction system, correlates data in said protocol statements conforming to said extended open network protocol with data fields in database files for a database coupled to said server, and receives data from said transaction system to provide said data to said client program.

18. The system of claim **17**, wherein said database files are passed to an application program for accessing said database coupled to said server.

19. A method for processing data over an open network for non-standard input/output (I/O) devices comprising the steps of:

coupling a server program to an open network;

coupling a non-standard I/O device to said open network;

communicating data conforming to an extended open network protocol between said server program and said non-standard I/O device;

coupling a common gateway interface to said server program, said common gateway for communicating data between a database and said server program;

providing data from protocol statements conforming to said extended network protocol to a transaction system, said protocol statements being received in a file from said client program;

receiving data from said transaction system and providing said data to said client program in a file; and

correlating data in extended open network protocol statements with data fields in database files.

20. The method of claim **19** further comprising the step of passing database files to an application program for accessing said database.

* * * * *

EXHIBIT B
PAGE 37 OF 37

EXHIBIT / ATTACHMENT

C

(To be scanned in place of tab)

US006366967B1

(12) **United States Patent**
Wagner

(10) **Patent No.: US 6,366,967 B1**
(45) **Date of Patent: *Apr. 2, 2002**

(54) **OPEN NETWORK SYSTEM FOR I/O OPERATION INCLUDING A COMMON GATEWAY INTERFACE AND AN EXTENDED OPEN NETWORK PROTOCOL WITH NON-STANDARD I/O DEVICES UTILIZING DEVICE AND IDENTIFIER FOR OPERATION TO BE PERFORMED WITH DEVICE**

(75) Inventor: **Richard Hiers Wagner**, Dunwoody, GA (US)

(73) Assignee: **Datascape, Inc.**, Atlanta, GA (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.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **09/314,266**

(22) Filed: **May 18, 1999**

Related U.S. Application Data

(63) Continuation of application No. 08/995,123, filed on Dec. 19, 1997, now Pat. No. 5,905,908, which is a continuation of application No. 08/493,772, filed on Jun. 22, 1995, now Pat. No. 5,742,845.

(51) **Int. Cl.**[7] **G06F 13/14**
(52) **U.S. Cl.** **710/33**; 710/20; 709/227
(58) **Field of Search** 710/1, 33, 20; 709/227

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,625,276 A 11/1986 Benton et al.
4,707,592 A 11/1987 Ware

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

WO WO 95/16971 6/1995

OTHER PUBLICATIONS

"Europay Members Forum", Europay International, May 8, 2001, (3 pages).

(List continued on next page.)

Primary Examiner—Thomas Lee
Assistant Examiner—Rehana Perveen
(74) *Attorney, Agent, or Firm*—Maginot, Addison & Moore

(57) **ABSTRACT**

An open network system for supporting input/output (I/O) operations for non-standard I/O devices are disclosed. The system includes a server coupled to a plurality of I/O device through an open network and an extended open system protocol that supports communication with devices that are not personal computers (PCs). These devices include magnetic stripe readers, check readers, smart card readers, credit card terminals, screen phone terminals, PIN pads, printers, and the like. The extended open network protocol includes tags which identify device and input operations and attributes which identify the location, data exchange method, and data variable names for the retrieval, acquisition, and submission of data between the server and I/O devices. Preferably, the open network protocol is implemented in a Hyper Text Transport Protocol (HTTP). Preferably, the system includes a common gateway interface (CGI) at the server which converts protocol statements communicated between the server and I/O devices to application language statements for providing data to an application program coupled to the server. Most preferably, the application statements and protocol statements are constructed in integrated statements with an editor. The editor ensures that data identifiers in the application and protocol statements are compatible. The integrated statements are then parsed by the editor to segregate the protocol statements from the application statements. The protocol statements are downloaded in a file to a client program at an I/O device for processing. The application statements are stored in a file for use by the application. In this manner, generation of the files for client and application processing are automatically done without the user ensuring the correlation of the data fields in the two files.

18 Claims, 25 Drawing Sheets

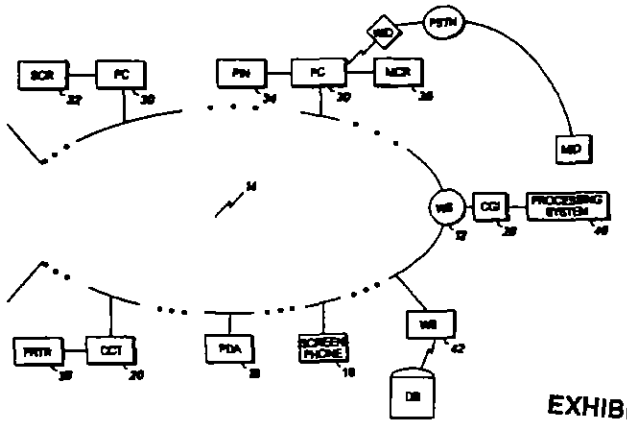

EXHIBIT C
PAGE 1 OF 39

U.S. PATENT DOCUMENTS

4,799,156	A		1/1989	Shavit et al.
4,885,777	A		12/1989	Takaragi et al.
4,962,531	A		10/1990	Sipman et al.
4,972,463	A		11/1990	Danielson et al.
5,018,196	A		5/1991	Takaragi et al.
5,025,373	A		6/1991	Keyser, Jr. et al.
5,046,094	A		9/1991	Kawamura et al.
5,148,481	A		9/1992	Abraham et al.
5,239,662	A		8/1993	Danielson et al.
5,256,863	A	*	10/1993	Ferguson et al. 235/383
5,276,312	A		1/1994	McCarthy
5,377,191	A	*	12/1994	Farrell et al. 370/94.1
5,379,344	A		1/1995	Larsson et al.
5,386,517	A		1/1995	Sheth et al.
5,461,217	A		10/1995	Claus
5,475,858	A		12/1995	Bush
5,530,844	A	*	6/1996	Phillips et al. 395/500
5,544,246	A		8/1996	Mandelbaum et al.
5,668,878	A		9/1997	Brands
5,732,219	A	*	3/1998	Blumer et al. 395/200.57
5,754,655	A		5/1998	Hughes et al.
5,756,978	A		5/1998	Soltesz et al.
5,778,067	A		7/1998	Jones et al.
5,815,577	A		9/1998	Clark
5,867,821	A		2/1999	Ballantyne et al.
RE36,310	E		9/1999	Bjerrum et al.

OTHER PUBLICATIONS

"NCR 2170 Hospitality Workstation Designed for Bars, Restaurants", AT&T New Release, Feb. 2, 1993, (2 pages).
"AT&T to Increase Smart Card Reliability, Expand Memory to 8KB", AT&T News Release, Apr. 8, 1994, (2 pages).
"AT&T and NCR Offer Smart Card Upgrade Kits for Bank ATMs", AT&T News Release, Nov. 17, 1993, (2 pages).
"The Smart Card Cashes In", The Economist, Jan. 29, 1994, (2 pages).
"NCR 7450 Terminal Brings ATM-style Interface to Point-of-Sale", AT&T News Release, Oct. 11, 1993, (4 pages).
"Mastercard Smart Card Milestones", MasterCard International Press Office, Mar. 5, 2001, (3 pages).
Svigals, Jerome, "Microchip Smart Cards Will Save Money", American Banker, Feb. 2, 1994, (1 page).
Barthel, Matt, "Fears of Crime at ATMs are on the Rise", American Banker, Mar. 30, 1994, (1 page).
"Smart Card Publication List", Smart Card Forum, Mar. 2, 2001, (7 pages).
"Appendix: Major Electronic Purse Projects, By Country", pp. 35–50 and 52–64, Date Unknown (29 pages).
O'Kelly, Robin, "Mondex 'Global Electronic Cash' Wins Helsinki Award", Sep. 8, 1996, (2 pages).
"White Paper – The Euro in the Electronic Purse", Apr. 2000, (5 pages).
"History of Gemplus", Gemplus, Mar. 2, 2001, (3 pages).
Pauli, Ralf and Koponen, Risto, "Toward Electronic Money", Bank of Finland Bulletin, Apr. 1997, (3 pages).
"A Frequently Asked Questions List (FAQ) for *alt.technology.smartcards*", Estonian Institute of Cybernetics, Jul. 4, 1995,(12 pages).
Husemann, Dirk, "The Smart Card: Don't Leave Home Without It", Institute of Electrical and Electronics Engineers, Inc., Sep. 18, 2000, (9 pages).
"Report to the Counil of The European Monetary Institute on Prepaid Cards", The Working Group on EU Payment Systems, May 1994, (12 pages).
Banerjee, Ram, "Smart Card Standards and Electronic Purse, A Review Paper", Aston University, May 1997, (20 pages).
Townend, Robin C., "Finance: History, Development & Market Overview", The International Smart Card Industry Guide, 1995–1996, (23 pages).
Monod, Elsbeth Marikje, "Case Study: The Multi-Functional Smart Carte Sante", The International Smart Card Industry Guide, 1995–1996, (8 pages).
Smith, Jack "Smart Card Industry Review 1995", Smart Card Monthly Newsletter, Jan. 1996, (6 pages).
Crotch-Harvey, Trevor, "Smart Cards in Telecoms", The International Smart Card Industry Guide, 1995–1996, (14 pages).
Farmer, Donna and Tonnac, Remy de, "What's So Smart About Smart Cards", Gemplus, Oct. 10, 2000, (1 page).
"P401 Definition of a pan-European IC Card", Eurescom, Mar. 5, 2001, (1 page).
"Innovonics, Inc. Selects AMP as Manufacturer for the PC Pay Module", Innovonics News Release, May 19, 1995, (3 pages).
Levy, Steven, "E-Money (That's What I Want)", Wired, Dec. 1994, (2 pages).
Fickenscher, Lisa, "Banc One Gears Up to Go National With Its Microchip Smart Cards", American Banker, Jan. 27, 1994 (2 pages).
Strachman, Daniel and Kutler, Jeffrey, "Convention Goers See Bank Future in Smart Cards", American Banker, Nov. 30, 1993, (1 page).
Meece, Mickey, "First Data Strengthens Position in Card Processing", American Banker, Dec. 21, 1993, (1 page).
Kutler, Jeffrey, "Prepaid Card, Even in Concept Stage, Sparking Interest Among Consumers", American Banker, Apr. 27, 1994, (1 page).
Kutler, Jeffrey, "Visa Starts Push for Smart Card to Replace Cash", American Banker, Mar. 22, 1994, (2 pages).
"AT&T Introduces Enchanced Multimedia Interface for ATM Networks", AT&T News Release, May 4, 1994, (2 pages).
"Trendar and AT&T Test Contactless Smart Cards at Truck Stops", AT&T News Release, Feb. 26, 1993, (3 pages).
"NCR Announces 5980 Signature Capture System for Retailers", AT&T News Release, Sep. 30, 1991, (2 pages).
Merckling, Roger and Anderson, Anne "Smart Card Introduction", Mar. 1994, (16 pages).
Merckling, Roger and Anderson, Anne, "DCE Smart Card Integration", Mar. 1994, (16 pages).
"Chapter 2–Technology", Smart Card Primer, The Schuler Consultancy, ©1993, pp. 13–24 and 26–29, (16 pages).
"CP8® PRODUCTS, PASCAM, Network Security Processor", Jan. 1994, (4 pages).
"Notes on Open Networks and Device Addressing", IBM Corporation, Apr. 25, 2001, (19 pages).
Chakrabarti, Samdih, Knight, James and Thies, Bill, "A Centralized Smart Card System Utilizing Distributed Proxies", 2000, University of Leeds, (16 pages).
"Single-Issuer Multiple Uses", Smart Card Primer, The Schuler Consultancy, ©1993, (3 pages).
Sazegar, Hossein, "The Impact of Smart Cards in the Global Payphone Business", Schlumberger Technologies, ©1992, (11 pages).
Derksen, Eduard, "Money-Bit: The Open Off-Host Prepaid Card System", Ascom Autelca, Ltd., Date Unknown, (10 pages).

EXHIBIT C
PAGE 2 OF 39

Corzo, Gerardo, "Credisa's Elite Card System", Date Unknown, (5 pages).
Cordonnier, Vincent M., "Assessing the Future of Smart Cards", Cardtech, Date Unknown, (7 pages).
"NatWest Launch Mondex Global Electronic Cash", Smart Card News, Ltd., Dec. 1993, (7 pages).
Merkert, Sr., Robert J., "Stored Value Cards and The Electronic Purse: Developments in the Americas", Danyl Corporation, ©1995, (8 pages).
"CP8® Products, PinPadLINK, Readers/Encoders", Dactyl France, June 1993, (2 pages).
"Smart Security", Byte Talk, February 1994, (2 pages).
Verschuren, Ton, "Smart Access: Strong Authentication on the Web", Apr. 24, 2001, (10 pages).
"PKCS #11: New Member of Public Key Cyrptography", RSA Data Security, Inc., Jan. 12, 1994, (2 pages).
Petri, Steve, "An Introduction to Smarcards", Litronic, Inc., ©1998, (2 pages).
Violino, Bob, "The Cashless Society", Information Week, Oct. 11, 1993, (5 pages).
Merkert, Sr., Robert J., 'Keynote Presentation for the AIC Conference, Auckland, New Zealand', Danyl Corporation, © 1995, (4 pages).
"Videoway's Experience Shows a New Approach to the Interactive Opportunity", Inside Cable and Telecom Services, Jun. 1995, (3 pages).
Bull CP8® News, Products and Technologies, No. 22, Oct. 1994, (2 pages).
Bull CP8® Systems Product Sheet, Sep. 1994, (2 pages).
"One Store's Comprehensive Loyalty Program–The Takashimaya Experience", Verifone, Inc., ©1994, (4 pages).
"Mondex Payments Scheme", Smart Card News Ltd., Dec. 1993, (2 pages).
"Why Consumers Are Still Carrying Cash Everywhere", Philadelphia Inquirer, Jun. 11, 1995, (2 pages).
"Electronic Payment System Architecture for Point of Sale Application", IBM Technical Disclosure Bulletin, Jan. 1992, (1 page).
"Portable Self Checkout Retail System", IBM Technical Disclosure Bulletin, Jun. 1992, (3 pages).
"Transaction Completion Code Based on Digital Signatures", IBM Technical Disclosure Bulletin, Aug. 1985, (6 pages).
"Chapter 4–Uses", Smart Card Primer, The Schuler Consultancy, ©1993, pages 57–66, 80–83, and 87–91, (19 pages).
Figure 8.5 from p. 151 of Zoreda & Oton "Smart Cards", Artech House, ©1994, (1 page).
"Internet Music Library Process", Date Unknown, (1 page).
Bibliography for "CITM Working Paper WP–95–1005; Designing Electronic Catalogs for Business Value: Results from the CommerceNet Pilot", Date Unknown, (3 pages).
Van Hove, Leo, Bibliography on Electronic Purses, Date Unknown, (4 pages).
Bibliography for "CITM Working Paper WP–95–1006: Financial EDI Over the Internet: A Case Study", Date Unknown, (5 pages).
Bibliography for Hawkes, P. L., Davies, Donald W., and Price W. L. "Integrated Circuit Cards, Tags, and Tokens: New Technology and Applications", Oxford: Boston: BSP Professional, ©1990, (1 page).
Bibliography for Devargas, M., "Smart Cards and Memory Cards", Oxford; Blackwell NCC, ©1992, (1 page).
Bibliography for Svigals, Jerome, "Smart Cards: The New Bank Cards", New York: Macmillan; London: Collier Macmillan Publishers, ©1987,(1 page).
Bibliography for Bright, Roy, "Smart Cards: Principles, Practice, Applications", Chichester [England]: Ellis Horwood; New York: Halsted Press, ©1988, (1 page).
Bibliography for Svigals, Jerome, "Smart Cards: The Ultimate Personal Computer", New York: Macmillan Publishing Co.; London: Collier Macmillan, ©1985, (1 page).
Bibliography for "Applications of Computer Card Technology", Washington, D.C., U.S. Dept. of Labor, Office of the Inspector General, [1990], (1 page).
German article entitled "Erfahrungen Mit Einer Hochsicheren, Multifunktionalen Chipkarte fur Bankenanwendungen" Date Unknown, (21 pages).
Document entitled: "Dialog Search for U.S. Patent Number 5,905,908", Date Unknown, (59 pages).
UBI Consortium Inc. Document, Mar. 26, 2001 (2 pages).

* cited by examiner

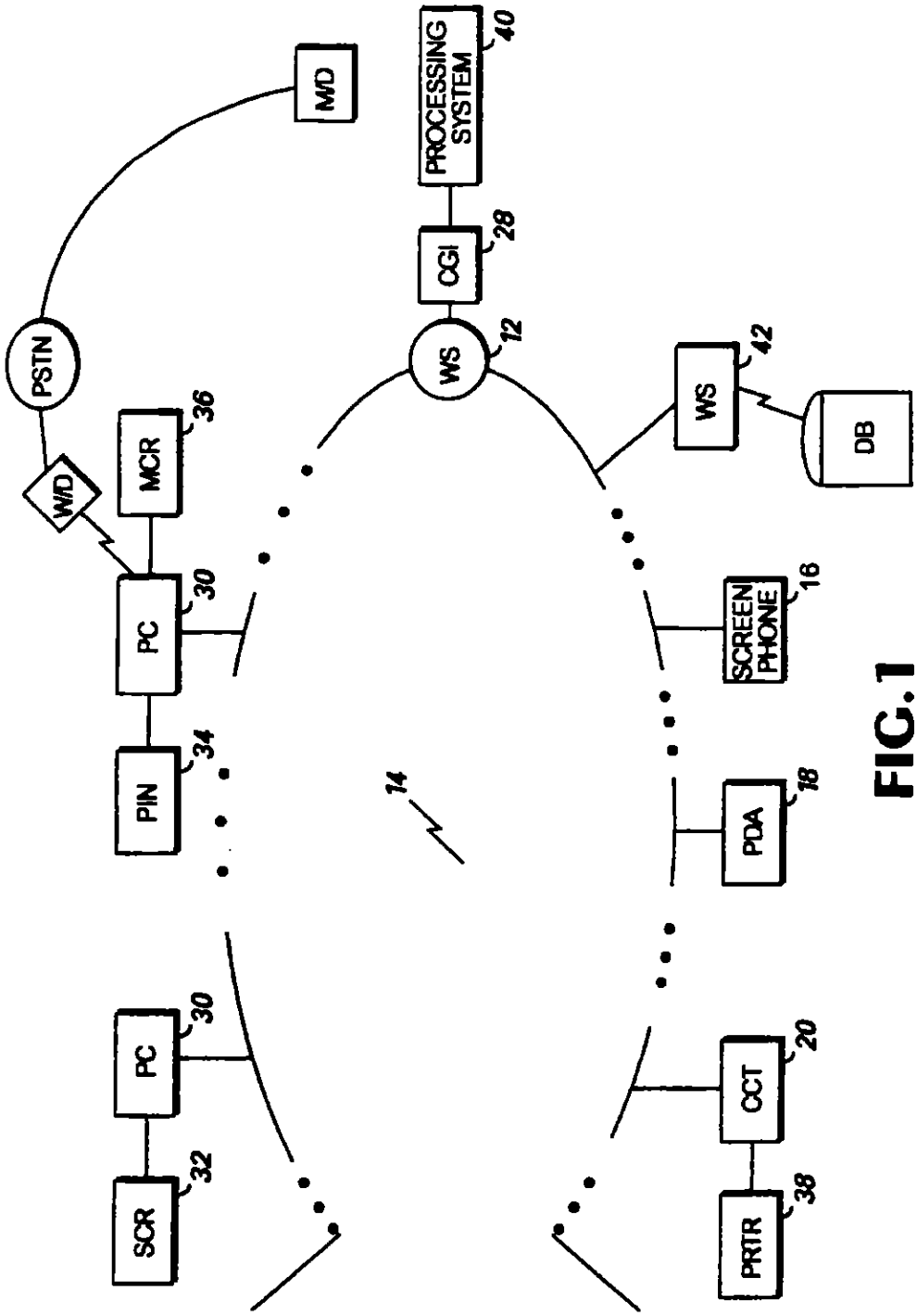

FIG. 1

EXHIBIT C
PAGE 4 OF 49

HTML+D Attributes		Description
<FORM ACTION=	"url"	To/From Web Server URL
	FROM "file name"	From Terminal Local File
	TO PRINTER	To Local Printer
	TO "file name"	To Terminal Local File
	FROM SCR	From Smart Card Reader
	TO SCR	To Smart Card Reader
METHOD=	"GET"	Retrieve Data
	"POST"	Store Data
	"PAYMENT"	Directive to deliver INPUT data to a private Payment Network for authorization and settlement.
	SQL <database name>	SQL statement database table

Attribute	Value	Description
<INPUT TYPE=	"text"	
	"password"	
	"checkbox"	
	"radio"	
	"submit"	
	"reset"	
NAME=	<field name>	
VALUE=	<initial value>	
CHECKED=		
SIZE=		
MAXLENGTH=		
>		

Attribute	HTML+D Value	Terminal Device
TYPE=	*"MSRT1"*	Mag Stripe Reader - Track 1
	"MSRT2"	Mag Stripe Reader - Track 2
	"KEY"	Terminal Command Keypad
	"PIN"	PIN Pad
	"BCW"	Bar Code Wand
	"MICR"	Check MICR Reader
	"AMT"	Dollar amount key input mask
	"INT"	Integer key input mask
	"LOCAL"	Input from Local Variable
	"AUTOSUBMIT"	Submit FORM to ACTION URL
NAME=	*ip_address*	Local Variable - Terminal's IP Address
	host_phone	Local Variable - Local Internet Access Phone Number
	tid	Local Variable - Terminal ID
	work_key	Local Variable - PIN encryption working key
	datetime	Local Variable - Date and time
	deposit_acct	Local Variable - Merchant Deposit Account

FIG.2

EXHIBIT C
PAGE 5 OF 39

SQL Statements

The following SQL commands represent a subset of the entire command set that varies by database vendor.

HTML+D Attributes	Description
SELECT, field_name,..*	Request field_name (one or many) from a database table
FROM=<table name>,..	Database table name
WHERE=<condition>	Conditional selection of data
name = "constant"	
name LIKE "constant"	
name IN "constant"	
AND	
OR	
ORDER=ASC	Request in ascending order
DESC	_descending
2	_by2's
GROUP=<name>	
INSERT TABLE=<table name>	Insert new data in database table
VALUES="constants"	
UPDATE FROM	
<table name>	Update field_name in database table
SET=field_name="constant"	
[WHERE=<condition>]	Update if WHERE clause is satisfied
DELETE FROM <table name>	Delete all columns that satisfy WHERE clause
[WHERE=<condition>]	
CREATE TABLE<table_name>	Create database table
PRIMARY KEY<name>	

FIG.3

EXHIBIT C
PAGE 9 OF 39

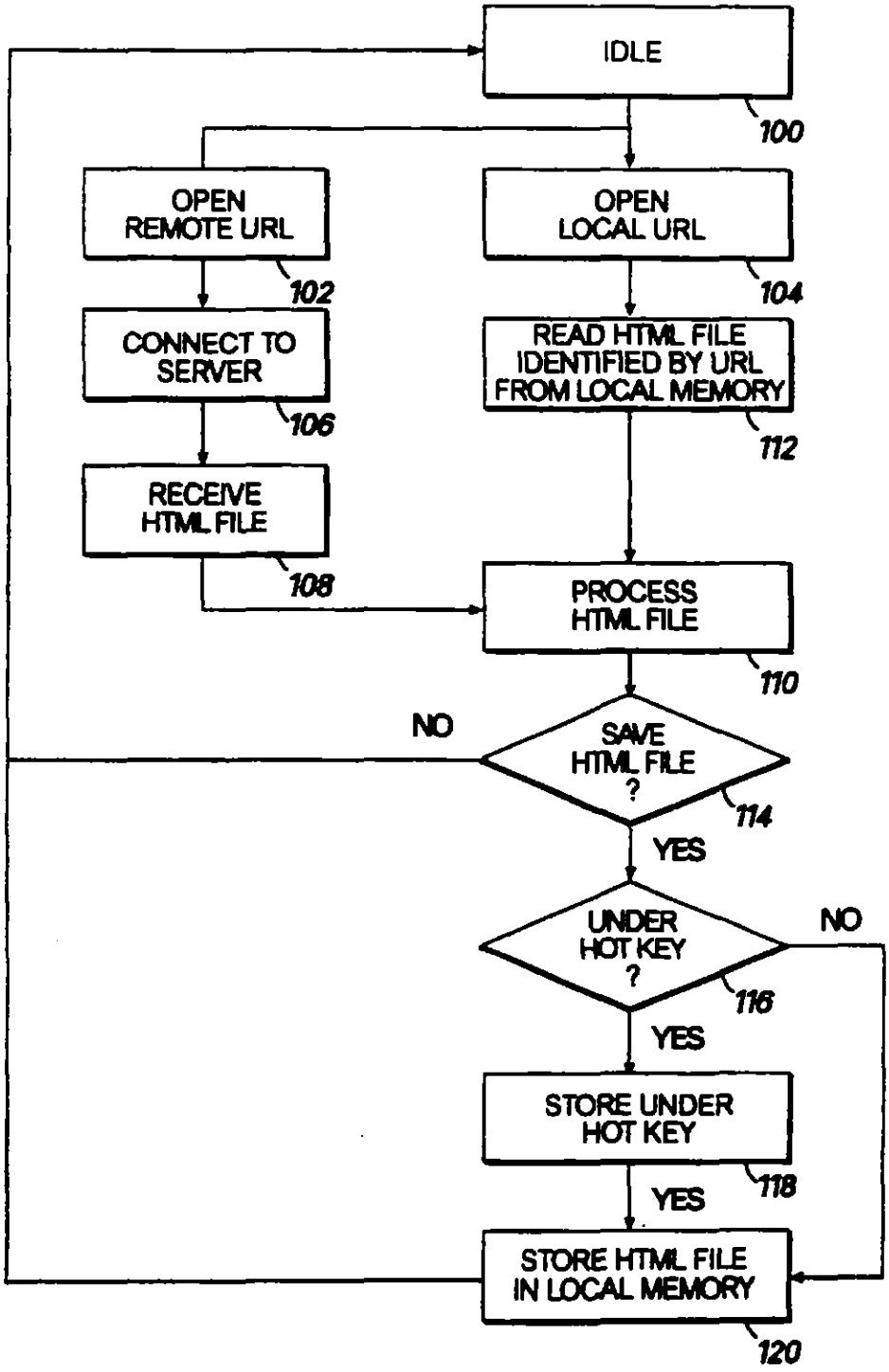

FIG.4

EXHIBIT C
PAGE 7 OF 39

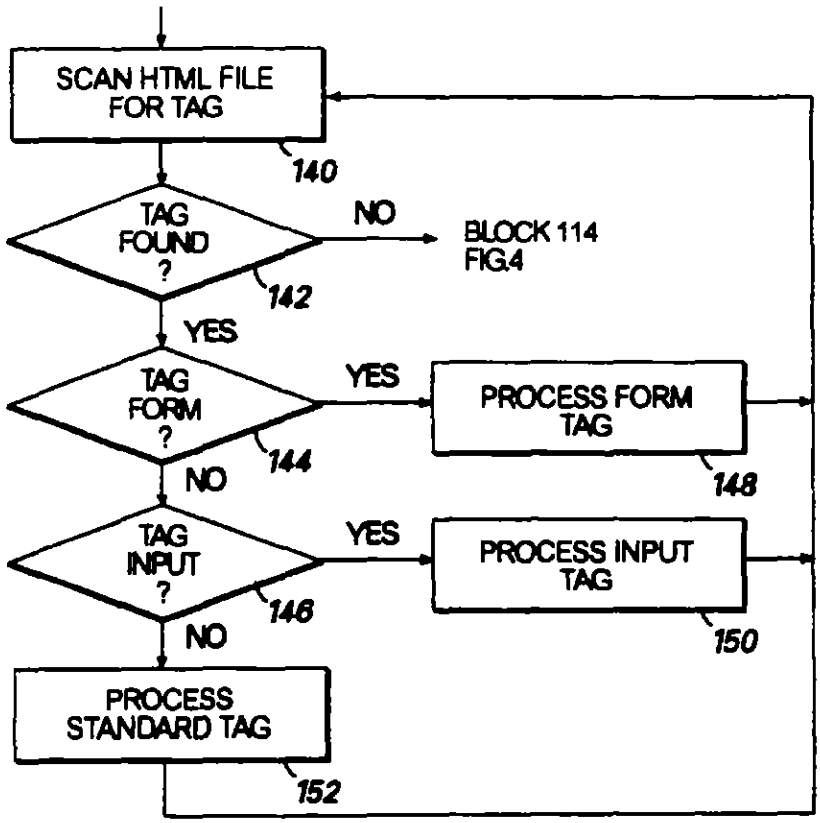

FIG.5

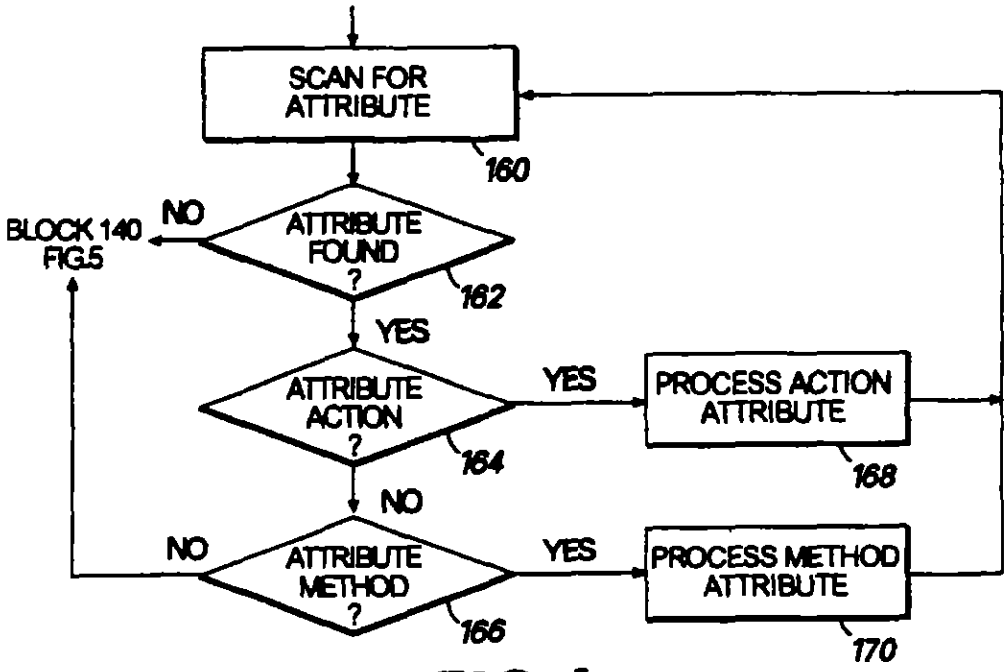

FIG.6

EXHIBIT C
PAGE 9 OF 39

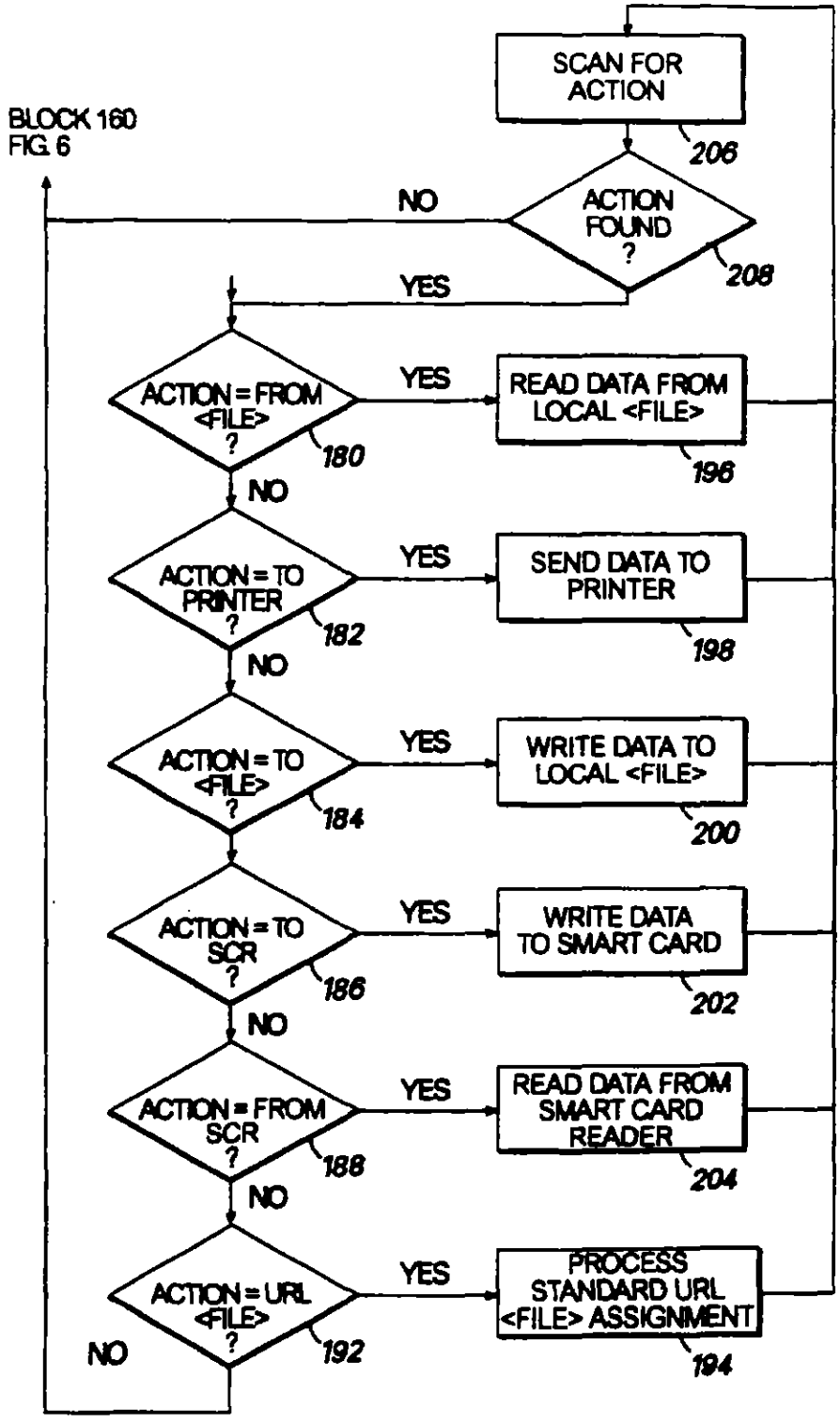

FIG.7

EXHIBIT C
PAGE 9 OF 39

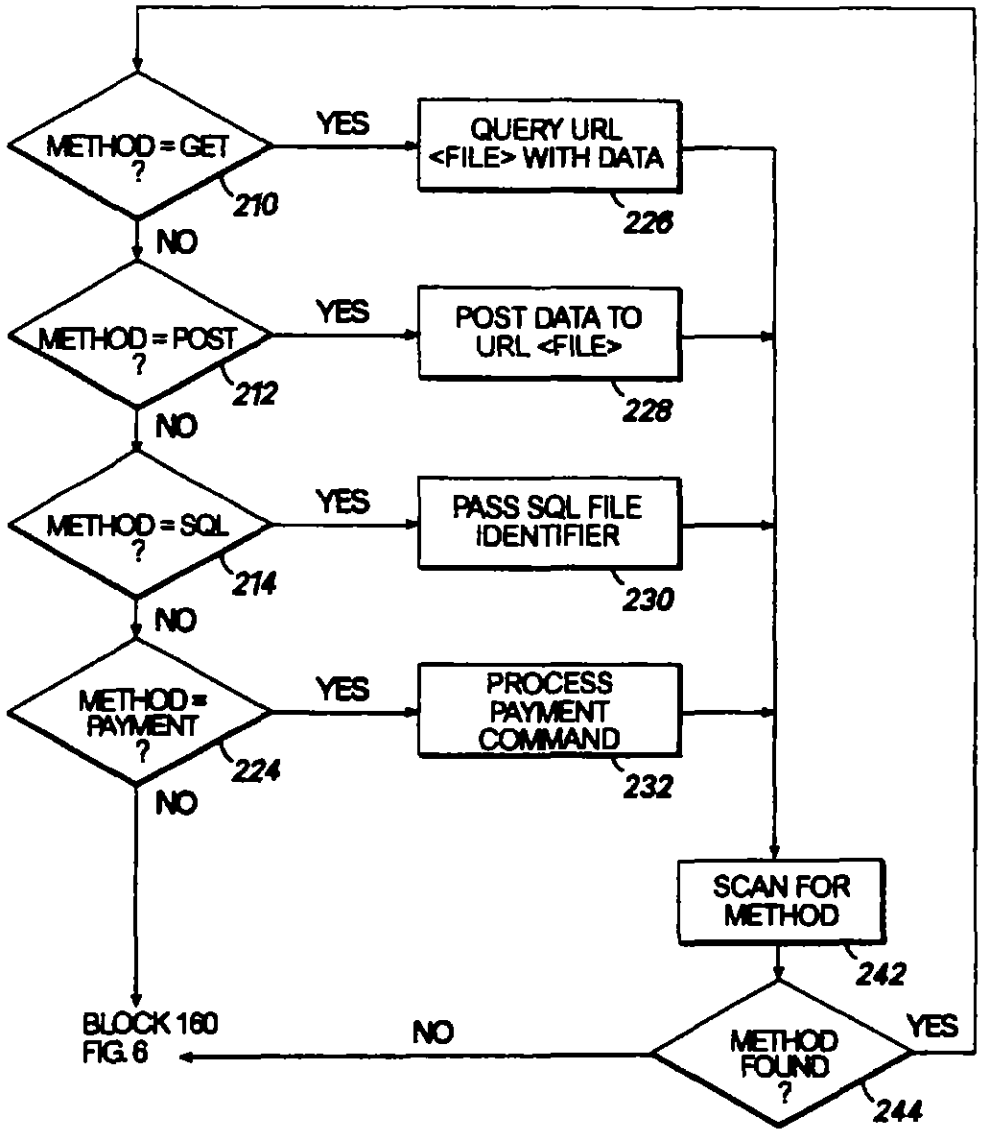

FIG.8

EXHIBIT C
PAGE 10 OF 39

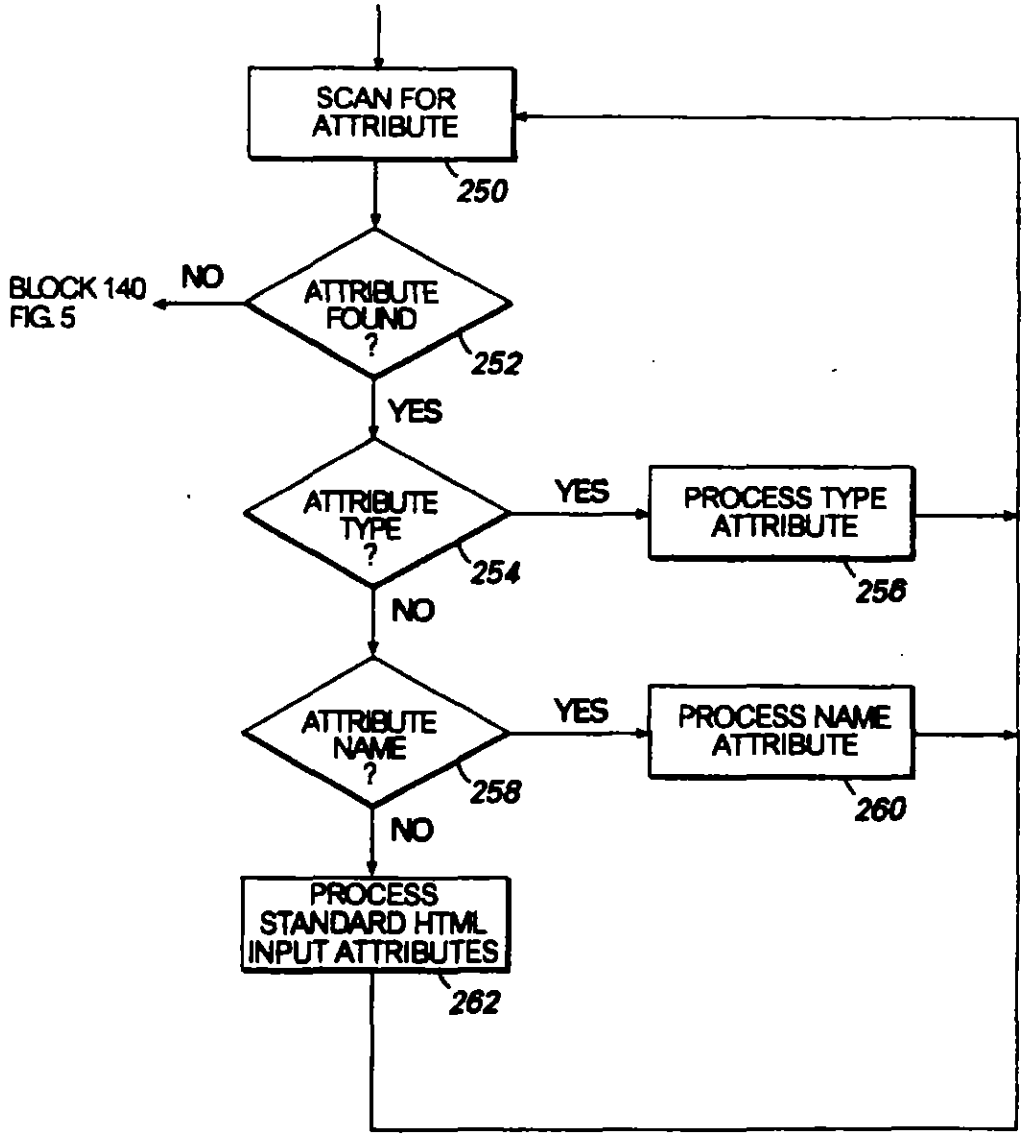

FIG.9

EXHIBIT C
PAGE 11 OF 39

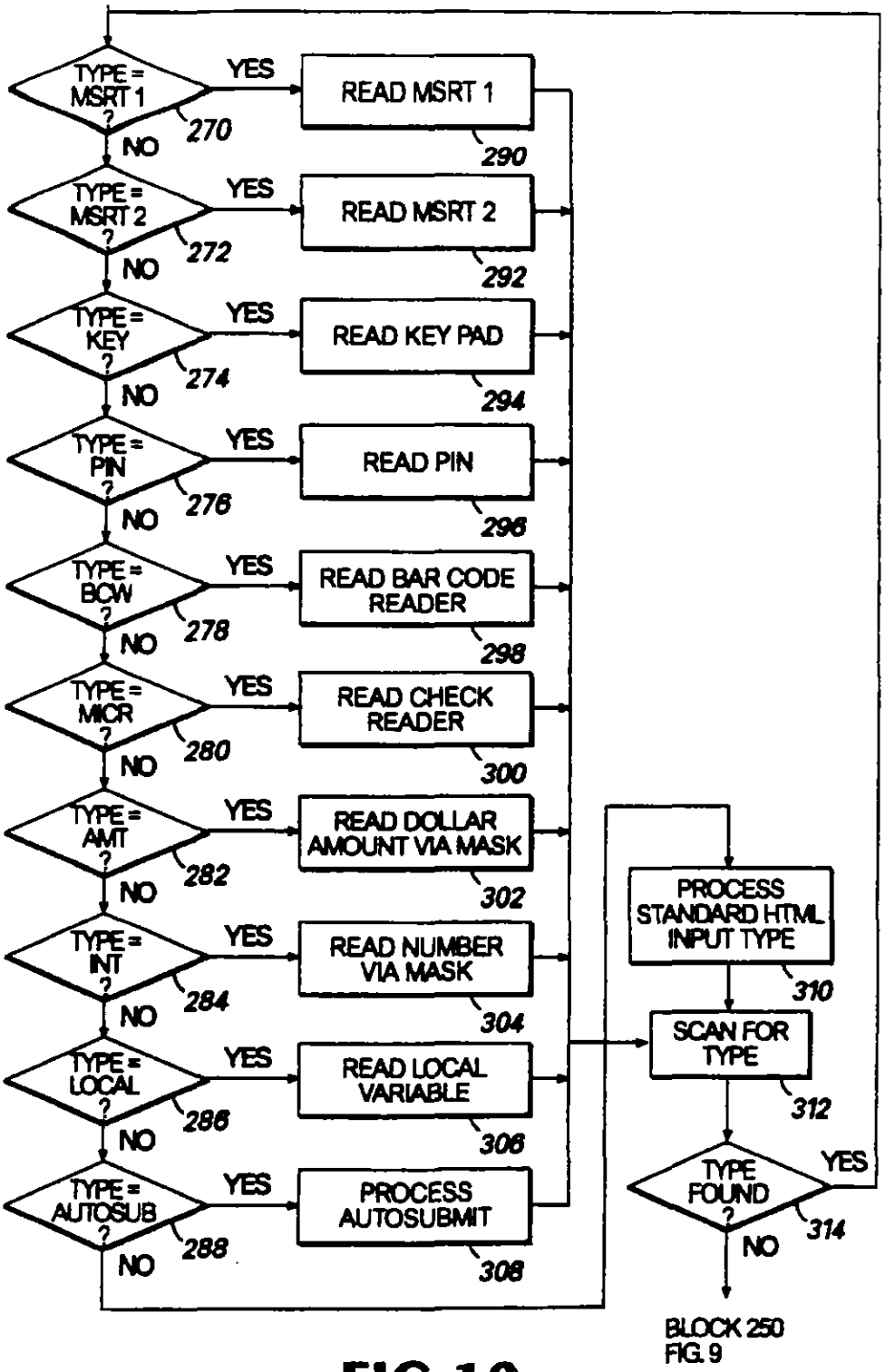

FIG.10

EXHIBIT C
PAGE 12 OF 39

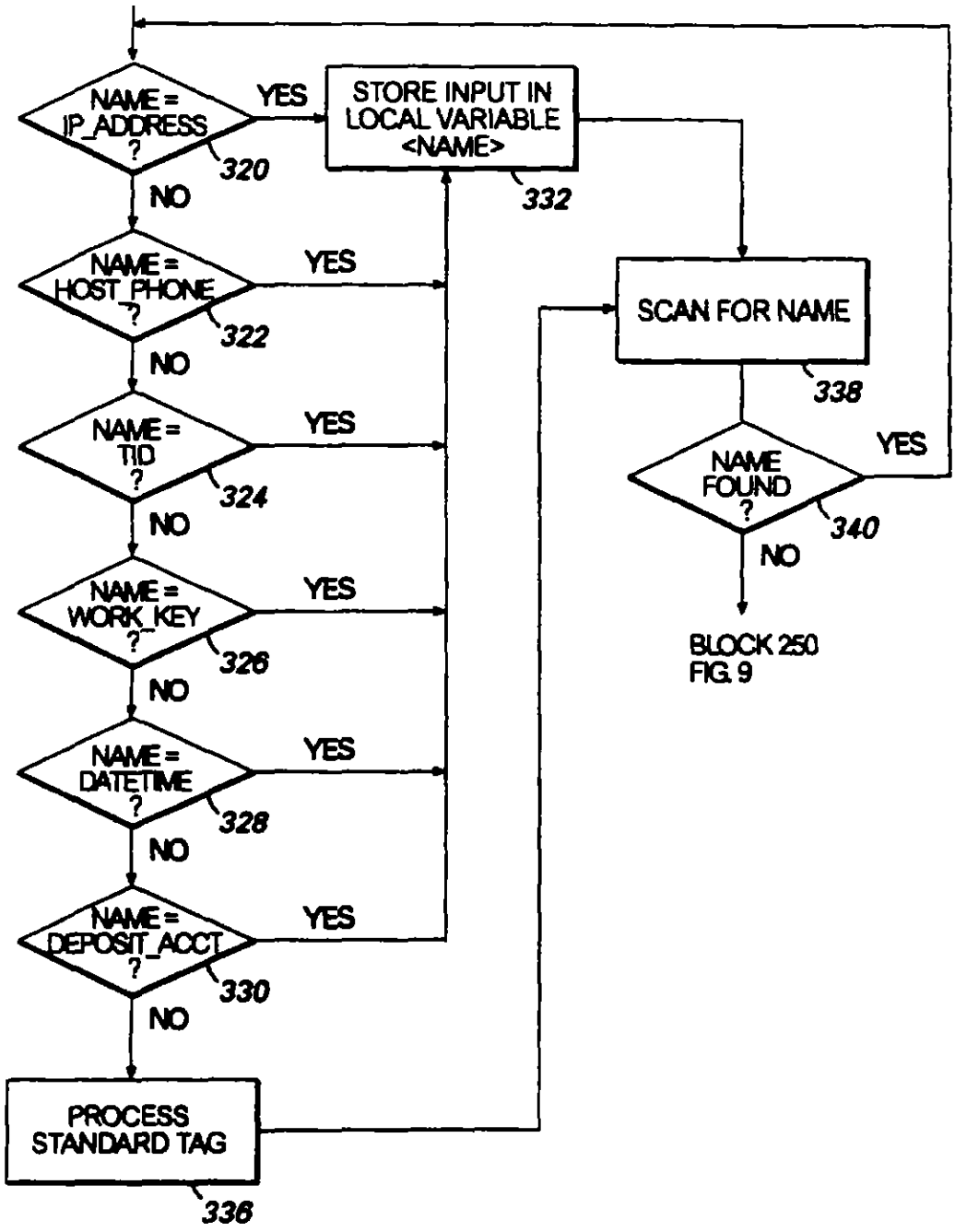

FIG.11

EXHIBIT C
PAGE 13 OF 34

1. <FORM ACTION=URL METHOD= GET>
2. <FORM ACTION=URL METHOD= POST>
3. <FORM ACTION=URL METHOD= SQL <database_name>

FIG.12

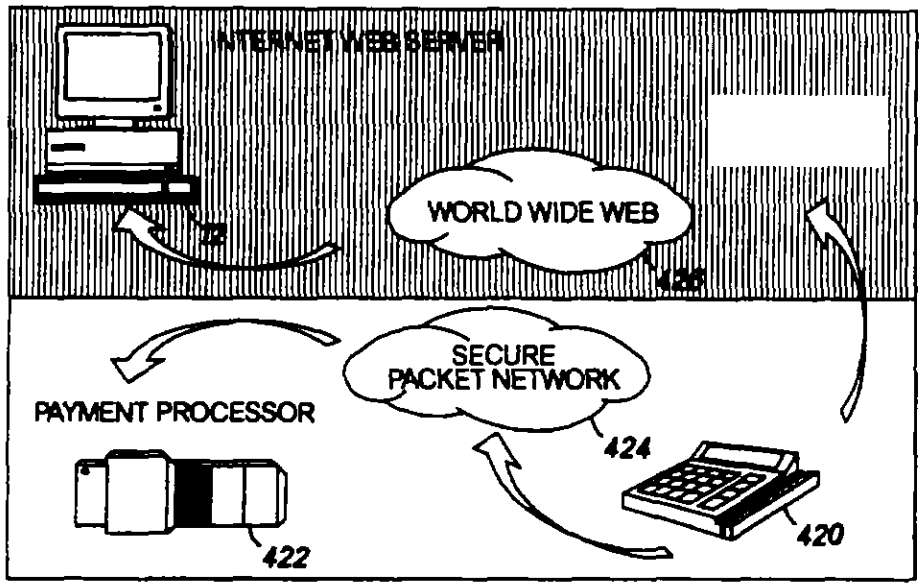

FIG.13A

<FORM ACTION=<filename> METHOD=PAYMENT>
<INPUT TYPE=AUTOSUBMIT>
</FORM>

FIG.13B

<FORM ACTION=dsinet METHOD=PAYMENT>
<INPUT TYPE=LOCAL NAME=DEPOSIT_ACCT VALUE=12345690234567890>
<INPUT TYPE=AUTOSUBMIT>
</FORM>

FIG.13C

EXHIBIT C
PAGE 14 OF 39

1.a. Transaction Request HTML+D

500

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
     METHOD=SQL
          "BEGIN TRAN
          IF NOT EXISTS ( SELECT substring(account, 1, 20) FROM auth_table)
               BEGIN
               INSERT TABLE=log_table VALUES=(getdate(),tid, substring (account, 1,20) ,
               substring( account, 22, 4), amount)
               SELECT * FROM log_table WHERE trandate = getdate()
               END
          ELSE SELECT * FROM error_table WHERE error_no=1
          COMMIT TRAN">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER ACCOUNT NUMBER:
<INPUT TYPE="MSRTZ" SIZE=40 NAME=account>
ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

1.b. Transaction Accepted Response

510

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
JUNE 1 1995            10:30AM       PURCHASE</P>
TERMINAL ID:                        999999999</P>
ACCOUNT NUMBER        99999999999999999999</P>
EXP DATE                                99/99</P>
AMOUNT                               $9999.99</P>
AUTH NUMBER                         999999999</P>
</p>
-------------------------------------------</p>
          CUSTOMER SIGNATURE </p>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

1.c. Transaction Declined or Submit Error Response

520

```
<HTML>
<BODY>
DECLINED<MESSAGE>
</BODY>
</HTML>
```

FIG.14

2.a. Transaction Request HTML+D

550

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
METHOD=SQL"BEGIN TRAN
        IF NOT EXISTS ( SELECT substring(account, 1, 20) FROM auth_table)
        BEGIN
                INSERT TABLE=log_table VALUES=(getdate(),tid, substring( account, 1,20),
                substring( account, 22, 4), amount)
        END
        ELSE  BEGIN
                        SELECT * FROM error_table WHERE error_no=1
                        RETURN
                END
        INSERT TABLE=order_table VALUES=( getdate(), cust_name,address,city, state, zip,
                part_code, unit_price, tax, ship_method, ship_chrg, unit_price + tax +
                ship_chrg, substring( account, 1, 20) ,substring( account, 22, 4))
        SELECT * FROM order_table WHERE trandate = getdate()
        COMMIT TRAN">
CUSTOMER NAME:
<INPUT TYPE="TEXT"SIZE=30 NAME=cust_name></p>
ADDRESS:
<INPUT TYPE="TEXT" SIZE=40 NAME=address></p>
CITY:
<INPUT TYPE="TEXT" SIZE=20 NAME=city><p>
STATE:
<INPUT TYPE="TEXT" SIZE=2 NAME=state>
ZIP:
<INPUT TYPE="TEXT" SIZE=10 NAME=address></p>
SCAN PART CODE:
<INPUT TYPE="BCW" SIZE=9 NAME=part_code></p>
ENTER UNIT PRICE:
<INPUT TYPE="AMT" SIZE=6 NAME=unit_price></p>
TAX:
<INPUT TYPE="AMT" SIZE=5 NAME=tax></p>
SHIPPING METHOD:
<INPUT TYPE="TEXT" SIZE=10 NAME=ship_method></p>
SHIPPING AMOUNT:
<INPUT TYPE="AMT" SIZE=5 NAME=ship_chrg></p>
SLIDE CARD:
<INPUT TYPE="MSRT2" SIZE=40 NAME=account></p>
<INPUT TYPE="SUBMIT" >
</FORM>
</BODY>
</HTML>
```

FIG.15A

EXHIBIT C
PAGE 16 OF 39

2.b. Transaction Accepted Response

555

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
ORDER # 9999999999 APPROVED</p>
JUNE 1 1995          10:30AM          PURCHASE</P>
TERMINAL ID:                          999999999</P>
NAME: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
ADDRESS:</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
CITY: XXXXXXXXXXXXXXXXXXXXXXXXX</p>
STATE: XX                 ZIP:XXXXXXXXXX</p>
ACCOUNT NUMBER:           9999999999999999999</p>
EXP DATE:                                99/99</p>
PART CODE:                           999999999</p>
UNIT PRICE:                           $9999.99</p>
SHIP METHOD:XXXXXX CHARGE             $9999.99</p>
TOTAL AMOUNT:                         $9999.99</p>
AUTH NUMBER:                         999999999</p>
</p>

- - - - - - - - - - - - - - - - - - - - - - - -</p>
          CUSTOMER SIGNATURE</p>

</FORM>
</BODY>
</HTML>
```

2.c. Transaction Declined or Submit Error Response

560

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.15B

3.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
METHOD SQL
    "INSERT TABLE=order_table VALUES=( getdate(), cust_name,address,city, state, zip,
        part_code, unit_price, tax, ship_method, ship_chrg,unit_price + tax +
        ship_chrg, substring( account, 1, 20) ,substring( account, 22, 4))
    SELECT * FROM order_table WHERE trandate = getdate()">
<INPUT TYPE="LOCAL" NAME=tid>
CUSTOMER NAME:
<INPUT TYPE="TEXT" SIZE=30 NAME=cust_name></p>
ADDRESS:
<INPUT TYPE="TEXT" SIZE=40 NAME=address></p>
CITY:
<INPUT TYPE="TEXT" SIZE=20 NAME=city><p>
STATE:
<INPUT TYPE="TEXT" SIZE=2 NAME=state>
ZIP:
<INPUT TYPE="TEXT" SIZE=10 NAME=address></p>
ENTER PART CODE:
<INPUT TYPE="TEXT" SIZE=10 NAME=part_code></p>
ENTER UNIT PRICE:
<INPUT TYPE="AMT" SIZE=8 NAME=unit_price></p>
TAX:
<INPUT TYPE="AMT" SIZE=5 NAME=tax></p>
SHIPPING METHOD:
<INPUT TYPE="TEXT" SIZE=10 NAME=ship_method></p>
SHIPPING AMOUNT:
<INPUT TYPE="AMT" SIZE=5 NAME=ship_chrg></p>
<INPUT TYPE="SUBMIT" >
</FORM>
</BODY>
</HTML>
```

FIG.16A

EXHIBIT C
PAGE 18 OF 39

3.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
ORDER # 9999999999 APPROVED</p>
JUNE 1 1995 10:30AM PURCHASE</P>
TERMINAL ID: 999999999</P>
NAME: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
ADDRESS:</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
CITY: XXXXXXXXXXXXXXXXXXXXXXXX</p>
STATE: XX ZIP XXXXXXXXX</p>
PART CODE: 999999999</p>
UNIT PRICE: $9999.99</p>
SHIP METHOD:XXXXXX CHARGE: $9999.99</p>
TOTAL AMOUNT: $9999.99</p>
</FORM>
<FORM ACTION=<file_name> METHOD=PAYMENT>
<INPUT TYPE=AUTOSUBMIT>
</FORM>
</BODY>
</HTML>
```

3.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
 DECLINED <error code>
</BODY>
</HTML>
```

FIG.16B

EXHIBIT C
PAGE 19 OF 39

4.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=SCR1 METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
SLIDE CARD:
<INPUT TYPE="MSRT2" SIZE=40 NAME=track2>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="LOCAL" NAME=work_key>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

4.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99 TIME:9999A</P>
ACCOUNT NUMBER 9999999999999999999</p>
EXP DATE: 99/99</p>
AMOUNT: $9999.99</p>
AUTH NUMBER 999999999</p>
</FORM>
APPROVED:999999999<P>
</BODY>
</HTML>
```

4.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.17A

EXHIBIT C
PAGE 20 OF 39

5.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=SCR2 METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
ENTER PIN:
<INPUT TYPE="PASSWORD" SIZE=4 NAME=pin>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="LOCAL" NAME=work_key>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

5.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99 TIME:9999A</P>
ACCOUNT NUMBER 9999999999999999999</P>
EXP DATE: 99/99</P>
AMOUNT $9999.99</P>
AUTH NUMBER 999999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

5.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.17B

EXHIBIT C
PAGE 21 OF 39

6.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=host_URL METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
SLIDE CARD
<INPUT TYPE="MSRT2" SIZE=40 NAME=track2>
<INPUT TYPE="PIN" SIZE=4 NAME=pin>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

6.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99 TIME:9999A</P>
TERMINAL ID: 999999999</P>
ACCOUNT NUMBER 9999999999999999999</P>
EXP DATE: 99/99</P>
AMOUNT $9999.99</P>
AUTH NUMBER 999999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

6.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
<INPUT TYPE="LOCAL" NAME=work_key VALUE="9999999999999999">
</BODY>
</HTML>
```

FIG.18

EXHIBIT C
PAGE 22 OF 39

7.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
      METHOD=SQL
            "IF EXISTS ( SELECT account FROM check_table)
                  SELECT * FROM check_table WHERE account = DDAaccount
            ELSE SELECT * FROM error_table WHERE error_no=1">
<INPUT TYPE="LOCAL" NAME=tid>
SCAN CHECK:
<INPUT TYPE="MICR" SIZE=20 NAME=account>
ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

7.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99                    TIME:9999</P>
TERMINAL ID:                     999999999</P>
ACCOUNT NUMBER  99999999999999999999</P>
AMOUNT                            $9999.99</P>
AUTH NUMBER                      999999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

7.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.19

EXHIBIT C
PAGE 23 OF 39

8.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
        METHOD=SQL
                *BEGIN TRAN
                IF NOT EXISTS (SELECT substring(account,51,20) FROM auth_table)
                        BEGIN
                        SELECT cur_bal FROM cust_tbl WHERE substring(account,51,20)=account
                        SELECT amount = amount - (points / .01)
                        SELECT cur_bal = cur_bal + (amount * .01)
                        UPDATE TABLE=cust_tbl VALUES=(getdate(), account, cur_bal - points)
                        SELECT * FROM log_table WHERE trandate = getdate()
                        INSERT TABLE=log_table VALUES=(getdate(), tid, substring(account,51,20),
                        substring(account,72,4), amount)
                        END
        ELSE SELECT * FROM error_table WHERE error_no=1"
        COMMIT TRAN">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER ACCOUNT NUMBER:
<INPUT TYPE="MSRT1" SIZE=90 NAME=account>
ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
REDEEM POINTS?
<INPUT TYPE="INT" SIZE=6 NAME=points>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

8.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
JUNE 1 1996        10:30AM          PURCHASE</P>
TERMINAL ID:                        999999999</P>
ACCOUNT NUMBER     99999999999999999999</P>
EXP DATE                            99/99</P>
AMOUNT                              $9999.99</P>
AUTH NUMBER                         999999999</P>
</p>
- - - - - - - - - - - - - - - - - - - - - - </p>
            CUSTOMER SIGNATURE</p>
</p>
THANK YOU!</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
POINTS REDEEMED:                    999999</p>
POINTS EARNED:                      999999</p>
CURRENT POINT BALANCE:              999999</p>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

8.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <MESSAGE>
</BODY>
</HTML>
```

FIG.20

EXHIBIT C
PAGE 21 OF 34

9.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
      METHOD=SQL
            "SELECT fields FROM table WHERE condition">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER SEARCH TABLE NAME:
<INPUT TYPE="TEXT" SIZE=10 NAME=table>
ENTER SEARCH FIELD NAMES:
<INPUT TYPE="TEXT" SIZE=100 NAME=fields>
ENTER SEARCH CONDITION:
<INPUT TYPE="TEXT" SIZE=50 NAME=condition>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

9.b. Transaction Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
FIELD1  FIELD2  FIELD3  ------  FIELDN  </p>
------  ------  ------          ------  </p>
XXXXX   XXXXX   XXXXX           XXXXX   </p>
XXXXX   XXXXX   XXXXX           XXXXX   </p>
•
•
XXXXX   XXXXX   XXXXX           XXXXX   </p>
</FORM>
</BODY>
</HTML>
```

FIG.21

EXHIBIT C
PAGE 25 OF 39

10.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM  ACTION=dbase_URL
        METHOD=SQL
        * INSERT TABLE=log_table VALUES=( getdate(), tid, gross_sales, opn_chks, voids,
        emp_disc, mgr_disc, vip_card, man_over, coupons, sales_tax, c_dep1, c_dep2,
        c_dep3, c_dep4, chg_fund,cc_dep, batch_no, chrg_sales, paid_outs, co_sales,
        cc_sales, te_sales, gross_sales - opn_chks - voids - emp_disc - mgr_disc - vip_card-
        man_over - coupons - sales_tax, gross_sales - opn_chks - voids -
        emp_disc - mgr_disc - vip_card - man_over - coupons - c_dep1 - c_dep2 -
        c_dep3 - c_dep4 - chg_fund - cc_dep - batch_no - chrg_sales - paid_outs)
        SELECT * FROM log_table WHERE trandate = getdate()">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER GROSS SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=gross_sales>
ENTER  OPEN CHECKS:
<INPUT TYPE="INT" SIZE=7 NAME=opn_chks>
ENTER VOIDS:
<INPUT TYPE="INT" SIZE=7 NAME=voids>
ENTER EMP DISCOUNTS:
<INPUT TYPE="INT" SIZE=7 NAME=emp_disc>
ENTER MGR DISCOUNT:
<INPUT TYPE="INT" SIZE=7 NAME=mgr_disc>
ENTER VIP CARD:
<INPUT TYPE="INT" SIZE=7 NAME=vip_card>
ENTER MANUAL OVERRINGS:
<INPUT TYPE="INT" SIZE=7 NAME=man_over>
ENTER COUPONS:
<INPUT TYPE="INT" SIZE=7 NAME=coupons>
ENTER SALES TAX:
<INPUT TYPE="AMT" SIZE=8 NAME=sales_tax>
ENTER CASH DEPOSIT 1:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep1>
ENTER CASH DEPOSIT 2:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep2>
ENTER CASH DEPOSIT 3:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep3>
ENTER CASH DEPOSIT 4:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep4>
ENTER CHANGE FUND:
<INPUT TYPE="AMT" SIZE=8 NAME=chg_fund>
```

FIG.22A

EXHIBIT C
PAGE 26 OF 39

```
ENTER  CC DEPOSIT:
<INPUT TYPE="AMT" SIZE=8 NAME=cc_dep>
ENTER BATCH #:
<INPUT TYPE="INT" SIZE=3 NAME=batch_no>
ENTER CHARGE SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=chrg_sales>
ENTER PAID OUTS:
<INPUT TYPE="INT" SIZE=8 NAME=paid_outs>
ENTER CARRY OUT SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=co_sales>
ENTER CREDIT CARD SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=cc_sales>
ENTER TAX EXEMPT SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=te_sales>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

10.b. Transaction Response

```
<HTML>
<BODY>
<FORM  ACTION=TO PRINTER METHOD=POST>
JUNE 1 1995        10:30AM          DAILY REPORT</P>
TERMINAL ID:                              999999999</P>
GROSS SALES                               999999.99</P>
VOIDS                          99          99999.99</P>
EMP DISCOUNTS                  99          99999.99</P>
MANAGER DISCOUNTS              99          99999.99</P>
VIP CARD                       99          99999.99</P>
COUPONS                        99          99999.99</P>
MANUAL OVERRINGS               99          99999.99</P>
SALES TAX                                 999999.99</P>
CASH DEPOSIT 1                            999999.99</P>
CASH DEPOSIT 2                            999999.99</P>
CASH DEPOSIT 3                            999999.99</P>
CASH DEPOSIT 4                            999999.99</P>
CASH DEPOSIT 5                            999999.99</P>
CHANGE FUND                               999999.99</P>
CC DEPOSIT                     999        999999.99</P>
CHARGE SALES                              999999.99</P>
PAID OUTS                       99          99999.99</P>
CARRY OUT SALES                           999999.99</P>
CREDIT CARD SALES                         999999.99</P>
TAX EXEMPT SALES                          999999.99</P>
- - - - - - - - - - - - - - - - - - - - - - - - - - - -</P>

NET SALES                                   9999999</P>
OVER/SHORT                                  9999999</P>
</FORM>
</BODY>
</HTML>
```

FIG.22B

EXHIBIT C
PAGE 27 OF 39

11.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM  ACTION=MAIL TO: mail_to>
 ENTER MAIL ADDRESS:
<INPUT TYPE="TEXT" SIZE=20 NAME=mail_to>
ENTER MESSAGE:
<INPUT TYPE="TEXT" SIZE=100>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

FIG.23

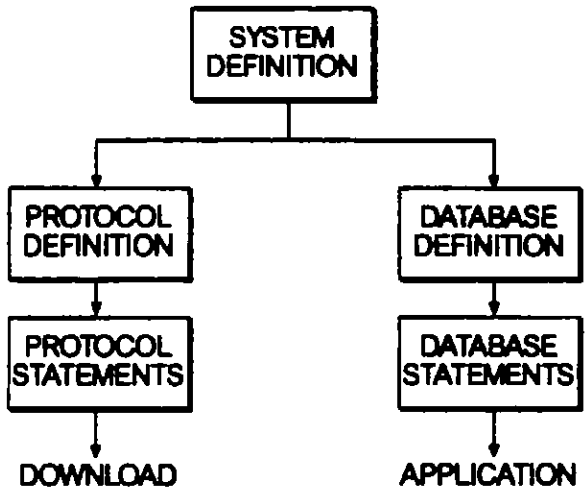

FIG.24A

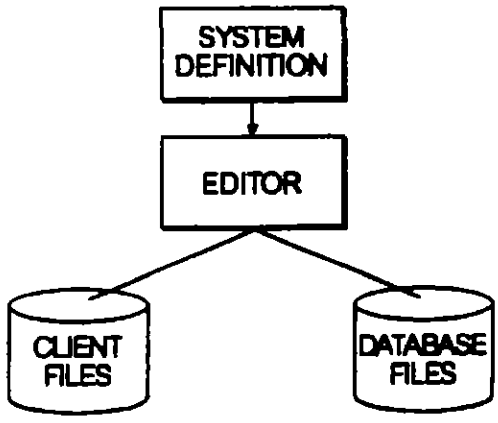

FIG.24B

EXHIBIT C
PAGE 28 OF 39

OPEN NETWORK SYSTEM FOR I/O OPERATION INCLUDING A COMMON GATEWAY INTERFACE AND AN EXTENDED OPEN NETWORK PROTOCOL WITH NON-STANDARD I/O DEVICES UTILIZING DEVICE AND IDENTIFIER FOR OPERATION TO BE PERFORMED WITH DEVICE

This application is a continuation of application Ser. No. 08/995,123 filed Dec. 19, 1997 (now U.S. Pat. No. 5,905,908), which is a continuation of application Ser. No. 08/493,772 filed Jun. 22, 1995 (now U.S. Pat. No. 5,742,845).

FIELD OF THE INVENTION

This invention relates to data transaction systems, and more particularly, to data transaction systems using non-standard input/output devices.

BACKGROUND OF THE INVENTION

Data transaction systems which communicate with a plurality of remote terminals to transfer information used to complete a transaction or compile a database are well known. Typically, such systems include a central transaction processing system which may maintain a database of information such as customer or consumer data. Exemplary information in such a database may include customer identification, customer account numbers, credit limits and/or account balances from which a customer may draw. The central transaction processing system is typically coupled to a plurality of remote transaction or data input terminals. Transaction computers may include special purpose devices such as automatic teller machines (ATMs), point of sale (POS) terminals, credit card terminals, and screen phone terminals. Screen phone terminals are devices which integrate a telephone with an ATM-like device and possibly a magnetic card swipe reader. Data input terminals may include personal computers (PCs) interfaced to data collection devices or special purpose data collection terminals or monitors.

In these known data transaction systems, a user usually initiates a transaction by requesting access to funds in an account or from a credit line maintained by the central processing system. The request is transmitted to the central processing system which performs a verification to determine whether the user is a valid user of the system, has an account within the system, and that the amount of the transaction is within the limits of the consumer's credit line or that the user has the requested funds available in an existing account monitored by the central processing system. The central processing system then transmits authorization for or denial of the transaction to the remote terminal. In response to the message from the central processing system, the remote terminal dispenses cash (for an ATM) or the merchant provides the goods being purchased to the user if the authorization message indicates that the consumer's funds will be transferred to the merchant's account. Similar communication exchanges occur in data systems where electronic documents and other information are provided to a central site for compilation or processing. Consequently, this background discussion applies to all such transaction and data systems. Though the remainder of the discussion is directed to transaction systems, the reader should appreciate that the comments also apply to data systems as well.

The remote terminals may be coupled to the central processing system in several ways. For example, in some

ATM systems, the ATMs are coupled to the central processing system through dedicated telephone or other data communication lines. These systems are preferred because they provide a relatively high degree of security since the dedicated data line coupling the central processing system to the ATM is not generally accessible by members of the public. The physical security of the dedicated data line is, however, expensive because no other traffic may utilize the line. Thus, the cost of leasing the dedicated line to an ATM with relatively low volumes of transactions may yield a high communication cost per transaction.

In an effort to reduce the communication cost per transaction, some transaction or data systems utilize telephone lines through a publicly-switched telephone network (PSTN) which may be accessed by other members of the public. Specifically, devices such as credit card terminals and screen phone terminals typically include a modem which converts the digital messages of the remote terminal into frequency modulated analog signals which may be transmitted over telephone lines to a modem at the central processing system. In other systems, the terminal may communicate digital data directly over ISDN lines of the PSTN to the central processing system. This line of communication between a remote terminal and the central processing system is performed by having the remote terminal dial a telephone number associated with the central processing system to establish communication with the central processing system. This type of communication path is relatively secure because the switching networks for the communication traffic through the PSTN are not readily accessible by the public and during the course of the financial transaction, only the central processing system and remote terminal are on the line.

Regardless of the communication method used to couple the central processing system to the remote terminals, the protocol and data formats used between the devices is typically proprietary. That is, the operator of each financial transaction system designs its own protocol and data message format for communication with the processor at the central site or generates a variant within a standard such as those established by the ANSI committee or the like for such communication. As a result, the remote terminals must include software that supports each operator's protocol and message formats in order to be compatible with an operator's central site. For example, application software in a credit terminal such as the TRANZ330, TRANZ380, or OMNI390 manufactured by VeriFone implement one or more of the communication protocols and formats for National Data Corporation (NDC), VISANET, MASTERCARD, BUYPASS, and National Bancard Corporation (NaBANCO) system processors in order to support transactions with the most popular transaction centers. Thus, the communication software absorbs a significant amount of terminal resources which could be used to support other terminal operations.

A related problem arises from the expanding home banking market. A customer of home banking system typically uses a screen phone terminal or a personal computer (PC) having a modem to establish communication through a PSTN to a central transaction processing system. Again, the operator of the central processing system must provide information regarding the data message formats for communicating with the central processing system to a vendor of software for the home banking terminals or must provide that software to its customers. As a result, home banking customers must purchase software to communicate with each banking system of which the customer wants to be a

member. This cost and the need to install additional communication programs may make some consumers reluctant to be a member of more than one banking system or to change banking systems.

A communication system becoming increasingly popular and which provides standardized communication is the Internet. The Internet is an open network of networks which communicate through a variety of physical communication devices such as telephone lines, direct communication lines, and the like. Each network is coupled to the main Internet network for communication through a host computer supporting a TCP/IP router or bridger. The host computer typically includes a program, frequently called a Web server, which acts as a gateway to resources at the host computer which may be resident on the host computer or a network coupled to the host computer. Each server has an address identifying the location of the resources available through the Web server. The router recognizes communication for the server and directs the message to the server or it recognizes that the communication should be forwarded to another server. As a result, communication within the Internet may be point-to-point, but more likely, the communication path is a somewhat circuitous one with the information passing through the routers of multiple servers before reaching its final destination.

A number of message protocols and formats have been developed for the Internet. The physical communication protocol and data message format is the Transport Control Protocol/Internet Protocol (TCP/IP). The TCP/IP protocol involves multiple layers of encapsulating headers containing communication information which are used to provide byte streams or datagram communications to computers on the networks coupled to the Internet. Encapsulated within TCP/IP headers are protocols which are used to format the data messages or transfer data from one computer to another computer coupled to the Internet. These protocols include File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), Telnet, and Hyper Text Transport Protocol (HTTP). The advantage of these protocols is that each provides a standardized communication format for transferring information between computers on the Internet. These protocols are typically called open system protocols ask they are publicly known and may be utilized by any programmer to develop programs for communicating with another computer coupled to the Internet. These non-proprietary protocols have contributed to the acceptance of using the Internet as an open network for coupling computer networks together.

While the Internet provides an open network for computer communication with publicly accessible protocols and formats, the Internet suffers from a number of limitations which preclude its effective use as a transaction or data system which uses non-standard I/O terminals and devices. First, circuitous communication presents a number of security issues for such a system. For example, a Web server could incorporate a router which examines the address of each message coming through it and upon recognizing an address associated with a central transaction processing system, copy the data message for the unauthorized retrieval of customer-sensitive information such as account numbers and personal identification numbers (PINs) which may be contained in the message.

A second limitation of open networks such as the Internet is that communication on such networks is only supported for computers acting as servers or clients. Specifically, all of the protocols and formats are constructed for standard input/output (I/O) operations for a PC terminal. That is, text information is directed to a standard monitor screen, user input is expected from a standard keyboard, and files are transferred to standard peripherals such as a hard disk or diskette drive. Especially absent is the ability in open network protocols for communication with devices that only use communication interfaces such as RS-232C. As a result, communication over the Internet is primarily performed with standard PCs through network communication methods and interfaces.

This presents a number of problems for home banking or for interfacing non-standard I/O terminals such as credit card terminals or screen phones to open networks such as the Internet either directly or through a PC. Generally, non-standard I/O devices are devices which interface to a PC through a port not normally used for networks, such as a RS-232C port, or are devices which have limited input and output capabilities such as small screen displays or ten keypads. These devices are not supported on the Internet because servers use protocols that communicate with PCs supporting standard QWERTY keyboards and standard monitors. Consequently, users are limited to entering account numbers and the like through a keyboard of a PC-like device for processing at a central transaction processing system. To request a transaction, one need only have a person's credit card account number. If the credit card number had to be input through a magnetic card reader, unauthorized access to a customer's account would be less likely since physical possession of the credit card would be required to initiate the transaction.

Another limitation of the standard I/O devices currently supported by the open network protocols is the lack of encryption. For example, PIN pads, which are typically incorporated in ATMs, automatically encrypt in hardware a PIN entered by a user. Such devices typically encrypt the number by implementing a data encryption standard (DES) algorithm in hardware before the PIN is transmitted or stored. When a standard keyboard is used to input the PIN, no hardware encryption is performed and, as a result, an unencrypted copy of the PIN is provided to the memory of the PC. Storage of unencrypted PINs is in contravention of current banking regulations. If PIN pads could be read via Internet protocols, then such a lapse in PIN security would be less likely to occur.

Another I/O device not supported on open networks are smart cards which are increasing in use. Smart cards include a processor and memory in which information regarding the amount of funds in a particular account, a transaction history, account numbers, and customer data may be stored. The card may be read through a smart card reader which is a computer having a processor and memory but usually provided with non-QWERTY keypads and limited displays. A transaction processor may validate a card owner through a PIN provided through a keypad, determine the amount of money remaining on the card and debit the card itself for a transaction amount by communicating with the smart card reader with one of the proprietary protocols discussed above. Such information is not readily obtainable by the owner of the card and so cannot be entered through a keyboard or the like. Smart card readers are non-standard devices which may be coupled to a PC through a COMM1 or COMM2 port. However, none of the standard protocols and message formats for open network communications currently provide I/O operations for such devices.

All systems which attempt to provide three party communication to execute an electronic transaction suffer from a number of limitations which present risks greater than those in a normal transaction performed at the point of sale.

In a typical point of sale (POS) transaction, the consumer hands a debit or credit card to a merchant's agent who may examine the card for security markings such as holograms, watermarks, or a cardholder signature. The agent then places the card into a reader for acquiring information from the card and, in some cases, have the consumer enter a PIN into a PIN entry device which encrypts the PIN in a hardware implemented scheme. If the PIN is entered, it is transmitted with the information from the card to a processing center, typically in one of the formats discussed above, under a X.25 protocol or the like. The processing center returns an authorization granted or denied message. The reader typically has a printer coupled to it through an RS-232C port or the like and a purchase agreement is printed. The consumer signs the agreement, the merchant's agent may verify the signature, and the merchant retains an original of the agreement and the consumer a copy. In this scenario, the merchant has initiated the communication to the processing center. The safeguards noted above permit the processing center to charge a merchant a lower processing fee than when a consumer initiates a transaction. Consumer initiated transactions present a greater risk because the consumer provides an agent an account number in a telephone conversation or non-encrypted DTMF transmission. Thus, there is no card inspection, signature verification, or PIN verification. As a result, such transactions are limited to credit cards because debit cards require that the cardholder be present to enter a PIN into an appropriate PIN entry device.

What is needed is a system that permits consumers remote from a merchant to order goods and present payment in a secured manner so the merchant's risk and processing costs, as well as a cardholder's exposure to fraud, is reduced. What is needed is a way for a processing center to communicate through an open network with non-standard I/O devices such as credit card terminals, personal digital assistants, and screen phone terminals or with non-standard I/O devices coupled to the open network through a PC or the like. What is needed is a transaction or data system which utilizes an open network such as the Internet to support electronic transactions or data compilation in a secure manner without undue limitation as to the devices with which communication may be made.

SUMMARY OF THE INVENTION

The present invention provides transaction and data systems which may be implemented on an open network such as the Internet. The system comprises a server for communicating in an open network protocol and a plurality of input/output (I/O) devices coupled to the server through an open network, the I/O devices communicating with the server in the extended open network protocol that supports communication with non-standard I/O devices over the open network. The system of the present invention provides a server with the capability of communicating with a number of I/O devices useful in transaction and data systems which heretofore have been unsupported on an open network system such as the Internet.

The system of the present invention is implemented by extending present open network communication protocols and data message formats to communicate with non-standard I/O devices either coupled to an open network as a client or coupled to an open network through a client, such as a PC, credit card terminal, screen phone, or PDA. That is, commands which are compatible with the communication schema of a presently-implemented protocol for the Internet are used and additions are made to commands implemented within the control structure of that existing protocol to support non-standard I/O device communication. At the server, the extended protocol is further supported by a common gateway interface (CGI) which converts the communication from a non-standard I/O device to a format which is compatible with a transaction or data application program which may be executed on the server or a computer coupled to the server. In this manner, the CGI permits the processing of the extended capability commands to be segregated from the communication functions performed by the server.

Preferably, the server and the I/O devices communicate through an Internet protocol and most preferably, the Hyper Text Transport Protocol (HTTP), to exchange data between an application program and non-standard I/O devices over an open network. Although HTTP is the preferred protocol used to implement the present invention, other protocols such as Telnet or SMTP, for example, may also be extended in a similar manner. Specifically, the HTTP protocol is expanded to communicate with printers, magnetic card readers, credit card terminals, smart card readers, check readers, PIN pads, bar-code readers, PDAs, or the like, and includes a command which instructs a non-standard I/O device to disconnect from the open network and re-couple to a transaction processing system to transfer funds from a consumer account to a merchant account through a PSTN or dedicated data line. By using these extended capability commands within HTTP, a processing system may operate on an open network such as the Internet and communicate with transaction or other data I/O devices which have not previously been able to couple to such open networks. Such a system may be used to execute a transaction between a consumer and a merchant so the merchant receives remittance information in a timely manner. The system permits the consumer to initiate a transaction and order from a merchant and then use a more secure link supported by PIN entry devices or the like to reduce the risk of fraud for the transaction.

Because the server may communicate through such open networks with non-standard I/O devices, the transaction or data processing system is available for the ever-expanding market available through the Internet. Such a system is able to communicate with non-standard I/O devices in myriad locations such as retail establishments or in consumers' homes. For example, a consumer may utilize the standard capability of an Internet protocol to communicate with a server that provides information regarding services or goods for sale over the Internet and then consummate a sales transaction by using the extended capability of the Internet protocol. Such a home consumer could provide transaction data through a smart card reader coupled to a COMM1 or COMM2 port of a PC. A database program executing at the server for the central processing site may accept product ordering information from a non-standard keypad or touch screen associated with a screen phone terminal at the remote site and then communicate with the smart card reader to consummate the transaction. Such a transaction system requires that the consumer have physical possession of the smart or credit card and not simply knowledge of the account number. Likewise, the server would be able to communicate with a PIN pad or the like to ensure the hardware encryption of PINs and other data before it is transmitted to the server site. Such a system is less susceptible to consumer fraud.

Another feature of the present invention is a PAYMENT command implemented in the extended Internet protocol that directs a non-standard I/O device or a PC interfaced with such devices to communicate with a transaction pro-

cessor through an alternative communication link. In one form, the PAYMENT command is used by a merchant terminal to submit a consumer's account number with a merchant deposit account number through a PSTN network or the like to the processing center. In another form of the PAYMENT command, a client program in a consumer's terminal receives an account number for a merchant account from a merchant's server with the PAYMENT command. On receipt of this command, the client program suspends its operation and passes the account number to a conventional bank processing program co-resident in memory. The bank processing program establishes a standard communication link with a transaction processing system through a dedicated data line or a PSTN network. Using that communication link, the bank processing program executes a commercial transaction using a standard VISA protocol or the like. The consumer may use a magnetic stripe reader and a PIN entry device to improve the security of the data transmission. The transaction center may transmit remittance data over the open network to the merchant so the merchant is apprised of payment and ships the ordered product. Once this consumer initiated transaction is complete, the bank processing program terminates and returns control to the client program which may terminate communication with the open network or retrieve information from another server on the open network for another transaction. In this way, the user may use the open network for non-confidential communication such as collecting product information, pricing, and product availability. This information may be collected quickly and efficiently using the extended Internet protocol. The conventional bank processing program and more secure communication links may then be used for the confidential information required for the transaction. Thus, the present invention is able to combine the features and advantages of the Internet with the more secure communication link and data security enhancing devices of systems presently known.

Preferably, an editor is provided which permits a user to define an application database table with data fields, define client application data fields, and define the integrated forms for communicating data between the defined database tables and a client application. The editor verifies the syntax of the user generated integrated forms containing extended Internet protocol statements and client application statements. The editor ensures that the variable names for the client application and the data fields for the database application correspond. Following the generation of the integrated form, the editor parses the integrated form to segregate the database language statements from the extended Internet protocol statements. A database language identifier is substituted in the Internet protocol statements for the database statements contained in the integrated form. The Internet protocol statements are downloaded as a file which is interpreted by the client program for the collection and submission of data from non-standard I/O devices to the database application. The database language statements segregated from the extended Internet protocol statements are placed in a second file which is named to correspond to the database table defined by the user. The CGI application recognizes the database language identifier contained in the returned forms of the Internet protocol statements. The CGI application correlates the database identifier with the file previously generated by the editor which contains the database command statements. The application then inserts the data from the returned form into the database command statements and provides the re-integrated database command statements to the database application. In this manner, the database may be queried by or retrieve data from the non-standard I/O device.

In the most preferred embodiment, the editor permits a user to develop integrated forms comprised of the extended HTML language and standard query language (SQL) database application statements. In this manner, the user does not have to manually generate the SQL commands, the HTML commands, and carefully correlate the data fields of the two commands in order to implement a transaction between a client and a database.

These and other advantages and features of the present invention may be discerned from reviewing the accompanying drawings and the detailed description of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention may take form in various components and arrangement of components and in various steps and arrangement of steps. The drawings are only for purposes of illustrating a preferred embodiment and are not to be construed as limiting the invention.

FIG. 1 is a diagram of an open network system in which the present invention is utilized;

FIG. 2 is a diagram of the format of the FORM and INPUT tags implemented in the preferred embodiment of the present invention;

FIG. 3 is a diagram of the preferred SQL commands supported in the preferred embodiment of the present invention;

FIG. 4 is a flowchart of the high level processing of the client program which interprets the HTML files of the preferred embodiment of the present invention;

FIG. 5 is a flowchart of the HTML file processing performed by the client program of the preferred embodiment of the present invention;

FIG. 6 is a flowchart of the attribute processing for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. 7 is a flowchart of the processing of the ACTION attribute for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. 8 is a flowchart of the processing for the METHOD attribute for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. 9 is a flowchart of the attribute processing for the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. 10 is a flowchart of the processing for the TYPE attribute for the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. 11 is a flowchart of the processing for the NAME attribute of the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. 12 is a diagram of the format for the ACTION attribute for the FORM tag performed by the common gateway interface between the Web server and an application program;

FIG. 13A is a diagram of the possible communication paths which may be used by an I/O device according to the principles of the present invention;

FIG. 13B shows an exemplary FORM tag and INPUT tag for the PAYMENT method implemented in a merchant's terminal according to the principles of the present invention;

FIG. 13C shows an exemplary FORM tag and INPUT tag for the PAYMENT method implemented in a consumer's terminal according to the principles of the present invention;

FIG. 14 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to

EXHIBIT C
PAGE 32 OF 39

generate the HTML files for the client program and the SQL files for the application program for a card initiated payment authorization and capture transaction;

FIG. 15 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a bar code reader input with card-initiated payment authorization transaction;

FIG. 16 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a key input order with secure payment transaction;

FIG. 17A shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a smart card debit (Type 1) transaction;

FIG. 17B shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a smart card debit (Type 2) transaction;

FIG. 18 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a debit card transaction;

FIG. 19 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a check verification transaction;

FIG. 20 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a customer frequency transaction;

FIG. 21 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for an item search transaction;

FIG. 22 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for retail store end of day reporting;

FIG. 23 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a store reporting an e-mail transaction;

FIG. 24A is a diagram of a manual development process for the files interpreted by the client program and the files interpreted by the application program in accordance with the principles of the present invention; and

FIG. 24B is a diagram of the generation of the files interpreted by the client program and the files interpreted by application program performed by an editor constructed in accordance with the principles of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

A transaction or data system constructed in accordance with the principles of the present invention is shown is FIG. 1. The system **10** includes a Web server **12** which is coupled to an open network **14** such as the Internet for communication with various I/O devices and terminals. For example, the I/O devices which may be coupled directly to network **14** include standard I/O devices already supported by Internet protocols such as PCs **30** and non-standard I/O devices such as a screen phone terminal **16**, a personal digital assistant (PDA) **18**, and a credit card terminal **20**. Other exemplary non-standard I/O devices such as smart card reader **32**, personal identification number (PIN) pad **34**, magnetic card swipe reader **36**, printer **38**, or the like, may be coupled to PCs through non-standard I/O ports such as COMM1 and COMM2 ports or to other non-standard I/O devices such as phone terminal **16**, PDA **18**, or credit card terminal **20**. Typically, these devices are coupled to PCs or devices **16**,**18**, or **20** through an interface such as a RS-232C interface. Merchants or other vendors may use a Web server **2** to couple to network **14** to communicate with the devices and processing system **40**.

The Web server **12** is preferably coupled to a Common Gateway Interface (CGI) application **28** which converts and communicates the data and commands between the devices on network **14** and the processing system **40** so the I/O devices do not have to use the database command language to interact with the database. System **40** and the devices may communicate directly if they are implemented in the same language or if a user implements a communication interface such as CGI **28** that correlates data fields in the client with those in system **40**. Server **12**, CGI **28**, and the applications supporting system **40** may all reside on a single host computer or they may reside on separate computers coupled together by a local area network (LAN) or a wide area network (WAN). Preferably, the application interfaces with a database which supports Open Data Base Connectivity (ODBC) and Structured Query Language (SQL).

The communication sessions between the I/O devices coupled to the open network **14** and the Web server **12** are generally conducted in the same fashion as Internet protocol communication sessions are currently performed. That is, the I/O device establishes a communication connection with Web server **12**, sends a request to the Web server, the Web server responds to the request and the I/O device or server closes the connection. Preferably, the non-standard I/O devices or PCs interfaced to such devices selectively couple to a local access port on the open network **14** through a local modem/ISDN connection. In this manner, the device is only coupled to the open network **14** when a transaction or a data operation is to be performed. While connected to the open network **14**, a device may access a number of servers to accomplish a purpose. For example, a device may couple to a local access port and communicate with a first server to check inventory levels at a site, communicate with a second server to order stock for the inventory, and communicate with a third server to settle payment for the ordered goods. When all aspects of the transaction are complete, the connection with the local access port is terminated. In the preferred embodiment of the present invention, the protocol used to transport data messages between Web server **12** and the I/O devices coupled to the open network **14** is the Hyper Text Transport Protocol (HTTP), although other open system protocols utilized on the Internet may be used.

In standard HTTP protocol, a client program executing in one of the I/O devices may initiate communication with a server by sending a query message of the format:

http://<host>:<port>/<path>?<search part>

The message identifies the client as seeking communication with a HTTP server at the host address on the specified

EXHIBIT C
PAGE 33 OF 39

port. In the HTTP protocol, the default value for the port is 80 and the host address is the Internet protocol (IP) address of the type well-known in the art. The path value selects the file in the HTTP server which is activated in response to the message and the search part specifies a query for the selected file. In the initial communication, the query may be omitted so that the selected host file responds to the client program before a query is processed.

In the present invention, the client program uses a similar message to initiate a transaction or data operation, except that database commands are preferably embedded in a file at the server **12** and not in the "search part" of the command, although search parts may be constructed in accordance with the principles of the present invention that support non-standard I/O devices. Preferably, the client program interprets Hyper Text Markup Language (HTML) files containing HTML commands for communicating data between non-standard I/O devices and server **12**. Most preferably, the HTML commands contain identifiers which are used by the CGI to place data returned in the forms of the HTML commands into database commands for queries or data insertions for the database. HTML is a command language well known for the retrieval and display of electronic documents for standard I/O devices such as PCs supported by full screen monitors, QWERTY keyboards, and standard peripherals such as hard disk drives and diskette drives. Standard HTML commands use text and previously known commands that reference Universal Resource Locators (URLs) to support the communication of electronic documents. These documents are files which may contain HTML commands, text, audio, video, or image data. The present invention extends HTML with commands that support communication between the server and the non-standard I/O devices.

In the HTTP protocol, data may be obtained during a communication session by using a tag called a FORM as part of the file defined by <path> in the command discussed above. The FORM format for standard HTTP is:

```
<FORM ACTION="URL"
    METHOD=GET POST
>
Command
</FORM>
```

where "|" is an "OR" operator. The commands supported by standard HTTP are INPUT, SELECT, and TEXTAREA. Additionally, standard HTTP permits the inclusion of text data in the command area. In the present invention, HTML has been extended to support new ACTIONs, METHODs, and INPUTs.

In accordance with the principles of the present invention, tags are preferably used to identify device transfers and input operations. Preferably, the FORM tag is used to identify device transfers and ACTION and METHOD attributes further identify the device operation. As shown in FIG. 2, the extended ACTION field may include a FROM and TO attribute for accessing a local terminal file or smart card reader or a TO PRINTER attribute for directing output data to a printer local to the I/O device. The FROM and TO attributes for accessing local files and smart card readers and for directing output data to a local printer have previously been unsupported in any Internet protocol. As a result, the server **12** may access non-standard I/O peripherals for any of the I/O devices used in the transaction or data system **10**. The ACTION="URL" is a part of standard HTTP and is well known.

The METHOD attributes may include the GET, POST, PAYMENT, or SQL methods. The GET and POST methods are currently supported in standard HTTP and are well known. The PAYMENT attribute is a directive to deliver data retrieved by an INPUT command to a private payment network for authorization and settlement and is not available in current Internet protocols. This directive is used by the client program to activate a conventional financial transaction application which communicates with the transaction system over a dedicated data line or PSTN in a known protocol such as VISA. Such an attribute is used where the more secure physical connection between remote site and transaction system and data encryption devices or the like are preferred. The SQL method preferably identifies a database language file which CGI **28** uses to correlate data in the HTML FORM to an insertion or query command contained in the file.

The preferred format for the INPUT tag which is used to identify input operations is also shown in FIG. **2**. The TYPE and NAME attributes are used to define a non-standard I/O device or local storage variable for the input of data. The TYPE field values "text," "password," "checkbox," "radio," "submit," and "reset" are previously known, as are the attributes NAME, VALUE, CHECKED, SIZE, and MAXLENGTH. To support the extended capability of the present invention, the TYPE attribute preferably includes attributes MSRT1 for reading track 1 of a magnetic swipe reader, MSRT2 for reading a magnetic swipe reader track 2, KEY for reading input from a terminal command keypad, PIN for reading a personal identification number pad, BCW for reading a bar code wand, MICR for reading a check magnetic code reader, ATM for reading a dollar amount via a key input mask, INT for reading an integer via a key input mask, LOCAL for reading input from a variable in the local storage of an I/O device, and AUTOSUBMIT for returning a FORM with information to the server.

The NAME attribute used with the INPUT tag identifies reserved word names for local storage in the device executing the client program. Preferably, the NAME attribute identifies ip_address, host_phone, tid, work_key, datetime, and deposit_acct as local storage areas in the local device for the terminal's Internet Protocol (IP) address, Internet access phone number, terminal ID, PIN encryption working key, date/time, and merchant account number, respectively. These attributes are used with the INPUT tag to read non-standard I/O devices which may be coupled to open network **14**. For example, an INPUT TYPE=MSRT1 attribute causes the client program residing within a magnetic stripe reader to input data from track 1 of a stripe reader and insert that data into a FORM which is returned to Web server **12** for processing by an INPUT TYPE=AUTOSUBMIT statement.

Preferably, the database language commands which may be embedded in the extended HTML are SQL commands such as those shown in FIG. **3**, although other database languages may be used. The SELECT command may include the names of data fields in a database so the device on network **14** may request a data item from a database at the central processing system. The database table is identified by the FROM attribute and the conditional selection of data from an identified database table may be defined by a WHERE attribute. Additionally, records may be requested from an identified database in ascending or descending order or in groups of two records at a time using the ORDER attribute. Additionally, the SELECT field command with the GROUP attribute provides I/O devices with the capability of retrieving records grouped under an identified name.

EXHIBIT C
PAGE 34 OF 39

Additionally, the I/O devices may either insert new data into an identified database with the INSERT attribute or update data already existing in a database with the UPDATE attribute. The values for the INSERT attribute may be identified with the VALUES attribute, and the SET and WHERE attributes may be used to define and conditionally update values in the identified database. Preferably, the present invention implements two DELETE and CREATE attributes. The DELETE attribute deletes all items in an identified column of a database table which may satisfy a condition defined by a WHERE attribute. The CREATE attribute creates a database table having a primary key identified by the PRIMARY KEY attribute.

Preferably, the server program executes on a computer system having at least an Intel 80386 or better processor with at least 4 megabytes of RAM and at least 3 megabytes of hard disk space available. The computer system running the server may operate any known server platform operating system such as WINDOWS 3.1, WINDOWS 95, or WINDOWS NT, UNIX, AIX, and others. The non-standard I/O devices require a processor of a Z80A type or better, at least 32K bytes of RAM, and at least 32K bytes of ROM. The device includes a modem capable of at least 1200 bits-per-second (bps) but other modem speeds may be used for communication between client and server. Alternatively, the device may be coupled to a LAN which in turn is coupled to the Internet for communication with server **12**. A typical non-standard device which executes the client program is a VeriFone OMNI390, OMNI395, or VuFone terminal. OMNI390, OMNI395, and VuFone are trademarks of VeriFone, Inc., of Redwood City, Calif. Other exemplary devices include Phillips Screen phone, Hypercomm T7 terminal, and Apple Computer Newton MessagePad.

To build the preferred HTML files which CGI **28** preferably uses to implement the client program and database application, the user preferably uses an off-line editor. The files generated by the editor are preferably comprised of an integrated statements formed from HTML statements and database statements for retrieving and writing data with the database. Exemplary files showing such integrated statements for performing transactions are depicted in FIGS. **14–23B**. After such a file is generated, the editor parses the integrated statements into HTML statements and into database statements such as SQL commands. The HTML files required by the client program to support communication with a transaction or data processing center may be downloaded to a device or PC for execution. The files containing the database application statements used by the CGI interface to communicate data with the database application program preferably reside on server **12**. Preferably, the database files used by the CGI interface include SQL commands for the application program interfaced to an ODBC compliant database.

The general format of the HTML commands in the HTML files used for communication with a client program and server are of the general format: TAG ATTRIBUTE. Preferably, the TAG field may be one of FORM, INPUT, SQL, or TEXTAREA. The ATTRIBUTE field value depends upon the TAG value. Preferably, the FORM tag may include the ACTION or METHOD attributes where the ACTION attributes include the FROM<file>, TO PRINTER, TO<file>, and TO SCR values noted above, as well as the standard HTML ACTION value of URL=<file>. The METHOD attributes include the PAYMENT and SQL attributes noted above, as well as the standard HTML METHOD values of GET and POST. Also in accordance with the principles of the present invention, the INPUT tag may include TYPE, NAME, VALUE, CHECKED, SIZE, and MAXLENGTH attributes. These attributes are previously supported for the INPUT tag in HTML, however, the present invention further includes TYPE values of MSRT1, MSRT2, KEY, PIN, BCW, MICR, AMT, INT, LOCAL, and AUTOSUB, as well as the standard HTML TYPE values of TEXT, PASSWORD, CHECKBOX, RADIO BUTTON, SUBMIT, and RESET. The present invention also supports NAME attributes of IP_ADDRESS, HOST_PHONE, TID, WORK_KEY, DATETIME, and DEPOSIT_ACCT to identify local storage areas as well as standard HTML NAME attribute <Field_NM> to identify a FORM variable.

The preferred high level processing of the client program is shown in FIG. **4**. That processing includes an idle step (Block **100**) in which the program performs general housekeeping tasks such as maintaining internal time, scanning for input which may activate the device, or other known functions. Further processing is activated by some operator action at the device or PC which causes the device to either open a remote URL (Block **102**) or open a local URL (Block **104**). If a remote URL is required, the device transmits a message of the format discussed previously which is routed through the open network and delivered to a server **12** for a transaction or data processing system (Block **106**). The HTML file selected at the server **12** is identified by the remote URL in the initial communication between the device and server **12** and that URL is used to return the selected HTML file to the device for processing (Blocks **108**, **110**).

FIG. **4** also shows that an operator may initiate an open local URL function by typing in a command or by pushing a hot key which is associated with a local URL. The I/O device reads the HTML file identified by the URL from local memory (Block **112**) and passes the HTML file to the function for processing HTML files (Block **110**). After a file is processed (Block **110**), the client program determines whether the HTML file is to be stored (Block **114**). If it is not, the process returns to the idle processing (Block **100**). Otherwise, the process determines whether the HTML file is to be associated with a hot key (Block **116**) and, if it is, it stores the file and generates the link between a hot key and the stored file (Blocks **118**, **120**). If the HTML file is only to be stored, no association is made with a hot key and the file is simply stored in local memory (Block **20**). The client program then returns to idle processing (Block **100**).

The high-level processing for the HTML file (Block **110**, FIG. **4**) is shown in further detail in FIG. **5**. The process begins by scanning the HTML file for a TAG (Block **140**). If no TAG is found, the file is not in proper format for processing and processing returns to Block **114** discussed in FIG. **4** above. If a TAG is found (Block **142**), the process determines whether the TAG is a FORM TAG (Block **144**) or an INPUT TAG (Block **146**). If it is a FORM TAG, then the FORM TAG is processed and the program continues by looking for other TAGS to process (Block **140**). If the TAG is an INPUT TAG, the INPUT TAG is processed (Block **150**) and the program continues by looking for other TAGS to process (Block **140**). If the TAG is one of the standard HTML TAGS, the program implements the TAG in standard known ways (Block **152**) and then scans for other TAGs to process (Block **140**).

Processing the ATTRIBUTES used to implement a FORM TAG is shown in FIG. **6**. That process continues by scanning the HTML file for an attribute (Block **160**). If an attribute is not found (Block **162**), the program returns to scan for other TAGS (Block **140**, FIG. **5**). If an attribute is found, the program determines whether it is an ACTION

EXHIBIT C
PAGE 35 OF 39

attribute (Block 164) or a METHOD attribute (Block 166). Depending on the type of attribute, the appropriate function for processing the attribute is executed (Blocks 168 or 170) and scanning for additional attributes continues (Block 160). If the attribute is not an ACTION or METHOD attribute, there is an error in the file and processing returns to scan for other TAGs.

The processing for the ACTION attribute is shown in FIG. 7. There, the ACTION attribute is examined to determine whether it is a FROM<file> (Block 180), TO PRINTER (Block 182), TO<file> (Block 184), TO SCR (Block 186), FROM SCR (Block 188) or a URL=<file> (Block 192). The URL=<file> ACTION is a standard HTML action which is processed in a known way (Block 194). The FROM <file> action is processed by reading data from a file associated with the I/O device or PC interfaced to the I/O device (Block 196). The TO PRINTER action results in data in the FORM being sent to the printer (Block 198) while the TO <file>action results in data in the FORM being written to a local file (Block 200). The TO SCR action causes data to be written to the smart card via a smart card reader (Block 202) and the FROM SCR reads data from a smart card through a smart card reader (Blook 204). After the appropriate action processing takes place, the HTML file is scanned for additional ACTION values to perform (Block 206), and if one is found, the process continues. If no attribute is located (Block 208), the process returns to scan for other attributes (Block 160, FIG. 6).

The processing for the METHOD attributes for FORM tags are shown in FIG. 8. The process determines which type of METHOD is present in the FORM and then properly processes the attribute. For the GET and POST methods (Blocks 210, 212) the processing is the same as that performed in standard HTML (Blocks 226, 228). That is, for the GET method, the identified URL<file> is queried for data while the POST attribute causes data to be transferred to the URL<file>. The preferred METHOD attributes extending the HTML implementation of the present invention are SQL (Block 214), and PAYMENT (Block 224) attributes. The SQL attribute is preferably not expanded into a SQL command at the client, but rather is expanded by the CGI 28 at server 12 by correlating the data or variable field names in a returned form with the SQL commands stored at the server. This processing is done in a manner described in more detail below. The client program passes the SQL file identifier to the server 12 (Block 230). The processing of the PAYMENT command (Block 232) is discussed in more detail below. The HTML file is scanned for other METHODs (Block 242, 244), and, if one is found, the processing continues by identifying the METHOD (Blocks 210–224). Otherwise (Block 244), the process returns to scan the HTML file for other ACTION or METHOD attributes (Block 160, FIG. 6).

Processing for the INPUT tag is shown in FIG. 9. The process scans the HTML file following the INPUT tag for attributes (Block 250). If no attributes are found (Block 252), the process continues by scanning the HTML file for other tags to process (Block 140, FIG. 5). If an attribute is found and it is a TYPE attribute (Block 254), it is processed (Block 256), and if the attribute is a NAME attribute (Block 258), it is processed (Block 260). Both the TYPE and NAME processing is shown in more detail in FIGS. 10 and 11, respectively. If the attribute is neither a NAME or TYPE attribute, it is a standard attribute for an INPUT tag supported by standard HTML and is processed in a known manner (Block 262). Following processing of the INPUT attribute, the HTML file is scanned for other attributes to process (Block 250).

Processing for the TYPE attribute is shown in FIG. 10. The process first identifies the TYPE attribute for the INPUT tag and then performs the appropriate processing. The new TYPE attributes of the preferred embodiment of the present invention are MSRT1 (Block 270), MSRT2 (Block 272), KEY (Block 274), PIN (Block 276), BCW (Block 278), MICR (Block 280), AMT (Block 282), INT (Block 284), LOCAL (Block 286), and AUTOSUB (Block 288). If the TYPE attribute is not one of these, it is a standard HTML type attribute that is processed in a known manner (Block 310). Each of the new HTML TYPES supported by the present invention causes an I/O operation with a non-standard device. Specifically, these operations are the reading of Track 1 of the magnetic stripe reader (Block 290), the reading of the second track of the magnetic stripe reader (Block 292), the reading of a keypad (Block 294), the reading of an encrypted PIN through a PIN entry device (Block 296), the reading of a bar code through a bar code reader (Block 298), the reading of encoded data on a check through a magnetic check reader (Block 300), the reading of a dollar amount from a keypad through a key input mask (Block 302), the reading of a number from a keypad through a key input mask (Block 304), the reading of data from a local variable (Block 306) and the submission of the data read from one of these devices in a FORM returned to the server 12 (Block 308). The data mask for AMT constrains the dollar amount read to a predetermined number of characters with only two characters following the decimal point. The data mask for INT ensures the number is an integer value within a predetermined range. Processing continues by scanning the HTML file for other TYPE attributes (Block 312) and, if another TYPE attribute is found (Block 314), processing continues by determining the TYPE attribute and performing the appropriate processing. Otherwise, the process returns to scan the HTML file for other attributes (Block 250, FIG. 9).

The NAME attribute processing is performed in accordance with the process shown in FIG. 11. That process examines the NAME attribute to determine if the variable name identified by the attribute is IP_ADDRESS, HOST_PHONE, TID, WORK_KEY, DATETIME, or DEPOSIT_ACCT (Blocks 320, 322, 324, 326, 328, 330). If they are, the INPUT value resulting from one of the INPUTS in a FORM of the HTML file is stored in a local variable identified by the NAME attribute. Following storage (Block 332), the file is scanned for other NAME attributes (Block 328) and, if there are none (Block 332), processing continues by scanning for other attributes for the INPUT tag (Block 250, FIG. 9). If the NAME attribute is a standard HTML INPUT NAME, it is processed by known methods (Block 336). Processing then continues by scanning for other NAME attributes to process (Block 338, 340). Otherwise, the process returns to scan the HTML file for other attributes (Block 250, FIG. 9).

CGI 28 receives Internet protocol statements in a file transmitted from a client program and provides data from those statements to the application(s) implementing system 40 and receives the output of system 40 and provides them to the client program in a file. CGI 28 may be implemented by a program developed by a user using a manual development method as shown in FIG. 24A. That method requires a user to generate a system definition from which a file statement definition for the client and application are developed to implement the transactional or data system. Using the file statement definitions, the user generates the files for the client and database programs which are interpreted by the respective programs to implement transactions or data

EXHIBIT C
PAGE 36 OF 39

processing. This process requires the user to not only have knowledge regarding the transaction or data process but specific details of the interaction between the client and database. The user is further required to resolve and correlate all data identifiers in the statements for the client and database environments.

Preferably, CGI **28** is developed with an editor that only requires the user to define the system with statements which are an integration of the protocol statements and the database language. The process implemented by this editor is shown in FIG. **24**B. Examples of such integrated statements for files which implement a specific transaction are shown in FIGS. **14** to **23**B. The editor verifies the syntax of the integrated statements and correlates the data variables of the protocol statements with the data fields of the database. Following the generation of the integrated statements, the editor segregates the protocol statements from the database language statements. The protocol statements are stored in files which are identified as being for a particular transaction or data process and the database statements are stored in files which are identified as being for a particular transaction or data process on an identified database table. The editor places a database file identifier in the protocol statements which contained embedded database statements. The database file identifiers are used by CGI **28** to select the file for the appropriate transaction so CGI **28** may correlate data variables in the protocol statements with data fields in the database files. The files containing statements to be interpreted by the client program are then downloaded to the appropriate terminals, and the database files containing database language statements are stored on the system executing the CGI **28**.

Alternatively, the editor of the present invention may parse integrated statements which are segregated into source code statements for first and second processors, such an editor further includes a compiler to generate executable code for each processor and, if the processors execute differing source code, a compiler for each source code language. The executable code may then be downloaded to the respective processors for execution.

More specifically, the editor preferably places the database statements for one of the transactions of the preferred embodiment in a file identified by the database name following SQL in FIG. **12**. The attributes and tags forming the HTML statements for one of the transactions of the preferred embodiment are placed in a file generally denoted as <html_file>.HTM. The name <html_file> is a name which identifies one of the transactions. Where SQL statements are in the fields of the integrated statements shown in FIGS. **14** to **23**B, the string "<html_file>.SQL" is substituted as the database name in the statements of the <html_file>.HTM file. When the CGI executable file is initiated and parses the returning forms, the returned data is placed in the corresponding "<html_file>.SQL" file which is passed to the application program as a command line argument. In this manner, an abbreviated form for the SQL commands may be communicated over the open network between the client and CGI and the CGI may be able to expand those abbreviated SQL commands into the appropriate SQL commands which the application program requires to manipulate the ODBC database.

To effectuate a transaction, for example, an operation at a terminal with non-standard I/O devices may activate a terminal file with a hot key or other action. In processing the activated file, the client program may acquire data which is stored in a local variable or accessible through a non-standard I/O device. This data may then be stored in a FORM and submitted to a server file at a processing system address. The server file activates CGI **28** which retrieves data from the FORM and incorporates it into database statements in the database file for the appropriate transaction and database. If the database statement is a query, the requested data is returned to the CGI in the database file and the CGI places it in the corresponding FORM variables so the server may return the data to the terminal. If the database statement provides data to a database to obtain an authorization, for example, the action performed by the database application in response to the data is placed in the corresponding FORM and returned to the terminal. In this way, data is exchanged between the terminal and the database application. This exchange is supported by CGI **28** even though the server/client communication is performed in an open system protocol, such as HTTP, and the database application is performed in another language, such as SQL. CGI **28** is able to convert and exchange the data between the client and database without the user having to specifically design and implement a conversion program.

The communication paths available for a device implementing the present invention are shown in FIG. **13**A. As shown there, an I/O device **420** is coupled through the WorldWide Web open network **426** to an Internet Web server **12**. This connection may be implemented with the preferred extended capability HTML described above. Although HTML files may be encrypted to enhance the security of the document as it is communicated across the Internet, the operator of the system may choose to utilize a more secure physical connection between the device **420** and the Web server **12**. To obtain this alternative connection, the PAYMENT command for the METHOD attribute is preferably used. One form of the PAYMENT command is for a merchant's terminal and the other is for a consumer's terminal. In either terminal, the client program which supports the extended capability HTML operates independently but co-resident in memory with a certified bank card authorization and capture application, which may be provided by a financial institution or a bank card processor.

For the form of the command shown in FIG. **13**B, the client program in the merchant terminal suspends its execution and passes the terminal identifier, stored locally, which identifies the merchant's account and the consumer account information read via a magnetic stripe reader or the like, to the bank card application. The bank card application communicates this information via a PSTN **424** or the like to a transaction processor **422**. The processor **422** authorizes or denies the transaction and, if authorized, a printer at the merchant terminal prints a purchase agreement which the consumer may execute to complete the transaction.

In response to a HTML file having a FORM with an ACTION attribute equal to an executable file name for a bank card application program or the like, a METHOD attribute with a field value of PAYMENT, and an INPUT tag with a TYPE attribute of LOCAL_NAME which identifies a deposit only account supplied by a merchant (as shown in FIG. **13**C), the client program is suspended and control is transferred to the bank processing application. The bank processing application then uses a modem or ISDN D channel using T3 POS protocol or the like to connect to a secure packet network **424** to connect in a virtual point-to-point manner with a payment processor through a PSTN network or the like. This physical connection provides an additional security element to the encrypted data for the transaction of account information, PIN numbers encrypted by PIN pads provided at the consumer site, and other sensitive information. The bank processor **422** may submit

EXHIBIT C
PAGE 37 OF 39

remittance data to the merchant, via the Web or otherwise. After receiving the remittance data, the merchant may ship the product to the consumer. Thus, in this manner, the I/O device may communicate with a plurality of Web servers to "shop" for a best price, delivery date, or other relevant information for selecting a preferred transaction, and then execute the PAYMENT method to utilize a more secure physical communication connection and data security devices to consummate the financial elements of the transaction with less risk and costs for the merchant, consumer, and bank processor.

The preferred integrated HTML/SQL statements which support a card initiated payment authorization and capture transaction are shown in FIG. **14**. A first file **500** includes statements which identify the URL database from which the non-standard I/O device seeks authorization for a transaction. The prompts to the operator to enter the account number and amount of the transaction are supported by the INPUT tags which read the second track of the magnetic stripe reader to accept a number of up to 40 characters and assign that information from that track to a variable, and to input the up to 8 characters from the keyboard or the like into a variable called AMOUNT. The INPUT tag with the TYPE attribute of AUTOSUBMIT returns the form to the server for processing in accordance with the method defined in the returned form. As shown in FIG. **14**, that METHOD statement causes CGI **28** to incorporate returned data into SQL commands which query the database as to whether the subfield of the track 2 data representing the account number is present in the authorization table of the database. If the data is not present, then a new record is inserted into a table labeled "log_table". The new record consists of the account number and the amount returned in the FORM. Based upon the results of this processing, the application program supplies the data fields to the FORM which will be returned to the client program for printing the transaction record. That file **510** is shown in FIG. **14**. The ACTION attribute TO PRINTER and the POST METHOD causes the data in the next eight lines to be directed to the printer coupled to the non-standard I/O device for printing the transaction form. The customer may then execute the printed form to complete the transaction. If the transaction is declined or an error is otherwise encountered, the file **520** is used to return a denial to the client program.

In a similar manner, the preferred integrated statements for a bar code order input with card-initiated payment authorization is shown in FIG. **15**. The file **550**, supported by the present invention which implements the transaction request, is again directed to the proper database by the ACTION attribute. The necessary customer information such as name and address may be input through a standard keyboard. The HTML command in the present invention also permits the form to receive the bar code, unit price, and credit card information in a manner similar to that discussed above for the magnetic card reader. Once this information is returned to the server and CGI interface, it is processed by the application program in accordance with the METHOD identified in the returned form. The method of HTML file **550** also creates a database order_table having the information shown in the method. Again, if the transaction is approved, the data for the order and customer acceptance of the order is provided in HTML file **555**, which is directed by the ACTION attribute to the printer at the non-standard I/O device. If the account number is not in the authorization database, the authorization declined or error response is provided in correspondence with the statements in file **560**.

In a similar manner, FIGS. **16–22** show the integrated statements for a transaction request, authorization response, or authorization declined response files for key input order with secure payment transaction (FIG. **16**), a smart card-debit (Type 1) transaction (FIG. **17A**), a smart card debit (Type 2) transaction (FIG. **17B**), a debit card transaction (FIG. **18**), a check verification transaction (FIG. **19**), a customer frequency transaction (FIG. **20**), an item search transaction for which there is no denial (FIG. **21**), retail store end of day reporting (FIG. **22**) and a store reporting an e-mail transaction (FIG. **23**).

While the present invention has been illustrated by the description of a preferred and alternative embodiments and processes, and while the preferred and alternative embodiments and processes have been described in considerable detail, it is not the intention of the applicant to restrict or in any way limit the scope of the appended claims to such detail. Additional advantages and modifications will readily appear to those skilled in the art. For example, rather than expanding HTTP to support non-standard I/O devices, the FTP, POP, SMTP, TELNET or other protocols may be expanded in like manner to couple non-standard I/O devices to the Internet. Similarly, the preferred implementation of the present invention supports a variety of non-standard I/O devices and I/O operations. An Internet protocol may be constructed in accordance with the principles of the present invention to support only selected I/O devices or operations disclosed in the present application. The invention in its broadest aspects is therefore not limited to the specific details, preferred embodiment, and illustrative examples shown and described. Accordingly, departures may be made from such details without departing from the spirit or scope of applicant's general inventive concept.

What is claimed is:

1. A method for communicating between a client program controlling in a non-standard input/output (I/O) device and a server over an open network comprising:

activating a non-standard I/O device to assign data obtained by a non-standard I/O device to a variable name in a file comprised of extended open network protocol statements; and

sending a file having said assigned data to a server to perform a data operation in accordance with said extended open network protocol statements.

2. The method of claim 1 further comprising:

identifying in one of said extended open network protocol statements an I/O operation for said non-standard I/O device to obtain said data for said assignment.

3. The method of claim 1 further comprising:

identifying a processing method for said data operation with said assigned data.

4. The method of claim 1 further comprising:

receiving a file comprised of extended open network protocol statements generated by said server from said data operation; and

performing a data operation with said non-standard I/O device in accordance with at least one of said generated extended open network protocol statements in said received file.

5. The method of claim **4** wherein said data operation is performed by executing an executable file identified by one of said generated extended open network protocol statements in said received file.

6. A method for communicating between a client program controlling a non-standard input/output (I/O) device and a server over an open network comprising:

generating a file comprising extended open network protocol statements, at least one of which identifies a data operation for obtaining data from a non-standard I/O device; and

sending said file to a client program controlling said non-standard I/O device.

7. The method of claim 6 further comprising:

receiving data obtained by said non-standard I/O device performing said identified data operation.

8. The method of claim 7 further comprising:

performing a data operation with said data received from said non-standard I/O device;

generating extended open network protocol statements from said data operation; and

sending a file comprising said generated extended open network protocol statements to a client program controlling said non-standard I/O device.

9. The method of claim 8 wherein said data operation identification is achieved by identifying an executable file name in one of said extended open network protocol statements.

10. A system for communicating between a client program controlling a non-standard input/output (I/O) device and a server over an open network comprising:

means for activating a non-standard I/O device to assign data obtained by a non-standard I/O device to a variable name in a file comprising extended open network protocol statements; and

means for sending a file having said assigned data to a server to perform a data operation.

11. The system of claim 10 further comprising:

means for identifying in one of said extended open network protocol statements an I/O operation for said non-standard I/O device to obtain said data for said assignment.

12. The system of claim 11 further comprising:

means for identifying a processing method for said data operation with said assigned data.

13. The system of claim 11 further comprising:

means for receiving a file comprising extended open network protocol statements generated by said server from said data operation; and

means for performing a data operation at said non-standard I/O device in accordance with at least one of said generated extended open network protocol statements in said received file.

14. The system of claim 13 further comprising:

an executable file for performing said data operation, said executable file being identified by one of said generated extended open network protocol statements in said received file.

15. A system for communicating between a client program controlling a non-standard input/output (I/O) device and a server over an open network comprising:

means for generating a file comprising extended open network protocol statements, at least one of which identifies a data operation for obtaining data from said non-standard I/O device; and

means for sending said file to a client program controlling said non-standard I/O device.

16. The system of claim 15 further comprising:

means for receiving data obtained by said non-standard I/O device performing said data operation.

17. The system of claim 16 further comprising:

means for performing a data operation with said data received from non-standard I/O device;

means for generating extended open network protocol statements from said data operation; and

means for sending a file comprising said generated extended open network protocol statements to a client program controlling said non-standard I/O device.

18. The system of claim 17 wherein said means for performing a data operation is an executable file.

* * * * *

EXHIBIT C
PAGE 39 OF 39

EXHIBIT / ATTACHMENT

D

(To be scanned in place of tab)

US006684269B2

(12) **United States Patent**
Wagner

(10) **Patent No.: US 6,684,269 B2**
(45) **Date of Patent: *Jan. 27, 2004**

(54) **SYSTEM AND METHOD FOR ENABLING TRANSACTIONS BETWEEN A WEB SERVER AND A SMART CARD, TELEPHONE, OR PERSONAL DIGITAL ASSISTANT OVER THE INTERNET**

(75) Inventor: **Richard Hiers Wagner**, Dunwoody, GA (US)

(73) Assignee: **Datascape Inc.**, Atlanta, GA (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.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **10/213,959**

(22) Filed: **Aug. 7, 2002**

(65) **Prior Publication Data**

US 2002/0198837 A1 Dec. 26, 2002

Related U.S. Application Data

(63) Continuation of application No. 10/100,347, filed on Mar. 18, 2002, now abandoned, which is a continuation of application No. 09/907,076, filed on Jul. 17, 2001, now abandoned, which is a continuation of application No. 09/314,266, filed on May 18, 1999, now Pat. No. 6,366,967, which is a continuation of application No. 08/995,123, filed on Dec. 19, 1997, now Pat. No. 5,905,908, which is a continuation of application No. 08/493,772, filed on Jun. 22, 1995, now Pat. No. 5,742,845.

(51) **Int. Cl.**[7] **G06F 13/00**; G06F 13/14

(52) **U.S. Cl.** **710/33**; 710/11; 710/20; 709/203; 709/227; 709/228; 370/401

(58) **Field of Search** 710/11, 20, 33; 709/203, 227, 228; 370/401

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,668,653	A	6/1972	Fair et al.
4,410,962	A	10/1983	Daniels et al.
4,438,511	A	3/1984	Baran

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

WO	WO 95/16971	6/1995
WO	WO 96/41286	12/1996
WO	WO 97/01137	1/1997

OTHER PUBLICATIONS

"Viola in a Nutshell: The Viola World Wide Web Toolkit", http://www.xcf.berkeley.edu/~wei/viola/book/, Apr. 1995 (176 pages).

(List continued on next page.)

Primary Examiner—Rehana Perveen
(74) *Attorney, Agent, or Firm*—Maginot, Moore & Beck LLP; David M. Lockman

(57) **ABSTRACT**

An open network system for supporting input/output (I/O) operations for non-standard I/O devices are disclosed. The system includes a server coupled to a plurality of I/O devices through an open network and an extended open system protocol that supports communication with devices that are not personal computers (PCs). These devices include magnetic stripe readers, check readers, smart card readers, credit card terminals, screen phone terminals, PIN pads, printers, and the like. The extended open network protocol includes tags which identify device and input operations and attributes which identify the location, data exchange method, and data variable names for the retrieval, acquisition, and submission of data between the server and I/O devices. Preferably, the open network protocol is implemented in a Hyper Text Transport Protocol (HTTP).

54 Claims, 25 Drawing Sheets

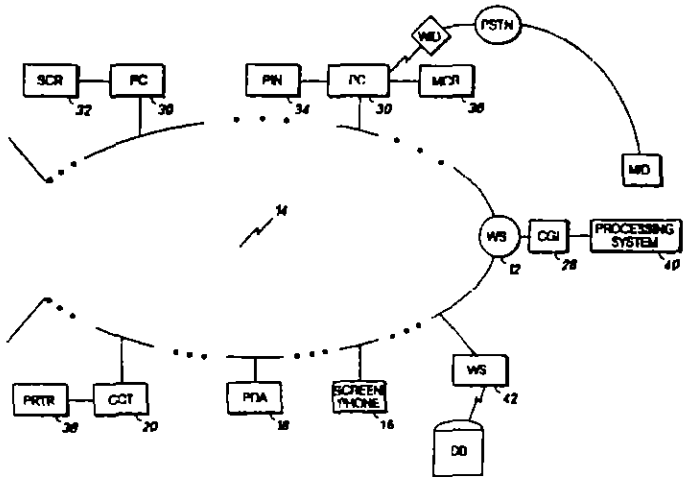

EXHIBIT D
PAGE 1 OF 47

U.S. PATENT DOCUMENTS

4,577,152	A	3/1986	Macovski
4,620,153	A	10/1986	Hino
4,625,171	A	11/1986	Sekihara et al.
4,625,276	A	11/1986	Benton et al.
4,707,592	A	11/1987	Ware
4,724,521	A	2/1988	Carron et al.
RE32,701	E	6/1988	Moran
4,755,940	A	7/1988	Brachtl et al.
4,774,655	A	9/1988	Kollin et al.
4,787,035	A	11/1988	Bourne
4,799,156	A	1/1989	Shavit et al.
4,815,029	A	3/1989	Barker et al.
4,847,604	A	7/1989	Doyle
4,885,777	A	12/1989	Takaragi et al.
4,901,223	A	2/1990	Rhyne
4,912,309	A	3/1990	Danielson et al.
4,926,325	A	5/1990	Benton et al.
4,942,532	A	7/1990	Mori
4,942,552	A	7/1990	Merrill et al.
4,947,028	A	8/1990	Gorog
4,949,248	A	8/1990	Caro
4,962,531	A	10/1990	Sipman et al.
4,965,742	A	10/1990	Skeirik
4,972,463	A	11/1990	Danielson et al.
5,014,221	A	5/1991	Mogul
5,018,196	A	5/1991	Takaragi et al.
5,021,949	A	6/1991	Morten et al.
5,025,373	A	6/1991	Keyser, Jr. et al.
5,036,461	A	7/1991	Elliott et al.
5,036,484	A	7/1991	McCoy et al.
5,046,094	A	9/1991	Kawamura et al.
5,047,923	A	9/1991	Elstner et al.
5,083,262	A	1/1992	Haff, Jr.
5,103,478	A	4/1992	Matyas et al.
5,113,517	A	5/1992	Beard et al.
5,119,465	A	6/1992	Jack et al.
5,146,499	A	9/1992	Geffrotin
5,146,553	A	9/1992	Noguchi et al.
5,148,481	A	9/1992	Abraham et al.
5,149,945	A	9/1992	Johnson et al.
5,161,222	A	11/1992	Montejo et al.
5,179,690	A	1/1993	Ishikawa
5,195,130	A	3/1993	Weiss et al.
5,202,828	A	4/1993	Vertelney et al.
5,204,947	A	4/1993	Bernstein et al.
5,206,951	A	4/1993	Khoyi et al.
5,212,369	A	5/1993	Karlisch et al.
5,220,380	A	6/1993	Hirata et al.
5,220,501	A	6/1993	Lawlor et al.
5,239,662	A	8/1993	Danielson et al.
5,243,284	A	9/1993	Noll
5,256,863	A	10/1993	Ferguson et al.
5,272,754	A	12/1993	Boerbert
5,274,821	A	12/1993	Rouquie
5,276,312	A	1/1994	McCarthy
5,277,192	A	1/1994	Cumoulin
5,278,955	A	1/1994	Forte et al.
5,278,972	A	1/1994	Baker et al.
5,294,782	A	3/1994	Kumar
5,297,249	A	3/1994	Bernstein et al.
5,307,411	A	4/1994	Anvret et al.
5,307,499	A	4/1994	Yin
5,309,351	A	5/1994	McCain et al.
5,321,806	A	6/1994	Meinerth et al.
5,321,808	A	6/1994	Rupp
5,347,580	A	9/1994	Molva et al.
5,347,632	A	9/1994	Filepp et al.
5,349,678	A	9/1994	Morris et al.
5,351,296	A	9/1994	Sullivan
5,367,572	A	11/1994	Weiss
5,367,635	A	11/1994	Bauer et al.
5,373,561	A	12/1994	Haber et al.
5,377,191	A	12/1994	Farrell et al.
5,379,344	A	1/1995	Larsson et al.
5,386,517	A	1/1995	Sheth et al.
5,390,314	A	2/1995	Swanson
5,418,908	A	5/1995	Keller et al.
5,461,217	A	10/1995	Claus
5,475,585	A	12/1995	Bush
5,475,858	A	12/1995	Gupta et al.
5,497,487	A	3/1996	Fortier
5,499,343	A	3/1996	Pettus
5,510,710	A	4/1996	Nauerth
5,511,122	A	4/1996	Atkinson
5,517,569	A	5/1996	Clark
5,521,980	A	5/1996	Brands
5,526,037	A	6/1996	Cortjens et al.
5,526,409	A	6/1996	Conrow et al.
5,530,844	A	6/1996	Phillips et al.
5,544,161	A	8/1996	Bigham et al.
5,544,246	A	8/1996	Mandelbaum et al.
5,544,320	A	8/1996	Konrad
5,546,582	A	8/1996	Brockmeyer et al.
5,548,721	A	8/1996	Denslow
5,550,561	A	8/1996	Ziarno
5,550,984	A	8/1996	Gelb
5,552,586	A	9/1996	Kalman
5,557,203	A	9/1996	Nauerth
5,563,878	A	10/1996	Blakeley et al.
5,570,465	A	10/1996	Tsakanikas
5,577,233	A	11/1996	Goettelmann et al.
5,581,478	A	12/1996	Cruse et al.
5,581,686	A	12/1996	Koppolu et al.
5,581,708	A	12/1996	Iijima
5,583,940	A	12/1996	Vidrascu et al.
5,586,036	A	12/1996	Pinstov
5,586,175	A	12/1996	Hogan et al.
5,590,038	A	12/1996	Pitroda
5,590,181	A	12/1996	Hogan et al.
5,590,196	A	12/1996	Moreau
5,590,197	A	12/1996	Chen et al.
5,590,199	A	12/1996	Krajewski, Jr. et al.
5,594,910	A	1/1997	Filepp et al.
5,602,905	A	2/1997	Mettke
5,602,915	A	2/1997	Campana et al.
5,602,918	A	2/1997	Chen et al.
5,604,802	A	2/1997	Holloway
5,606,493	A	2/1997	Duscher et al.
5,613,012	A	3/1997	Hoffman et al.
5,613,096	A	3/1997	Danknick
5,615,251	A	3/1997	Hogan et al.
5,623,547	A	4/1997	Jones et al.
5,630,101	A	5/1997	Sieffert
5,633,919	A	5/1997	Hogan et al.
5,638,430	A	6/1997	Hogan et al.
5,640,193	A	6/1997	Wellner
5,642,485	A	6/1997	Deaton et al.
5,649,114	A	7/1997	Deaton et al.
5,649,118	A	7/1997	Carlisle et al.
5,652,876	A	7/1997	Ashe et al.
5,655,008	A	8/1997	Futch et al.
5,664,110	A	9/1997	Green et al.
5,666,412	A	9/1997	Handelman et al.
5,668,878	A	9/1997	Brands
5,677,953	A	10/1997	Dolphin
5,677,955	A	10/1997	Doggett et al.
5,678,002	A	10/1997	Fawcett et al.
5,679,945	A	10/1997	Renner et al.
5,687,322	A	11/1997	Deaton et al.
5,689,799	A	11/1997	Dougherty et al.
5,694,471	A	12/1997	Chen et al.

5,694,546	A	12/1997	Reisman
5,696,903	A	12/1997	Mahany
5,703,795	A	12/1997	Mankovitz
5,706,273	A	1/1998	Guerreri
5,708,780	A	1/1998	Levergood et al.
5,708,782	A	1/1998	Larson et al.
5,708,825	A	1/1998	Sotomayor
5,710,884	A	1/1998	Dedrick
5,715,314	A	2/1998	Payne et al.
5,722,754	A	3/1998	Langner
5,724,506	A	3/1998	Cleron et al.
5,729,549	A	3/1998	Kostreski et al.
5,732,219	A	3/1998	Blumer et al.
5,734,720	A	3/1998	Salganicoff
5,742,845	A	4/1998	Wagner
5,742,931	A	4/1998	Spiegelhoff et al.
5,748,908	A	5/1998	Yu
5,754,655	A	5/1998	Hughes et al.
5,756,978	A	5/1998	Soltesz et al.
5,758,257	A	5/1998	Herz et al.
5,778,067	A	7/1998	Jones et al.
5,781,189	A	7/1998	Holleran et al.
5,799,156	A	8/1998	Hogan et al.
5,805,719	A	9/1998	Pare, Jr. et al.
5,805,807	A	9/1998	Hanson et al.
5,809,415	A	9/1998	Rossmann
5,815,577	A	9/1998	Clark
5,815,657	A	9/1998	Williams et al.
5,819,092	A	10/1998	Ferguson et al.
5,826,245	A	10/1998	Sandberg-Diment
5,838,906	A	11/1998	Doyle et al.
5,850,442	A	12/1998	Muftic
5,851,149	A	12/1998	Xidos et al.
5,854,833	A	12/1998	Hogan et al.
5,867,821	A	2/1999	Ballantyne et al.
5,870,552	A	2/1999	Dozier et al.
5,870,724	A	2/1999	Lawlor et al.
5,873,099	A	2/1999	Hogan et al.
5,884,292	A	3/1999	Baker et al.
5,889,863	A	3/1999	Weber
5,931,917	A	8/1999	Nguyen et al.
5,933,816	A	8/1999	Zeanah et al.
RE36,310	E	9/1999	Bjerrum et al.
5,953,731	A	9/1999	Glaser
5,960,216	A	9/1999	Vishlitzky et al.
5,963,205	A	10/1999	Sotomayor
5,973,731	A	10/1999	Schwab
5,978,569	A	11/1999	Traeger
5,978,773	A	11/1999	Hudetz et al.
5,982,891	A	11/1999	Ginter et al.
5,983,004	A	11/1999	Shaw et al.
5,987,312	A	11/1999	Dekker
5,996,076	A	11/1999	Rowney et al.
6,085,224	A	7/2000	Wagner
6,112,191	A	8/2000	Burke
6,175,922	B1	1/2001	Wang
6,182,052	B1	1/2001	Fulton et al.
6,205,437	B1	3/2001	Gifford
6,208,904	B1	3/2001	Mullen, Jr.
6,212,575	B1	4/2001	Cleron et al.
6,226,744	B1	5/2001	Murphy et al.
6,226,752	B1	5/2001	Gupta et al.
6,253,027	B1	6/2001	Weber et al.
6,275,869	B1	8/2001	Sieffert et al.
6,279,112	B1	8/2001	O'Toole, Jr. et al.
6,282,656	B1	8/2001	Wang
6,283,375	B1	9/2001	Wilz, Sr. et al.
6,314,456	B1	11/2001	Van Andel et al.

OTHER PUBLICATIONS

O'Reilly & Associates, Inc., ViolaWWW, Version 3.3 Beta, Features List, http://www.xcf.berkeley.edu/~wei/viola/README.txt, Apr. 10, 1995 (2 pages).

Wei, Pei, "A Brief Overview of the VIOLA Engine, and its Applications",http://www.xcf.berkeley.edu/~wei/viola/violaIntro.html, Aug. 16, 1994, (13 pages).

Wei, Pei, "WWW Browsers: Extensibility Issues", O'Reilly & Associates, Sep. 20, 1994, (13 pages).

"Basic HTTP as defined in 1992", W3C, http://www.w3.org/Protocols/HTTP/HTTP2.html, 1992, (125 pages).

McCool, Rob, "CGI/1.0: last call", Dec. 4, 1993 (1 page).

"Olde Viola Montage", http://www.xcf.berkeley.edu/~wei/viola/vintage/montage.html, 1991, (5 pages).

Masinter, Larry, "Naming and resource location", Stanford Computer Forum WWW Workshop, Sep. 20, 1994, (1 page).

Cousins, Steve and Shoham Yoav, "Software Agents on the WWW", Stanford Computer Forum WWW Workshop, Sep. 20, 1994, (1 page).

Winograd, Terry, "Web Structure and Meta-Information", Stanford Computer Forum WWW Workshop, Sep. 20, 1994, (1 page).

Shafrir, Eviatar, "Interacting with Information", Stanford Computer Forum WWW Workshop, http://www-pcd.stanford.edu/workshop/abstracts/shafrir.html, Sep. 20, 1994 (2 pages).

Gruber, Tom, "Virtual Documents", Stanford Computer Forum WWW Workshop, Sep. 20, 1994, (21 pages).

Keller, Arthur M., "CIT's Work on CommerceNet", Center for Information Technology, Stanford University, http://cit.stanford.edu/cit/comercenet.html, Oct. 1994, (6 pages).

Shapiro,Ehud, "Enhancing the WWW with co-presence", Stanford Computer Forum WWW Workshop, Sep. 20, 1994, (9 pages).

"Readings for Stanford WWW Workshop", Stanford Computer Forum, http://www-pcd.stanford.edu/workshop/readings.html, Sep. 20, 1994, (3 pages).

"WorldWide Web-Summary", http://www.w3.org/Summary.html, 1992, (2 pages).

"The Original HTTP as defined in 1991", http://www.w3.org/Protocols/HTTP/AsImplemented.html, 1991, (2 pages).

"WorldWide Web Seminar", http://www.w3.org/Talks/General/About.html, 1993, (15 pages).

White, Sean, "Online Communities on the WWW", Stanford Computer Forum WWW Workshop, Sep. 20, 1994, (1 page).

"The Common Gateway Interface", http://hoohoo.ncsa.uiuc.edu/cgi/, 1994, (23 pages).

"W3C: Object MetaInformation", http://www.w3.org/Protocols/HTTP/Object Headers.html, 1992, (9 pages).

"Hypertext Markup Language (HTML)", http://www.w3.org/MarkUp/1995-archive/1, 1995, (66 pages).

McCool, Rob, "Re: CGP/1.0 specification", Nov. 19, 1993 (2 pages).

McCool, Rob, "Revised CG1/1.0 specification", Nov. 19, 1993, http://www.webhistory.org/www-talk.1993q4/0540.html, (1 page).

Citation List of References noting "The Common Gateway Interface", CiteSeer, 2002, (6 pages).

Robinson, David and Coar, Ken, "The Common Gateway, Interface (CGI) Version 1.1", Apr. 16, 2003, (30 pages).

EXHIBIT D
PAGE 3 OF 47

Varela, Carlos and Hayes, Caroline, "Providing Data on the Web: From Examples to Programs", Dept. Computer Science, University of Illinois at Urbana–Champaign, http://archive.ncsa.uiuc edu/SDG/IT94/Proceedings/DDay/varela/, Oct. 17, 1994, (26 pages).

Dynamic Information Systems Coporation, "Developing JDBC Servlet Applications", http://www.disc.com/jdbc-serv.html, May 8, 2002, (12 pages).

ProSolutions, Inc. Active Server Pages (ASP) Technology, http://www.prosolutionsinc.com/asp.htm, 2002, (2 pages).

Rasmussen, B.F., "WDB—A Web Interface to Sybase", http://www.stsci.edu/stsci/meetings/adassIV/rasmussen-b.html, 1995, (4 pages).

Hume. Tom, "WAP: Where we've come from, where we are, where we're going." Available online at http://www.tomhume.org/stories/2002/02/04/wapWhereWeveComeFromWhereWeAreW .(18 pages).

Goldman, Chris, "Browsing for Perfection," ARC Group in the Press, Feb. 1, 2001. Available online at http://www.aregroup.com/homepage.nsf/PRSARC/ar010201, (4 pages).

"Handheld Devices Markup Language (HDML) specification," Apr. 11, 1997, Available online at http://www.w3.org/TR/hdml20–3.html through http://www.w3.org/TR/hdml20–9.html. (33 pages).

Hinden, R., "Simple Internet Protocol Plus White Paper, "Network Working Group. Oct. 1994, Available online at ftp://flp.rfc–editor.org/in–notes/rfc1710.txt, (22 pages).

McCarthy, Vance, "Novell to link wireless laptops, PDAs to LANs (portable computers, personal digital assistants, local area networks),"InfoWorld, Sep. 13, 1993 v15 n37 p1, (2 pages).

Schwartz, Barry K. and Stephen B. Weinstein, "Dual–Media Messaging Using Screen Telephones on the Telephone Network," Bell Communications Research, IEEE 1993, pp. 1138–1188. (6 pages).

Verjinski, Richard D., "PHASE. A Portable Host Access System Environment," Unisys Defense Systems. IEEE 1989, pp. 806–809, (4 pages).

Klemets, Anders. Gerald Q. Maquire, Frank Reichert, and Mark T. Smith. "MINT—A Mobile Internet Router." IEEE 1993, pp. 70–74, (5 pages).

Perkins, Charles E. and Pravin Bhagwat, "A Mobile Networking System based on Internet Protocol," IEEE Personal Communications, First Quarter 1994, pp. 32–41, (10 pages).

Connolly, T., P. Amer, and P. Conrad, "An Extension to TCP: Partial Order Service," Network Working Group. Nov. 1994, Available online at ftp://ftp.rfc–editor.org/in–notes/rfc1693.txt, (14 pages).

Furniss, P., "Octet Sequences for Upper–Layer OSI to Support Basic Communications Applications," Network Working Group, Oct. 1994, Available online at ftp://ftp.rfc–editor.org/in–notes/rfc1698.txt. (28 pages).

Graves, C., T. Butts, and M. Angel, "TN3270 Extensions for LUname and Printer Selection," Network Working Group, Jul. 1994, Available online at ftp://ftp.rfc–editor.org/in–notes/rfc1646.txt, (13 pages).

Klensin, J., N. Freed, M. Rose, E. Stefferud, and D. Crocker. "SMTP Service Extensions." Network Working Group, Feb. 1993, Available online at ftp://ftp.rfc–editor.org/in–notes/rfc1425.txt, (10 pages).

Wang, Z., "EIP: The Extended Internet Protocol A Framework for Maintaining Backward Compatibility." Network Working Group, Nov. 1992, Available online at ftp://ftp.rfc–editor.org/in–notes/rfc1385.txt, (16 pages).

Gwinn, A., "Simple Network Paging Protocol—Version 2," Network Working Group, Jul. 1994. Available online at ftp://ftp.rfc–editor.org/in–notes/rfc1645.txt, (15 pages).

Braden, B., J. Postel, and Y. Rekhter, "Internet Architecture Extensions for Shared Media." Network Working Group, May 1994, Available online at ftp://ftp.rfc–editor.org/in–notes/rfc1620.txt, (18 pages).

Delgrossi, Luca, Christian Halstrick, Ralf Guido Herrtwich, and Heinrich Stüttgen. "HeiTP—A Transport Protocol for ST–II," IBM European Networking Center, IEEE 1992, pp. 1369–1373, (5 pages).

Tsang, P. W. M. and R. W. C. Wang. "Development of a Distributive Lighting Control System Using Local Operation Network," IEEE Transactions on Consumer Electronics, vol. 40, No. 4, Nov. 1994, (11 pages).

Adomeit, Reinhard and Bernhard Holtkamp, "A Reference Framework for Integrated POI/POS Systems." Fraunhofer Institute for Software Engineering and Systems Engineering. (4 pages).

Caneve, Maurizio, and Roberto Gagliardi, "A WAIS–Based Hypermedia Document System," (4 pages).

Schmidt, Claudia, Burkhard Stiller, and Martina Zitterbart, "Towards Flexible Service–integrated Communication Subsystems," Jul. 1, 1994, (4 pages).

Plagemann, Thomas, Janusz Waclawczyk, and Bernhard Plattner, "Management of Configurable Protocols for Multimedia Applications," (4 pages).

"Frame relay attracts chains: networks at Kash N Karry, Office Max (network communications protocol)," Chain Store Age Executive with Shopping Center Age, Dec. 1994 v70, (4 pages).

Cummings, Joanne, "What's Hot What's Not '95," Network World, Dec. 26, 1994/Jan. 2, 1995. (5 pages).

Aronin, Mindy, "XcelleNet launches fifth–generation RemoteWare 2.0 for remote and mobile applications," Business Wire, Nov. 28, 1994, (4 pages).

Baum, David, "Designing mobile applications (case studies of companies developing mobile programs)," InfoWorld, Nov. 28, 1994 v16 n48 p73, (7 pages).

Moeller, Michael, "Wireless standard to ease application portability, extensions will provide a common layer for CDPD, RAM, Ardis software," PC Week, Nov. 21, 1994, v11 n46 p1, (3 pages).

Petreley, Nicholas, Nancy Durlester, and Laura Wonnacott, "The interoperability headache: linking disparate clients and servers," InfoWorld, Nov. 15, 1993 v15 n46 p124, (12 pages).

Miscellaneous List of Article Abstracts, (50 pages).

Rescorla et al., "The Secure HyperText Transfer Protocol", Network Working Group, © Aug. 1999, (42 pages).

Berners–Lee et al., "HyterText Markup Language Specification—2.0", Nov. 29, 1994, Available online at http://marc.merlins.org/~merlin/htmlearn/html2.html., Aug. 21, 2002, (69 pages).

Raggett, Dave, "HyperText Markup Language Specification Version 3.0", Mar. 28, 1995, Available online at http://www.w3.or./MarkUp/html3/html3.txt., Aug. 21, 2002, (181 pages).

Merle et al., "Integration of heterogeneous environments: World Wide Web, Smart Card and Corba," Date Unknown, (16 pages).

EXHIBIT D
PAGE 4 OF 47

"ProSolutions, Inc. ASP vs. CGI 'Active Server pages (ASP) Technology'", E Business Solutions, © 1995–2002, Available online at http://www.prosolutionsinc.com/asp.htm., May 8, 2002, (2 pages).

Kirby et al., "Developing JDBC Servlet Applications", Dynamic Information Systems Corporation, Available online at http://www.disc.com/jdbcserv.html., May 8, 2002, (12 pages).

Pasian et al., "A Generalized Mosaic–to–SQL Interface with Extensions to Distributed Archives", Astronomical Data Analysis Software and Systems IV ASP Conference Series, vol. 77, 1995, Available online at http://www.stsci.edu/stsci/meetings/adassIV/pasianf.html., May 7, 2002, (5 pages).

Ng, Jason, "GSQL—a Mosaic–SQL gateway", Dec. 1993, Available online at http://archive.ncsa.uiuc.edu/SDG/People/jason/pub/gsql/starthere.html., May 7, 2002, (16 pages).

Rasmussen, B. F., "WDB—A Web Interface to Sybase", Astronomical Data Analysis Software and Systems IV ASP Conference Series, vol. 77, 1995, Available online at http://www.stsci.edu/stsci/meetings/addasIV/rasmussenb.html., May 7, 2002, (7 pages).

Connolly, Daniel W., "Paul Burchard on HTML 2.0 FORMs", Jul. 11, 1994, Available online at http://lists.w3.org/Archives/Public/www–html/1994Jul/0014.html., Dec. 3, 2001, (6 pages).

Hackborn, Dianne, "Re: Java Applet element proposal", Jun. 8, 1995, Available online at http://ksi.cpsc.ucalagary.ca/archives/HTML–WG/html–wg–95q2.messages/1228.html., Nov. 1, 2001, (2 pages).

Kaye, Walter Ian, "Re: Protocol registry? (fwd. Java Applet element proposal)", Jun. 5, 1995, Available online at http://list.w3.org/Archives/Public/www–html/1995Jun/0069.html., Aug. 14, 2001, (3 pages).

"The APP HTML Tag", Hot Java, Sun Microsystems, Inc., © 1995, Available online at http://www–math.uni–paderborn.de/dictionaire/Dictionaries/HTML–Dictionary/j . . . /index.html., Apr. 24, 2002, (2 pages).

Yourdon, Edward, "Java and the new Internet programming paradigm", © 2002, Available online at http://www.javaworld.com/javaworld/jw–08–1996/jw–08–yourdonbook_p.html., Apr. 29, 2002, (20 pages).

Merckling et al., "Smart Card Introduction", Mar. 1994, Available online at http://www.opengroup.org/rfc/mirror–rfc/rfc57.0.txt., Mar. 1, 2001, (16 pages).

Bellovin et al., "Limitations of the Kerberos Authentication System", USENIX, 1991, (16 pages).

"Guideline for the Use of Advanced Authentication Technology Alternatives", Federal Information Processing Standards Publication 190, Sep. 28, 1994, Available online at http://www.itl.nist.gov/fipspubs/fip190.htm., Mar. 21, 2002, (49pages).

"HTML + (Hypertext markup format)", Nov. 8, 1993, Available online at http://www.w3.org/MarkUp/HTMLPlus/htmlplus_2.html., Mar. 28, 2002, (89 pages).

Everhart, Craig, "Re: Interesting uses of networking", Jun. 11, 1990, Available online at http://www.cs.ucsd.edu/users/bsy/coke.history.txt., Sep. 17, 2002 (4 pages).

Kotanchik, J., Kerberos and Two–Factor Authentication, Security Dynamics, Mar. 1994, (11 pages).

McLaughlin, L. III, ed., "Line Printer Daemon Protocol", Network Printing Working Group, Aug. 1990, Available online at http://www.ietf.org/rfc/rfc1179.txt., Sep. 17, 2002, (14 pages).

Gutfreund, Yechezkal–Shimon, "Mosaic Accessories", Jan. 26, 1994, Available online at file://C\WINDOWS/TEMP/WWW–Talk%20Jan–Mar%201994%20Mosaic%20Accessories.htm., Apr. 1, 2002, (8 pages).

Gessler et al., "PDAs as mobile WWW browsers", Date Unknown, Telecooperation Office, University of Karlsruhe, Available online at file://C:\WINDOWS/TEMP/gwpring/PDAs%20as%20mobile%20WWW%20browsers.htm., Mar. 28, 2002, (9 pages).

Rose, M., "Principles of Operation for the TPC.INT Subdomain: Radio Paging—Technical Procedures", Dover Beach Consulting, Inc., Jan. 1994, (6 pages).

Workshop on Network and Distributed System Security, Privacy and Security Research Group, Feb. 11–12, 1993, (6 pages).

Malamud et al., "Principles of Operation for the TPC.INT Subdomain: Remote Printing—Technical Procedures", Dover Beach Consulting, Inc., Oct. 1993, (11 pages).

Louis, Tristan, "Java on the Menu; Causing Jitters?," Jun. 1995, Available online at http://www.tnl.net/who/bibliography/javajitters.asp., Apr. 29, 2002, (2 pages).

Allen, Terry, "Java Applet element proposal," Jun. 1, 1995, Available online at http://ksi.cpsc.ucalgary.ca/archives/HTML–WG/html–wg–95q2.messages/1119.html., Nov. 16, 2001, (4 pages).

Hansen, Wilfred J., "Enhancing Documents with Embedded Programs: How Ness Extends Insets in the Andrew Toolkit", IEEE 1990, pp. 23–32, (10 pages).

Tani, M., Yamaashi, K., Tanikoshi, K., Futakawa, M., and Tanufuji, S., "Object–Oriented Video: Interaction with Real–World Objects Through Live Video", ACM 1992, pp. 593–598, (6 pages).

Crowley, T., Milazzo, P., Baker, E., Forsdick, H., and Tomlinson, R., "MMConf: An Infrastructure for Building Shared Multimedia Applications", ACM 1990, pp. 329–342, (14 pages).

Davis, H., Hall, W., Heath, I., Hill, G., and Wilkins, R., "Towards an Integrated Information Environment with Open Hypermedia Systems", ACM 1992, pp. 181–191, (10 pages).

Ferrara, Fabrizio M., "The KIM Query System: An Iconic Interface for the Unified Access to Distributed Multimedia Databases", SIGGHI Bulletin, 1994, vol. 26, No. 3, pp. 30–39, (10 pages).

Gibbs, Simon, "Composite Multimedia and Active Objects", OOPSLA, 1991, pp. 97–112, (16 pages).

Davis, H., Hall, W., Pickering, A., and Wilkins, R., "Microcosm: An Open Hypermedia System" ACM 1993, (1 page).

Cullen, J., Day, B. and Planes, M., "The Use of FTAM to Access Graphical Pictures Across Wide Area Networks", Elsevier Science Publishers B.V. 1992, pp. 377–383, (7 pages).

Vaziri, Arsi, "Scientific Visualization in High–Speed Network Environments", Elsevier Science Publishers B.V. 1991, pp. 111–129, (19 pages).

Lashkari, Y., Rao, S., Mhaskar, V. and Shelat, A., "PLX: A Proposal to Implement a General Broadcasting Facility in a Distributed Environment Running X Windows", Comput. And Graphics, vol. 16, No. 2, 1992, pp. 143–149, (7 pages).

Kirste, Thomas, "SpacePicture– An Interactive Hypermedia Satellite Image Archival System", Comput. & Graphics, vol. 17, No. 3, 1993, pp. 251–260, (10 pages).

EXHIBIT D
PAGE 5 OF 47

Coulson, G., Blair, G., Davies, N. and Williams, N., "Extensions to ANSA for Multimedia Computing", Elsevier Science Publishers B.V. 1992, pp. 305–322, (18 pages).
Le Huynh, D., Jensen, M., Larsen, R., Southard, J. and Wang, F., "PIX: An Object–Oriented Network Graphics Environment", Comput. & Graphics, vol. 17, No. 3, 1993, pp. 295–304, (10 pages).
Berners–Lee, T.J., Cailliau, R., and Groff, J., "The World–Wide Web", Elsevier Science Publishers B.V. 1992, pp. 454–459, (6 pages).
Shackelford, D., Smith, J., and Smith, F., "The Architecture and Implementation of a Distributed Hypermedia Storage System", Nov. 1993, pp. 1–13, (13pages).
Labriola Don, "Whiteboard Software: Remote Possibilities", PC Magazine, Jun. 14, 1994, pp. 223–224, (2 pages).
"How We Evaluate the Whiteboard Software", PC Magazine, Jun. 14, 1994, pp. 225–228, (4 pages).
Udell, John, "Visual Basic Custom Controls Meet OLE", BYTE, Mar. 1994, pp. 197–198 and 200, (3 pages).
Sarna, D. and Febish, G., "OLE Gains Without (Much) Pain", DATAMATION, Jun. 15, 1994, pp. 31, (1 page).
Scrupski, Susan, "Outsourcing: It Could be Good", DATAMATION, Jun. 15, 1994, pp. 32–33 and 114, (3 pages).
Rizzo, John, "What's OpenDoc?", MacUser, Apr. 1994, pp. 119–123, (3pages).
Cello Browser Release 1.01a, ftp.law.cornell.edu/pub/LI1/cello , Mar. 16, 1994, (8 pages).
"Why Jim Clark Loves Mosaic", Wired, Oct. 1994, pp. 118–121, 150–151 and 157, (7 pages).
Marovac, N., and Osburn, L., "Hypernet: A Tool to Choreograph Worldwide Distributed Hypermedia Documents", Comput. & Graphics, vol. 16, No. 2, 1992, pp. 197–202, (6 pages).
Moeller, M., and Leach, N., "Microsoft Maps New OCX Plan; ActiveX seen as Web Content Platform", PC Week, Mar. 11, 1996, vol. 13, No. 10, pp. 1–2, (2 pages).
"OLE 2.0: Death to Monoliths?", BYTE, Mar. 1994, pp. 121–122, (2 pages).
"Netscape Communications Offers New Network Navigator Free on the Internet", http://home.netscape.com/newsref/pr/newsrelease1.html, Jan. 29, 1998.
Sackman, Gleason, "Remote Tele–Excavation via the Web", Tele–robotics, vol. 11, Issue 9, (1page).
Wolf, Gary, "The (Second Phase of) the Revolution has Begun", WIRED, Oct. 1994, pp. 16–17, (2 pages).
Wynne, B. and Marovac, N., "Lean Management, Group Support Systems, and Hypermedia: A Combination Whose Time Has Come", IEEE, 1993, pp. 112–121, (10 pages).
Moran, Patrick J. "Tele–Nicer–Slicer–Dicer: A New Tool for the Visualization of Large Volumetric Data", Biomedical Imaging Group, pp. 1–7, (7 pages).
Hansen, W., Consortium, A, "Andrew as a Multiple Environment for Visual Languages", IEEE, 1993, pp. 256–260, (5 pages).
Doyle, M., "The Virtual Embryo: VR Applications in Human Development Anatomy", Jan. 27, 1994, pp. 1–9, 9 pages).
Doyle, M., "A Knowledge Management Environment through the World Wide Web", Jan. 7, 1994.
Oliver, D. et al., "Netscape™ Unleashed",.
Ang, C., Martin, D., and Doyle, M., "Intergrated Control of Distributed Volume Visualization Through the World–Wide–Web", 1994, pp. 1–10 and 1, (11 pages).
"Europay Members Forum", Europay International, May 8, 2001, (3 pages).
"NCR 2170 Hospitality Workstation Designed for Bars, Restaurants", AT&T News Release, Feb. 2, 1993, (2 pages).
"AT&T to Increase Smart Card Reliability, Expand Memory to 8KB", AT&T News Release, Apr. 8, 1994, (2 pages).
"AT&T and NCR Offer Smart Card Upgrade Kits for Bank ATMs", AT&T News Release, Nov. 17, 1993, (2 pages).
"The Smart Card Cashes In", The Economist, Jan. 29, 1994, (2 pages).
"NCR 7450 Terminal Brings ATM–style Interface to Point–of–Sale", AT&T News Release, Oct. 11, 1993, (4 pages).
"Mastercard Smart Card Milestones", MasterCard International Press Office, Mar. 5, 2001, (3 pages).
Svigals, Jerome, "Microchip Smart Cards Will Save Money", American Banker, Feb. 2, 1994, (1 page).
Barthel, Matt, "Fears of Crime at ATMs are on the Rise", American Banker, Mar. 30, 1994, (1 page).
"Smart Card Publication List", Smart Card Forum, Mar. 2, 2001, (7 pages).
"Appendix: Major Electronic Purse Projects, By Country", pp. 35–50 and 52–64, Date Unknown (29 pages).
O'Kelly, Robin, "Mondex 'Global Electronic Cash' Wins Helsinki Award", Sep. 8, 1996, (2 pages).
"White Paper—The Euro in the Electronic Purse", Apr. 2000, (5 pages).
"History of Gemplus", Gemplus, Mar. 2, 2001, (3 pages).
Pauli, Ralf and Koponen, Risto, "Toward Electronic Money", Bank of Finland Bulletin, Apr. 1997, (3 pages).
"A Frequently Asked Questions List (FAQ) for alt.technology.smartcards", Estonian Institute of Cybernetics, Jul. 4, 1995, (12 pages).
Husemann, Dirk, "The Smart Card: Don't Leave Home Without It", Institute of Electrical and Electronics Engineers, Inc., Sep. 18, 2000, (9 pages).
"Report to the Council of The European Monetary Institute on Prepaid Cards", The Working Group on EU Payment Systems, May 1994, (12 pages).
Banerjee, Ram, "Smart Card Standards and Electronic Purse, A Review Paper", Aston University, May 1997, (20 pages).
Townend, Robin C., "Finance: History, Development & Market Overview", The International Smart Card Industry Guide, 1995–1996, (23 pages).
Monod, Elsbeth Marikje, "Case Study: The Multi–Functional Smart Carte Sante", The International Smart Card Industry Guide, 1995–1996, (8 pages).
Smith, Jack, "Smart Card Industry Review 1995", Smart Card Monthly Newsletter, Jan. 1996, (6 pages).
Crotch–Harvey, Trevor, "Smart Cards in Telecoms", The International Smart Card Industry Guide, 1995–1996, (14 pages).
Farmer, Donna and Tonnac, Remy de, "What's So Smart About Smart Cards", Gemplus, Oct. 10, 2000, (1 page).
"P401 Definition of a pan–European IC Card", Eurescom, Mar. 5, 2001, (1 page).
"Innovonics, Inc. Selects AMP as Manufacturer for the PC Pay Module", Innovonics News Release, May 19, 1995, (3 pages).
Levy, Steven, "E–Money (That's What I Want)", Wired, Dec., 1994, (2 pages).

EXHIBIT D
PAGE 6 OF 47

Fickenscher, Lisa, "Banc One Gears Up to Go National With Its Microchip Smart Cards", American Banker, Jan. 27, 1994 (2 pages).
Strachman, Daniel and Kutler, Jeffrey, "Convention Goers See Bank Future in Smart Cards", American Banker, Nov. 30, 1993, (1 page).
Meece, Mickey, "First Data Strengthens Position in Card Processing", American Banker, Dec. 21, 1993, (1 page).
Kutler, Jeffrey, "Prepaid Cards, Even in Concept Stage, Sparking Interest Among Consumers", American Banker, Apr. 27, 1994, (1 page).
Kutler, Jeffrey, "Visa Starts Push for Smart Card to Replace Cash", American Banker, Mar. 22, 1994, (2 pages).
"AT&T Introduces Enhanced Multimedia Interface for ATM Networks", AT&T News Release, May 4, 1994, (2 pages).
"Trendar and AT&T Test Contactless Smart Cards at Truck Stops", AT&T News Release, Feb. 26, 1993, (3 pages).
"NCR Announces 5980 Signature Capture System for Retailers", AT&T News Release, Sep. 30, 1991, (2 pages).
Merckling, Roger and Anderson, Anne, "Smart Card Introduction", Mar. 1994,(16 pages).
Merckling, Roger and Anderson, Anne, "DCE Smart Card Integration", Mar. 1994, (16 pages).
"Chapter 2—Technology", Smart Card Primer, The Schuler Consultancy, ©1993, pp. 13–24 and 26–29, (16 pages).
"CP8® PRODUCTS, PASCAM, Network Security Processor", Jan. 1994, (4 pages).
"Notes on Open Networks and Device Addressing", IBM Corporation, Apr. 25, 2001, (19 pages).
Chakrabarti, Samdih, Knight, James and Thies, Bill, "A Centralized Smart Card System Utilizing Distributed Proxies", 2000, University of Leeds, (16 pages).
"Single–Issuer Multiple Uses", Smart Card Primer, The Schuler Consultancy, ©1993, (3 pages).
Sazegar, Hossein, "The Impact of Smart Cards in the Global Payphone Business", Schlumberger Technologies, ©1992, (11 pages).
Derksen, Eduard, "Money–Bit: The Open Off–Host Prepaid Card System", Ascom Autelca, Ltd., Date Unknown, (10 pages).
Corzo, Gerardo, "Credisa's Elite Card System", Date Unknown, (5 pages).
Cordonnier, Vincent M., "Assessing the Future of Smart Cards", Cardtech, Date Unknown, (7 pages).
"NatWest Launch Mondex Global Electronics Cash", Smart Card News, Ltd., Dec. 1993, (7 pages).
Merkert, Sr., Robert J., "Stored Value Cards and The Electronic Purse: Developments in the Americas", Danyl Corporation, ©1995, (8 pages).
"CP8® Products, PinPadLINK, Readers/Encoders", Dactyl France, Jun. 1993, (2 pages).
"Smart Security", Byte Talk, Feb., 1994, (2 pages).
Verschuren, Ton, "Smart Access: Strong Authentication on the Web", Apr. 24, 2001, (10 pages).
"PKCS#11: New Member of Public Key Cyrptography", RSA Data Security, Inc., Jan. 12, 1994, (2 pages).
Petri, Steve, "An Introduction to Smartcards", Litronic, Inc., ©1998, (2 pages).
Violino, Bob, "The Cashless Society", InformationWeek, Oct. 11, 1993, (5 pages).
Merkert, Sr., Robert J., 'Keynote Presentation for the AIC Conference, Auckland, New Zealand', Danyl Corporation, ©1995, (4 pages).
"Videoway's Experience Shows a New Approach to the Interactive Opportunity", Inside Cable and Telecom Services, Jun. 1995, (3 pages).
Bull CP8® News, Products and Technologies, No. 22, Oct. 1994, (2 pages).
Bull CP8® Systems Product Sheet, Sep. 1994, (2 pages).
"One Store's Comprehensive Loyalty Program—The Takashimaya Experience", Verifone, Inc., ©1994, (4 pages).
"Mondex Payments Scheme", Smart Card News Ltd., Dec. 1993, (2 pages).
"Why Consumers Are Still Carrying Cash Everywhere", Philadelphia Inquirer, Jun. 11, 1995, (2 pages).
"Electronic Payment System Architecture for Point of Sale Application", IBM Technical Disclosure Bulletin, Jan. 1992, (1 page).
"Portable Self Checkout Retail System", IBM Technical Disclosure Bulletin, Jun. 1992, (3 pages).
"Transaction Completion Code Based on Digital Signatures", IBM Technical Disclosure Bulletin, Aug. 1985, (6 pages).
"Chapter 4—Uses", Smart Card Primer, The Schuler Consultancy, ©1993, pp. 57–66, 80–83, and 87–91, (19 pages).
Figure 8.5 from p. 151 of Zoreda & Oton "Smart Cards", Artech House, ©1994, (1 page).
"Internet Music Library Process", Date Unknown, (1 page).
Bibliography for "CITM Working Paper WP–95–1005: Designing Electronic Catalogs for Business Value: Results from the CommerceNet Pilot", Date Unknown, (3 pages).
Van Hove, Leo, Bibliography on Electronic Purses, Date Unknown, (4 pages).
Bibliography for "CITM Working Paper WP–95–1006: Financial EDI Over the Internet: A Case Study", Date Unknown, (5 pages).
Bibliography for Hawkes, P.L., Davies, Donald W., and Price, W.L., "Integrated Circuit Cards, Tags, and Tokens: New Technology and Applications", Oxford; Boston: BSP Professional, ©1990, (1 page).
Bibliography for Devargas, M., "Smart Cards and Memory Cards", Oxford; Blackwell NCC, ©1992, (1 page).
Bibliography for Svigals, Jerome, "Smart Cards: The New Bank Cards", New York: Macmillan; London: Collier Macmillan Publishers, ©1987, (1 page).
Bibliography for Bright, Roy, "Smart Cards: Principles, Practice, Applications", Chichester [England]: Ellis Horwood; New York: Halsted Press, ©1988, (1 page).
Bibliography for Svigals, Jerome, "Smart Cards: The Ultimate Personal Computer", New York: Macmillan Publishing Co.; London: Collier Macmillan, ©1985, (1 page).
Bibliography for "Applications of Computer Card Technology", Washington, D.C., U.S. Dept. of Labor, Office of the Inspector General, [1990], (1 page).
German article entitled "Erfahrungen Mit Einer Hochsicheren, Multifunktionalen Chipkarte fur Bankenanwendungen" Date Unknown, (21 pages).
Document entitled: "Dialog Search for U.S. patent No. 5,905,908", Date Unknown, (59 pages).
UBI Consortium Inc. Document, Mar. 26, 2001 (2 pages).
HTML file entitled: "The Common Gateway Interface", available at httpd@ncsa.uiuc.edu, circa, Jun. 1995.
HTML file entitled: "Critique of Secure–HTTP", available at shttp://www.spyglass.com/secure, Apr. 19, 1995.
Directory of abstracts of HTML files available from IEFT Secretariat—Corporation for National Research Initiatives; circa, Jun. 1995.

EXHIBIT D
PAGE 1 OF 47

HTML file entitled: "A Beginner's Guide to HTML" available from National Center for Supercomputing Applications/pubs@ncsa.uiuc.edu; circa, Jun. 1995.
Memo entitled: "Uniform Resource Locators (URL)", by URI working group of the Internet Engineering Task Force; Dec. 1994.
Press release entitled: "VeriFone to acquire Enterprise Integration Technologies (EIT), Internet Commerce Software and Services Leader", VeriFone, Inc. and Enterprise Integration Technologies; Redwood, California; Aug. 21, 1995.
Spyglass white paper entitled: "Electronic Commerce Standards for the WWW", Spyglass, Inc. http://spyglass.com; Aug. 21, 1995.
Memo entitled: "The Secure Hypertext Transfer Protocol"; E. Rescorla and A. Schiffman; Jul. 1995.
Article entitled: "Cash on the Wirehead", Andrew Singleton, BYTE Magazine, Jun. 1995.
"Beyond the Web: Excavating the Real World Via Mosaic", Goldberg et al., 2nd International WWW Conference.
"Magnetic Resonance Angiography", Radiology 1986; 161, p. 717–720, Dumoulin et al.
"Velocity and Diffusion Imaging in Dynamic NMR Microscopy", JMR 91, p. 326–352, 1991, Callaghan et al.
"Reduced Circular Field-of-View Imaging", MRM 40:474–480, 1998, Scheffler et al.
"Measurement of Flow Using Back Projection Reconstruction: Display of Limited Regions of the Body", Young et al.
Salsman, James P., "Form–based Device Input and Upload in HTML", available at http://www.w3.org/1999/07/NOTE–device–upload–19990706, Cisco Systems, ©1999, Jul. 6, 1999 (9 pages).
Memo entitled: "Form–based File Upload in HTML", E. Nebel and L. Masinter, Network Working Group of Xerox Corporation, Nov. 1995 (12 pages).
HTML file entitled: "HTML 3.0 Draft (Expired!) Materials", available at http://www.w3.org/MarkUp/html3, Dec. 21, 1995 (4 pages).
Berners–Lee, T. and Connolly, D., HTML file entitled: "HyperText Markup Language Specification—2.0", available available at http://www.w3.org/MarkUp/html–spec., Nov. 1995 (10 pages).
Raggett, David, HTML file entitled: "A Review of the HTML + Document Format", Hewlett Packard Laboratories, date unknown (13 pages).
Raggett, David, HTML file entitled: "10–Fill–out Forms and Input Fields", available at http://www.w3.org/MarkUp HTMLPlus/htmlplus_41.html, Nov. 8, 1993 (5 pages).
Raggett, David, HTML File entitled: "ISO 8879:1986", available at http://www.w3.org/MarkUp/htmlplus_paper/htmlplus.dtd, Apr. 5, 1994 (17 pages).
Riordan, T., "A Toll Collector on the Information Highway", *The New York Times*, Sunday, Aug. 22, 1993, (3 pages).
Burchard et al., "Compount Documents in HTML", Internet Draft Memo, Date Unknown, (14 pages).
Press Release, "Netscape Introduces Netscape Navigator 2.0", Netscape Company Press Relations, Sep. 18, 1995, (3 pages).
Stabell, B., The EMBED Tag by Netscape (v2.0), Sep. 19, 1995, (9 pages).
Article entitled, "Eolas Sues Microsoft for Infringement of Patent for Fundamental Web Browser Technology That Makes "Plug–Ins" and "Applets" Possible", Chicago, Illinois, Feb. 2, 1999, (2 pages).
Sottong, S., "Quick HTML Basics", Date Unknown, (12 pages).
Ang et al., "Integrated Control of Distributed Volume Visualization Through the World–Wide–Web", Oct. 30, 2001, (8 pages).
Krajewski, Jr., et al., "Applicability of Smart Cards to Network User Authentication", *Computing Systems*, vol. 7, No. 1, Winter 1994, (15 pages).
"Financial Institution Sign–On Authentication for Wholesale Financial Transactions", American National Standards Institute, X9.26, Feb. 28, 1990, (35 pages).
Kohl et al., "The Evoluation of *Kerberos* Authentication Service", Date Unknown, (15 pages).
Cordonnier et al., "Time as an Aid to Improving Security in Smart Cards", Date Unknown, (14 pages).
Cordonnier et al., "The Concept of Suspicion: A New Security Model for Identification Management in Smart Cards", Date Unknown, (14 pages).
Haykin et al., "Computer Science and Technology—Smart Card Technology: New Methods for Computer Access Control", U.S. Department of Commerce, National Technical Information Service, Sep. 1998, (55 pages).
Rescorla et al., "The Secure HyperText Transfer Protocol", Internet Draft Memo, Dec. 1994, (29 pages).
Takei, D., "Videotex Information System and Credit System Connecting with MARS–301 of JNR", *Japanese Railway Engineering*, No. 95, Sep. 1985, (5 pages).
Pongratz et al., "IC Cards in Videotex Systems", *Smart Card 2000: The Future of IC Cards*, International Conference, Oct. 19–20, 1987, (9 pages)>.
Neuman, B.C., "Security, Payment, and Privacy for Network Commerce", *IEEE Journal on Selected Areas in Communication*, vol. 13, No. 8, Oct. 1995, (8 pages).
Zilles, S.N., "Catalog–Based Order Entry System", IBM Technical Disclosure Bulletin, vol. 25, No. 11B, Apr. 1983, (3 pages).
Article entitled, "PKCS #11: New Member of Public Key Cryptography", available online at http://www.rsasecurity-.com/news/pr/940112–1.html., Jan. 12, 1994, (12 pages).
Kessler, J., "The French Minitel: Is There Life Outside of the 'US ASCII' Internet? A Challenge of Convergence", *D–Lib Magazine*, Dec. 1995, (10 pages).
Schöter et al., "Digital Money Online, A Review of Some Existing Technologies", Intertrader Ltd., ©Feb. 1997, (60 pages).
Sutherland, E., "The Quickening Pace of Technology", ©1995, (3 pages).
Article entitled "NetBill: An Internet Commerce System Optimized for Network Delivered Services", Date Unknown, (9 pages).
Reagle, Jr., et al., expanded bibliography for "Papers Describing the Operation of Electronic Commerce Protocols", Date Unknown, (3 pages).
Article entitled "ISO/IEC 7816 Part 4: Interindustry Command for Interchange", Nov. 26, 1998, (77 pages).
1991 Statement on Role of NIST, available online at http://csrc.nist.gov/publications/nistbul/cs191–02.txt, Jun. 17, 1992, (38 pages).
Kirkman, P., "Electronic Funds Transfer Systems: The Revolution in Cashless Banking and Payment Methods", ©1987, (14 pages).
Fancher, C., "Smart Cards: As Potential Applications Grow, Computers in the Wallet are Making Unobtrusive Inroads", *Scientific American*, Date Unknown, (10 pages).

EXHIBIT D
PAGE 8 OF 47

Walsh, E., "Smart Card Standards", Datapro Information Services, Aug. 25, 2000, (10 pages).
Article entitled "The Birth of Smart Cards: 1980", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1981", Le Courier de la Monétique, Date Unknown, (3 pages).
Article entitled "The Birth of Smart Cards: 1982", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1983", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1984", Le Courier de la Monétique, Date Unknown, (3 pages).
Article entitled "The Birth of Smart Cards: 1985", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1986", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1987", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1988", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1989", Le Courier de la Monétique, Date Unknown, (2 pages).
Doyle, M., University of California, San Diego, Office of Continuing Medical Education, Disclosure Statement, Nov. 30, 1993, (10 pages).
Article entitled "Workshop on Network and Distributed System Security", Privacy and Research Group, Feb. 11–12, 1993, (5 pages).
Krajewski, Jr., M., "Concept for a Smart Card Kerberos", Jan. 22, 2002, (8 pages).
Kohl et al., "The Kerberos Network Authentication Service (V5)", Sep. 1993, (103 pages).
Nebel, E., "File Upload in HTML Forms", Sep. 23, 1994, (4 pages).
Raggett, D., E–mail memo entitled "Re: Request for New Forms Submission Consensus", Oct. 12, 1993, (4 pages).
Gaskell et al., "Improved Security for Smart Card Use in DCE", Feb. 1995, (17 pages).
Pato, J., "Using Pre–Authentication to Avoid Password Guessing Attacks", Jun. 1993, (5 pages).
Shipplett, M., E–mail memo entitled "Re: Insecure WWW Access Authorization Protocol?", Mar. 7, 1994, (3 pages).
"Financial Transaction Card Originated Messages—Interchange Message Specifications", International Standard, Second Edition, Dec. 15, 1993, (128 pages).
Various articles on the subject "1994 NIST Guideline for Advanced Authentication (Smart Cards)", Date Unknow, (73 pages).
Allen et al., "Smart Cards: Seizing Strategic Business Opportunities", The Smart Card Forum, ©1997 The McGraw–Hill Companies, (323 pages).
Fielding et al., "Hypertext Transfer Protocol—HTTP/1.1", Jan. 1997, (162 pages).
Branwyn, G., "M•O•S•A•T•I•C Quick Tour for Windows: Accessing & Navigating the Internet's World Wide Web", ©1994 Ventana Press, Inc., (215 pages).
Shelly et al., "Netscape Navigator: An Introduction", ©1996 Boyd & Fraser Publishing Company, (138 pages).
James, P., "Official Netscape Navigator 2.0 Book: Windows Edition", ©1996, Ventana Press, Inc., (692 pages).
Article entitled "A Smart Card for On–Line Home Services", ChipCord News, Issue No. 32, Jul. 1990, (2 pages).
Bosak et al., "XML and the Second Generation Web", *Scientific American*, May 1999, (5 pages).
Beners–Lee et al., "RFC1866–Hypertext Markup Language–2.0", Nov. 1995, (58 pages).

EXHIBIT D
PAGE 9 OF 47

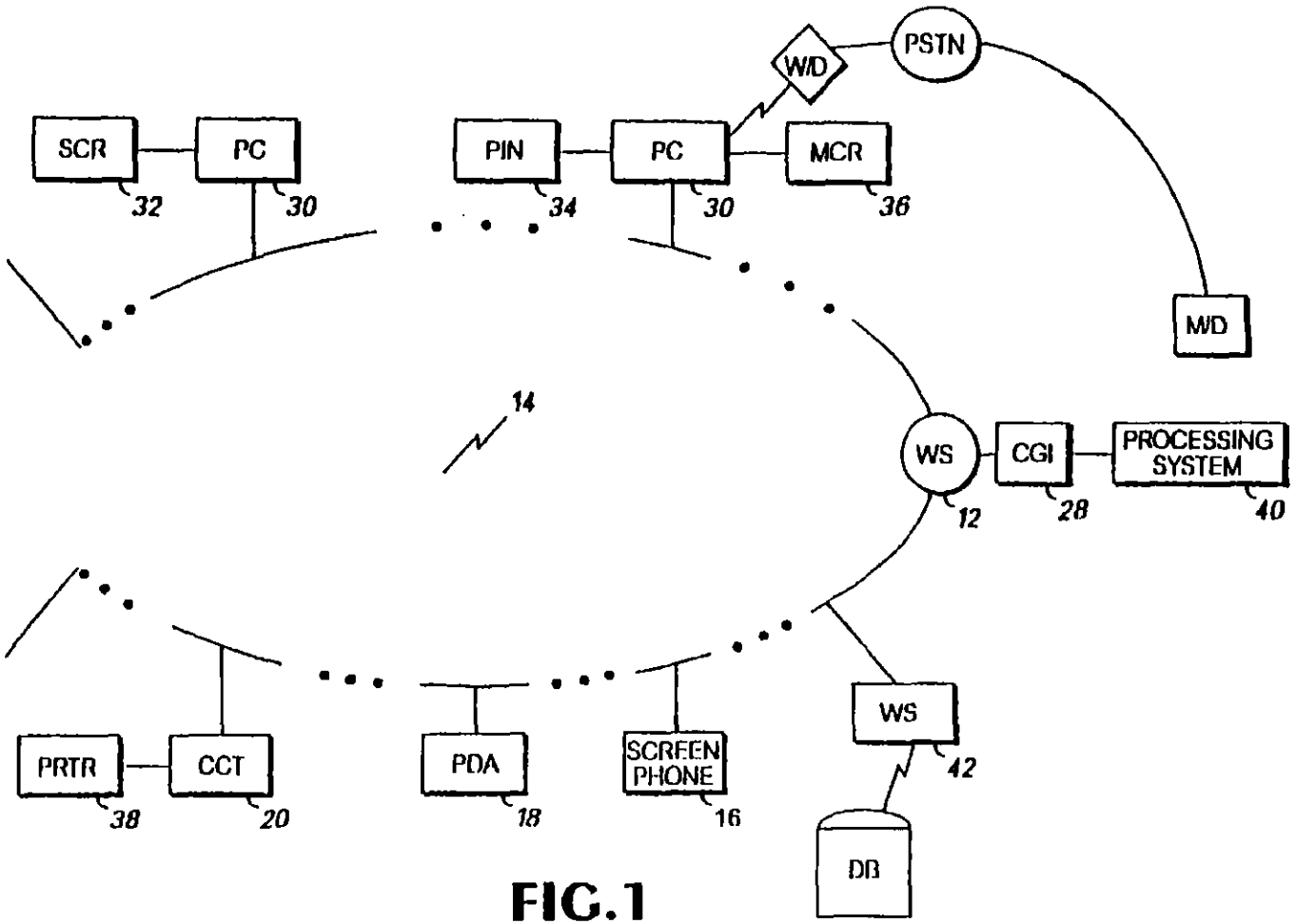

FIG.1

EXHIBIT D
PAGE 10 OF 31

HTML+D Attributes		Description
<FORM ACTION = "url"		To/From Web Server URL
	FROM "file name"	From Terminal Local File
	TO PRINTER	To Local Printer
	TO "file name"	To Terminal Local File
	FROM SCR	From Smart Card Reader
	TO SCR	To Smart Card Reader
METHOD =	"GET"	Retrieve Data
	"POST"	Store Data
	"PAYMENT"	Directive to deliver INPUT data to a private Payment Network for authorization and settlement.
	SQL <database name>	SQL statement database table

Attribute	Value	Description
<INPUT TYPE =	"text"	
	"password"	
	"checkbox"	
	"radio"	
	"submit"	
	"reset"	
NAME =	<field name>	
VALUE =	<initial value>	
CHECKED =		
SIZE =		
MAXLENGTH =		
>		

Attribute	HTML+DValue	Terminal Device
TYPE =	*"MSRT1"*	Mag Stripe Reader - Track 1
	"MSRT2"	Mag Stripe Reader - Track 2
	"KEY"	Terminal Command Keypad
	"PIN"	PIN Pad
	"BCW"	Bar Code Wand
	"MICR"	Check MICR Reader
	"AMT"	Dollar amount key input mask
	"INT"	Integer key input mask
	"LOCAL"	Input from Local Variable
	"AUTOSUBMIT"	Submit FORM to ACTION URL
NAME =	*ip_address*	Local Variable - Terminal's IP Address
	host_phone	Local Variable - Local Internet Access Phone Number
	tid	Local Variable - Terminal ID
	work_key	Local Variable - PIN encryption working key
	datetime	Local Variable - Date and time
	deposit_acct	Local Variable - Merchant Deposit Account

FIG.2

EXHIBIT D
PAGE 11 OF 47

SQL Statements

The following SQL commands represent a subset of the entire command set that varies by database vendor.

HTML+D Attributes	Description
*SELECT *, field_name,..*	Request field_name (one or many) from a database table
FROM=<table name>,..	Database table name
WHERE=<condition>	Conditional selection of data
name = "constant"	
name LIKE "constant"	
name IN "constant"	
AND	
OR	
ORDER=ASC	Request in ascending order
DESC	_descending
2	_by2's
GROUP=<name>	
INSERT TABLE=<table name>	Insert new data in database table
VALUES="constants"	
UPDATE FROM	
<table name>	Update field_name in database table
SET=field_name="constant"	
[WHERE=<condition>]	Update if WHERE clause is satisfied
DELETE FROM <table name>	Delete all columns that satisfy
[WHERE=<condition>]	WHERE clause
CREATE TABLE<table _name>	Create database table
PRIMARY KEY<name>	

FIG.3

EXHIBIT D
PAGE 12 OF 47

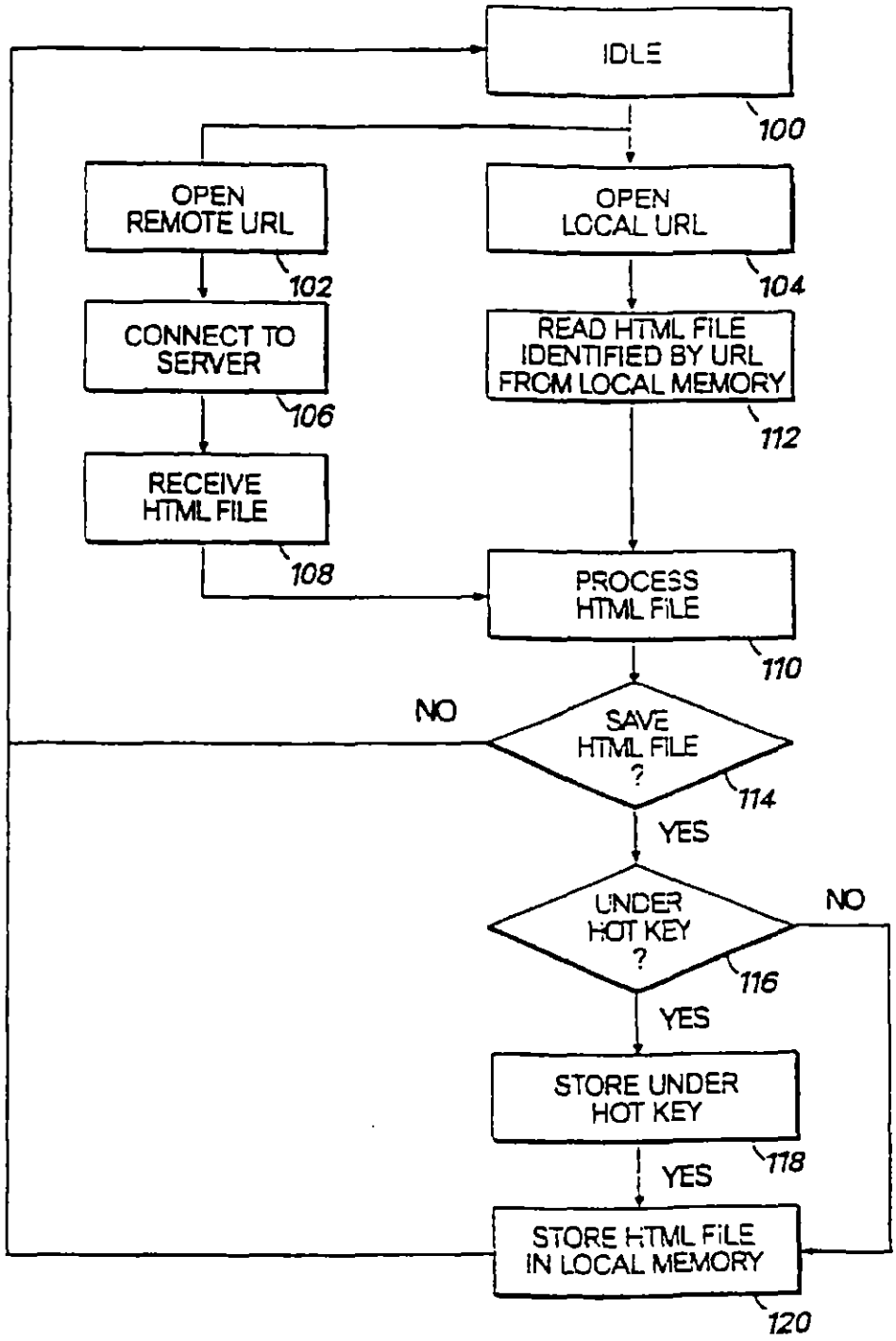

FIG.4

EXHIBIT D
PAGE 13 OF 47

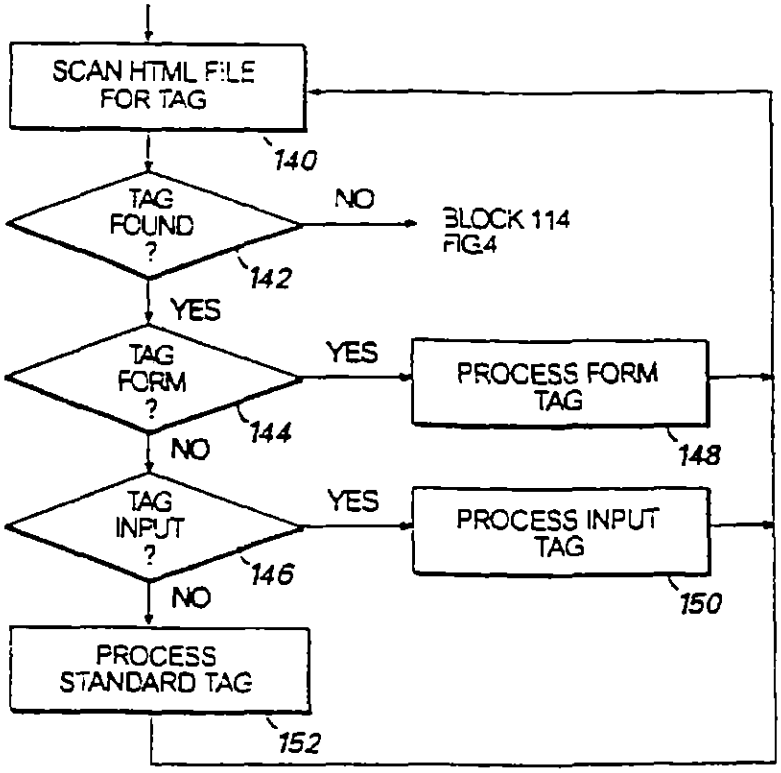

FIG.5

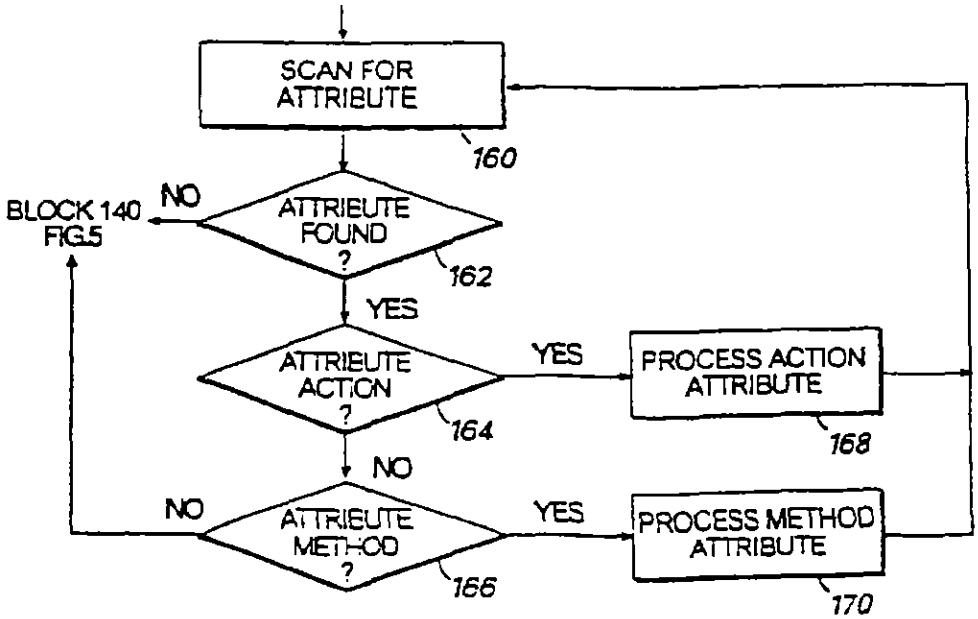

FIG.6

EXHIBIT D
PAGE 14 OF 47

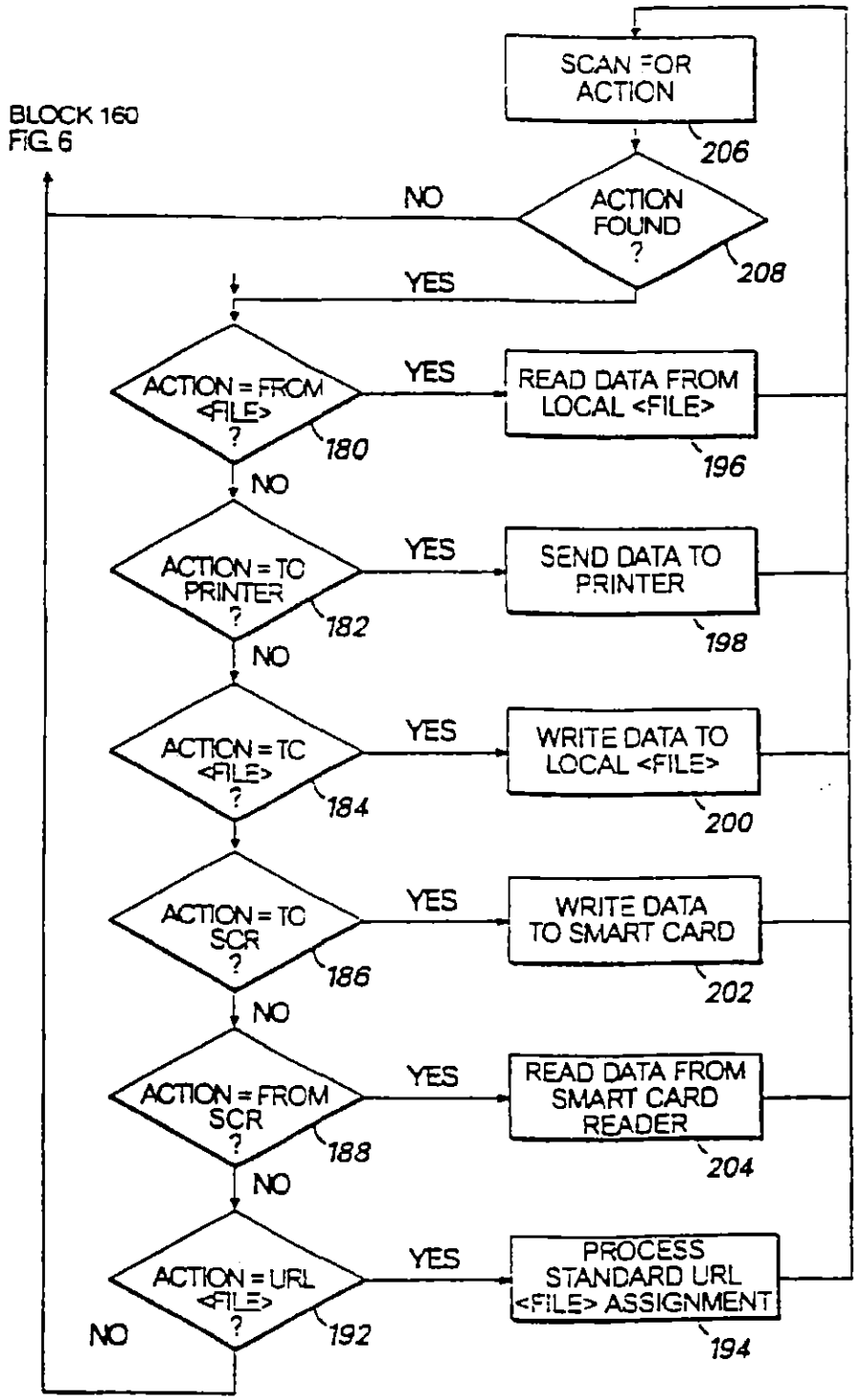

FIG. 7

EXHIBIT D
PAGE 15 OF 47

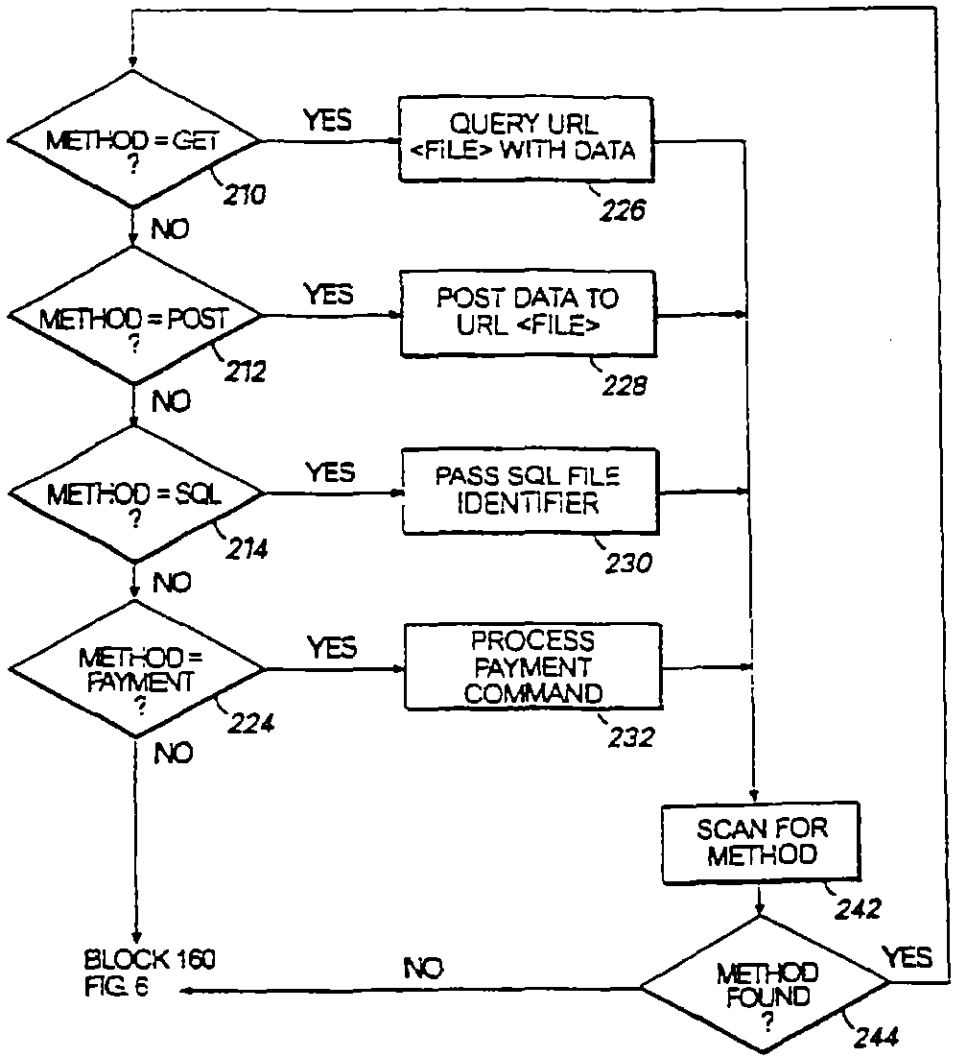

FIG. 8

EXHIBIT D
PAGE 16 OF 47

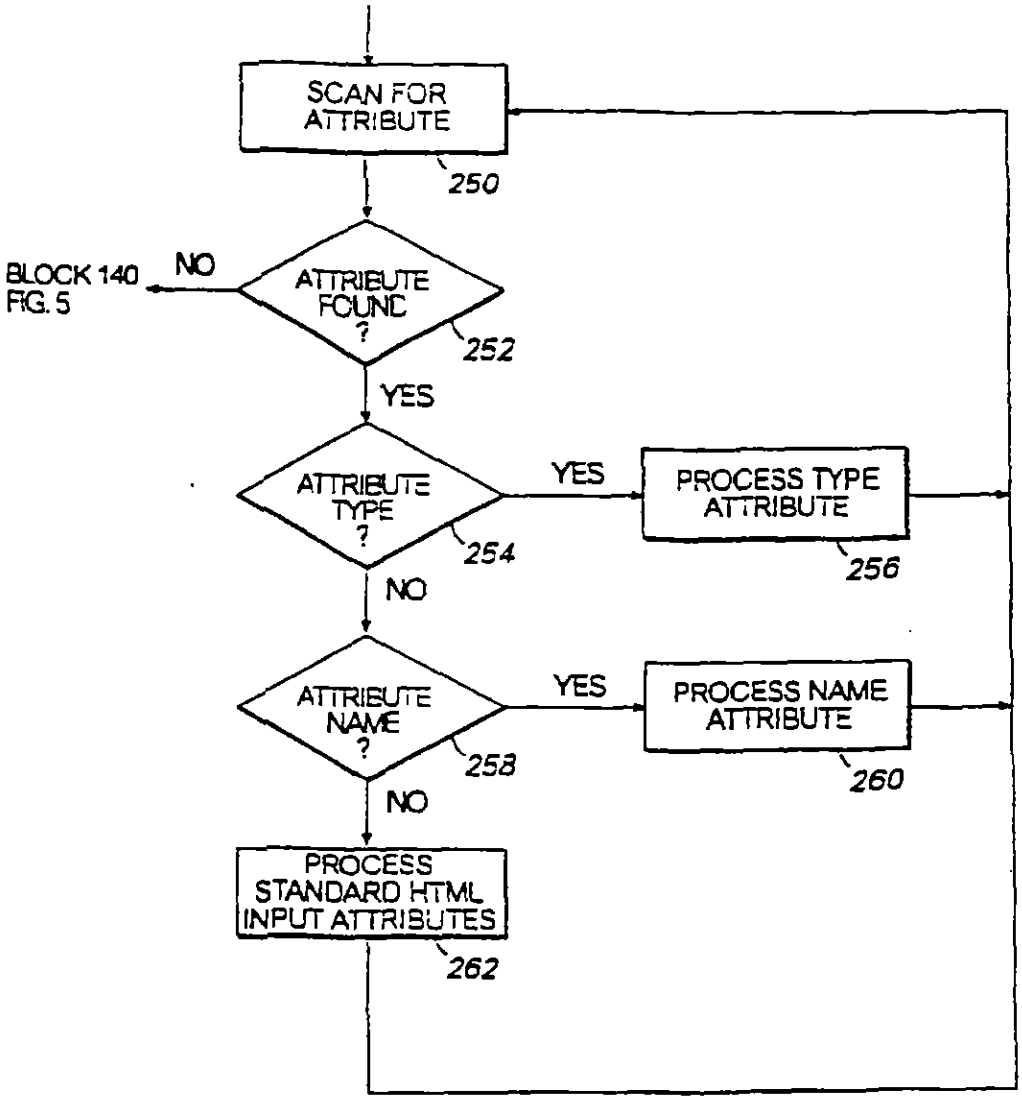

FIG. 9

EXHIBIT D
PAGE 17 OF

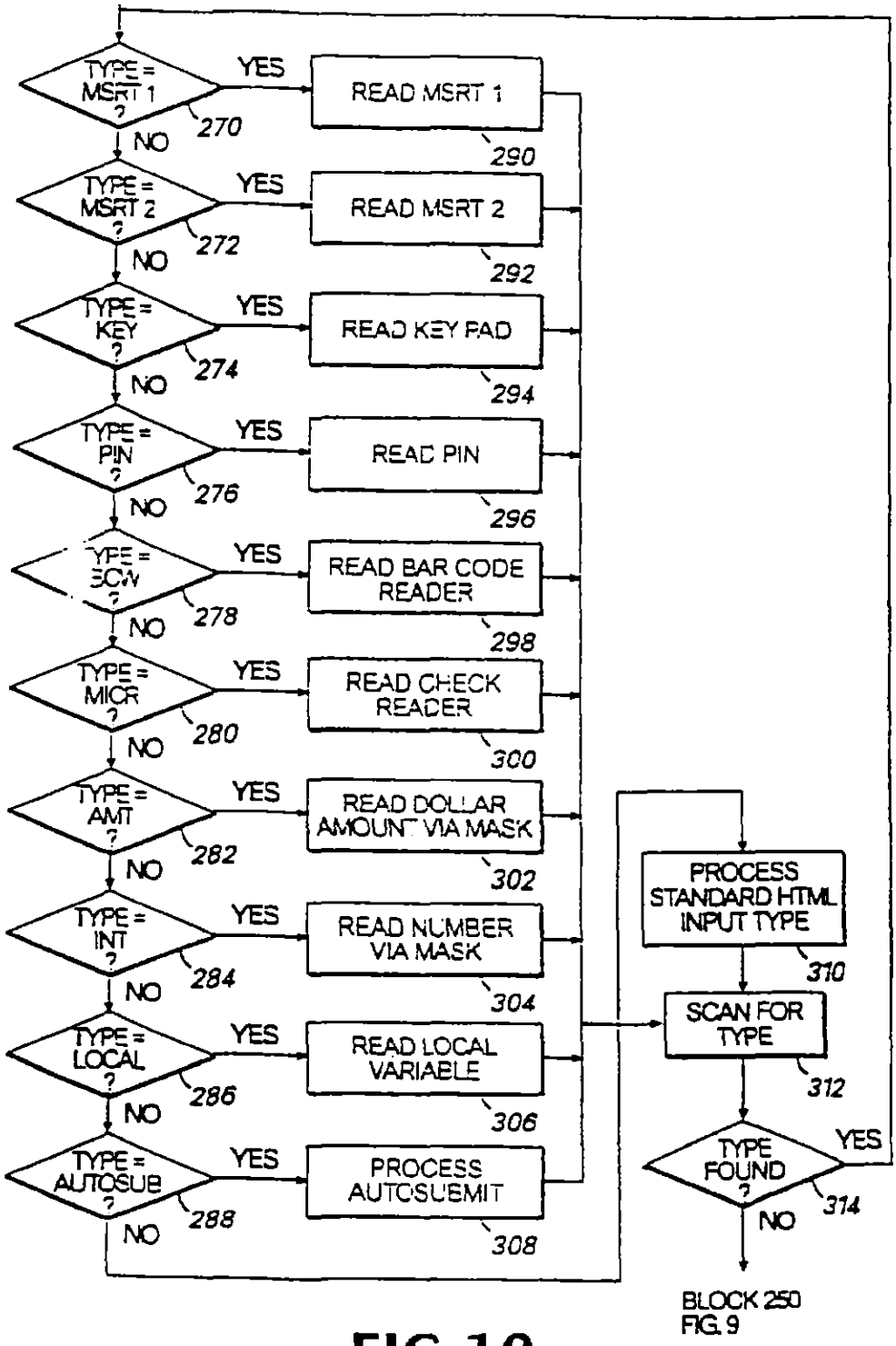

FIG.10

EXHIBIT D
PAGE 18 OF 47

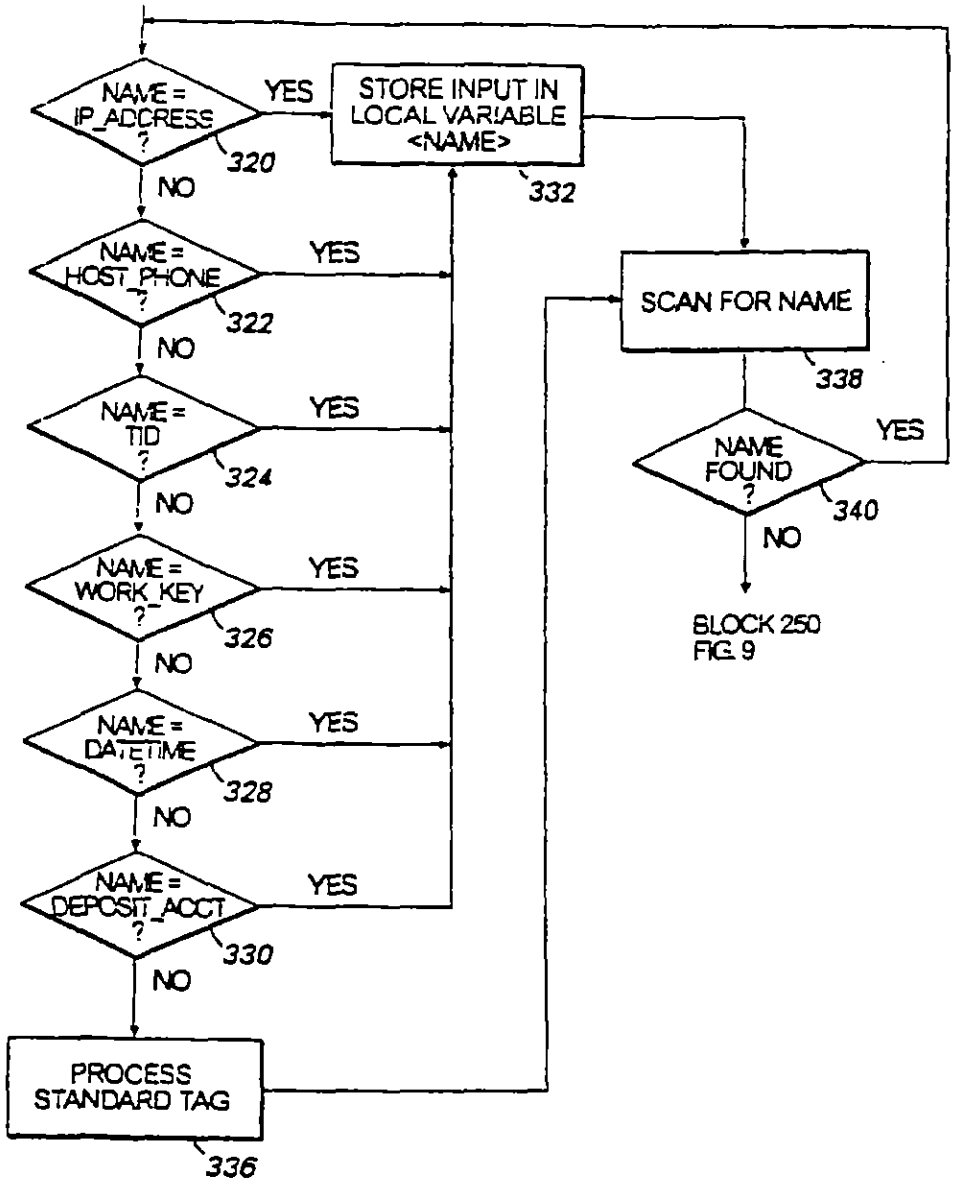

FIG.11

EXHIBIT D
PAGE 19 OF 47

1. <FORM ACTION=URL METHOD= GET>
2. <FORM ACTION=URL METHOD= POST>
3. <FORM ACTION=URL METHOD= SQL <database_name>

FIG.12

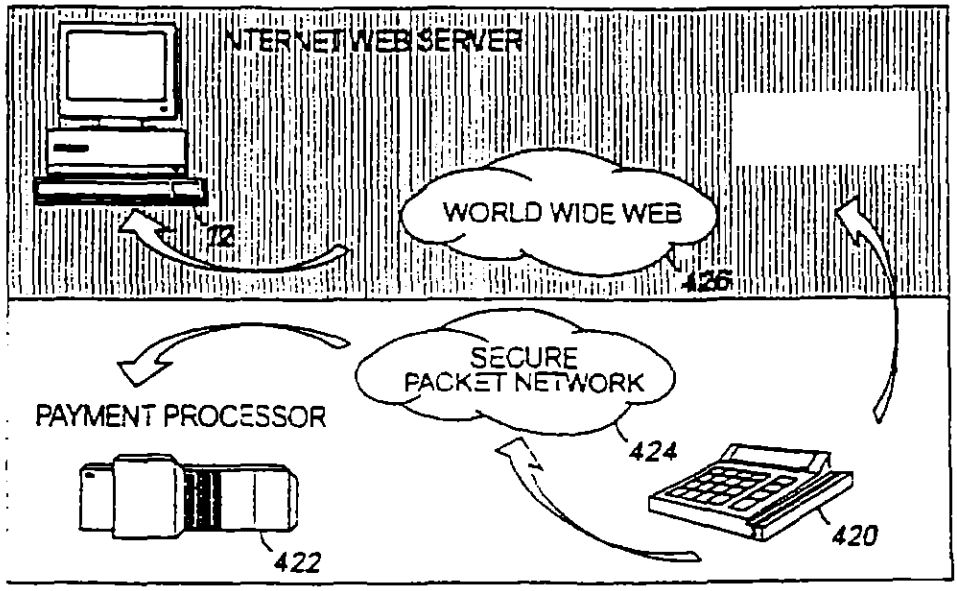

FIG.13A

<FORM ACTION=<filename> METHOD=PAYMENT>
<INPUT TYPE=AUTOSUBMIT>
</FORM>

FIG.13B

<FORM ACTION=dsinet METHOD=PAYMENT>
<INPUT TYPE=LOCAL NAME=DEPOSIT_ACCT VALUE=1234568902345678 90>
<INPUT TYPE=AUTOSUBMIT>
</FORM>

FIG.13C

EXHIBIT D
PAGE 20 OF 47

1.a. Transaction Request HTML+D

500

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
     METHOD=SQL
          "BEGIN TRAN
          IF NOT EXISTS ( SELECT substring(account, 1, 20) FROM auth_table)
               BEGIN
               INSERT TABLE=log_table VALUES=(getdate(),tid, substring (account, 1,20) ,
               substring( account, 22, 4), amount)
               SELECT * FROM log_table WHERE trandate = getdate()
               END
          ELSE SELECT * FROM error_table WHERE error_no=1
          COMMIT TRAN">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER ACCOUNT NUMBER:
<INPUT TYPE="MSRT2" SIZE=40 NAME=account>
ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

1.b. Transaction Accepted Response

510

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
JUNE 1 1996              10:30AM          PURCHASE<P>
TERMINAL ID:                              999999999</P>
ACCOUNT NUMBER          99999999999999999999</P>
EXP DATE:                                     99/99</P>
AMOUNT                                     $9999.99</P>
AUTH NUMBER                               999999999</P>
<p>
--------------------------------------------</p>
          CUSTOMER SIGNATURE </p>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

1.c. Transaction Declined or Submit Error Response

520

```
<HTML>
<BODY>
DECLINED<MESSAGE>
</BODY>
</HTML>
```

FIG.14

2a. Transaction Request HTML+D

550

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
METHOD=SQL"BEGIN TRAN
        IF NOT EXISTS ( SELECT substring(account, 1, 20) FROM auth_table)
        BEGIN
                INSERT TABLE=log_table VALUES=(getdate(),tid, substring( account, 1,20),
                substring( account, 22, 4), amount)
        END
        ELSE  BEGIN
                        SELECT * FROM error_table WHERE error_no=1
                        RETURN
                END
        INSERT TABLE=order_table VALUES=( getdate(), cust_name,address,city, state, zip,
                part_code, unit_price, tax, ship_method, ship_chrg, unit_price + tax +
                ship_chrg, substring( account, 1, 20) ,substring( account, 22, 4))
        SELECT * FROM order_table WHERE trandate = getdate()
        COMMIT TRAN">
CUSTOMER NAME:
<INPUT TYPE="TEXT"SIZE=30 NAME=cust_name><p>
ADDRESS:
<INPUT TYPE="TEXT" SIZE=40 NAME=address><p>
CITY:
<INPUT TYPE="TEXT" SIZE=20 NAME=city><p>
STATE:
<INPUT TYPE="TEXT" SIZE=2 NAME=state>
ZIP:
<INPUT TYPE="TEXT" SIZE=10 NAME=address><p>
SCAN PART CODE:
<INPUT TYPE="BCW" SIZE=9 NAME=part_code><p>
ENTER UNIT PRICE:
<INPUT TYPE="AMT" SIZE=8 NAME=unit_price><p>
TAX:
<INPUT TYPE="AMT" SIZE=5 NAME=tax><p>
SHIPPING METHOD:
<INPUT TYPE="TEXT" SIZE=10 NAME=ship_method><p>
SHIPPING AMOUNT:
<INPUT TYPE="AMT" SIZE=5 NAME=ship_chrg><p>
SLIDE CARD:
<INPUT TYPE="MSRTZ" SIZE=40 NAME=account><p>
<INPUT TYPE="SUBMIT">
</FORM>
</BODY>
</HTML>
```

FIG.15A

EXHIBIT D
PAGE 22 OF 47

2.b. Transaction Accepted Response

555

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
ORDER # 9999999999 APPROVED</p>
JUNE 1 1996        10:30AM       PURCHASE</P>
TERMINAL ID:                     999999999</P>
NAME: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
ADDRESS:</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
CITY: XXXXXXXXXXXXXXXXXXXXXXXXX</p>
STATE: XX              ZIP:XXXXXXXXXX</p>
ACCOUNT NUMBER:        9999999999999999999</p>
EXP DATE:                          99/99</p>
PART CODE:                     999999999</p>
UNIT PRICE:                     $9999.99</p>
SHIP METHOD:XXXXXXX CHARGE      $9999.99</p>
TOTAL AMOUNT:                   $9999.99</p>
AUTH NUMBER:                   999999999</p>
<p>

- - - - - - - - - - - - - - - - - - - - - - -</p>
          CUSTOMER SIGNATURE</p>

</FORM>
</BODY>
</HTML>
```

2.c. Transaction Declined or Submit Error Response

560

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG. 15B

3.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
METHOD SQL
      "INSERT TABLE=order_table VALUES=( getdate(), cust_name,address,city, state, zip,
            part_code, unit_price, tax, ship_method, ship_chrg,unit_price + tax +
            ship_chrg, substring( account, 1, 20) ,substring( account, 22, 4))
      SELECT * FROM order_table WHERE trandate = getdate()">
<INPUT TYPE="LOCAL" NAME=tid>
CUSTOMER NAME:
<INPUT TYPE="TEXT" SIZE=30 NAME=cust_name></p>
ADDRESS:
<INPUT TYPE="TEXT" SIZE=40 NAME=address></p>
CITY:
<INPUT TYPE="TEXT" SIZE=20 NAME=city><p>
STATE:
<INPUT TYPE="TEXT" SIZE=2 NAME=state>
ZIP:
<INPUT TYPE="TEXT" SIZE=10 NAME=address></p>
ENTER PART CODE:
<INPUT TYPE="TEXT" SIZE=10 NAME=part_code></p>
ENTER UNIT PRICE:
<INPUT TYPE="AMT" SIZE=8 NAME=unit_price></p>
TAX:
<INPUT TYPE="AMT" SIZE=5 NAME=tax></p>
SHIPPING METHOD:
<INPUT TYPE="TEXT" SIZE=10 NAME=ship_method></p>
SHIPPING AMOUNT:
<INPUT TYPE="AMT" SIZE=5 NAME=ship_chrg></p>
<INPUT TYPE="SUBMIT" >
</FORM>
</BODY>
</HTML>
```

FIG.16A

EXHIBIT D
PAGE 24 OF 47

3.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
ORDER # 9999999999 APPROVED</p>
JUNE 1 1995 10:30AM PURCHASE</P>
TERMINAL ID: 999999999</P>
NAME: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
ADDRESS:</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
CITY: XXXXXXXXXXXXXXXXXXXXXXX</p>
STATE: XX ZIP:XXXXXXXXX</p>
PART CODE: 999999999</p>
UNIT PRICE: $9999.99</p>
SHIP METHOD:XXXXXX CHARGE: $9999.99</p>
TOTAL AMOUNT: $9999.99</p>
</FORM>
<FORM ACTION=<file_name> METHOD=PAYMENT>
<INPUT TYPE=AUTOSUBMIT>
</FORM>
</BODY>
</HTML>
```

3.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.16B

EXHIBIT D
PAGE 25 OF 47

4.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=SCR1 METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
SLIDE CARD:
<INPUT TYPE="MSRT2" SIZE=40 NAME=track2>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="LOCAL" NAME=work_key>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

4.b. Transaction Acceptec HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99 TIME:9999A</P>
ACCOUNT NUMBER 9999999999999999999</p>
EXP DATE: 99/99</p>
AMOUNT: $9999.99</p>
AUTH NUMBER 999999999</p>
</FORM>
APPROVED999999999</P>
</BODY>
</HTML>
```

4.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.17A

EXHIBIT D
PAGE 26 OF 47

5.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=SCR2 METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
ENTER PIN:
<INPUT TYPE="PASSWORD" SIZE=4 NAME=pin>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="LOCAL" NAME=work_key>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

5.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99 TIME:9999A</P>
ACCOUNT NUMBER 9999999999999999999</P>
EXP DATE: 99/99</P>
AMOUNT $9999.99</P>
AUTH NUMBER 999999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

5.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.17B

EXHIBIT D
PAGE 27 OF 37

6.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=host_URL METHOD=POST>
<INPUT TYPE="LOCAL" NAME=id>
SLIDE CARD
<INPUT TYPE="MSRTZ" SIZE=40 NAME=track2>
<INPUT TYPE="PIN" SIZE=4 NAME=pin>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

6.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99                    TIME:9999A</P>
TERMINAL ID:                     999999999</P>
ACCOUNT NUMBER  9999999999999999999</P>
EXP DATE:                        99/99</P>
AMOUNT                           $9999.99</P>
AUTH NUMBER                      999999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

6.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
<INPUT TYPE="LOCAL" NAME=work_key VALUE="9999999999999999">
</BODY>
</HTML>
```

FIG.18

EXHIBIT D
PAGE 29 OF 47

7.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
      METHOD=SQL
            "IF EXISTS ( SELECT account FROM check_table)
                  SELECT * FROM check_table WHERE account = DDAaccount
            ELSE SELECT * FROM error_table WHERE error_no=1">
<INPUT TYPE="LOCAL" NAME=id>
SCAN CHECK:
<INPUT TYPE="MICR" SIZE=20 NAME=account>
ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

7.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99                    TIME:9999</P>
TERMINAL ID:                     99999999</P>
ACCOUNT NUMBER  999999999999999999</P>
AMOUNT                           $9999.99</P>
AUTH NUMBER                      99999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

7.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.19

EXHIBIT D
PAGE 29 OF 47

8.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
        METHOD=SQL
            "BEGIN TRAN
             IF NOT EXISTS ( SELECT substring(account51,20) FROM auth_table)
                    BEGIN
                    SELECT cur_bal FROM cust_tbl WHERE substring(account51,20)=account
                    SELECT amount = amount - ( points / .01 )
                    SELECT cur_bal = cur_bal - ( amount * .01 )
                    UPDATE TABLE=cust_tbl VALUES=( getdate(), account, cur_bal - points )
                    SELECT * FROM log_table WHERE trandate = getdate()
                    INSERT TABLE=log_table VALUES=( getdate(), tid, substring(account51,20),
                    substring( account, 72, 4), amount)
                    END
        ELSE SELECT * FROM error_table WHERE error_no=1"
        COMMIT TRAN">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER ACCOUNT NUMBER:
<INPUT TYPE="MSRT1" SIZE=90 NAME=account>
ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
REDEEM POINTS?
<INPUT TYPE="INT" SIZE=6 NAME=points>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

8.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
JUNE 1 1995      10:30AM          PURCHASE</P>
TERMINAL ID:                      999999999</P>
ACCOUNT NUMBER      99999999999999999999</P>
EXP DATE                              99/99</P>
AMOUNT                             $9999.99</P>
AUTH NUMBER                       999999999</P>
</p>
- - - - - - - - - - - - - - - - - - - - - - </p>
          CUSTOMER SIGNATURE</p>
</p>
THANK YOU!</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
POINTS REDEEMED:                    999999</p>
POINTS EARNED:                      999999</p>
CURRENT POINT BALANCE:              999999</p>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

8.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <MESSAGE>
</BODY>
</HTML>
```

FIG. 20

EXHIBIT D
PAGE 30 OF 47

9.a Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
      METHOD=SQL
            "SELECT fields FROM table WHERE condition">
<INPUT TYPE="LOCAL" NAME=id>
ENTER SEARCH TABLE NAME:
<INPUT TYPE="TEXT" SIZE=10 NAME=table>
ENTER SEARCH FIELD NAMES:
<INPUT TYPE="TEXT" SIZE=100 NAME=fields>
ENTER SEARCH CONDITION:
<INPUT TYPE="TEXT" SIZE=50 NAME=condition>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

9.b Transaction Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
FIELD1  FIELD2  FIELD3  ----  FIELDN  </p>
------  ------  ------        ------  </p>
XXXXX   XXXXX   XXXXX         XXXXX   </p>
XXXXX   XXXXX   XXXXX         XXXXX   </p>
.
.
XXXXX   XXXXX   XXXXX         XXXXX   </p>
</FORM>
</BODY>
</HTML>
```

FIG.21

EXHIBIT D
PAGE 31 OF 47

10.a Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
      METHOD=SQL
      " INSERT TABLE=log_table VALUES=( getdate(), tid, gross_sales, opn_chks, voids,
      emp_disc, mgr_disc, vip_card, man_over, coupons, sales_tax, c_dep1, c_dep2,
      c_dep3, c_dep4, chg_fund,cc_dep, batch_no, chrg_sales, paid_outs, co_sales,
      cc_sales, te_sales, gross_sales - opn_chks - voids - emp_disc - mgr_disc - vip_card-
      man_over - coupons - sales_tax, gross_sales - opn_chks - voids -
      emp_disc - mgr_disc - vip_card - man_over - coupons - c_dep1 - c_dep2 -
      c_dep3 - c_dep4 - chg_fund - cc_dep - batch_no - chrg_sales - paid_outs)
      SELECT * FROM log_table WHERE trandate = getdate()">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER GROSS SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=gross_sales>
ENTER OPEN CHECKS:
<INPUT TYPE="INT" SIZE=7 NAME=opn_chks>
ENTER VOIDS:
<INPUT TYPE="INT" SIZE=7 NAME=voids>
ENTER EMP DISCOUNTS:
<INPUT TYPE="INT" SIZE=7 NAME=emp_disc>
ENTER MGR DISCOUNT:
<INPUT TYPE="INT" SIZE=7 NAME=mgr_disc>
ENTER VIP CARD:
<INPUT TYPE="INT" SIZE=7 NAME=vip_card>
ENTER MANUAL OVERRINGS:
<INPUT TYPE="INT" SIZE=7 NAME=man_over>
ENTER COUPONS:
<INPUT TYPE="INT" SIZE=7 NAME=coupons>
ENTER SALES TAX:
<INPUT TYPE="AMT" SIZE=8 NAME=sales_tax>
ENTER CASH DEPOSIT 1:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep1>
ENTER CASH DEPOSIT 2:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep2>
ENTER CASH DEPOSIT 3:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep3>
ENTER CASH DEPOSIT 4:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep4>
ENTER CHANGE FUND:
<INPUT TYPE="AMT" SIZE=8 NAME=chg_fund>
```

FIG.22A

EXHIBIT D
PAGE 32 OF 47

```
ENTER CC DEPOSIT:
<INPUT TYPE="AMT" SIZE=8 NAME=cc_dep>
ENTER BATCH #:
<INPUT TYPE="INT" SIZE=3 NAME=batch_no>
ENTER CHARGE SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=chrg_sales>
ENTER PAID OUTS:
<INPUT TYPE="INT" SIZE=8 NAME=paid_outs>
ENTER CARRY OUT SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=co_sales>
ENTER CREDIT CARD SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=cc_sales>
ENTER TAX EXEMPT SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=te_sales>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

10.b. Transaction Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
JUNE 1 1995      10:30AM      DAILY REPORT</P>
TERMINAL ID:                  9999999999</P>
GROSS SALES                   999999.99</P>
 VOIDS                  99      99999.99</P>
 EMP DISCOUNTS          99      99999.99</P>
 MANAGER DISCOUNTS      99      99999.99</P>
 VIP CARD               99      99999.99</P>
 COUPONS                99      99999.99</P>
 MANUAL OVERRINGS       99      99999.99</P>
 SALES TAX                     999999.99</P>
 CASH DEPOSIT 1                999999.99</P>
 CASH DEPOSIT 2                999999.99</P>
 CASH DEPOSIT 3                999999.99</P>
 CASH DEPOSIT 4                999999.99</P>
 CASH DEPOSIT 5                999999.99</P>
 CHANGE FUND                   999999.99</P>
 CC DEPOSIT             999    999999.99</P>
 CHARGE SALES                  999999.99</P>
 PAID OUTS               99     99999.99</P>
 CARRY OUT SALES               999999.99</P>
 CREDIT CARD SALES             999999.99</P>
 TAX EXEMPT SALES              999999.99</P>
 ----------------------------------------</P>

 NET SALES                       99999999</P>
 OVER/SHORT                      99999999</P>
</FORM>
</BODY>
</HTML>
```

FIG.22B

EXHIBIT D
PAGE 33 OF 47

11.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM  ACTION=MAIL TO: mail_to>
 ENTER MAIL ADDRESS:
<INPUT TYPE="TEXT" SIZE=20 NAME=mail_to>
ENTER MESSAGE:
<INPUT TYPE="TEXT" SIZE=100>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

FIG.23

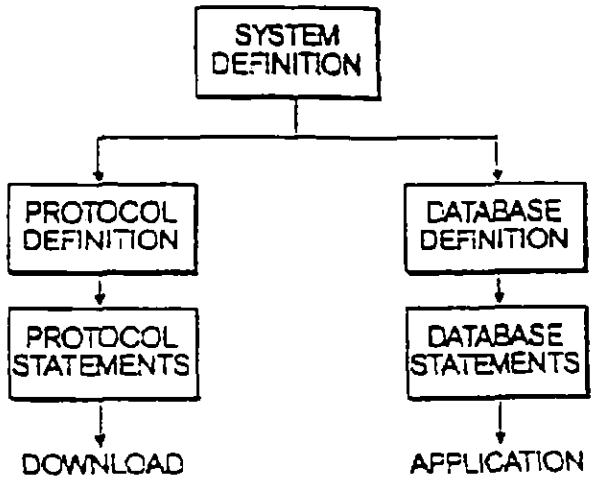

FIG.24A

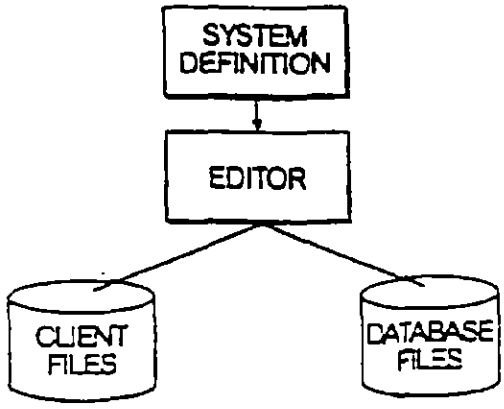

FIG.24B

EXHIBIT D
PAGE 34 OF 47

SYSTEM AND METHOD FOR ENABLING TRANSACTIONS BETWEEN A WEB SERVER AND A SMART CARD, TELEPHONE, OR PERSONAL DIGITAL ASSISTANT OVER THE INTERNET

This application is a continuation of co-pending application Ser. No. 10/100,347, filed on Mar. 18, 2002 (now pending), which is a continuation of application Ser. No. 09/907,076, filed on Jul. 17, 2001 (now pending), which is a continuation of application Ser. No. 09/314,266, filed on May 18, 1999 (now U.S. Pat. No. 6,366,967), which is a continuation of application Ser. No. 08/995,123 filed Dec. 19, 1997 (now U.S. Pat. No. 5,905,908), which is a continuation of application Ser. No. 08/493,772 filed Jun. 22, 1995 (now U.S. Pat. No. 5,742,845).

FIELD OF THE INVENTION

This invention relates to data transaction systems, and more particularly, to data transaction systems using non-standard input/output devices.

BACKGROUND OF THE INVENTION

Data transaction systems which communicate with a plurality of remote terminals to transfer information used to complete a transaction or compile a database are well known. Typically, such systems include a central transaction processing system which may maintain a database of information such as customer or consumer data. Exemplary information in such a database may include customer identification, customer account numbers, credit limits and/or account balances from which a customer may draw. The central transaction processing system is typically coupled to a plurality of remote transaction or data input terminals. Transaction computers may include special purpose devices such as automatic teller machines (ATMs), point of sale (POS) terminals, credit card terminals, and screen phone terminals. Screen phone terminals are devices which integrate a telephone with an ATM-like device and possibly a magnetic card swipe reader. Data input terminals may include personal computers (PCs) interfaced to data collection devices or special purpose data collection terminals or monitors.

In these known data transaction systems, a user usually initiates a transaction by requesting access to funds in an account or from a credit line maintained by the central processing system. The request is transmitted to the central processing system which performs a verification to determine whether the user is a valid user of the system, has an account within the system, and that the amount of the transaction is within the limits of the consumer's credit line or that the user has the requested funds available in an existing account monitored by the central processing system. The central processing system then transmits authorization for or denial of the transaction to the remote terminal. In response to the message from the central processing system, the remote terminal dispenses cash (for an ATM) or the merchant provides the goods being purchased to the user if the authorization message indicates that the consumer's funds will be transferred to the merchant's account. Similar communication exchanges occur in data systems where electronic documents and other information are provided to a central site for compilation or processing. Consequently, this background discussion applies to all such transaction and data systems. Though the remainder of the discussion is directed to transaction systems, the reader should appreciate that the comments also apply to data systems as well.

The remote terminals may be coupled to the central processing system in several ways. For example, in some ATM systems, the ATMs are coupled to the central processing system through dedicated telephone or other data communication lines. These systems are preferred because they provide a relatively high degree of security since the dedicated data line coupling the central processing system to the ATM is not generally accessible by members of the public. The physical security of the dedicated data line is, however, expensive because no other traffic may utilize the line. Thus, the cost of leasing the dedicated line to an ATM with relatively low volumes of transactions may yield a high communication cost per transaction.

In an effort to reduce the communication cost per transaction, some transaction or data systems utilize telephone lines through a publicly-switched telephone network (PSTN) which may be accessed by other members of the public. Specifically, devices such as credit card terminals and screen phone terminals typically include a modem which converts the digital messages of the remote terminal into frequency modulated analog signals which may be transmitted over telephone lines to a modem at the central processing system. In other systems, the terminal may communicate digital data directly over ISDN lines of the PSTN to the central processing system. This line of communication between a remote terminal and the central processing system is performed by having the remote terminal dial a telephone number associated with the central processing system to establish communication with the central processing system. This type of communication path is relatively secure because the switching networks for the communication traffic through the PSTN are not readily accessible by the public and during the course of the financial transaction, only the central processing system and remote terminal are on the line.

Regardless of the communication method used to couple the central processing system to the remote terminals, the protocol and data formats used between the devices is typically proprietary. That is, the operator of each financial transaction system designs its own protocol and data message format for communication with the processor at the central site or generates a variant within a standard such as those established by the ANSI committee or the like for such communication. As a result, the remote terminals must include software that supports each operator's protocol and message formats in order to be compatible with an operator's central site. For example, application software in a credit terminal such as the TRANZ330, TRANZ380, or OMNI390 manufactured by VeriFone implement one or more of the communication protocols and formats for National Data Corporation (NDC), VISANET, MASTERCARD, BUYPASS, and National Bancard Corporation (NaBANCO) system processors in order to support transactions with the most popular transaction centers. Thus, the communication software absorbs a significant amount of terminal resources which could be used to support other terminal operations.

A related problem arises from the expanding home banking market. A customer of home banking system typically uses a screen phone terminal or a personal computer (PC) having a modem to establish communication through a PSTN to a central transaction processing system. Again, the operator of the central processing system must provide information regarding the data message formats for communicating with the central processing system to a vendor of software for the home banking terminals or must provide that software to its customers. As a result, home banking

customers must purchase software to communicate with each banking system of which the customer wants to be a member. This cost and the need to install additional communication programs may make some consumers reluctant to be a member of more than one banking system or to change banking systems.

A communication system becoming increasingly popular and which provides standardized communication is the Internet. The Internet is an open network of networks which communicate through a variety of physical communication devices such as telephone lines, direct communication lines, and the like. Each network is coupled to the main Internet network for communication through a host computer supporting a TCP/IP router or bridger. The host computer typically includes a program, frequently called a Web server, which acts as a gateway to resources at the host computer which may be resident on the host computer or a network coupled to the host computer. Each server has an address identifying the location of the resources available through the Web server. The router recognizes communication for the server and directs the message to the server or it recognizes that the communication should be forwarded to another server. As a result, communication within the Internet may be point-to-point, but more likely, the communication path is a somewhat circuitous one with the information passing through the routers of multiple servers before reaching its final destination.

A number of message protocols and formats have been developed for the Internet. The physical communication protocol and data message format is the Transport Control Protocol/Internet Protocol (TCP/IP). The TCP/IP protocol involves multiple layers of encapsulating headers containing communication information which are used to provide byte streams or datagram communications to computers on the networks coupled to the Internet. Encapsulated within TCP/IP headers are protocols which are used to format the data messages or transfer data from one computer to another computer coupled to the Internet. These protocols include File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), Telnet, and Hyper Text Transport Protocol (HTTP). The advantage of these protocols is that each provides a standardized communication format for transferring information between computers on the Internet. These protocols are typically called open system protocols as they are publicly known and may be utilized by any programmer to develop programs for communicating with another computer coupled to the Internet. These non-proprietary protocols have contributed to the acceptance of using the Internet as an open network for coupling computer networks together.

While the Internet provides an open network for computer communication with publicly accessible protocols and formats, the Internet suffers from a number of limitations which preclude its effective use as a transaction or data system which uses non-standard I/O terminals and devices. First, circuitous communication presents a number of security issues for such a system. For example, a Web server could incorporate a router which examines the address of each message coming through it and upon recognizing an address associated with a central transaction processing system, copy the data message for the unauthorized retrieval of customer-sensitive information such as account numbers and personal identification numbers (PINs) which may be contained in the message.

A second limitation of open networks such as the Internet is that communication on such networks is only supported for computers acting as servers or clients. Specifically, all of the protocols and formats are constructed for standard input/output (I/O) operations for a PC terminal. That is, text information is directed to a standard monitor screen, user input is expected from a standard keyboard, and files are transferred to standard peripherals such as a hard disk or diskette drive. Especially absent is the ability in open network protocols for communication with devices that only use communication interfaces such as RS-232C. As a result, communication over the Internet is primarily performed with standard PCs through network communication methods and interfaces.

This presents a number of problems for home banking or for interfacing non-standard I/O terminals such as credit card terminals or screen phones to open networks such as the Internet either directly or through a PC. Generally, non-standard I/O devices are devices which interface to a PC through a port not normally used for networks, such as a RS-232C port, or are devices which have limited input and output capabilities such as small screen displays or ten keypads. These devices are not supported on the Internet because servers use protocols that communicate with PCs supporting standard QWERTY keyboards and standard monitors. Consequently, users are limited to entering account numbers and the like through a keyboard of a PC-like device for processing at a central transaction processing system. To request a transaction, one need only have a person's credit card account number. If the credit card number had to be input through a magnetic card reader, unauthorized access to a customer's account would be less likely since physical possession of the credit card would be required to initiate the transaction.

Another limitation of the standard I/O devices currently supported by the open network protocols is the lack of encryption. For example, PIN pads, which are typically incorporated in ATMs, automatically encrypt in hardware a PIN entered by a user. Such devices typically encrypt the number by implementing a data encryption standard (DES) algorithm in hardware before the PIN is transmitted or stored. When a standard keyboard is used to input the PIN, no hardware encryption is performed and, as a result, an unencrypted copy of the PIN is provided to the memory of the PC. Storage of unencrypted PINs is in contravention of current banking regulations. If PIN pads could be read via Internet protocols, then such a lapse in PIN security would be less likely to occur.

Another I/O device not supported on open networks are smart cards which are increasing in use. Smart cards include a processor and memory in which information regarding the amount of funds in a particular account, a transaction history, account numbers, and customer data may be stored. The card may be read through a smart card reader which is a computer having a processor and memory but usually provided with non-QWERTY keypads and limited displays. A transaction processor may validate a card owner through a PIN provided through a keypad, determine the amount of money remaining on the card and debit the card itself for a transaction amount by communicating with the smart card reader with one of the proprietary protocols discussed above. Such information is not readily obtainable by the owner of the card and so cannot be entered through a keyboard or the like. Smart card readers are non-standard devices which may be coupled to a PC through a COMM1 or COMM2 port. However, none of the standard protocols and message formats for open network communications currently provide I/O operations for such devices.

All systems which attempt to provide three party communication to execute an electronic transaction suffer from

a number of limitations which present risks greater than those in a normal transaction performed at the point of sale. In a typical point of sale (POS) transaction, the consumer hands a debit or credit card to a merchant's agent who may examine the card for security markings such as holograms, watermarks, or a cardholder signature. The agent then places the card into a reader for acquiring information from the card and, in some cases, have the consumer enter a PIN into a PIN entry device which encrypts the PIN in a hardware implemented scheme. If the PIN is entered, it is transmitted with the information from the card to a processing center, typically in one of the formats discussed above, under a X.25 protocol or the like. The processing center returns an authorization granted or denied message. The reader typically has a printer coupled to it through an RS-232C port or the like and a purchase agreement is printed. The consumer signs the agreement, the merchant's agent may verify the signature, and the merchant retains an original of the agreement and the consumer a copy. In this scenario, the merchant has initiated the communication to the processing center. The safeguards noted above permit the processing center to charge a merchant a lower processing fee than when a consumer initiates a transaction. Consumer initiated transactions present a greater risk because the consumer provides an agent an account number in a telephone conversation or non-encrypted DTMF transmission. Thus, there is no card inspection, signature verification, or PIN verification. As a result, such transactions are limited to credit cards because debit cards require that the cardholder be present to enter a PIN into an appropriate PIN entry device.

What is needed is a system that permits consumers remote from a merchant to order goods and present payment in a secured manner so the merchant's risk and processing costs, as well as a cardholder's exposure to fraud, is reduced. What is needed is a way for a processing center to communicate through an open network with non-standard I/O devices such as credit card terminals, personal digital assistants, and screen phone terminals or with non-standard I/O devices coupled to the open network through a PC or the like. What is needed is a transaction or data system which utilizes an open network such as the Internet to support electronic transactions or data compilation in a secure manner without undue limitation as to the devices with which communication may be made.

SUMMARY OF THE INVENTION

The present invention provides transaction and data systems which may be implemented on an open network such as the Internet. The system comprises a server for communicating in an open network protocol and a plurality of input/output (I/O) devices coupled to the server through an open network, the I/O devices communicating with the server in the extended open network protocol that supports communication with non-standard I/O devices over the open network. The system of the present invention provides a server with the capability of communicating with a number of I/O devices useful in transaction and data systems which heretofore have been unsupported on an open network system such as the Internet.

The system of the present invention is implemented by extending present open network communication protocols and data message formats to communicate with non-standard I/O devices either coupled to an open network as a client or coupled to an open network through a client, such as a PC, credit card terminal, screen phone, or PDA. That is, commands which are compatible with the communication schema of a presently-implemented protocol for the Internet are used and additions are made to commands implemented within the control structure of that existing protocol to support non-standard I/O device communication. At the server, the extended protocol is further supported by a common gateway interface (CGI) which converts the communication from a non-standard I/O device to a format which is compatible with a transaction or data application program which may be executed on the server or a computer coupled to the server. In this manner, the CGI permits the processing of the extended capability commands to be segregated from the communication functions performed by the server.

Preferably, the server and the I/O devices communicate through an Internet protocol and most preferably, the Hyper Text Transport Protocol (HTTP), to exchange data between an application program and non-standard I/O devices over an open network. Although HTTP is the preferred protocol used to implement the present invention, other protocols such as Telnet or SMTP, for example, may also be extended in a similar manner. Specifically, the HTTP protocol is expanded to communicate with printers, magnetic card readers, credit card terminals, smart card readers, check readers, PIN pads, bar-code readers, PDAs, or the like, and includes a command which instructs a non-standard I/O device to disconnect from the open network and re-couple to a transaction processing system to transfer funds from a consumer account to a merchant account through a PSTN or dedicated data line. By using these extended capability commands within HTTP, a processing system may operate on an open network such as the Internet and communicate with transaction or other data I/O devices which have not previously been able to couple to such open networks. Such a system may be used to execute a transaction between a consumer and a merchant so the merchant receives remittance information in a timely manner. The system permits the consumer to initiate a transaction and order from a merchant and then use a more secure link supported by PIN entry devices or the like to reduce the risk of fraud for the transaction.

Because the server may communicate through such open networks with non-standard I/O devices, the transaction or data processing system is available for the ever-expanding market available through the Internet. Such a system is able to communicate with non-standard I/O devices in myriad locations such as retail establishments or in consumers' homes. For example, a consumer may utilize the standard capability of an Internet protocol to communicate with a server that provides information regarding services or goods for sale over the Internet and then consummate a sales transaction by using the extended capability of the Internet protocol. Such a home consumer could provide transaction data through a smart card reader coupled to a COMM1 or COMM2 port of a PC. A database program executing at the server for the central processing site may accept product ordering information from a non-standard keypad or touch screen associated with a screen phone terminal at the remote site and then communicate with the smart card reader to consummate the transaction. Such a transaction system requires that the consumer have physical possession of the smart or credit card and not simply knowledge of the account number. Likewise, the server would be able to communicate with a PIN pad or the like to ensure the hardware encryption of PINs and other data before it is transmitted to the server site. Such a system is less susceptible to consumer fraud.

Another feature of the present invention is a PAYMENT command implemented in the extended Internet protocol

that directs a non-standard I/O device or a PC interfaced with such devices to communicate with a transaction processor through an alternative communication link. In one form, the PAYMENT command is used by a merchant terminal to submit a consumer's account number with a merchant deposit account number through a PSTN network or the like to the processing center. In another form of the PAYMENT command, a client program in a consumer's terminal receives an account number for a merchant account from a merchant's server with the PAYMENT command. On receipt of this command, the client program suspends its operation and passes the account number to a conventional bank processing program co-resident in memory. The bank processing program establishes a standard communication link with a transaction processing system through a dedicated data line or a PSTN network. Using that communication link, the bank processing program executes a commercial transaction using a standard VISA protocol or the like. The consumer may use a magnetic stripe reader and a PIN entry device to improve the security of the data transmission. The transaction center may transmit remittance data over the open network to the merchant so the merchant is apprised of payment and ships the ordered product. Once this consumer initiated transaction is complete, the bank processing program terminates and returns control to the client program which may terminate communication with the open network or retrieve information from another server on the open network for another transaction. In this way, the user may use the open network for non-confidential communication such as collecting product information, pricing, and product availability. This information may be collected quickly and efficiently using the extended Internet protocol. The conventional bank processing program and more secure communication links may then be used for the confidential information required for the transaction. Thus, the present invention is able to combine the features and advantages of the Internet with the more secure communication link and data security enhancing devices of systems presently known.

Preferably, an editor is provided which permits a user to define an application database table with data fields, define client application data fields, and define the integrated forms for communicating data between the defined database tables and a client application. The editor verifies the syntax of the user generated integrated forms containing extended Internet protocol statements and client application statements. The editor ensures that the variable names for the client application and the data fields for the database application correspond. Following the generation of the integrated form, the editor parses the integrated form to segregate the database language statements from the extended Internet protocol statements. A database language identifier is substituted in the Internet protocol statements for the database statements contained in the integrated form. The Internet protocol statements are downloaded as a file which is interpreted by the client program for the collection and submission of data from non-standard I/O devices to the database application. The database language statements segregated from the extended Internet protocol statements are placed in a second file which is named to correspond to the database table defined by the user. The CGI application recognizes the database language identifier contained in the returned forms of the Internet protocol statements. The CGI application correlates the database identifier with the file previously generated by the editor which contains the database command statements. The application then inserts the data from the returned form into the database command statements and provides the re-integrated database command statements to the database application. In this manner, the database may be queried by or retrieve data from the non-standard I/O device. In the most preferred embodiment, the editor permits a user to develop integrated forms comprised of the extended HTML language and standard query language (SQL) database application statements. In this manner, the user does not have to manually generate the SQL commands, the HTML commands, and carefully correlate the data fields of the two commands in order to implement a transaction between a client and a database.

These and other advantages and features of the present invention may be discerned from reviewing the accompanying drawings and the detailed description of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention may take form in various components and arrangement of components and in various steps and arrangement of steps. The drawings are only for purposes of illustrating a preferred embodiment and are not to be construed as limiting the invention.

FIG. 1 is a diagram of an open network system in which the present invention is utilized;

FIG. 2 is a diagram of the format of the FORM and INPUT tags implemented in the preferred embodiment of the present invention;

FIG. 3 is a diagram of the preferred SQL commands supported in the preferred embodiment of the present invention;

FIG. 4 is a flowchart of the high level processing of the client program which interprets the HTML files of the preferred embodiment of the present invention;

FIG. 5 is a flowchart of the HTML file processing performed by the client program of the preferred embodiment of the present invention;

FIG. 6 is a flowchart of the attribute processing for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. 7 is a flowchart of the processing of the ACTION attribute for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. 8 is a flowchart of the processing for the METHOD attribute for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. 9 is a flowchart of the attribute processing for the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. 10 is a flowchart of the processing for the TYPE attribute for the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. 11 is a flowchart of the processing for the NAME attribute of the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. 12 is a diagram of the format for the ACTION attribute for the FORM tag performed by the common gateway interface between the Web server and an application program;

FIG. 13A is a diagram of the possible communication paths which may be used by an I/O device according to the principles of the present invention;

FIG. 13B shows an exemplary FORM tag and INPUT tag for the PAYMENT method implemented in a merchant's terminal according to the principles of the present invention;

FIG. 13C shows an exemplary FORM tag and INPUT tag for the PAYMENT method implemented in a consumer's terminal according to the principles of the present invention;

FIG. **14** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a card initiated payment authorization and capture transaction;

FIGS. **15A–15B** show exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a bar code reader input with card-initiated payment authorization transaction;

FIGS. **16A–16B** show exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a key input order with secure payment transaction;

FIG. **17A** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a smart card debit (Type 1) transaction;

FIG. **17B** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a smart card debit (Type 2) transaction;

FIG. **18** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a debit card transaction;

FIG. **19** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a check verification transaction;

FIG. **20** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a customer frequency transaction;

FIG. **21** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for an item search transaction;

FIGS. **22A–22B** show exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for retail store end of day reporting;

FIG. **23** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a store reporting an e-mail transaction;

FIG. **24A** is a diagram of a manual development process for the files interpreted by the client program and the files interpreted by the application program in accordance with the principles of the present invention; and

FIG. **24B** is a diagram of the generation of the files interpreted by the client program and the files interpreted by application program performed by an editor constructed in accordance with the principles of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

A transaction or data system constructed in accordance with the principles of the present invention is shown is FIG. **1**. The system **10** includes a Web server. **12** which is coupled to an open network **14** such as the Internet for communication with various I/O devices and terminals. For example, the I/O devices which may be coupled directly to network **14** include standard I/O devices already supported by Internet protocols such as PCs **30** and non-standard I/O devices such as a screen phone terminal **16**, a personal digital assistant (PDA) **18**, and a credit card terminal **20**. Other exemplary non-standard I/O devices such as smart card reader **32**, personal identification number (PIN) pad **34**, magnetic card swipe reader **36**, printer **38**, or the like, may be coupled to PCs through non-standard I/O ports such as COMM1 and COMM2 ports or to other non-standard I/O devices such as phone terminal **16**, PDA **18**, or credit card terminal **20**. Typically, these devices are coupled to PCs or devices **16**, **18**, or **20** through an interface such as a RS-232C interface. Merchants or other vendors may use a Web server **2** to couple to network **14** to communicate with the devices and processing system **40**.

The Web server **12** is preferably coupled to a Common Gateway Interface (CGI) application **28** which converts and communicates the data and commands between the devices on network **14** and the processing system **40** so the I/O devices do not have to use the database command language to interact with the database. System **40** and the devices may communicate directly if they are implemented in the same language or if a user implements a communication interface such as CGI **28** that correlates data fields in the client with those in system **40**. Server **12**, CGI **28**, and the applications supporting system **40** may all reside on a single host computer or they may reside on separate computers coupled together by a local area network (LAN) or a wide area network (WAN). Preferably, the application interfaces with a database which supports Open Data Base Connectivity (ODBC) and Structured Query Language (SQL).

The communication sessions between the I/O devices coupled to the open network **14** and the Web server **12** are generally conducted in the same fashion as Internet protocol communication sessions are currently performed. That is, the I/O device establishes a communication connection with Web server **12**, sends a request to the Web server, the Web server responds to the request and the I/O device or server closes the connection. Preferably, the non-standard I/O devices or PCs interfaced to such devices selectively couple to a local access port on the open network **14** through a local modem/ISDN connection. In this manner, the device is only coupled to the open network **14** when a transaction or a data operation is to be performed. While connected to the open network **14**, a device may access a number of servers to accomplish a purpose. For example, a device may couple to a local access port and communicate with a first server to check inventory levels at a site, communicate with a second server to order stock for the inventory, and communicate with a third server to settle payment for the ordered goods. When all aspects of the transaction are complete, the connection with the local access port is terminated. In the preferred embodiment of the present invention, the protocol used to transport data messages between Web server **12** and the I/O devices coupled to the open network **14** is the Hyper Text Transport Protocol (HTTP), although other open system protocols utilized on the Internet may be used.

In standard HTTP protocol, a client program executing in one of the I/O devices may initiate communication with a server by sending a query message of the format:

http://<host>:<port>/<path>?<search part>

The message identifies the client as seeking communication with a HTTP server at the host address on the specified

EXHIBIT D
PAGE 99 OF 47

port. In the HTTP protocol, the default value for the port is 80 and the host address is the Internet protocol (IP) address of the type well-known in the art. The path value selects the file in the HTTP server which is activated in response to the message and the search part specifies a query for the selected file. In the initial communication, the query may be omitted so that the selected host file responds to the client program before a query is processed.

In the present invention, the client program uses a similar message to initiate a transaction or data operation, except that database commands are preferably embedded in a file at the server 12 and not in the "search part" of the command, although search parts may be constructed in accordance with the principles of the present invention that support non-standard I/O devices. Preferably, the client program interprets Hyper Text Markup Language (HTML) files containing HTML commands for communicating data between non-standard I/O devices and server 12. Most preferably, the HTML commands contain identifiers which are used by the CGI to place data returned in the forms of the HTML commands into database commands for queries or data insertions for the database. HTML is a command language well known for the retrieval and display of electronic documents for standard I/O devices such as PCs supported by full screen monitors, QWERTY keyboards, and standard peripherals such as hard disk drives and diskette drives. Standard HTML commands use text and previously known commands that reference Universal Resource Locators (URLs) to support the communication of electronic documents. These documents are files which may contain HTML commands, text, audio, video, or image data. The present invention extends HTML with commands that support communication between the server and the non-standard I/O devices.

In the HTTP protocol, data may be obtained during a communication session by using a tag called a FORM as part of the file defined by <path> in the command discussed above. The FORM format for standard HTTP is:

```
<FORM ACTION="URL"
      METHOD=GET|POST
>
Command
</FORM>
```

where "|" is an "OR" operator. The commands supported by standard HTTP are INPUT, SELECT, and TEXTAREA. Additionally, standard HTTP permits the inclusion of text data in the command area. In the present invention, HTML has been extended to support new ACTIONs, METHODs, and INPUTs.

In accordance with the principles of the present invention, tags are preferably used to identify device transfers and input operations. Preferably, the FORM tag is used to identify device transfers and ACTION and METHOD attributes further identify the device operation. As shown in FIG. 2, the extended ACTION field may include a FROM and TO attribute for accessing a local terminal file or smart card reader or a TO PRINTER attribute for directing output data to a printer local to the I/O device. The FROM and TO attributes for accessing local files and smart card readers and for directing output data to a local printer have previously been unsupported in any Internet protocol. As a result, the server 12 may access non-standard I/O peripherals for any of the I/O devices used in the transaction or data system 10. The ACTION="URL" is a part of standard HTTP and is well known.

The METHOD attributes may include the GET, POST, PAYMENT, or SQL methods. The GET and POST methods are currently supported in standard HTTP and are well known. The PAYMENT attribute is a directive to deliver data retrieved by an INPUT command to a private payment network for authorization and settlement and is not available in current Internet protocols. This directive is used by the client program to activate a conventional financial transaction application which communicates with the transaction system over a dedicated data line or PSTN in a known protocol such as VISA. Such an attribute is used where the more secure physical connection between remote site and transaction system and data encryption devices or the like are preferred. The SQL method preferably identifies a database language file which CGI 28 uses to correlate data in the HTML FORM to an insertion or query command contained in the file.

The preferred format for the INPUT tag which is used to identify input operations is also shown in FIG. 2. The TYPE and NAME attributes are used to define a non-standard I/O device or local storage variable for the input of data. The TYPE field values "text," "password," "checkbox," "radio," "submit," and "reset" are previously known, as are the attributes NAME, VALUE, CHECKED, SIZE, and MAXLENGTH. To support the extended capability of the present invention, the TYPE attribute preferably includes attributes MSRT1 for reading track 1 of a magnetic swipe reader, MSRT2 for reading a magnetic swipe reader track 2, KEY for reading input from a terminal command keypad, PIN for reading a personal identification number pad, BCW for reading a bar code wand, MICR for reading a check magnetic code reader, ATM for reading a dollar amount via a key input mask, INT for reading an integer via a key input mask, database with the INSERT attribute or update data already existing in a database with the UPDATE attribute. The values for the INSERT attribute may be identified with the VALUES attribute, and the SET and WHERE attributes may be used to define and conditionally update values in the identified database. Preferably, the present invention implements two DELETE and CREATE attributes. The DELETE attribute deletes all items in an identified column of a database table which may satisfy a condition defined by a WHERE attribute. The CREATE attribute creates a database table having a primary key identified by the PRIMARY KEY attribute.

Preferably, the server program executes on a computer system having at least an Intel 80386 or better processor with at least 4 megabytes of RAM and at least 3 megabytes of hard disk space available. The computer system running the server may operate any known server platform operating system such as WINDOWS 3.1, WINDOWS 95, or WINDOWS NT, UNIX AIX, and others. The non-standard I/O devices require a processor of a Z80A type or better, at least 32K bytes of RAM, and at least 32K bytes of ROM. The device includes a modem capable of at least 1200 bits-per-second (bps) but other modem speeds may be used for communication between client and server. Alternatively, the device may be coupled to a LAN which in turn is coupled to the Internet for communication with server 12. A typical non-standard device which executes the client program is a VeriFone OMNI390, OMNI395, or VuFone terminal. OMNI390, OMNI395, and VuFone are trademarks of VeriFone, Inc., of Redwood City, Calif. Other exemplary devices include Phillips Screen phone, Hypercomm T7 terminal, and Apple Computer Newton MessagePad.

To build the preferred HTML files which CGI 28 preferably uses to implement the client program and database

EXHIBIT D
PAGE 40 OF 47

application, the user preferably uses an off-line editor. The files generated by the editor are preferably comprised of an integrated statements formed from HTML statements and database statements for retrieving and writing data with the database. Exemplary files showing such integrated statements for performing transactions are depicted in FIGS. 14–23B. After such a file is generated, the editor parses the integrated statements into HTML statements and into database statements such as SQL commands. The HTML files required by the client program to support communication with a transaction or data processing center may be downloaded to a device or PC for execution. The files containing the database application statements used by the CGI interface to communicate data with the database application program preferably reside on server 12. Preferably, the database files used by the CGI interface include SQL commands for the application program interfaced to an ODBC compliant database.

The general format of the HTML commands in the HTML files used for communication with a client program and server are of the general format: TAG ATTRIBUTE. Preferably, the TAG field may be one of FORM, INPUT, SQL, or TEXTAREA. The ATTRIBUTE field value depends upon the TAG value. Preferably, the FORM tag may include the ACTION or METHOD attributes where the ACTION attributes include the FROM<file>, TO PRINTER TO<file>, and TO SCR values noted above, as well as the standard HTML ACTION value of URL=<file>. The METHOD attributes include the PAYMENT and SQL attributes noted above, as well as the standard HTML METHOD values of GET and POST. Also in accordance with the principles of the present invention, the INPUT tag may include TYPE, NAME, VALUE, CHECKED, SIZE, and MAXLENGTH attributes. These attributes are previously supported for the INPUT tag in HTML, however, the present invention further includes TYPE values of MSRT1, MSRT2, KEY, PIN, BCW, MICR, AMT, INT, LOCAL, and AUTOSUB, as well as the standard HTML TYPE values of TEXT, PASSWORD, CHECKBOX, RADIO BUTTON, SUBMIT, and RESET. The present invention also supports NAME attributes of IP_ADDRESS, HOST_PHONE, TID, WORK_KEY, DATETIME, and DEPOSIT_ACCT to identify local storage areas as well as standard HTML NAME attribute <Field_NM> to identify a FORM variable.

The preferred high level processing of the client program is shown in FIG. 4. That processing includes an idle step (Block 100) in which the program performs general housekeeping tasks such as maintaining internal time, scanning for input which may activate the device, or other known functions. Further processing is activated by some operator action at the device or PC which causes the device to either open a remote URL (Block 102) or open a local URL (Block 104). If a remote URL is required, the device transmits a message of the format discussed previously which is routed through the open network and delivered to a server 12 for a transaction or data processing system (Block 106). The HTML file selected at the server 12 is identified by the remote URL in the initial communication between the device and server 12 and that URL is used to return the selected HTML file to the device for processing (Blocks 108, 110).

FIG. 4 also shows that an operator may initiate an open local URL function by typing in a command or by pushing a hot key which is associated with a local URL. The I/O device reads the HTML file identified by the URL from local memory (Block 112) and passes the HTML file to the function for processing HTML files (Block 110). After a file is processed (Block 110), the client program determines whether the HTML file is to be stored (Block 114). If it is not, the process returns to the idle processing (Block 100). Otherwise, the process determines whether the HTML file is to be associated with a hot key (Block 116) and, if it is, it stores the file and generates the link between a hot key and the stored file (Blocks 118, 120). If the HTML file is only to be stored, no association is made with a hot key and the file is simply stored in local memory (Block 20). The client program then returns to idle processing (Block 100).

The high-level processing for the HTML file (Block 110, FIG. 4) is shown in further detail in FIG. 5. The process begins by scanning the HTML file for a TAG (Block 140). If no TAG is found, the file is not in proper format for processing and processing returns to Block 114 discussed in FIG. 4 above. If a TAG is found (Block 142), the process determines whether the TAG is a FORM TAG (Block 144) or an INPUT TAG (Block 146). If it is a FORM TAG, then the FORM TAG is processed and the program continues by looking for other TAGS to process (Block 140). If the TAG is an INPUT TAG, the INPUT TAG is processed (Block 150) and the program continues by looking for other TAGS to process (Block 140). If the TAG is one of the standard HTML TAGS, the program implements the TAG in standard known ways (Block 152) and then scans for other TAGs to process (Block 140).

Processing the ATTRIBUTES used to implement a FORM TAG is shown in FIG. 6. That process continues by scanning the HTML file for an attribute (Block 160). If an attribute is not found (Block 162), the program returns to scan for other TAGS (Block 140, FIG. 5). If an attribute is found, the program determines whether it is an ACTION attribute (Block 164) or a METHOD attribute (Block 166). Depending on the type of attribute, the appropriate function for processing the attribute is executed (Blocks 168 or 170) and scanning for additional attributes continues (Block 160). If the attribute is not an ACTION or METHOD attribute, there is an error in the file and processing returns to scan for other TAGs.

The processing for the ACTION attribute is shown in FIG. 7. There, the ACTION attribute is examined to determine whether it is a FROM<file> (Block 180), TO PRINTER (Block 182), TO<file> (Block 184), TO SCR (Block 186), FROM SCR (Block 188) or a URL=<file> (Block 192). The URL=<file> ACTION is a standard HTML action which is processed in a known way (Block 194). The FROM <file> action is processed by reading data from a file associated with the I/O device or PC interfaced to the I/O device (Block 196). The TO PRINTER action results in data in the FORM being sent to the printer (Block 198) while the TO <file> action results in data in the FORM being written to a local file (Block 200). The TO SCR action causes data to be written to the smart card via a smart card reader (Block 202) and the FROM SCR reads data from a smart card through a smart card reader (Block 204). After the appropriate action processing takes place, the HTML file is scanned for additional ACTION values to perform (Block 206), and if one is found, the process continues. If no attribute is located (Block 208), the process returns to scan for other attributes (Block 160, FIG. 6).

The processing for the METHOD attributes for FORM tags are shown in FIG. 8. The process determines which type of METHOD is present in the FORM and then properly processes the attribute. For the GET and POST methods (Blocks 210, 212) the processing is the same as that performed in standard HTML (Blocks 226, 228). That is, for the GET method, the identified URL<file> is queried for data

EXHIBIT D
PAGE 41 OF 47

while the POST attribute causes data to be transferred to the URL<file>. The preferred METHOD attributes extending the HTML implementation of the present invention are SQL (Block 214), and PAYMENT (Block 224) attributes. The SQL attribute is preferably not expanded into a SQL command at the client, but rather is expanded by the CGI 28 at server 12 by correlating the data or variable field names in a returned form with the SQL commands stored at the server. This processing is done in a manner described in more detail below. The client program passes the SQL file identifier to the server 12 (Block 230). The processing of the PAYMENT command (Block 232) is discussed in more detail below. The HTML file is scanned for other METHODs (Block 242, 244), and, if one is found, the processing continues by identifying the METHOD (Blocks 210–224). Otherwise (Block 244), the process returns to scan the HTML file for other ACTION or METHOD attributes (Block 160, FIG. 6).

Processing for the INPUT tag is shown in FIG. 9. The process scans the HTML file following the INPUT tag for attributes (Block 250). If no attributes are found (Block 252), the process continues by scanning the HTML file for other tags to process (Block 140, FIG. 5). If an attribute is found and it is a TYPE attribute (Block 254), it is processed (Block 256), and if the attribute is a NAME attribute (Block 258), it is processed (Block 260). Both the TYPE and NAME processing is shown in more detail in FIGS. 10 and 11, respectively. If the attribute is neither a NAME or TYPE attribute, it is a standard attribute for an INPUT tag supported by standard HTML and is processed in a known manner (Block 262). Following processing of the INPUT attribute, the HTML file is scanned for other attributes to process (Block 250).

Processing for the TYPE attribute is shown in FIG. 10. The process first identifies the TYPE attribute for the INPUT tag and then performs the appropriate processing. The new TYPE attributes of the preferred embodiment of the present invention are MSRT1 (Block 270), MSRT2 (Block 272), KEY (Block 274), PIN (Block 276), BCW (Block 278), MICR (Block 280), AMT (Block 282), INT (Block 284), LOCAL (Block 286), and AUTOSUB (Block 288). If the TYPE attribute is not one of these, it is a standard HTML type attribute that is processed in a known manner (Block 310). Each of the new HTML TYPES supported by the present invention causes an I/O operation with a non-standard device. Specifically, these operations are the reading of Track 1 of the magnetic stripe reader (Block 290), the reading of the second track of the magnetic stripe reader (Block 292), the reading of a keypad (Block 294), the reading of an encrypted PIN through a PIN entry device (Block 296), the reading of a bar code through a bar code reader (Block 298), the reading of encoded data on a check through a magnetic check reader (Block 300), the reading of a dollar amount from a keypad through a key input mask (Block 302), the reading of a number from a keypad through a key input mask (Block 304), the reading of data from a local variable (Block 306). and the submission of the data read from one of these devices in a FORM returned to the server 12 (Block 308). The data mask for AMT constrains the dollar amount read to a predetermined number of characters with only two characters following the decimal point. The data mask for INT ensures the number is an integer value within a predetermined range. Processing continues by scanning the HTML file for other TYPE attributes (Block 312) and, if another TYPE attribute is found (Block 314), processing continues by determining the TYPE attribute and performing the appropriate processing. Otherwise, the process returns to scan the HTML file for other attributes (Block 250, FIG. 9).

The NAME attribute processing is performed in accordance with the process shown in FIG. 11. That process examines the NAME attribute to determine if the variable name identified by the attribute is IP_ADDRESS, HOST_PHONE, TID, WORK_KEY, DATETIME, or DEPOSIT_ACCT (Blocks 320, 322, 324, 326, 328, 330). If they are, the INPUT value resulting from one of the INPUTS in a FORM of the HTML file is stored in a local variable identified by the NAME attribute. Following storage (Block 332), the file is scanned for other NAME attributes (Block 328) and, if there are none (Block 332), processing continues by scanning for other attributes for the INPUT tag (Block 250, FIG. 9). If the NAME attribute is a standard HTML INPUT NAME, it is processed by known methods (Block 336). Processing then continues by scanning for other NAME attributes to process (Block 338, 340). Otherwise, the process returns to scan the HTML file for other attributes (Block 250, FIG. 9).

CGI 28 receives Internet protocol statements in a file transmitted from a client program and provides data from those statements to the application(s) implementing system 40 and receives the output of system 40 and provides them to the client program in a file. CGI 28 may be implemented by a program developed by a user using a manual development method as shown in FIG. 24A. That method requires a user to generate a system definition from which a file statement definition for the client and application are developed to implement the transactional or data system. Using the file statement definitions, the user generates the files for the client and database programs which are interpreted by the respective programs to implement transactions or data processing. This process requires the user to not only have knowledge regarding the transaction or data process but specific details of the interaction between the client and database. The user is further required to resolve and correlate all data identifiers in the statements for the client and database environments.

Preferably, CGI 28 is developed with an editor that only requires the user to define the system with statements which are an integration of the protocol statements and the database language. The process implemented by this editor is shown in FIG. 24B. Examples of such integrated statements for files which implement a specific transaction are shown in FIGS. 14 to 23B. The editor verifies the syntax of the integrated statements and correlates the data variables of the protocol statements with the data fields of the database. Following the generation of the integrated statements, the editor segregates the protocol statements from the database language statements. The protocol statements are stored in files which are identified as being for a particular transaction or data process and the database statements are stored in files which are identified as being for a particular transaction or data process on an identified database table. The editor places a database file identifier in the protocol statements which contained embedded database statements. The database file identifiers are used by CGI 28 to select the file for the appropriate transaction so CGI 28 may correlate data variables in the protocol statements with data fields in the database files. The files containing statements to be interpreted by the client program are then downloaded to the appropriate terminals, and the database files containing database language statements are stored on the system executing the CGI 28.

Alternatively, the editor of the present invention may parse integrated statements which are segregated into source code statements for first and second processors, such an editor further includes a compiler to generate executable

code for each processor and, if the processors execute differing source code, a compiler for each source code language. The executable code may then be downloaded to the respective processors for execution.

More specifically, the editor preferably places the database statements for one of the transactions of the preferred embodiment in a file identified by the database name following SQL in FIG. 12. The attributes and tags forming the HTML statements for one of the transactions of the preferred embodiment are placed in a file generally denoted as <html_file>.HTM. The name <html_file> is a name which identifies one of the transactions. Where SQL statements are in the fields of the integrated statements shown in FIGS. 14 to 23B, the string "<html_file>.SQL" is substituted as the database name in the statements of the <html_file>.HTM file. When the CGI executable file is initiated and parses the returning forms, the returned data is placed in the corresponding "<html_file>.SQL" file which is passed to the application program as a command line argument. In this manner, an abbreviated form for the SQL commands may be communicated over the open network between the client and CGI and the CGI may be able to expand those abbreviated SQL commands into the appropriate SQL commands which the application program requires to manipulate the ODBC database.

To effectuate a transaction, for example, an operation at a terminal with non-standard I/O devices may activate a terminal file with a hot key or other action. In processing the activated file, the client program may acquire data which is stored in a local variable or accessible through a non-standard I/O device. This data may then be stored in a FORM and submitted to a server file at a processing system address. The server file activates CGI **28** which retrieves data from the FORM and incorporates it into database statements in the database file for the appropriate transaction and database. If the database statement is a query, the requested data is returned to the CGI in the database file and the CGI places it in the corresponding FORM variables so the server may return the data to the terminal. If the database statement provides data to a database to obtain an authorization, for example, the action performed by the database application in response to the data is placed in the corresponding FORM and returned to the terminal. In this way, data is exchanged between the terminal and the database application. This exchange is supported by CGI **28** even though the server/client communication is performed in an open system protocol, such as HTTP, and the database application is performed in another language, such as SQL. CGI **28** is able to convert and exchange the data between the client and database without the user having to specifically design and implement a conversion program.

The communication paths available for a device implementing the present invention are shown in FIG. 13A. As shown there, an I/O device **420** is coupled through the Worldwide Web open network **426** to an Internet Web server **12**. This connection may be implemented with the preferred extended capability HTML described above. Although HTML files may be encrypted to enhance the security of the document as it is communicated across the Internet, the operator of the system may choose to utilize a more secure physical connection between the device **420** and the Web server **12**. To obtain this alternative connection, the PAYMENT command for the METHOD attribute is preferably used. One form of the PAYMENT command is for a merchant's terminal and the other is for a consumer's terminal. In either terminal, the client program which supports the extended capability HTML operates independently but co-resident in memory with a certified bank card authorization and capture application, which may be provided by a financial institution or a bank card processor.

For the form of the command shown in FIG. 13B, the client program in the merchant terminal suspends its execution and passes the terminal identifier, stored locally, which identifies the merchant's account and the consumer account information read via a magnetic stripe reader or the like, to the bank card application. The bank card application communicates this information via a PSTN **424** or the like to a transaction processor **422**. The processor **422** authorizes or denies the transaction and, if authorized, a printer at the merchant terminal prints a purchase agreement which the consumer may execute to complete the transaction.

In response to a HTML file having a FORM with an ACTION attribute equal to an executable file name for a bank card application program or the like, a METHOD attribute with a field value of PAYMENT, and an INPUT tag with a TYPE attribute of LOCAL_NAME which identifies a deposit only account supplied by a merchant (as shown in FIG. 13C), the client program is suspended and control is transferred to the bank processing application. The bank processing application then uses a modem or ISDN D channel using T3 POS protocol or the like to connect to a secure packet network **424** to connect in a virtual point-to-point manner with a payment processor through a PSTN network or the like. This physical connection provides an additional security element to the encrypted data for the transaction of account information, PIN numbers encrypted by PIN pads provided at the consumer site, and other sensitive information. The bank processor **422** may submit remittance data to the merchant, via the Web or otherwise. After receiving the remittance data, the merchant may ship the product to the consumer. Thus, in this manner, the I/O device may communicate with a plurality of Web servers to "shop" for a best price, delivery date, or other relevant information for selecting a preferred transaction, and then execute the PAYMENT method to utilize a more secure physical communication connection and data security devices to consummate the financial elements of the transaction with less risk and costs for the merchant, consumer, and bank processor.

The preferred integrated HTML/SQL statements which support a card initiated payment authorization and capture transaction are shown in FIG. 14. A first file **500** includes statements which identify the URL database from which the non-standard I/O device seeks authorization for a transaction. The prompts to the operator to enter the account number and amount of the transaction are supported by the INPUT tags which read the second track of the magnetic stripe reader to accept a number of up to 40 characters and assign that information from that track to a variable, and to input the up to 8 characters from the keyboard or the like into a variable called AMOUNT. The INPUT tag with the TYPE attribute of AUTOSUBMIT returns the form to the server for processing in accordance with the method defined in the returned form. As shown in FIG. 14, that METHOD statement causes CGI **28** to incorporate returned data into SQL commands which query the database as to whether the subfield of the track 2 data representing the account number is present in the authorization table of the database. If the data is not present, then a new record is inserted into a table labeled "log_table". The new record consists of the account number and the amount returned in the FORM. Based upon the results of this processing, the application program supplies the data fields to the FORM which will be returned to the client program for printing the transaction record. That

EXHIBIT D
PAGE 43 OF 47

file **510** is shown in FIG. **14**. The ACTION attribute TO PRINTER and the POST METHOD causes the data in the next eight lines to be directed to the printer coupled to the non-standard I/O device for printing the transaction form. The customer may then execute the printed form to complete the transaction. If the transaction is declined or an error is otherwise encountered, the file **520** is used to return a denial to the client program.

In a similar manner, the preferred integrated statements for a bar code order input with card-initiated payment authorization is shown in FIG. **15**. The file **550**, supported by the present invention which implements the transaction request, is again directed to the proper database by the ACTION attribute. The necessary customer information such as name and address may be input through a standard keyboard. The HTML command in the present invention also permits the form to receive the bar code, unit price, and credit card information in a manner similar to that discussed above for the magnetic card reader. Once this information is returned to the server and CGI interface, it is processed by the application program in accordance with the METHOD identified in the returned form. The method of HTML file **550** also creates a database order_table having the information shown in the method. Again, if the transaction is approved, the data for the order and customer acceptance of the order is provided in HTML file **555**, which is directed by the ACTION attribute to the printer at the non-standard I/O device. If the account number is not in the authorization database, the authorization declined or error response is provided in correspondence with the statements in file **560**.

In a similar manner, FIGS. **16–22** show the integrated statements for a transaction request, authorization response, or authorization declined response files for key input order with secure payment transaction (FIG. **16**), a smart card-debit (Type 1) transaction (FIG. **17**A), a smart card debit (Type 2) transaction (FIG. **17**B), a debit card transaction (FIG. **18**), a check verification transaction (FIG. **19**), a customer frequency transaction (FIG. **20**), an item search transaction for which there is no denial (FIG. **21**), retail store end of day reporting (FIG. **22**) and a store reporting an e-mail transaction (FIG. **23**).

While the present invention has been illustrated by the description of a preferred and alternative embodiments and processes, and while the preferred and alternative embodiments and processes have been described in considerable detail, it is not the intention of the applicant to restrict or in any way limit the scope of the appended claims to such detail. Additional advantages and modifications will readily appear to those skilled in the art. For example, rather than expanding HTTP to support non-standard I/O devices, the FTP, POP, SMTP, TELNET or other protocols may be expanded in like manner to couple non-standard I/O devices to the Internet. Similarly, the preferred implementation of the present invention supports a variety of non-standard I/O devices and I/O operations. An Internet protocol may be constructed in accordance with the principles of the present invention to support only selected I/O devices or operations disclosed in the present application. The invention in its broadest aspects is therefore not limited to the specific details, preferred embodiment, and illustrative examples shown and described. Accordingly, departures may be made from such details without departing from the spirit or scope of applicant's general inventive concept.

What is claimed is:

1. A Internet processing system comprising:

a Web server for communicating in an extended Internet protocol; and

a plurality of input/output (I/O) devices coupled to the Web server through the Internet, the I/O devices communicating with the Web server in the extended Internet protocol that supports communication with non-standard I/O devices;

wherein the extended Internet protocol further comprising:

tags for identifying one of the I/O devices and input operation to be performed with the one of the I/O devices;

action attributes for defining the identified device operation to be performed with a local resource for one of the I/O devices; and

method attributes for defining a data transfer method for providing data between the Web server and the I/O devices.

2. A Internet processing system comprising:

a Web server program coupled to the Internet;

a smart card coupled to the Internet; and

a client program for communicating data in an extended Internet protocol between the Web server program and the smart card, the extended Internet protocol including one identifier for the smart card for a transaction and an identifier for an operation to be performed with the identified smart card.

3. A method for processing data over the Internet for a smart card comprising:

coupling a Web server to the Internet;

coupling a smart card to the Internet;

communicating data conforming to an extended Internet protocol between the Web server program and the smart card;

identifying the smart card in a protocol statement conforming to the extended Internet protocol; and

identifying an operation to be performed with the identified smart card.

4. An Internet processing system comprising:

a Web server program coupled to the Internet, the Web server program including a common gateway interface;

a smart card coupled to the Internet;

a client program for communicating data in an extended Internet protocol between the Web server program and the smart card, the client program communicating the data in files having protocol statements conforming to the extended Internet protocol; and

the common gateway interface providing data from the protocol statements conforming to the extended Internet protocol to a transaction system, correlates data in the protocol statements conforming to the extended Internet protocol with data fields in database files for a database coupled to the Web server, and receives data from the transaction system to provide the data to the client program.

5. A method for processing data over the Internet for a smart card comprising:

coupling a Web server program to the Internet;

coupling a smart card to the Internet;

communicating data conforming to an extended Internet protocol between the Web server program and the smart card;

coupling a common gateway interface to the Web server program, the common gateway for communicating data between a database and the Web server program;

providing data from protocol statements conforming to the extended network protocol to a transaction system,

EXHIBIT D
PAGE 44 OF 47

the protocol statements being received in a file from the client program;

receiving data from the transaction system and providing the data to the client program in a file; and

correlating data in extended Internet protocol statements with data fields in database files.

6. A method for communicating between a client program controlling a smart card and a Web server over the Internet comprising:

activating a smart card to assign data obtained by the smart card to a variable name in a file comprised of extended Internet protocol statements; and

sending a file having the assigned data to a Web server to perform a data operation in accordance with the extended Internet protocol statements.

7. A method for communicating between a client program controlling a smart card and a Web server over the Internet comprising:

generating a file comprising extended Internet protocol statements, at least one of which identifies a data operation for obtaining data from a smart card; and

sending the file to a client program controlling the smart card.

8. A system for communicating between a client program controlling a smart card and a Web server over the Internet comprising:

means for generating a file comprising extended Internet protocol statements, at least one of which identifies a data operation for obtaining data from a smart card; and

means for sending the file to a client program controlling the smart card.

9. A Internet processing system comprising:

a Web server program coupled to the Internet;

a telephone coupled to the Internet; and

a client program for communicating data in an extended Internet protocol between the Web server program and the telephone, the extended network protocol including one identifier for the telephone for a transaction and an identifier for an operation to be performed with the identified telephone.

10. A method for processing data over the Internet for a telephone comprising:

coupling a Web server to the Internet;

coupling a telephone to the Internet;

communicating data conforming to an extended Internet protocol between the Web server program and the telephone;

identifying the telephone in a protocol statement conforming to the extended network protocol; and

identifying an operation to be performed with the identified telephone.

11. An Internet processing system comprising:

a Web server program coupled to the Internet, the Web server program including a common gateway interface;

a telephone coupled to the Internet;

a client program for communicating data in an extended Internet protocol between the Web server program and the telephone, the client program communicating the data in files having protocol statements conforming to the extended Internet protocol; and

the common gateway interface provides data from the protocol statements conforming to the extended Internet protocol to a transaction system, correlates data in the protocol statements conforming to the extended Internet protocol with data fields in database files for a database coupled to the Web server, and receives data from the transaction system to provide the data to the client program.

12. A method for processing data over the Internet for a telephone comprising:

coupling a Web server program to the Internet;

coupling a telephone to the Internet;

communicating data conforming to an extended Internet protocol between the Web server program and the telephone;

coupling a common gateway interface to the Web server program, the common gateway for communicating data between a database and the Web server program;

providing data from protocol statements conforming to the extended network protocol to a transaction system, the protocol statements being received in a file from the client program;

receiving data from the transaction system and providing the data to the client program in a file; and

correlating data in extended Internet protocol statements with data fields in database files.

13. A method for communicating between a client program controlling a telephone and a Web server over the Internet comprising:

activating a telephone to assign data obtained by the telephone to a variable name in a file comprised of extended Internet protocol statements; and

sending a file having the assigned data to a Web server to perform a data operation in accordance with the extended Internet protocol statements.

14. A method for communicating between a client program controlling a telephone and a Web server over the Internet comprising:

generating a file comprising extended Internet protocol statements, at least one of which identifies a data operation for obtaining data from a telephone; and

sending the file to a client program controlling the telephone.

15. A system for communicating between a client program controlling a telephone and a Web server over the Internet comprising:

means for generating a file comprising extended Internet protocol statements, at least one of which identifies a data operation for obtaining data from a telephone; and

means for sending the file to a client program controlling the telephone.

16. A Internet processing system comprising:

a Web server program coupled to the Internet;

a personal digital assistant (PDA) coupled to the Internet; and

a client program for communicating data in an extended Internet protocol between the Web server program and the personal digital assistant (PDA), the extended Internet protocol including one identifier for the personal digital assistant (PDA) for a transaction and an identifier for an operation to be performed with the identified personal digital assistant (PDA).

17. A method for processing data over the Internet for a personal digital assistant (PDA) comprising:

coupling a Web server to the Internet;

coupling a personal digital assistant (PDA) to the Internet;

communicating data conforming to an extended Internet protocol between the Web server program and the personal digital assistant (PDA);

EXHIBIT D
PAGE 45 OF 47

identifying the personal digital assistant (PDA) in a protocol statement conforming to the extended Internet protocol; and

identifying an operation to be performed with the identified personal digital assistant (PDA).

18. An Internet processing system comprising:

a Web server program coupled to the Internet, the Web server program including a common gateway interface;

a personal digital assistant (PDA) coupled to the Internet;

a client program for communicating data in an extended Internet protocol between the Web server program and the personal digital assistant (PDA), the client program communicating the data in files having protocol statements conforming to the extended Internet protocol; and

the common gateway interface provides data from the protocol statements conforming to the extended Internet protocol to a transaction system, correlates data in the protocol statements conforming to the extended Internet protocol with data fields in database files for a database coupled to the Web server, and receives data from the transaction system to provide the data to the client program.

19. A method for processing data over the Internet for a personal digital assistant (PDA) comprising:

coupling a Web server program to the Internet;

coupling a personal digital assistant (PDA) to the Internet;

communicating data conforming to an extended Internet protocol between the Web server program and the personal digital assistant (PDA);

coupling a common gateway interface to the Web server program, the common gateway for communicating data between a database and the Web server program;

providing data from protocol statements conforming to the extended Internet protocol to a transaction system, the protocol statements being received in a file from the client program;

receiving data from the transaction system and providing the data to the client program in a file; and

correlating data in extended Internet protocol statements with data fields in database files.

20. A method for communicating between a client program controlling a personal digital assistant (PDA) and a Web server over the Internet comprising:

activating a personal digital assistant (PDA) to assign data obtained by the personal digital assistant (PDA) to a variable name in a file comprised of extended Internet protocol statements; and

sending a file having the assigned data to a Web server to perform a data operation in accordance with the extended Internet protocol statements.

21. A method for communicating between a client program controlling a personal digital assistant (PDA) and a Web server over the Internet comprising:

generating a file comprising extended Internet protocol statements, at least one of which identifies a data operation for obtaining data from a personal digital assistant (PDA); and

sending the file to a client program controlling the personal digital assistant (PDA).

22. A system for communicating between a client program controlling a personal digital assistant (PDA) and a Web server over the Internet comprising:

means for generating a file comprising extended Internet protocol statements, at least one of which identifies a data operation for obtaining data from a personal digital assistant (PDA); and

means for sending the file to a client program controlling the personal digital assistant (PDA).

23. A system for supporting communication between a Web server and a smart card over the Internet comprising:

a Web server that processes extended Internet protocol statements, the Web server being communicatively coupled to the Internet;

a smart card communicatively coupled to the Internet; and

a client program for processing extended Internet protocol statements so that the smart card may communicate with the Web server.

24. The method of claim 23 wherein the extended Internet protocol statements are extended Hyper Text Transport Protocol (HTTP) statements.

25. The method of claim 23 wherein the extended Internet protocol statements are extended Hyper Text Markup Language (HTML) command statements.

26. A client program for processing extended Internet protocol statements so a smart card may communicate with a Web server over the Internet comprising:

means for receiving extended Internet protocol statements over the Internet; and

means for processing the received extended Internet protocol statements to control operations associated with a smart card.

27. The method of claim 26 wherein the extended Internet protocol statements are extended Hyper Text Transport Protocol (HTTP) statements.

28. The method of claim 26 wherein the extended Internet protocol statements are extended Hyper Text Markup Language (HTML) command statements.

29. A method for supporting communication between a Web server and a smart card over the Internet comprising:

processing extended Internet protocol statements at a Web server communicatively coupled to the Internet;

communicatively coupling a smart card to the Internet; and

processing extended Internet protocol statements with a client program so that the smart card may communicate with the Web server.

30. The method of claim 29 wherein the extended Internet protocol statements are extended Hyper Text Transport Protocol (HTTP) statements.

31. The method of claim 29 wherein the extended Internet protocol statements are extended Hyper Text Markup Language (HTML) command statements.

32. A method for processing extended Internet protocol statements so a smart card may communicate with a Web server over the Internet comprising:

receiving extended Internet protocol statements over the Internet; and

processing the received extended Internet statements to control an operation associated with a smart card.

33. The method of claim 32 wherein the extended Internet protocol statements are extended Hyper Text Transport Protocol (HTTP) statements.

34. The method of claim 32 wherein the extended Internet protocol statements are extended Hyper Text Markup Language (HTML) command statements.

35. A system for supporting communication between a Web server and a telephone over the Internet comprising:

a Web server that processes extended Internet protocol statements, the Web server being communicatively coupled to the Internet;

EXHIBIT D
PAGE 46 OF 47

a telephone communicatively coupled to the Internet; and

a client program for processing extended Internet protocol statements so that the telephone may communicate with the Web server.

36. The method of claim 35 wherein the extended Internet protocol statements are extended Hyper Text Transport Protocol (HTTP) statements.

37. The method of claim 35 wherein the extended Internet protocol statements are extended Hyper Text Markup Language (HTML) command statements.

38. A client program for processing extended Internet protocol statements so a telephone may communicate with a Web server over the Internet comprising:

means for receiving extended Internet protocol statements over the Internet; and

means for processing the received extended Internet protocol statements to control operations associated with a telephone.

39. The method of claim 38 wherein the extended Internet protocol statements are extended Hyper Text Transport Protocol (HTTP) statements.

40. The method of claim 38 wherein the extended Internet protocol statements are extended Hyper Text Markup Language (HTML) command statements.

41. A method for supporting communication between a Web server and a telephone over the Internet comprising:

processing extended Internet protocol statements at a Web server communicatively coupled to the Internet;

communicatively coupling a telephone to the Internet; and

processing extended Internet protocol statements with a client program so that the telephone may communicate with the Web server.

42. A method for processing extended Internet protocol statements so a telephone may communicate with a Web server over the Internet comprising:

receiving extended Internet protocol statements over the Internet; and

processing the received extended Internet statements to control an operation associated with a telephone.

43. The method of claim 42 wherein the extended Internet protocol statements are extended Hyper Text Transport Protocol (HTTP) statements.

44. The method of claim 42 wherein the extended Internet protocol statements are extended Hyper Text Markup Language (HTML) command statements.

45. A system for supporting communication between a Web server and a personal digital assistant (PDA) over the Internet comprising:

a Web server that processes extended Internet protocol statements, the Web server being communicatively coupled to the Internet;

a personal digital assistant (PDA) communicatively coupled to the Internet; and

a client program for processing extended Internet protocol statements so that the personal digital assistant (PDA) may communicate with the Web server.

46. The method of claim 45 wherein the extended Internet protocol statements are extended Hyper Text Transport Protocol (HTTP) statements.

47. The method of claim 45 wherein the extended Internet protocol statements are extended Hyper Text Markup Language (HTML) command statements.

48. A client program for processing extended Internet protocol statements so a personal digital assistant (PDA) may communicate with a Web server over the Internet comprising:

means for receiving extended Internet protocol statements over the Internet; and

means for processing the received extended Internet protocol statements to control operations associated with a personal digital assistant (PDA).

49. The method of claim 48 wherein the extended Internet protocol statements are extended Hyper Text Transport Protocol (HTTP) statements.

50. The method of claim 48 wherein the extended Internet protocol statements are extended Hyper Text Markup Language (HTML) command statements.

51. A method for supporting communication between a Web server and a personal digital assistant (PDA) over the Internet comprising:

processing extended Internet protocol statements at a Web server communicatively coupled to the Internet;

communicatively coupling a personal digital assistant (PDA) to the Internet; and

processing extended Internet protocol statements with a client program so that the personal digital assistant (PDA) may communicate with the Web server.

52. A method for processing extended Internet protocol statements so a personal digital assistant (PDA) may communicate with a Web server over the Internet comprising:

receiving extended Internet protocol statements over the Internet; and

processing the received extended Internet statements to control an operation associated with a personal digital assistant (PDA).

53. The method of claim 52 wherein the extended Internet protocol statements are extended Hyper Text Transport Protocol (HTTP) statements.

54. The method of claim 52 wherein the extended Internet protocol statements are extended Hyper Text Markup Language (HTML) command statement.

* * * * *

EXHIBIT D
PAGE 47 OF 47

EXHIBIT / ATTACHMENT

E

(To be scanned in place of tab)

US006694387B2

(12) **United States Patent**
Wagner

(10) **Patent No.: US 6,694,387 B2**
(45) **Date of Patent: *Feb. 17, 2004**

(54) **SYSTEM FOR ENABLING SMART CARD TRANSACTIONS TO OCCUR OVER THE INTERNET AND ASSOCIATED METHOD**

(75) Inventor: **Richard Hiers Wagner**, Dunwoody, GA (US)

(73) Assignee: **Datascape, Inc.**, Atlanta, GA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 62 days.
This patent is subject to a terminal disclaimer.

(21) Appl. No.: **10/100,347**

(22) Filed: **Mar. 18, 2002**

(65) **Prior Publication Data**

US 2002/0138430 A1 Sep. 26, 2002

Related U.S. Application Data

(63) Continuation of application No. 09/907,076, filed on Jul. 17, 2001, which is a continuation of application No. 09/314,266, filed on May 18, 1999, now Pat. No. 6,366,967, which is a continuation of application No. 08/995,123, filed on Dec. 19, 1997, now Pat. No. 5,905,908, which is a continuation of application No. 08/493,772, filed on Jun. 22, 1995, now Pat. No. 5,742,845.

(51) **Int. Cl.**[7] **G06F 13/00**; G06F 13/20

(52) **U.S. Cl.** **710/33**; 710/11; 710/20; 709/203; 709/227; 709/228; 705/26; 370/401

(58) **Field of Search** 710/11, 20, 33; 709/203, 227, 228; 705/26; 370/401

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,668,653	A	6/1972	Fair et al.
4,410,962	A	10/1983	Daniels et al.
4,438,511	A	3/1984	Baran
4,577,152	A	3/1986	Macovski
4,620,153	A	10/1986	Hino
4,625,171	A	11/1986	Sekihara et al.
4,625,276	A	11/1986	Benton et al.
4,707,592	A	11/1987	Ware
4,724,521	A	2/1988	Carron et al.
RE32,701	E	6/1988	Moran
4,755,940	A	7/1988	Brachtl et al.
4,774,655	A	9/1988	Kollin et al.
4,787,035	A	11/1988	Bourne
4,799,156	A	1/1989	Shavit et al.
4,815,029	A	3/1989	Barker et al.

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

WO	WO 95/16971	6/1995
WO	WO 96/41286	12/1996
WO	WO 97/01137	1/1997

OTHER PUBLICATIONS

Riordan, T., "A Toll Collector on the Information Highway", *The New York Times*, Sunday, Aug. 22, 1993, (3 pages).
Burchard et al., "Compount Documents in HTML", Internet Draft Memo, Date Unknown, (14 pages).

(List continued on next page.)

Primary Examiner—Rehana Perveen
(74) *Attorney, Agent, or Firm*—Maginot, Moore & Beck; David M. Lockman

(57) **ABSTRACT**

An open network system for supporting input/output (I/O) operations for non-standard I/O devices are disclosed. The system includes a server coupled to a plurality of I/O devices through an open network and an extended open system protocol that supports communication with devices that are not personal computers (PCs). These devices include magnetic stripe readers, check readers, smart card readers, credit card terminals, screen phone terminals, PIN pads, printers, and the like. The extended open network protocol includes tags which identify device and input operations and attributes which identify the location, data exchange method, and data variable names for the retrieval, acquisition, and submission of data between the server and I/O devices. Preferably, the open network protocol is implemented in a Hyper Text Transport Protocol (HTTP).

8 Claims, 25 Drawing Sheets

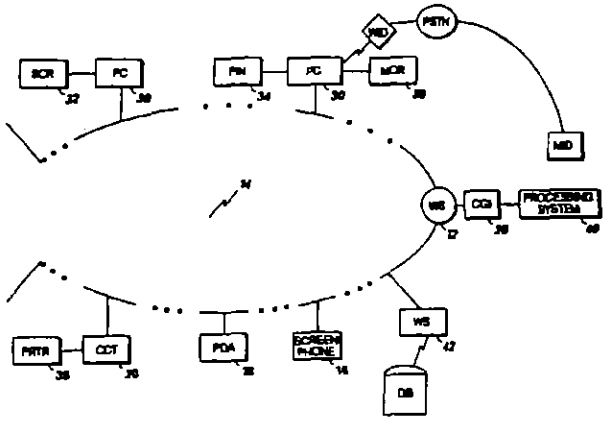

EXHIBIT E
PAGE 1 OF 43

U.S. PATENT DOCUMENTS

4,847,604	A	7/1989	Doyle
4,885,777	A	12/1989	Takaragi et al.
4,901,223	A	2/1990	Rhyne
4,912,309	A	3/1990	Danielson et al.
4,926,325	A	5/1990	Benton et al.
4,942,532	A	7/1990	Mori
4,942,552	A	7/1990	Merrill et al.
4,947,028	A	8/1990	Gorog
4,949,248	A	8/1990	Caro
4,962,531	A	10/1990	Sipman et al.
4,965,742	A	10/1990	Skeirik
4,972,463	A	11/1990	Danielson et al.
5,014,221	A	5/1991	Mogul
5,018,196	A	5/1991	Takaragi et al.
5,021,949	A	6/1991	Morten et al.
5,025,373	A	6/1991	Keyser, Jr. et al.
5,036,461	A	7/1991	Elliott et al.
5,036,484	A	7/1991	McCoy et al.
5,046,094	A	9/1991	Kawamura et al.
5,047,923	A	9/1991	Elstner et al.
5,083,262	A	1/1992	Haff, Jr.
5,103,478	A	4/1992	Matyas et al.
5,113,517	A	5/1992	Beard et al.
5,119,465	A	6/1992	Jack et al.
5,146,499	A	9/1992	Geffrotin
5,146,553	A	9/1992	Noguchi et al.
5,148,481	A	9/1992	Abraham et al.
5,149,945	A	9/1992	Johnson et al.
5,161,222	A	11/1992	Montejo et al.
5,179,690	A	1/1993	Ishikawa
5,195,130	A	3/1993	Weiss et al.
5,202,828	A	4/1993	Vertelney et al.
5,204,947	A	4/1993	Bernstein et al.
5,206,951	A	4/1993	Khoyi et al.
5,212,369	A	5/1993	Karlisch et al.
5,220,380	A	6/1993	Hirata et al.
5,220,501	A	6/1993	Lawlor et al.
5,239,662	A	8/1993	Danielson et al.
5,243,284	A	9/1993	Noll
5,256,863	A	10/1993	Ferguson et al.
5,272,754	A	12/1993	Boerbert
5,274,821	A	12/1993	Rouquie
5,276,312	A	1/1994	McCarthy
5,277,192	A	1/1994	Cumoulin
5,278,955	A	1/1994	Forte et al.
5,278,972	A	1/1994	Baker et al.
5,294,782	A	3/1994	Kumar
5,297,249	A	3/1994	Bernstein et al.
5,307,499	A	4/1994	Yin
5,309,351	A	5/1994	McCain et al.
5,317,411	A	5/1994	Yoshida
5,321,806	A	6/1994	Meinerth et al.
5,321,808	A	6/1994	Rupp
5,347,580	A	9/1994	Molva et al.
5,347,632	A	9/1994	Filepp et al.
5,349,678	A	9/1994	Morris et al.
5,351,296	A	9/1994	Sullivan
5,367,572	A	11/1994	Weiss
5,367,635	A	11/1994	Bauer et al.
5,373,561	A	12/1994	Haber et al.
5,377,191	A	12/1994	Farrell et al.
5,379,344	A	1/1995	Larsson et al.
5,386,517	A	1/1995	Sheth et al.
5,390,314	A	2/1995	Swanson
5,418,908	A	5/1995	Keller et al.
5,461,217	A	10/1995	Claus
5,475,585	A	12/1995	Bush
5,475,858	A	12/1995	Gupta et al.
5,497,487	A	3/1996	Fortier
5,499,343	A	3/1996	Pettus
5,510,710	A	4/1996	Nauerth
5,511,122	A	4/1996	Atkinson
5,517,569	A	5/1996	Clark
5,521,980	A	5/1996	Brands
5,526,037	A	6/1996	Cortjens et al.
5,526,409	A	6/1996	Conrow et al.
5,530,844	A	6/1996	Phillips et al.
5,544,161	A	8/1996	Bigham et al.
5,544,246	A	8/1996	Mandelbaum et al.
5,544,320	A	8/1996	Konrad
5,546,582	A	8/1996	Brockmeyer et al.
5,548,721	A	8/1996	Denslow
5,550,561	A	8/1996	Ziarno
5,550,984	A	8/1996	Gelb
5,552,586	A	9/1996	Kalman
5,557,203	A	9/1996	Nauerth
5,563,878	A	10/1996	Blakeley et al.
5,570,465	A	10/1996	Tsakanikas
5,577,233	A	11/1996	Goettelmann et al.
5,581,478	A	12/1996	Cruse et al.
5,581,686	A	12/1996	Koppolu et al.
5,581,708	A	12/1996	Iijima
5,583,940	A	12/1996	Vidrascu et al.
5,586,036	A	12/1996	Pintsov
5,586,175	A	12/1996	Hogan et al.
5,590,038	A	12/1996	Pitroda
5,590,181	A	12/1996	Hogan et al.
5,590,196	A	12/1996	Moreau
5,590,197	A	12/1996	Chen et al.
5,590,199	A	12/1996	Krajewski, Jr. et al.
5,594,910	A	1/1997	Filepp et al.
5,602,905	A	2/1997	Mettke
5,602,915	A	2/1997	Campana et al.
5,602,918	A	2/1997	Chen et al.
5,604,802	A	2/1997	Holloway
5,606,493	A	2/1997	Duscher et al.
5,613,012	A	3/1997	Hoffman et al.
5,613,096	A	3/1997	Danknick
5,615,251	A	3/1997	Hogan et al.
5,623,547	A	4/1997	Jones et al.
5,630,101	A	5/1997	Sieffert
5,633,919	A	5/1997	Hogan et al.
5,638,430	A	6/1997	Hogan et al.
5,640,193	A	6/1997	Wellner
5,642,485	A	6/1997	Deaton et al.
5,649,114	A	7/1997	Deaton et al.
5,649,118	A	7/1997	Carlisle et al.
5,652,876	A	7/1997	Ashe et al.
5,655,008	A	8/1997	Futch et al.
5,664,110	A	9/1997	Green et al.
5,666,412	A	9/1997	Handelman et al.
5,668,878	A	9/1997	Brands
5,677,953	A	10/1997	Dolphin
5,677,955	A	10/1997	Doggett et al.
5,678,002	A	10/1997	Fawcett et al.
5,679,945	A	10/1997	Renner et al.
5,687,322	A	11/1997	Deaton et al.
5,689,799	A	11/1997	Dougherty et al.
5,694,546	A	12/1997	Reisman
5,696,903	A	12/1997	Mahany
5,703,795	A	12/1997	Mankovitz
5,706,273	A	1/1998	Guerreri
5,708,780	A	1/1998	Levergood et al.
5,708,782	A	1/1998	Larson et al.
5,708,825	A	1/1998	Sotomayor
5,710,884	A	1/1998	Dedrick
5,715,314	A	2/1998	Payne et al.
5,722,754	A	3/1998	Langner
5,724,506	A	3/1998	Cleron et al.
5,729,549	A	3/1998	Kostreski et al.
5,732,219	A	3/1998	Blumer et al.

EXHIBIT E
PAGE 2 OF 43

5,734,720	A	3/1998	Salganicoff
5,742,845	A	4/1998	Wagner
5,742,931	A	4/1998	Spiegelhoff et al.
5,748,908	A	5/1998	Yu
5,754,655	A	5/1998	Hughes et al.
5,756,978	A	5/1998	Soltesz et al.
5,758,257	A	5/1998	Herz et al.
5,778,067	A	7/1998	Jones et al.
5,781,189	A	7/1998	Holleran et al.
5,799,156	A	8/1998	Hogan et al.
5,805,719	A	9/1998	Pare, Jr. et al.
5,805,807	A	9/1998	Hanson et al.
5,809,415	A	9/1998	Rossmann
5,815,577	A	9/1998	Clark
5,815,657	A	9/1998	Williams et al.
5,819,092	A	10/1998	Ferguson et al.
5,826,245	A	10/1998	Sandberg-Diment
5,838,906	A	11/1998	Doyle et al.
5,850,442	A	12/1998	Muftic
5,851,149	A	12/1998	Xidos et al.
5,854,833	A	12/1998	Hogan et al.
5,867,821	A	2/1999	Ballantyne et al.
5,870,552	A	2/1999	Dozier et al.
5,870,724	A	2/1999	Lawlor et al.
5,873,099	A	2/1999	Hogan et al.
5,884,292	A	3/1999	Baker et al.
5,889,863	A	3/1999	Weber
5,931,917	A	8/1999	Nguyen et al.
5,933,816	A	8/1999	Zeanah et al.
RE36,310	E	9/1999	Bjerrum et al.
5,953,731	A	9/1999	Glaser
5,960,216	A	9/1999	Vishlitzky et al.
5,963,205	A	10/1999	Sotomayor
5,973,731	A	10/1999	Schwab
5,978,569	A	11/1999	Traeger
5,978,773	A	11/1999	Hudetz et al.
5,982,891	A	11/1999	Ginter et al.
5,983,004	A	11/1999	Shaw et al.
5,987,312	A	11/1999	Dekker
5,996,076	A	11/1999	Rowney et al.
6,085,224	A	7/2000	Wagner
6,112,191	A	8/2000	Burke
6,175,922	B1	1/2001	Wang
6,182,052	B1	1/2001	Fulton et al.
6,205,437	B1	3/2001	Gifford
6,208,904	B1	3/2001	Mullen, Jr.
6,212,575	B1	4/2001	Cleron et al.
6,226,744	B1	5/2001	Murphy et al.
6,226,752	B1	5/2001	Gupta et al.
6,253,027	B1	6/2001	Weber et al.
6,275,869	B1	8/2001	Sieffert et al.
6,279,112	B1	8/2001	O'Toole, Jr. et al.
6,282,656	B1 *	8/2001	Wang
6,283,375	B1	9/2001	Wilz, Sr. et al.

OTHER PUBLICATIONS

Press Release, "Netscape Introduces Netscape Navigator 2.0", Netscape Company Press Relations, Sep. 18, 1995, (3 pages).

Stabell, B., The EMBED Tage by Netscape (v2.0), Sep. 19, 1995, (9 pages).

Article entitled, "Eolas Sues Microsoft for Infrigement of Patent for Fundamental Web Browser Technology That Makes "Plug–Ins" and "Applets" Possible", Chicago, Illinois, Feb. 2, 1999, (2 pages).

Sottong, S., "Quick HTML Basics", Date Unknown, (12 pages).

Ang et al., "Integrated Control of Distributed Volume Visualization Through the World–Wide–Web", Oct. 30, 2001, (8 pages).

Krajewski, Jr., et al., "Applicability of Smart Cards to Network User Authentication", *Computing Systems,* vol. 7, No. 1, Winter 1994, (15 pages).

"Financial Institution Sign–On Authentication for Wholesale Financial Transactions", American National Standards Institute, X9.26, Feb. 28, 1990 (35 pages).

Kohl et al., "The Evoluation of Kerberos Authentication Service", Date Uknown, (15 pages).

Cordonnier et al., "Time as an Aid to Improving Security in Smart Cards", Date Unknown, (14 pages).

Cordonnier et al., "The Concept of Suspicion: A New Security Model for Identification Management in Smart Cards", Date Unknown, (14 pages).

Haykin et al., "Computer Science and Technology—Smart Card Technology: New Methods for Computer Access Control", U.S. Department of Commerce, National Technical Information Service, Sep. 1998, (55 pages).

Rescorla et al., "The Secure HyperText Transfer Protocol", Internet Draft Memo, Dec. 1994, (29 pages).

Takei, D., "Videotex Information System and Credit System Connecting with MARS–301 of JNR", *Japanese Railway Engineering,* No. 95, Sep. 1985, (5 pages).

Pongratz et al., "IC Cards in Videotex Systems", *Smart Card 2000: The Future of IC Cards,* International Conference, Oct. 19–20, 1987, (9 pages).

Neuman, B.C., "Security, Payment, and Privacy for Network Commerce", *IEEE Journal on Selected Areas in Communication,* vol. 13, No. 8, Oct. 1995, (8 pages).

Zilles, S.N., "Catalog–Based Order Entry System", IBM Technical Disclosure Bulletin, vol. 25, No. 11B, Apr. 1983, (3 pages).

Article entitled, "PKCS #11: New Member of Public Key Cryptography", available online at http://www.rsasecurity.com/news/pr/940112–1.html., Jan. 12, 1994, (12 pages).

Kessler, J., "The French Minitel: Is There Life Outside of the 'US ASCII' Internet? A Challenge of Convergence", *D–Lib Magazine,* Dec. 1995, (10 pages).

Schöter et al., "Digital Money Online, A Review of Some Existing Technologies", Intertrader Ltd., ©Feb. 1997, (60 pages).

Sutherland, E., "The Quickening Pace of Technology", ©1995, (3 pages).

Article entitled "NetBill: An Internet Commerce System Optimized for Network Delivered Services", Date Unknown, (9 pages).

Reagle, Jr., et al., Expanded bibliography for "Papers Describing the Operation of Electronic Commerce Protocols", Date Unknown, (3 pages).

Article entitled "ISO/IEC 7816 part 4: Interindustry Command for Interchange", Nov. 26, 1998, (77 pages).*

1991 Statement on Role of NIST, available online at http://csrc.nist.gov/publications/nistbul/cs191–02.txt, Jun. 17, 1992, (38 pages).*

Kirkman, P., "Electronic Funds Transfer Systems: The Revolution in Cashless Banking and Payment Methods", ©1987, (14 pages).*

Fancher, C., "Smart Cards: As Potential Applications Grow, Computers in the Wallet are Making Unobtrusive Inroads", *Scientific American,* Date Unknown, (10 pages).

Walsh, E., "Smart Card Standards", Datapro Information Services, Aug. 25, 2000, (10 pages).

EXHIBIT 15
PAGE 3 OF 43

Article entitled "The Birth of Smart Cards: 1980", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1981", Le Courier de la Monétique, Date Unknown, (3 pages).
Article entitled "The Birth of Smart Cards: 1982", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1983", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1984", Le Courier de la Monétique, Date Unknown, (3 pages).
Article entitled "The Birth of Smart Cards: 1985", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1986", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1987", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1988", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1989", Le Courier de la Monétique, Date Unknown, (2 pages).
Doyle, M., University of California, San Diego, Office of Continuing Medical Education, Disclosure Statement, Nov. 30, 1993, (10 pages).
Article entitled "Workshop on Network and Distributed System Security", Privacy and Research Group, Feb. 11–12, 1993, (5 pages).
Krajewski, Jr., M., "Concept for a Smart Card Kerberos", Jan. 22, 2002, (8 pages).
Kohl et al., "The Kerberos Network Authentication Service (V5)", Sep. 1993, (103 pages).
Nebel, E., "File Upload in HTML Forms", Sep. 23, 1994, (4 pages).
Raggett, D., E-mail memo entitled "Re: Request for New Forms Submission Consensus", Oct. 12, 1993, (4 pages).
Gaskell et al., "Improved Security for Smart Card Use in DCE", Feb. 1995, (17 pages).
Pato, J., "Using Pre-Authentication to Avoid Passord Guessing Attacks", Jun. 1993, (5 pages).
Shipplett, M., E-mail memo entitled "Re: Insecure WWW Access Authorization Protocol?", Mar. 7, 1994, (3 pages).
"Financial Transaction Card Originated Messages—Interchange Message Specifications", International Standard, Second Edition, Dec. 15, 1993, (128 pages).
Various articles on the subject "1994 NIST Guideline for Advanced Authentication (Smart Cards)", Date Unknow, (73 pages).
Allen et al., "Smart Cards: Seizing Strategic Business Opportunities", The Smart Card Forum, ©1997 The McGraw-Hill Companies, (323 pages).
Fielding et al., "Hypertext Transfer Protocol—HTTP/1.1", Jan. 1997, (162 pages).
Branwyn, G., "M•O•S•A•I•C Quick Tour for Windows: Accessing & Navigating the Internet's World Wide Web", ©1994 Ventana Press, Inc., (215 pages).
Shelly et al., "Netscape Navigator: An Introduction", ©1996 Boyd & Fraser Publishing Company, (138 pages).
James, P., "Official Netscape Navigator 2.0 Book: Windows Edition", ©1996, Ventana Press, Inc., (692 pages).
Article entitled "A Smart Card for On-Line Home Services", ChipCord News, Issue No. 32, Jul. 1990, (2 pages).
Bosak et al., "XML and the Second Generation Web", *Scientific American*, May 1999, (5 pages).
Beners-Lee et al., "RFC1866-Hypertext Markup Language-2.0", Nov. 1995, (58 pages).
"Europay Members Forum", Europay International, May 8, 2001, (3 pages).
"NCR 2170 Hospitality Workstation Designed for Bars, Restaurants", AT&T News Release, Feb. 2, 1993, (2 pages).
"AT&T to Increase Smart Card Reliability, Expand Memory to 8KB", AT&T News Release, Apr. 8, 1994, (2 pages).
"AT&T and NCR Offer Smart Card Upgrade Kits for Bank ATMs", AT&T News Release, Nov. 17, 1993, (2 pages).
"The Smart Card Cashes In", The Economist, Jan. 29, 1994, (2 pages).
"NCR 7450 Terminal Brings ATM-style Interface to Point-of-Sale", AT&T News Release, Oct. 11, 1993, (4 pages).
"Mastercard Smart Card Milestones", MasterCard International Press Office, Mar. 5, 2001, (3 pages).
Svigals, Jerome, "Microchip Smart Cards Will Save Money", American Banker, Feb. 2, 1994, (1 pages).
Barthel, Matt, "Fears of Crime at ATMs are on the Rise", American Banker, Mar. 30, 1994, (1 pages).
"Smart Card Publication List", Smart Card Forum, Mar. 2, 2001, (7 pages).
"Appendix: Major Electronic Purse Projects, By Country", pp. 35–50 and 52–64, Date Unknown (29 pages).
O'Kelly, Robin, "Mondex 'Global Electronic Cash' Wins Helsinki Award", Sep. 8, 1996, (2 pages).
"White Paper—The Euro in the Electronic Purse", Apr. 2000, (5 pages).
"History of Gemplus", Gemplus, Mar. 2, 2001, (3 pages).
Pauli, Ralf and Koponen, Risto, "Toward Electronic Money", Bank of Finland Bulletin, Apr. 1997, (3 pages).
"A Frequently Asked Questions List (FAQ) for alt.technology.smartcards", Estonian Institute of Cybernetics, Jul. 4, 1995, (12 pages).
Husemann, Dirk, "The Smart Card: Don't Leave Home Without It", Institute of Electrical and Electronics Engineers, Inc., Sep. 18, 2000, (9 pages).
"Report to the Council of The European Monetary Institute on Prepaid Cards", The Working Group on EU Payment Systems, May 1994, (12 pages).
Banerjee, Ram, "Smart Card Standards and Electronic Purse, A Review Paper", Aston University, May 1997, (20 pages).
Townend, Robin C., "Finance: History, Development & Market Overview", The International Smart Card Industry Guide, 1995–1996, (23 pages).
Monod, Elsbeth Marikje, "Case Study: The Multi-Functional Smart Carte Sante", The International Smart Card Industry Guide, 1995–1996, (8 pages).
Smith, Jack, "Smart Card Industry Review 1995", Smart Card Monthly Newsletter, Jan. 1996, (6 pages).
Crotch-Harvey, Trevor, "Smart Cards in Telecoms", The International Smart Card Industry Guide, 1995–1996, (14 pages).
Farmer, Donna and Tonnac, Remy de, "What's So Smart About Smart Cards", Gemplus, Oct. 10, 2000, (1 pages).
"P401 Definition of a Pan-European IC Card",Eurescom, Mar. 5, 2001, (1 pages).
"Innovonics, Inc. Selects AMP as Manufacturer for the PC Pay Module", Innovonics News Release, May 19, 1995, (3 pages).
Levy, Steven, "E-Money (That's What I Want)", Wired, Dec., 1994, (2 pages).

EXHIBIT E
PAGE 4 OF 43

Fickenscher, Lisa, "Banc One Gears Up to Go National With Its Microchip Smart Cards", American Banker, Jan. 27, 1994 (2 pages).
Strachman, Daniel and Kutler, Jeffrey, "Conventional Goers See Bank Future in Smart Cards", American Banker, Nov. 30, 1993, (1 pages).
Meece, Mickey, "First Data Strengthens Position in Card Processing", American Banker, Dec. 21, 1993, (1 page).
Kutler, Jeffrey, "Prepaid Cards, Even in Concept Stage, Sparking Interest Among Consumers", American Banker, Apr. 27, 1994, (1 pages).
Kutler, Jeffrey, "Visa Starts Push for Smart Card to Replace Cash", American Banker, Mar. 22, 1994, (2 pages).
"AT&T Introduces Enhanced Multimedia Interface for ATM Networks", AT&T News Release, May 4, 1994 (2 pages).
"Trendar and AT&T Test Contactless Smart Cards at Truck Stops", AT&T News Release, Feb. 26, 1993, (3 pages).
"NCR Announces 5908 Signature Capture System for Retailers", AT&T News Release, Sep. 30, 1991 (2 pages).
Merckling, Roger and Anderson, Anne, "Smart Card Introduction", Mar. 1994,(16 pages).
Merckling, Roger and Anderson, Anne, "DCE Smart Card Integration", Mar. 1994, (16 pages).
"Chapter 2—Technology", Smart Card Primer, The Schuler Consultancy, ©1993, pp. 13–24 and 26–29, (16 pages).
"CP8® Products, PASCAM, Network Security Processor", Jan. 1994, (4 pages).
"Notes on Open Networks and Device Addressing", IBM Corporation, Apr. 25, 2001, (19 pages).
Chakrabarti, Samdih, Knight, James and Thies, Bill, "A Centralized Smart Card System Utilizing Distributed Proxies", 2000, University of Leeds, (16 pages).
"Single–Issuer Multiple Uses", Smart Card Primer, The Schuler Consultancy, ©1993, (3 pages).
Sazegar, Hossein, "The Impact of Smart Cards in the Global Payphone Business", Schlumberger Technologies, ©1992, (11 pages).
Derksen, Eduard, "Money–Bit: The Open Off–Host Prepaid Card System", Ascom Autelca, Ltd., Date Unknown, (10 pages).
Corzo, Gerardo, "Credisa's Elite Card System", Date Unknown, (5 pages).
Cordonnie, Vincent M., "Assessing the Future of Smart Cards", Cartech, Date Unknown, (7 pages).
"NatWest Launch Mondex Global Electronic Cash", Smart Card News, Ltd., Dec. 1993, (7 pages).
Merkert, Sr., Robert J., "Stored Value Cards and The Electronic Purse: Developments in the Americas", Danyl Corporation, ©1995, (8 pages).
"CP8® Products, PinPadLINK, Readers/Encoders", Dactyl France, Jun. 1993, (2 pages).
"Smart Security", Byte Talk, Feb., 1994, (2 pages).
Verschuren, Ton, "Smart Access: Strong Authentication on the Web", Apr. 24, 2001, (10 pages).
"PKCS#11: New Member of Public Key Cyrptography", RSA Data Security, Inc., Jan. 12, 1994, (2 pages).
Petri, Steve, "An Introduction to Smartcards", Litronic, Inc., #1998, (2 pages).
Violino, Bob, "The Cashless Society", InformationWeek, Oct. 11, 1993, (5 pages).
Merkert, Sr., Robert J., 'Keynote Presentation for the AIC Conference, Auckland, New Zealand', Danyl Corporation, ©1995, (4 pages).
"Videoway's Experience Shows a New Approach to the Interactive Opportunity", Inside Cable and Telecom Services, Jun. 1995, (3 pages).
Bull CP8® News, Products and Technologies, No. 22, Oct. 1994, (2 pages).
Bull CP8® Systems Product Sheet, Sep. 1994, (2 pages).
"One Store's Comprehensive Loyalty Program—The Takashimaya Experience", Verifone, Inc., ©1994, (4 pages).
"Mondex Payments Scheme", Smart Card News Ltd., Dec. 1993, (2 pages).
"Why Consumers Are Still Carrying Cash Everywhere", Philadelphia Inquirer, Jun. 11, 1995, (2 pages).
"Electronic Payment System Architecture for Point of Sale Application", IBM Technical Disclosure Bulletin, Jan. 1992, (1 pages).
"Portable Self Checkout Retail System", IBM Technical Disclosure Bulletin, Jun. 1992, (3 pages).
"Transaction Completion Code Based on Digital Signatures", IBM Technical Dislcosure Bulletin, Aug. 1985, (6 pages).
"Chapter 4—Uses", Smart Card Primer, The Schuler Consultancy, ©1993, pp. 57–66, 80–83, and 87–91, (19 pages).
Figure 8.5 from page 151 of Zoreda & Oton "Smart Cards", Artech House, ©1994, (1 pages).
"Internet Music Library Process", Date Unknown, (1 pages).
Bibliography for "CITM Working Paper WP–95–1005: Designing Electronic Catalogs for Business Value: Results from the CommerceNet Pilot", Date Unknown, (35 pages).
Van Hove, Leo, Bibliography on Electronic Purses, Date Unknown, (4 pages).
Bibliography for "CITM Working Paper WP–95–1006: Financial EDI Over the Internet: A Case Study", Date Unknown, (5 pages).
Bibliography for Hawkes, P.L., Davies, Donald. W., and Price, W.L., "Integrated Circuit Cards, Tags, and Tokens: New Technology and Applications", Oxford; Boston: BSP Professional, ©1990, (1 pages).
Bibliography for Devargas, M., "Smart Cards and Memory Cards", Oxford; Blackwell NCC, ©1992, (1 pages).
Bibliography for Svigals, Jerome, "Smart Cards: The New Bank Cards", New York: Macmillan; London: Collier Macmillan Publishers, ©1987, (1 pages).
Bibliography for Bright, Roy, "Smart Cards: Principles, Practice, Applications", Chichester [England]: Ellis Horwood: New York: Halsted Press, ©1988, (1 pages).
Bibliography for Svigals, Jerome, "Smart Cards: The Ultimate Personal Computer", New York: Macmillan Publishing Co.; London: Collier Macmillan, ©1985, (1 pages).
Bibliography for "Applications of Computer Card Technology", Washington, D.C., U.S. Dept. of Labor, Office of the Inspector General, [1990], (1 pages).
German article entitled "Erfahrungen Mit Einer Hochsicheren, Multifunktionalen Chipkarte fur Bankenanwendungen" Date Unknown, (21 pages).
Document entitled: "Dialog Search for U.S. Patent Number 5,905,908", Date Unknown, (59 pages).
UBI Consortium Inc. Document Mar. 26, 2001 (2 pages).
HTML file entitled: "The Common Gateway Interface", available at httpd@ncsa.uiuc.edu, circa, Jun. 1995.
HTML file entitled: "Critique of Secure–HTTP", available at shttp://www.spyglass.com/secure, Apr. 19, 1995.
Directory of abstracts of HTML files available from IEFT Secretariat—Corporation for National Research Initiatives; circa, Jun. 1995.

EXHIBIT E
PAGE 5 OF 43

HTML file entitled: "A Beginner's Guide to HTML" available from National Center for Supercomputing Applications/pubs@ncsa.uiuc.edu; circa, Jun. 1995.

Memo entitled: "Uniform Resource Locators (URL)", by URI working group of the Internet Engineering Task Force; Dec. 1994.

Press release entitled: "VeriFone to acquire Enterprise Integration Technologies (EIT), Internet Commerce Software and Services Leader", VeriFone, Inc. and Enterprise Integration Technologies; Redwood, California; Aug. 21, 1995.

Spyglass white paper entitled: "Electronic Commerce Standards for the WWW"; Spyglass, Inc. http://spyglass.com; Aug. 21, 1995.

Memo entitled: "The Secure Hypertext Transfer Protocol"; E. Rescorla and A. Schiffman; Jul. 1995.

Article entitled: "Cash on the Wirehead", Andrew Singleton, BYTE Magazine, Jun. 1995.

"Beyond the Web: Excavating the Real World Via Mosaic", Goldberg et al., 2nd International WWW Conference.

"Magnetic Resonance Angiography", Radiology 1986; 161, p. 717–720, Dumoulin et al.

"Velocity and Diffusion Imaging in Dynamic NMR Microscopy", JMR 91, p. 326–352, 1991, Callaghan et al.

"Reduced Circular Field–of–View Imaging", MRM 40:474–480, 1998, Scheffler et al.

"Measurement of Flow Using Back Projection Reconstruction: Display of Limited Regions of the Body", Young et al.

Salsman, James P., "Form–based Device Input and Upload in HTML", available at http://www.w3.org/1999/07/NOTE–device–upload–19990706, Cisco Systems, © 1999, Jul. 6, 1999 (9 pages).

Memo entitled: "Form–based File Upload in HTML", E. Nebel and L. Masinter, Network Working Group of Xerox Corporation, Nov. 1995 (12 pages).

HTML file entitled: "HTML 3.0 Draft (Expired!) Materials", available at http://www.w3.org/MarkUp/html3, Dec. 21, 1995 (4 pages).

Berners–Lee, T. and Connolly, D., HTML file entitled: "HyperText Markup Language Specification—2.0", available available at http://www.w3.org/MarkUp/html–spec., Nov. 1995 (10 pages).

Raggett, David, HTML file entitled: "A Review of the HTML + Document Format", Hewlett Packard Laboratories, date unknown (13 pages).

Raggett, David, HTML file entitled: "10–Fill–out Forms and Input Fields", available at http://www.w3.org/MarkUp HTMLPlus/htmlplus_41.html, Nov. 8, 1993 (5 pages).

Raggett, David, HTML File entitled: "ISO 8879:1986", available at http://www.w3.org/MarkUp/htmlplus_paper/htmlplus.dtd, Apr. 5, 1994 (17 pages).

Rescorla et al., "The Secure HyperText Transfer Protocol", Network Working Group, © Aug. 1999, (42 pages).

Berners–Lee et al., "HyterText Markup Language Specification –2.0", Nov. 29, 1994, Available online at *http://marc.merlins.org/–merlin/htmlearn/html2.html.*, Aug. 21, 2002, (69 pages).

Raggett, Dave, "HyperText Markup Language Specification Version 3.0", Mar. 28, 1995, Available online at *http://www.w3.or./MarkUp/html3/html3.txt.*, Aug. 21, 2002, (181 pages).

Merle et al., "Integration of heterogeneous environments: World Wide Web, Smart Card and Corba," Date Unknown, (16 pages).

"ProSolutions, Inc. ASP vs. CGI 'Active Server pages (ASP) Technology'", E Business Solutions, © 1995–2002, Available online at *http://www.prosolutionsinc.com/asp.htm.*, May 8, 2002, (2 pages).

Kirby et al., "Developing JDBC Servlet Applications", Dynamic Information Systems Corporation, Available online at *http://www.disc.com/jdbeserv.html.*, May 8, 2002, (12 pages).

Pasian et al., "A Generalized Mosaic–to–SQL Interface with Extensions to Distributed Archives", Astronomical Data Analysis Software and Systems IV ASP Conference Series, vol. 77, 1995, Available online at *http://www.stsci.edu/stsci/meetings/adassIV/pasianf.html.*, May 7, 2002, (5 pages).

Ng, Jason, "GSQL –a Mosaic–SQL gateway", Dec. 1993, Available online at *http://archive.ncsa/uiuc.edu/SDG/People/jason/pub/gsql/starthere.html.*, May 7, 2002, (16 pages).

Rasmussen, B. F., "WDB—A Web Interface to Sybase", Astronomical Data Analysis Software and Systems IV ASP Conference Series, vol. 77, 1995, Available online at *http://www.stsci.edu/stsci/meetings/assasIV/rasmussenb.html,*, May 7, 2002, (7 pages).

Connolly, Daniel W., "Paul Burchard on HTML 2.0 FORMs", Jul. 11, 1994, Available online at *http://lists.w3.org/Archives/Public/www–html/1994Jul./0014.html.*, Dec. 3, 2001, (6 pages).

Hackborn, Dianne, "Re: Java Applet element proposal", Jun. 8, 1995, Available online at *http://ksi.cpsc.uclagary.ca/archives/HTML–WG/html–wg–95q2.messages/1228.html.*, Nov. 1, 2001, (2 pages).

Kaye, Walter Ian, "Re: Protocol registry? (fwd. Java Applet element proposal)", Jun. 5, 1995, Available online at *http://lists.w3.org/Archives/Public/www–html/1995Jun./0069.html.*, Aug. 14, 2001, (3 pages).

"The APP HTML Tag", Hot Java, Sun Microsystems, Inc., © 1995, Available online at *http://www–math.uni–paderborn.de/dictionaire/Dictionaries/HTML–Dictionary/j.../index.html.*, Apr. 24, 2002, (2 pages).

Yourdon, Edward, "Java and the new Internet programming paradigm", © 2002, Available online at *http://www.javaworld.com/javaworld/jw–08–1996/jw–08–yourdonbook_p.html.*, Apr. 29, 2002, (20 pages).

Merckling et al., "Smart Card Introduction", Mar. 1994, Available online at *http://www.opengroup.org/rfc/mirrorrfc/rfc57.0txt.*, Mar. 1, 2001, (16 pages).

Bellovin et al., "Limitations of the Kerberos Authentication System", USENIX, 1991, (16 pages).

"Guideline for the Use of Advanced Authentication Technology Alternatives", Federal Information Processing Standards Publication 190, Sep. 28, 1994, Available online at *http://www.itl.nist.gov/fipspubs/fip190.htm.*, Mar. 21, 2002, (49pages).

"HTML + (Hypertext markup format)", Nov. 8, 1993, Available online at *http://www.w3.org/MarkUp/HTMLPlus/htmlplus_2.html.*, Mar. 28, 2002, (89 pages).

Everhart, Craig, "Re: Interesting uses of networking", Jun. 11, 1990, Available online at *http://www.cs.ucsd.edu/users/bsy/coke.history.txt.*, Sep. 17, 2002 (4 pages).

Kotanchik, J., Kerberos and Two–Factor Authentication, Security Dynamics, Mar. 1994, (11 pages).

McLaughlin, L. III, ed., "Line Printer Daemon Protocol", Network Printing Working Group, Aug. 1990, Available online at *http://www.ietf.org/rfc/rfc1179.txt.*, Sep. 17, 2002, (14 pages).

EXHIBIT E
PAGE 6 OF 43

Gutfreund, Yechezkal–Shimon, "Mosaic Accessories", Jan. 26, 1994, Available online at *file://C/WINDOWS/TEMP/WWW–Talk%20Jan–Mar%201994%20Mosaic%20Accessories.htm.*, Apr. 1, 2002, (8 pages).

Gessler et al., "PDAs as mobile WWW browsers", Date Unknown, Telecooperation Office, University of Karlsruhe, Available online as *file://C:/WINDOWS/TEMP/gwpring/PDAs%20as%20mobile%20WWW%20browsers.htm.*, Mar. 28, 2002, (9 pages).

Rose, M., "Principles of Operation for the TPC.INT Subdomain: Radio Paging —Technical Procedures", Dover Beach Consulting, Inc., Jan. 1994, (6 pages).

Workshop on Network and Distributed System Security, Privacy and Security Research Group, Feb. 11–12, 1993, (6 pages).

Malamud et al., "Principles of Operation for the TPC.INT Subdomain: Remote Printing —Technical Procedures", Dover Beach Consulting, Inc., Oct. 1993, (11 pages).

Louis, Tristan, "Java on the Menu; Causing Jitters?," Jun. 1995, Available online at *http://www.tnl.net/who/bibliography/javajitters.asp.*, Apr. 29, 2002, (2 pages).

Allen, Terry, "Java Applet element proposal," Jun. 1, 1995, Available online at *http://ksi.cpsc.ucalgary.ca/archives/HTML–WG/html–wg–95q2.messages/1119.html.*, Nov. 16, 2001, (4 pages).

Hume, Tom, "WAP: Where we've come from, where we are, where we're going," Available online at *http://www.tomhume.org/stories/2002/02/04/wapWhereWeveComeI rom-WhereWeAreW...*, (18 pages).

Goldman, Chris, "Browsing for Perfection," ARC Group in the Press, Feb. 1, 2001, Available online at *http://www.arcgroup.com/homepage.nsf/PRSARC/ar010201*, (4 pages).

"Handheld Devices Markup Language (HDML) specification," Apr. 11, 1997, Available online at *http://www.w3.org/TR/hdml20–3.html* through *http://www.w3.org/TR/hdml20–9.html*, (33 pages).

Hinden, R., "Simple Internet Protocol Plus White Paper," Network Working Group, Oct. 1994, Available online at *ftp://ftp.rfc–editor.org/in–notes/rfc1710.txt*, (22 pages).

McCarthy, Vance, "Novell to link wireless laptops, PDAs to LANs (portable computers, personal digital assistants, local area networks)," InfoWorld, Sep. 13, 1993 v15 n37 p1, (2 pages).

Schwartz, Barry K. and Stephen B. Weinstein, "Dual–Media Messaging Using Screen Telephones on the Telephone Network," Bell Communications Research, IEEE 1993, pp. 1138–1188, (6 pages).

Verjinski, Richard D., "PHASE, A Portable Host Access System Environment," Unisys Defense Systems, IEEE 1989, pp. 806–809, (4 pages).

Klemets, Anders, Gerald Q. Maguire, Frank Reichert, and Mark T. Smith, "MINT –A Mobile Internet Router," IEEE 1993, pp. 70–74, (5 pages).

Perkins, Charles E. and Pravin Bhagwat, "A Mobile Networking System based on Internet Protocol," IEEE Personal Communications, First Quarter 1994, pp. 32–41, (10 pages).

Connolly, T., P. Amer, and P. Conrad, "An Extension to TCP: Partial Order Service," Network Working Group, Nov. 1994, Available online at *ftp://ftp.rfc–editor.org/in–notes/rfc1693.txt*, (14 pages).

Furniss, P., "Octet Sequences for Upper–Layer OSI to Support Basic Communications Applications," Network Working Group, Oct. 1994, Available online at *ftp://ftp.rfc–editor.org/in–notes/rfc1698.txt*, (28 pages).

Graves, C., T. Butts, and M. Angel, "TN3270 Extensions for LUname and Printer Selection," Network Working Group, Jul. 1994, Available online at *ftp://ftp.rfc–editor.org/innotes/rfc1646.txt*, (13 pages).

Klensin, J., N. Freed, M. Rose, E. Stefferud, and D. Crocker, "SMTP Service Extensions," Network Working Group, Feb. 1993, Available online at *ftp://ftp.rfc–editor.org/in–notes/rfc1425.txt*, (10 pages).

Wang, Z., "EIP: The Extended Internet Protocol A Framework for Maintaining Backward Compatibility," Network Working Group, Nov. 1992, Available online at *ftp://ftp.rfc–editor.org/in–notes/rfc1385.txt*, (16 pages).

Gwinn, A., "Simple Network Paging Protocol –Version 2," Network Working Group, Jul. 1994, Available online at *ftp://ftp.rfc–editor.org/in–notes/rfc1645.txt*, (15 pages).

Braden, B., J. Postel, and Y. Rekhter, "Internet Architecture Extensions for Shared Media," Network Working Group, May 1994, Available online at *ftp://ftp.rfc–editor.org/in–notes/rfc1620.txt*, (18 pages).

Delgrossi, Luca, Christian Halstrick, Ralf Guido Herrtwich, and Heinrich Stüttgen, "HeiTP –A Transport Protocol for ST–II," IBM European Networking Center, IEEE 1992, pp. 1369–1373, (5 pages).

Tsang, P. W. M. and R. W. C. Wang, "Development of a Distributive Lighting Control System Using Local Operation Network," IEEE Transactions on Consumer Electronics, vol. 40, No. 4, Nov. 1994, (11 pages).

Adomeit, Reinhard and Bernhard Holtkamp, "A Reference Framework for Integrated POI/POS Systems," Fraunhofer Institute for Software Engineering and Systems Engineering, (4 pages).

Caneve, Maurizo, and Roberto Gagliardi, "A WAIS–Based Hypermedia Document System," (4 pages).

Schmidt, Claudia, Burkhard Stiller, and Martina Zitterbart, "Towards Flexible Service–integrated Communication Subsystems," Jul. 1, 1994, (4 pages).

Plagemann, Thomas, Janusz Waclawczyk, and Bernhard Plattner, "Management of Configurable Protocols for Multimedia Applications," (4 pages).

"Frame relay attracts chains; networks at Kash N' Karry, Office Max (network communications protocol)," Chain Store Age Executive with Shopping Center Age, Dec. 1994 v70, (4 pages).

Cummings, Joanne, "What's Hot What's Not '95," Network World, Dec. 26, 1994/Jan. 2, 1995, (5 pages).

Aronin, Mindy, "XcelleNet launches fifth–generation RemoteWare 2.0 for remote and mobile applications," Business Wire, Nov. 28, 1994, (4 pages).

Baum, David, "Designing mobile applications (case studies of companies developing mobile programs)," InfoWorld, Nov. 8, 1994 v16 n48 p73, (7 pages).

Moeller, Michael, "Wireless standard to ease application portability; extensions will provide a common layer for CDPD, RAM, Ardis software," PC Week, Nov. 21, 1994 v11 n46 p1, (3 pages).

Petreley, Nicholas, Nancy Durlester, and Laura Wonnacott, "The interoperability headache: linking disparate clients and servers," InfoWorld, Nov. 15, 1993 v15 n46 p124, (12 pages).

Miscellaneous List of Article Abstracts, (50 pages).

Hansen, Wilfred J., "Enhancing Documents with Embedded Programs: How Ness Extends Insets in the Andrew ToolKit", IEEE 1990, pp. 23–32, (10 pages).

EXHIBIT E
PAGE 7 OF 43

Tani, M., Yamaashi, K., Tanikoshi, K., Futakawa, M., and Tanufuji, S., "Object-Oriented Video: Interaction with Real-World Objects Through Live Video", ACM 1992, pp. 593–598, (6 pages).

Crowley, T., Milazzo, P., Baker, E., Forsdick, H., and Tomlinson, R., "MMConf: An Infrastructure for Building Shared Multimedia Applications", ACM 1990, pp. 329–342, (14 pages).

Davis, H., Hall, W., Heath, I., Hill, G., and Wilkins, R., "Towards an Intergrated Information Environment with Open Hypermedia Systems", ACM 1992, pp. 181–191, (10 pages).

Ferrara, Fabrizio M., "The KIM Query System: An Iconic Interface for the Unified Access to Distributed Multimedia Databases", SIGGHI Bulletin, 1994, vol. 26, No. 3, pp. 30–39, (10 pages).

Gibbs, Simon, "Composite Multimedia and Active Objects", OOPSLA, 1991, pp. 97–112, (16 pages).

Davis, H., Hall, W., Pickering, A., and Wilkins, R., "Microcosm: An Open Hypermedia System" ACM, 1993, (1 page).

Cullen, J., Day, B. and Planes, M., "The Use of FTAM to Access Graphical Pictures Across Wide Area Networks", Elsevier Science Publischers B.V., 1992, pp. 377–383, (7 pages).

Vaziri, Arsi, "Scientific Visualization in High-Speed Network Environments", Elsevier Science Publishers B.V., 1991, pp. 111–129, (19 pages).

Lashkari, Y., Rao, S., Mhaskar, V. and Shelat, A., "PLX: A Proposal to Implement a General Broadcasting Facility in a Distributed Environment Running X Windows", Comput. And Graphics, vol. 16, No. 2, 1992, pp. 143–149, (7 pages).

Kirste, Thomas, "SpacePicture–An Interactive Hypermedia Satellite Image Archival System", Comput. & Graphics, vol. 17, No. 3, 1993, pp. 251–260, (10 pages).

Coulson, G., Blair, G., Davies, N. and Williams, N., "Extensions to ANSA for Multimedia Computing", Elsevier Science Publishers B.V., 1992, pp. 305–322, (18 pages).

Le Huynh, D., Jensen, M., Larsen, R., Southard, J. and Wang, F., "PIX: An Object-Oriented Network Graphics Environment", Comput. & Graphics, vol. 17, No. 3, 1993, pp. 295–304, (10 pages).

Berners-Lee, T.J., Cailliau, R., and Groff, J., "The World-Wide Web", Elsevier Science Publishers B.V. 1992, pp. 454–459, (6 pages).

Shackelford, D., Smith, J., and Smith, F., "The Architecture and Implementation of a Distributed Hypermedia Storage System", Nov., 1993, pp. 1–13, (13 pages).

Labriola Don, "Whiteboard Software: Remote Possibilites", PC Magazine, Jun. 14, 1994, pp. 223–224, (2 pages).

"How We Evaluate the Whiteboard Software", PC Magazine, Jun. 14, 1994, pp. 225–228, (4 pages).

Udell, John, "Visual Basic Custom Controls Meet OLE", BYTE, Mar., 1994, pp. 197–198 and 200, (3 pages).

Sarna, D. and Febish, G., "OLE Gains Without (Much) Pain", DATAMATION, Jun. 15, 1994, p. 31, (1 page).

Scrupski, Susan, "Outsourcing: It Could be Good", DATAMATION, Jun. 15, 1994, pp. 32–33 and 114, (3 pages).

Rizzo, John, "What's OpenDoc?", MacUser, Apr. 1994, pp. 119–123, (3pages).

Cello Browser Release 1.01a, *ftp.law.cornell.edu/pub/L11/cello*, Mar. 16, 1994, (8 pages).

"Why Jim Clark Loves Mosaic", Wired, Oct. 1994, pp. 118–121, 150–151 and 157, (7 pages).

Marovac, N., and Osburn, L., "Hypernet: A Tool to Choreograph Worldwide Distributed Hypermedia Documents", Comput. & Graphics, vol. 16, No. 2, 1992, pp. 197–202, (6 pages).

Moeller, M., and Leach, N., "Microsoft Maps New OCX Plan; ActiveX seen as Web Content Platform", PC Week, Mar. 11, 1996, vol. 13, No. 10, pp. 1–2, (2 pages).

"OLE 2.0: Death to Monoliths?", BYTE, Mar. 1994, pp. 121–122, (2 pages).

"Netscape Communications Offers New Network Navigator Free on the Internet", *http://home.netscape.com/newsref/pr/newrelease.1.html*, Jan. 29, 1998.

Sackman, Gleason, "Remote Tele-Excavation via the Web", Tele-robotics, vol. 11, Issue 9, (1 page).

Wolf, Gary, "The (Second Phase of) the Revolution has Begun", WIRED, Oct. 1994, pp. 16–17, (2 pages).

Wynne, B. and Marovac, N., "Lean Management, Group Support Systems, and Hypermedia: A Combination Whose Time Has Come", IEEE, 1993, pp. 112–121, (10 pages).

Moran, Patrick J. "Tele-Nicer-Slicer-Dicer: A New Tool for the Visualization of Large Volumetric Data", Biomedical Imaging Group, pp. 1–7, (7 pages).

Hansen, W., Consortium, A, "Andrew as a Multiple Environment for Visual Languages", IEEE, 1993, pp. 256–260, (5 pages).

Doyle, M., "The Virtual Embryo: VR Applications in Human Development Anatomy", Jan. 27, 1994, pp. 1–9, 9 pages.

Doyle, M., "A Knowledge Management Environment through the World Wide Web", Jan. 7, 1994.

Oliver, D. et al., "Netscape™ Unleashed".

Ang, C., Martin, D., and Doyle, M., "Intergrated Control of Distributed Volume Visulazation Through the World-Wide-Web", 1994, pp. 1–10, and 1, (11 pages).

* cited by examiner

EXHIBIT E
PAGE 9 OF 43

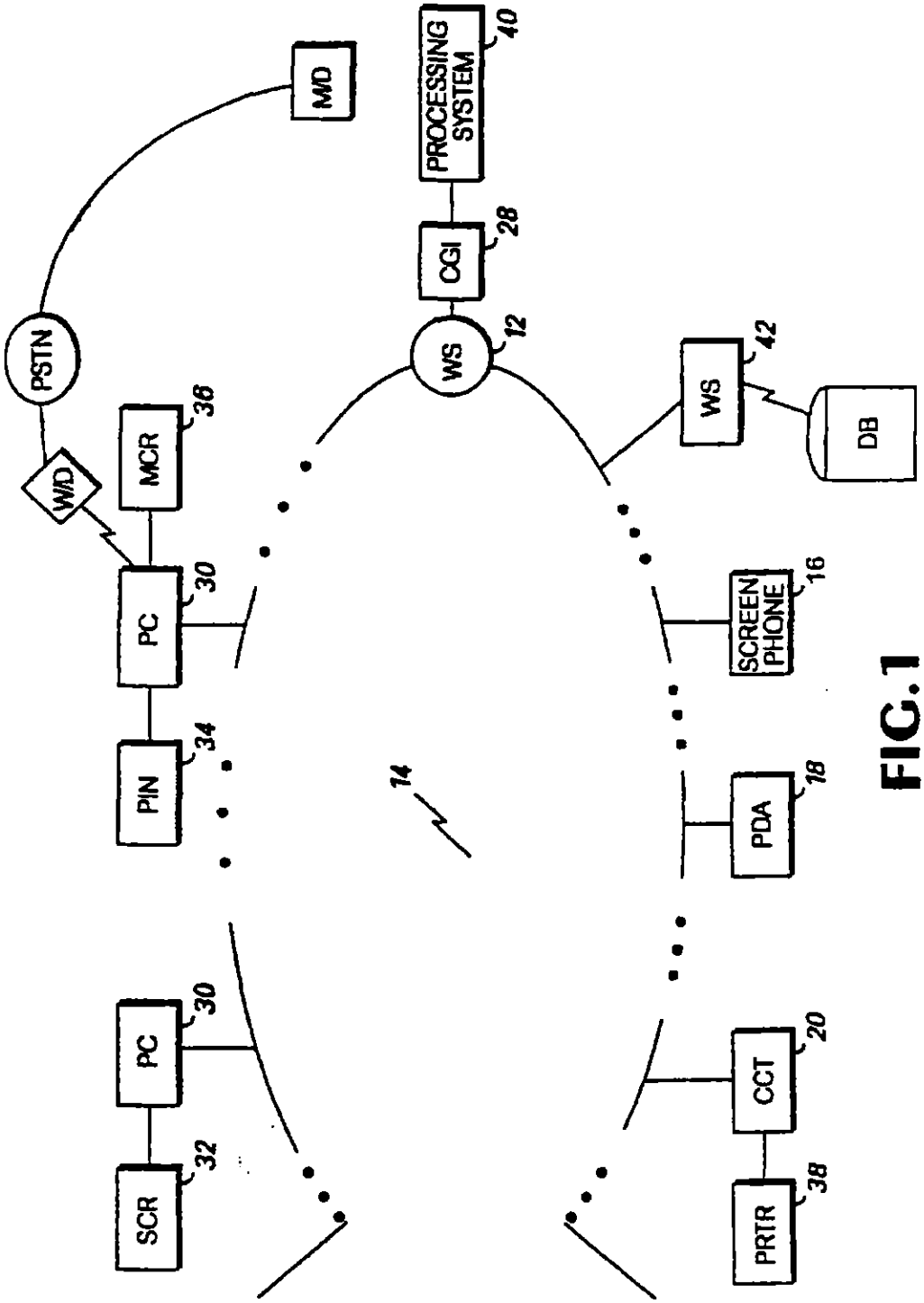

FIG.1

EXHIBIT E
PAGE 9 OF 43

HTML+D Attributes		Description
<FORM ACTION = "url"		To/From Web Server URL
	FROM "file name"	From Terminal Local File
	TO PRINTER	To Local Printer
	TO "file name"	To Terminal Local File
	FROM SCR	From Smart Card Reader
	TO SCR	To Smart Card Reader
METHOD = "GET"		Retrieve Data
	"POST"	Store Data
	"PAYMENT"	Directive to deliver INPUT data to a private Payment Network for authorization and settlement.
	SQL <database name>	SQL statement database table

Attribute	Value	Description
<INPUT TYPE =	"text"	
	"password"	
	"checkbox"	
	"radio"	
	"submit"	
	"reset"	
NAME =	<field name>	
VALUE =	<initial value>	
CHECKED =		
SIZE =		
MAXLENGTH =		
>		

Attribute	HTML+D Value	Terminal Device
TYPE =	*"MSRT1"*	Mag Stripe Reader - Track 1
	"MSRT2"	Mag Stripe Reader - Track 2
	"KEY"	Terminal Command Keypad
	"PIN"	PIN Pad
	"BCW"	Bar Code Wand
	"MICR"	Check MICR Reader
	"AMT"	Dollar amount key input mask
	"INT"	Integer key input mask
	"LOCAL"	Input from Local Variable
	"AUTOSUBMIT"	Submit FORM to ACTION URL
NAME =	*ip_address*	Local Variable - Terminal's IP Address
	host_phone	Local Variable - Local Internet Access Phone Number
	tid	Local Variable - Terminal ID
	work_key	Local Variable - PIN encryption working key
	datetime	Local Variable - Date and time
	deposit_acct	Local Variable - Merchant Deposit Account

FIG.2

EXHIBIT E
PAGE 10 OF 43

SQL Statements

The following SQL commands represent a subset of the entire command set that varies by database vendor.

HTML+D Attributes	Description
SELECT, field_name,_*	Request field_name (one or many) from a database table
FROM=<table name>,_	Database table name
WHERE=<condition>	Conditional selection of data
name = "constant"	
name LIKE "constant"	
name IN "constant"	
AND	
OR	
ORDER=ASC	Request in ascending order
DESC	_descending
2	_by2's
GROUP=<name>	
INSERT TABLE=<table name>	Insert new data in database table
VALUES="constants"	
UPDATE FROM	
<table name>	Update field_name in database table
SET=field_name="constant"	
[WHERE=<condition>]	Update if WHERE clause is satisfied
DELETE FROM <table name>	Delete all columns that satisfy
[WHERE=<condition>]	WHERE clause
CREATE TABLE<table_name>	Create database table
PRIMARY KEY<name>	

FIG.3

EXHIBIT E
PAGE 11 OF 43

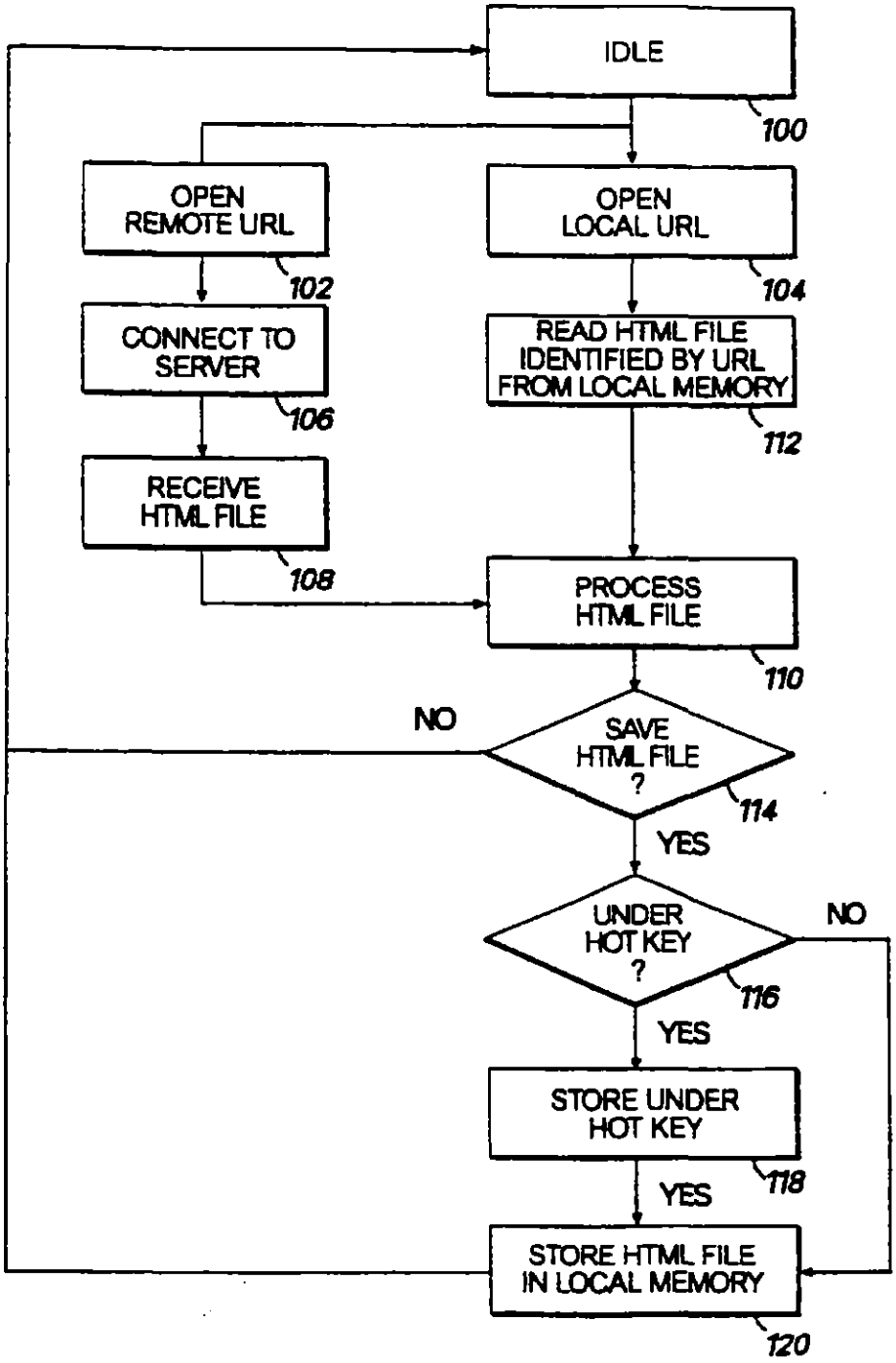

FIG.4

EXHIBIT E
PAGE 12 OF 43

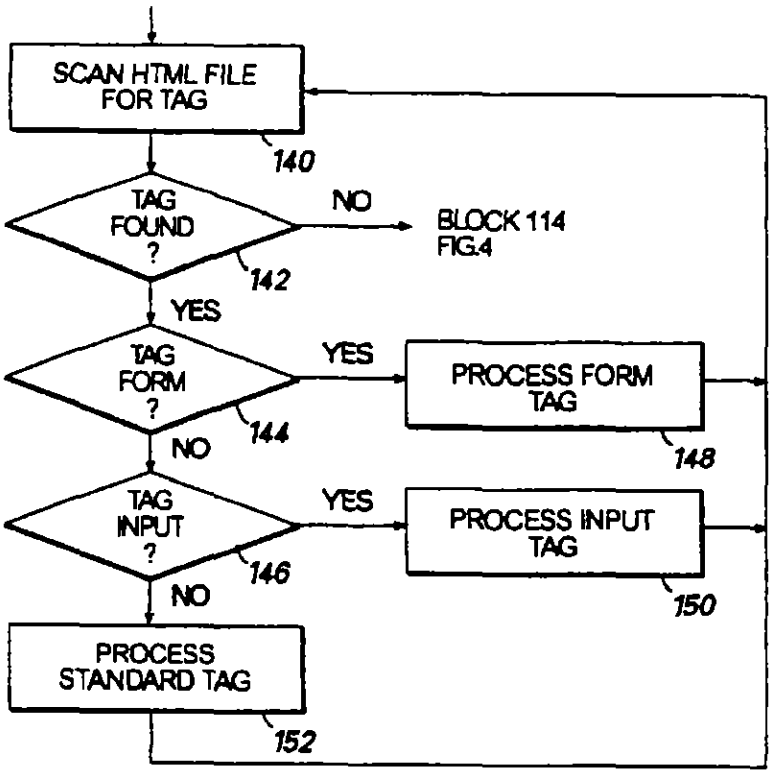

FIG.5

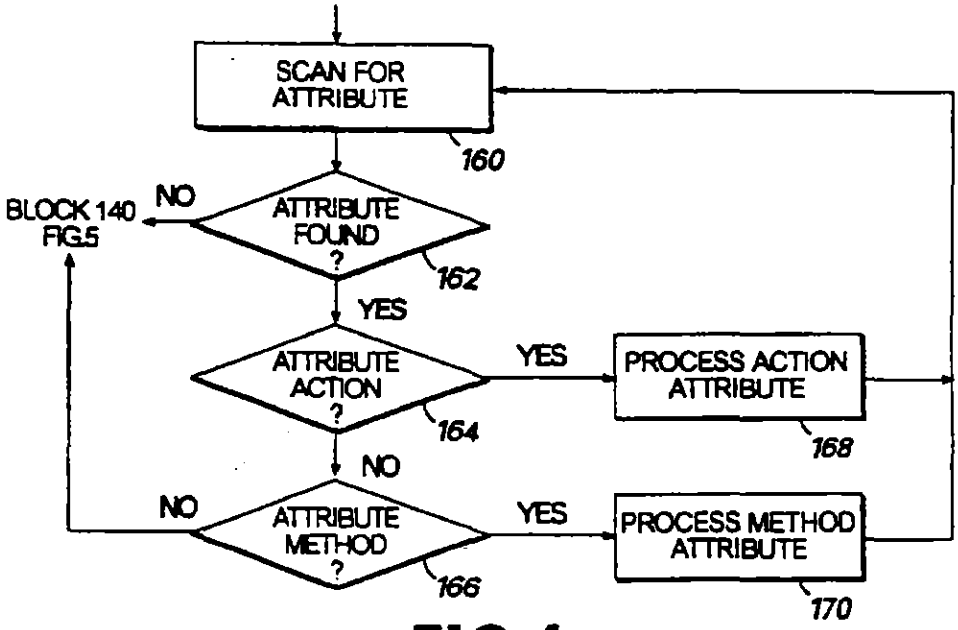

FIG.6

EXHIBIT E
PAGE 13 OF 43

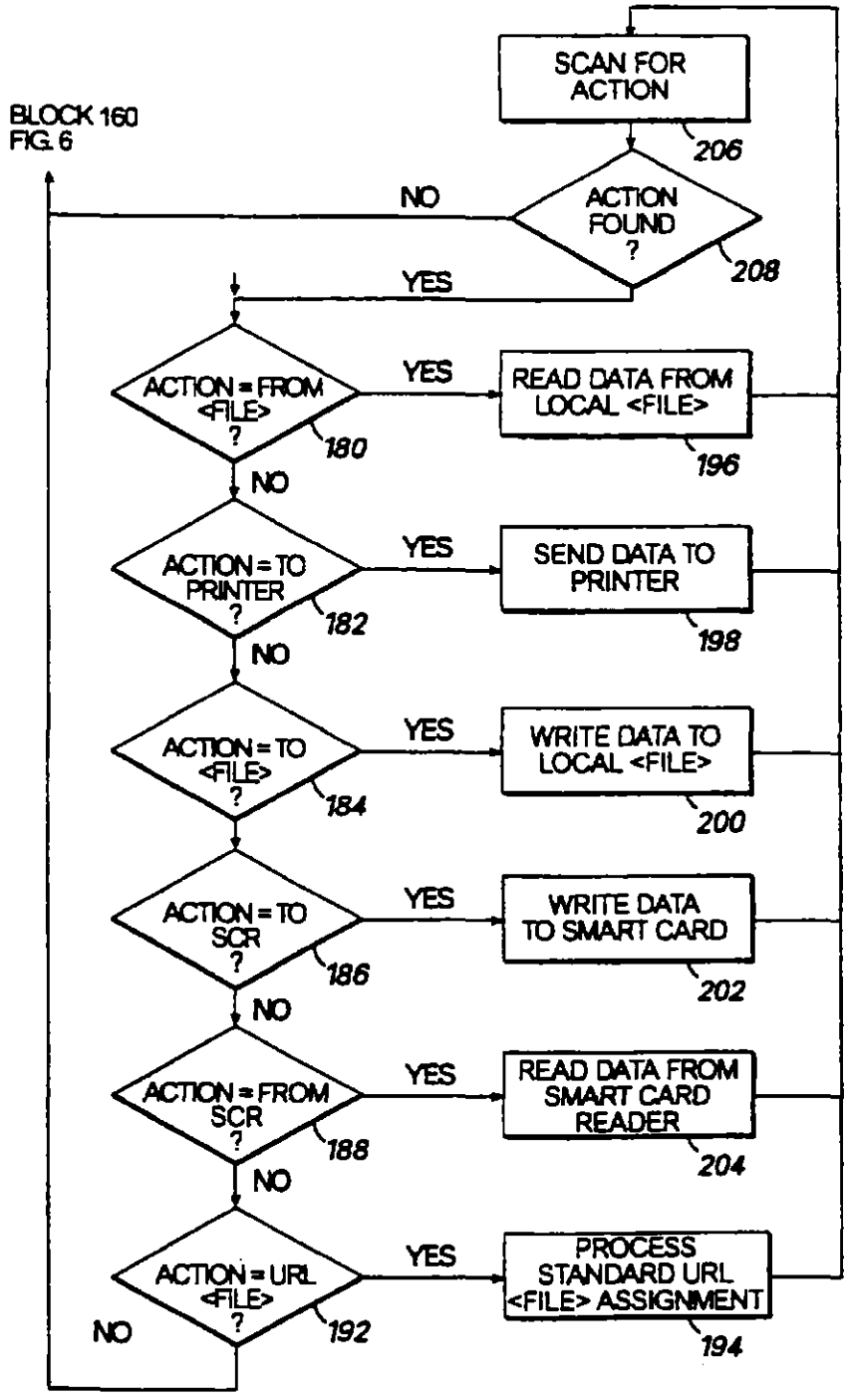

FIG. 7

EXHIBIT E
PAGE 14 OF 43

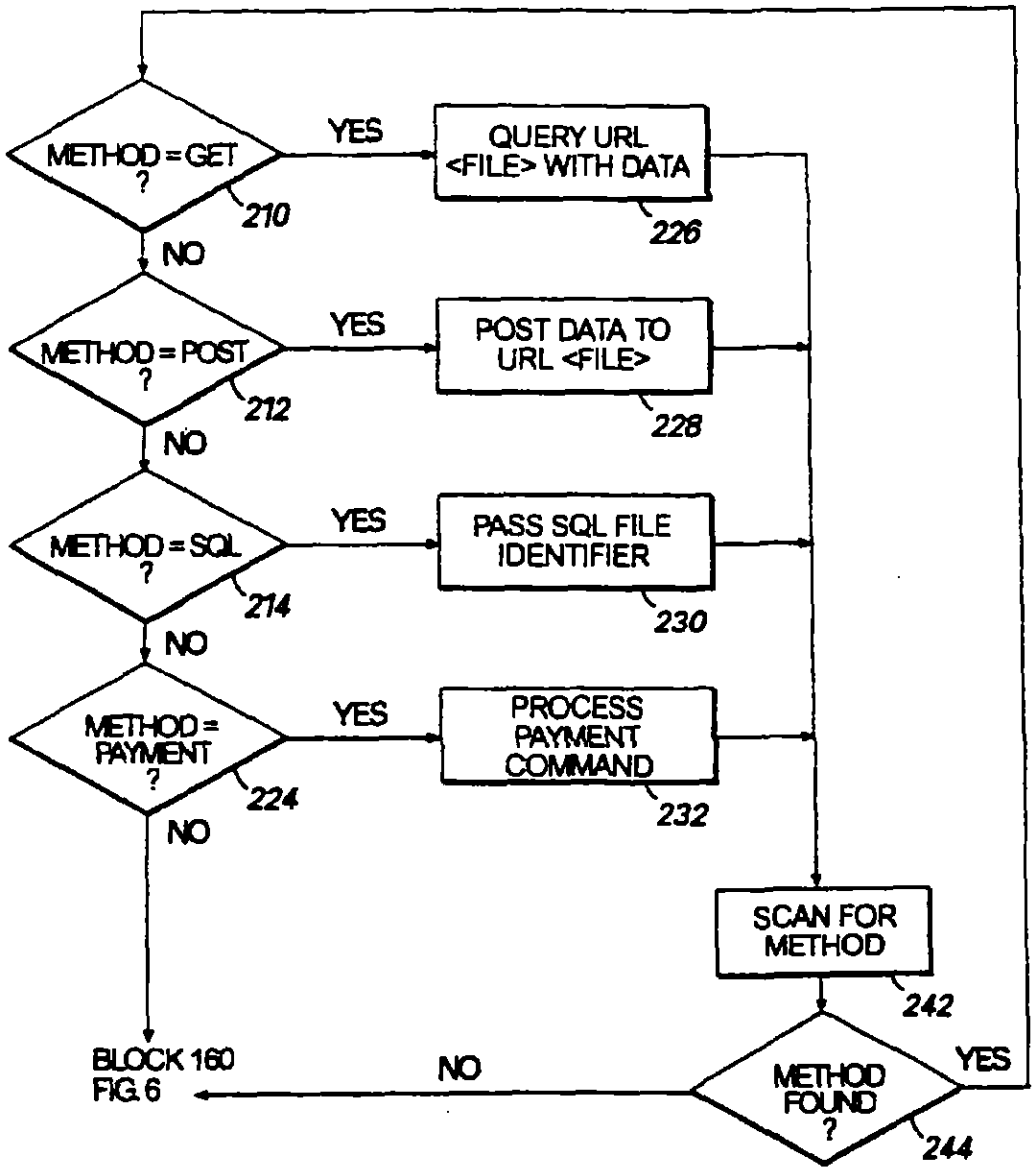

FIG.8

EXHIBIT E
PAGE 15 OF 43

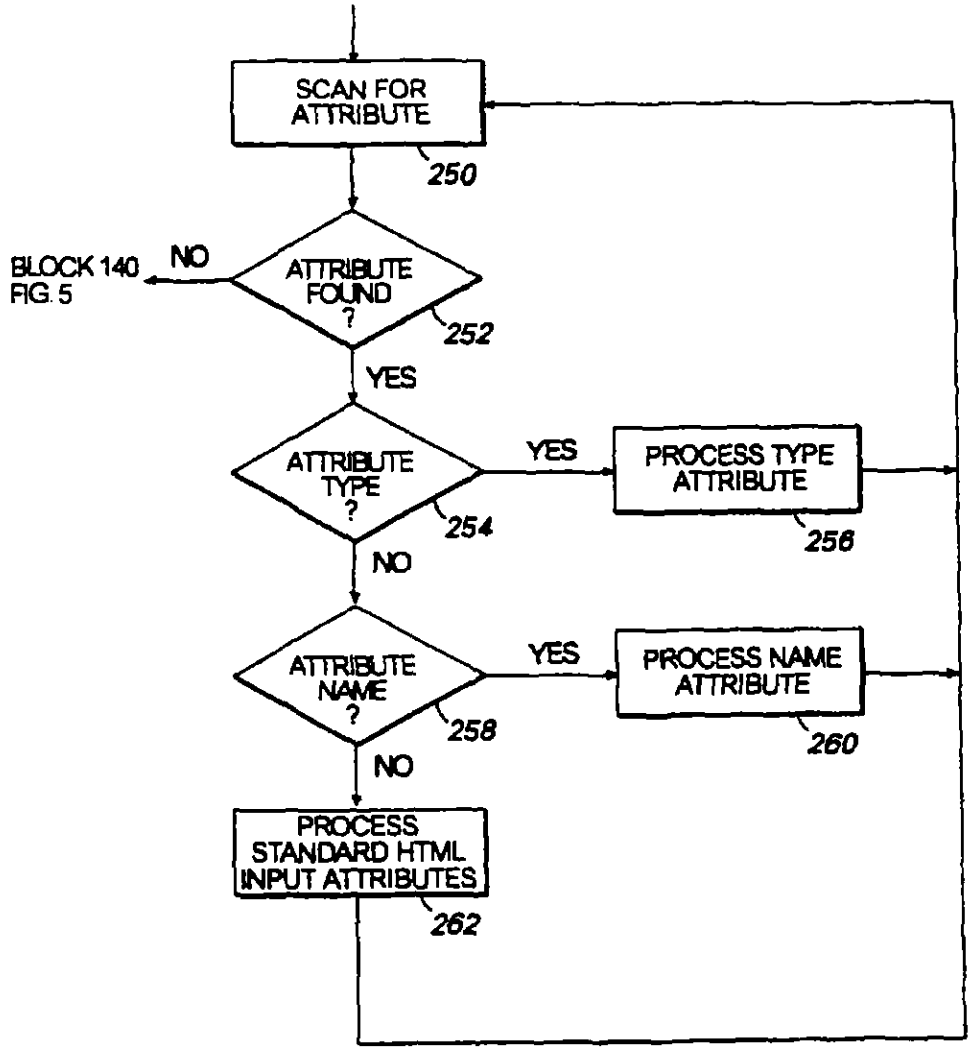
SCAN FOR ATTRIBUTE
250
BLOCK 140 FIG. 5
NO
ATTRIBUTE FOUND ?
252
YES
ATTRIBUTE TYPE ?
254
YES
PROCESS TYPE ATTRIBUTE
256
NO
ATTRIBUTE NAME ?
258
YES
PROCESS NAME ATTRIBUTE
260
NO
PROCESS STANDARD HTML INPUT ATTRIBUTES
262

FIG. 9

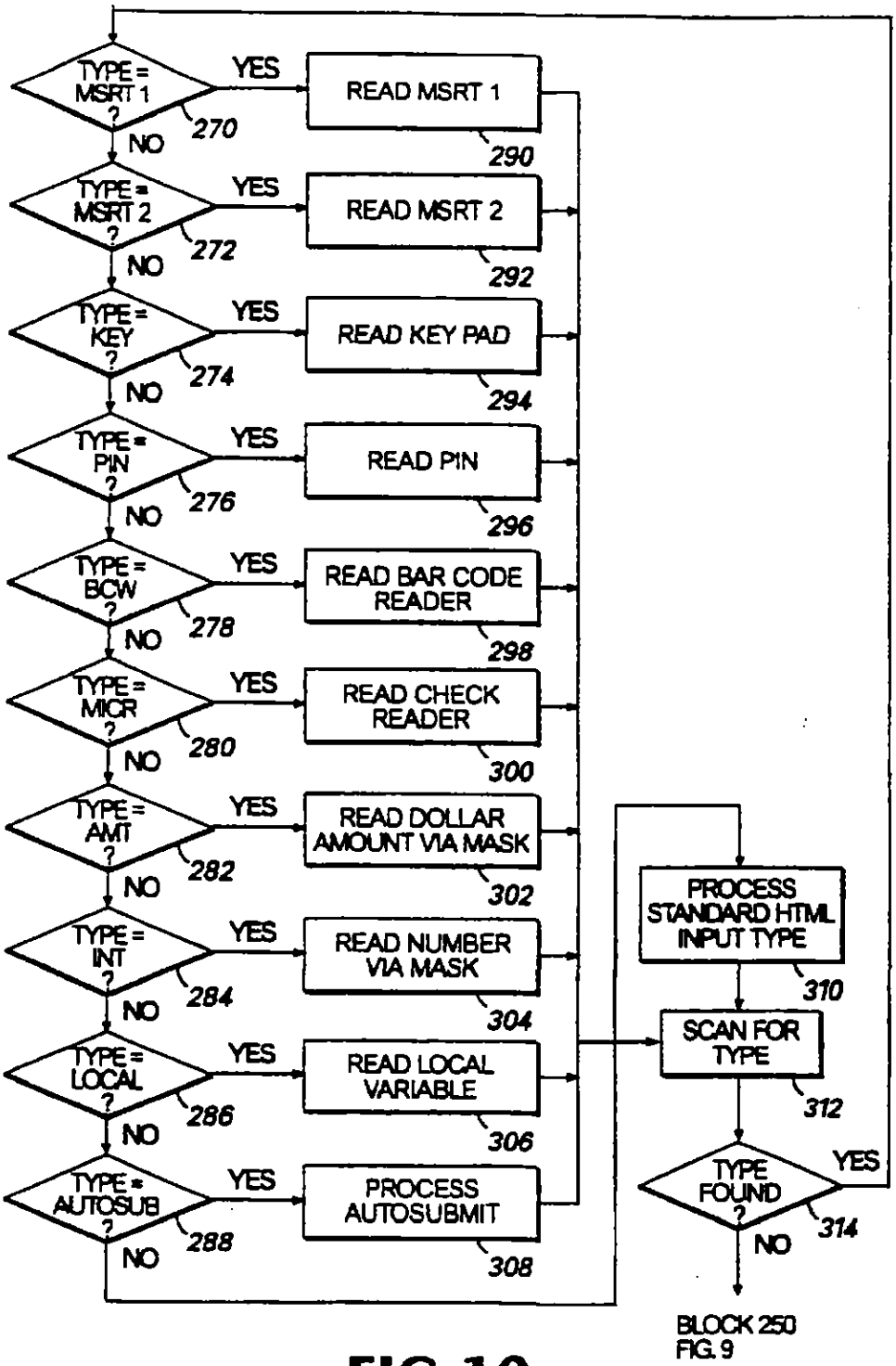

FIG.10

EXHIBIT E
PAGE 17 OF 43

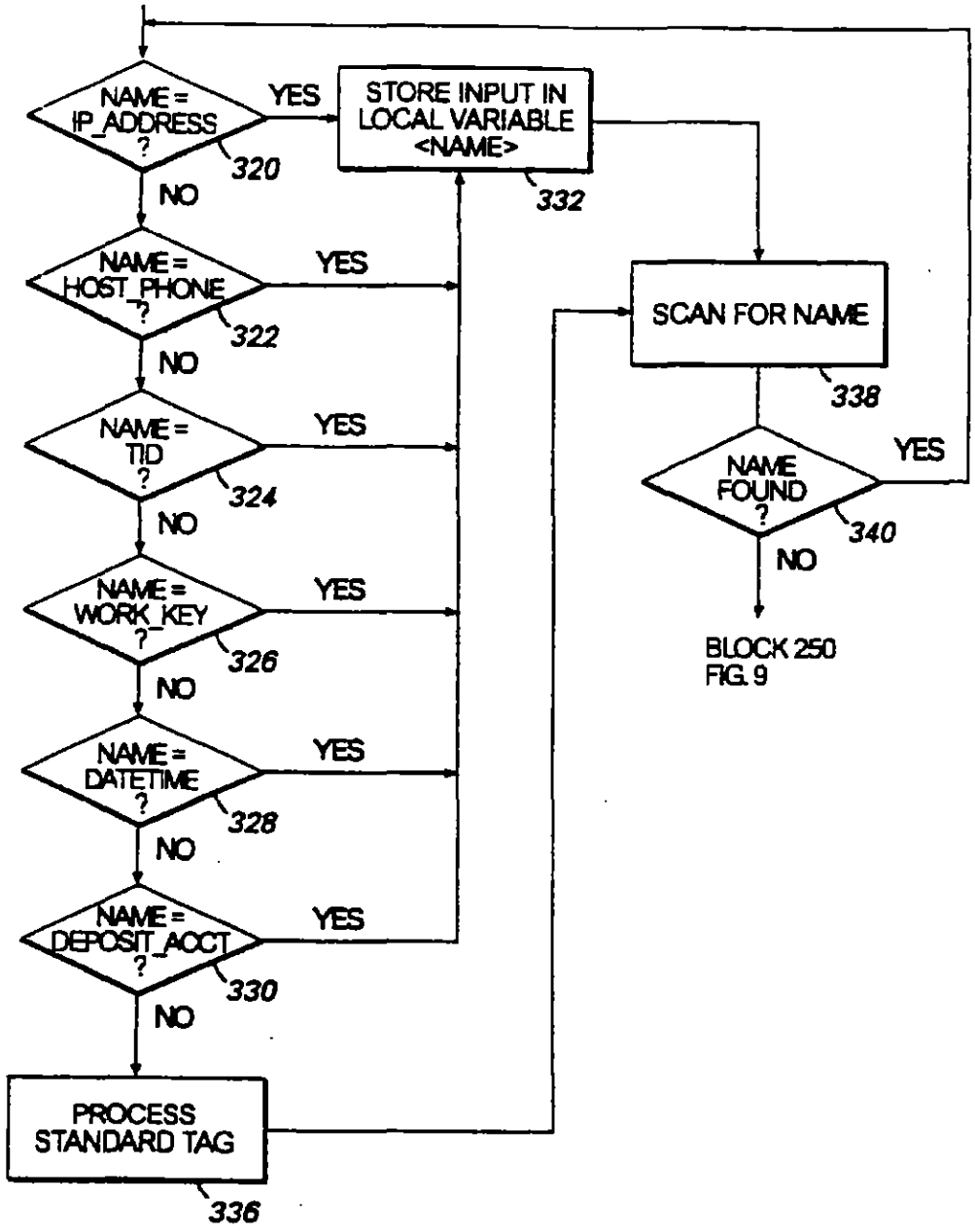

FIG.11

EXHIBIT E
PAGE 18 OF 43

```
1.  <FORM ACTION=URL      METHOD=  GET>
2.  <FORM ACTION=URL      METHOD=  POST>
3.  <FORM ACTION=URL      METHOD=  SQL <database_name>
```

FIG.12

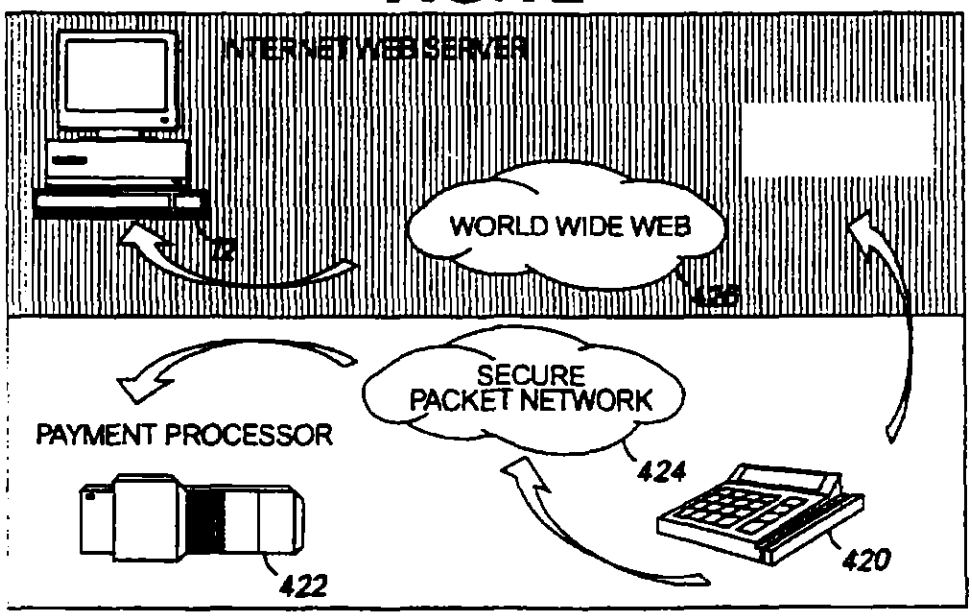

FIG.13A

```
<FORM ACTION=<filename>    METHOD=PAYMENT>
<INPUT TYPE=AUTOSUBMIT>
</FORM>
```

FIG.13B

```
<FORM ACTION=dsinet METHOD=PAYMENT>
<INPUT TYPE=LOCAL NAME=DEPOSIT_ACCT VALUE=1234568902345678 90>
<INPUT TYPE=AUTOSUBMIT>
</FORM>
```

FIG.13C

EXHIBIT E
PAGE 19 OF 43

1.a. Transaction Request HTML+D

500

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
    METHOD=SQL
        "BEGIN TRAN
        IF NOT EXISTS ( SELECT substring(account, 1, 20) FROM auth_table)
            BEGIN
            INSERT TABLE=log_table VALUES=(getdate(),tid, substring (account, 1,20) ,
            substring( account, 22, 4), amount)
            SELECT * FROM log_table WHERE trandate = getdate()
            END
        ELSE SELECT * FROM error_table WHERE error_no=1
        COMMIT TRAN">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER ACCOUNT NUMBER:
<INPUT TYPE="MSRT2" SIZE=40 NAME=account>
ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

1.b. Transaction Accepted Response

510

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
JUNE 1 1995            10:30AM        PURCHASE</P>
TERMINAL ID:                          999999999</P>
ACCOUNT NUMBER         99999999999999999999</P>
EXP DATE:                                 99/99</P>
AMOUNT                                 $9999.99</P>
AUTH NUMBER                           999999999</P>
</p>
-------------------------------------------</p>
        CUSTOMER SIGNATURE </p>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

1.c. Transaction Declined or Submit Error Response

520

```
<HTML>
<BODY>
DECLINED<MESSAGE>
</BODY>
</HTML>
```

FIG.14

EXHIBIT E
PAGE 20 OF 43

2.a. Transaction Request HTML+D

550

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
METHOD=SQL"BEGIN TRAN
        IF NOT EXISTS ( SELECT substring(account, 1, 20) FROM auth_table)
        BEGIN
                INSERT TABLE=log_table VALUES=(getdate(),tid, substring( account, 1,20),
                substring( account, 22, 4), amount)
        END
        ELSE  BEGIN
                        SELECT * FROM error_table WHERE error_no=1
                        RETURN
                END
        INSERT TABLE=order_table VALUES=( getdate(), cust_name,address,city, state, zip,
                part_code, unit_price, tax, ship_method, ship_chrg, unit_price + tax +
                ship_chrg, substring( account, 1, 20) ,substring( account, 22, 4))
        SELECT * FROM order_table WHERE trandate = getdate()
        COMMIT TRAN">
CUSTOMER NAME:
<INPUT TYPE="TEXT"SIZE=30 NAME=cust_name></p>
ADDRESS:
<INPUT TYPE="TEXT" SIZE=40 NAME=address></p>
CITY:
<INPUT TYPE="TEXT" SIZE=20 NAME=city><p>
STATE:
<INPUT TYPE="TEXT" SIZE=2 NAME=state>
ZIP:
<INPUT TYPE="TEXT" SIZE=10 NAME=address></p>
SCAN PART CODE:
<INPUT TYPE="BCW" SIZE=9 NAME=part_code></p>
ENTER UNIT PRICE:
<INPUT TYPE="AMT" SIZE=8 NAME=unit_price></p>
TAX:
<INPUT TYPE="AMT" SIZE=5 NAME=tax></p>
SHIPPING METHOD:
<INPUT TYPE="TEXT" SIZE=10 NAME=ship_method></p>
SHIPPING AMOUNT:
<INPUT TYPE="AMT" SIZE=5 NAME=ship_chrg></p>
SLIDE CARD:
<INPUT TYPE="MSRTZ" SIZE=40 NAME=account></p>
<INPUT TYPE="SUBMIT" >
</FORM>
</BODY>
</HTML>
```

FIG.15A

EXHIBIT E
PAGE 21 OF 43

2.b. Transaction Accepted Response

555

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
ORDER # 9999999999 APPROVED</p>
JUNE 1 1995        10:30AM        PURCHASE</P>
TERMINAL ID:                      99999999</P>
NAME: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
ADDRESS:</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
CITY: XXXXXXXXXXXXXXXXXXXXXXXXX</p>
STATE: XX                ZIP: XXXXXXXXXX</p>
ACCOUNT NUMBER:          9999999999999999999</p>
EXP DATE:                            99/99</p>
PART CODE:                        99999999</p>
UNIT PRICE:                       $9999.99</p>
SHIP METHOD:XXXXXXX CHARGE        $9999.99</p>
TOTAL AMOUNT:                     $9999.99</p>
AUTH NUMBER:                      99999999</p>
</p>

- - - - - - - - - - - - - - - - - - - - - - - </p>
          CUSTOMER SIGNATURE</p>

</FORM>
</BODY>
</HTML>
```

2.c. Transaction Declined or Submit Error Response

560

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.15B

EXHIBIT 6
PAGE 22 OF 43

3.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
METHOD SQL
      "INSERT TABLE=order_table VALUES=( getdate(), cust_name,address,city, state, zip,
            part_code, unit_price, tax, ship_method, ship_chrg,unit_price + tax +
            ship_chrg, substring( account, 1, 20) ,substring( account, 22, 4))
      SELECT * FROM order_table WHERE trandate = getdate()">
<INPUT TYPE="LOCAL" NAME=tid>
CUSTOMER NAME:
<INPUT TYPE="TEXT" SIZE=30 NAME=cust_name></p>
ADDRESS:
<INPUT TYPE="TEXT" SIZE=40 NAME=address></p>
CITY:
<INPUT TYPE="TEXT" SIZE=20 NAME=city><p>
STATE:
<INPUT TYPE="TEXT" SIZE=2 NAME=state>
ZIP:
<INPUT TYPE="TEXT" SIZE=10 NAME=address></p>
ENTER PART CODE:
<INPUT TYPE="TEXT" SIZE=10 NAME=part_code></p>
ENTER UNIT PRICE:
<INPUT TYPE="AMT" SIZE=8 NAME=unit_price></p>
TAX:
<INPUT TYPE="AMT" SIZE=5 NAME=tax></p>
SHIPPING METHOD:
<INPUT TYPE="TEXT" SIZE=10 NAME=ship_method></p>
SHIPPING AMOUNT:
<INPUT TYPE="AMT" SIZE=5 NAME=ship_chrg></p>
<INPUT TYPE="SUBMIT" >
</FORM>
</BODY>
</HTML>
```

FIG.16A

EXHIBIT E
PAGE 23 OF 43

3.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
ORDER # 9999999999 APPROVED</p>
JUNE 1 1995 10:30AM PURCHASE</P>
TERMINAL ID: 999999999</P>
NAME: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
ADDRESS:</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
CITY: XXXXXXXXXXXXXXXXXXXXXXXX</p>
STATE: XX ZIP:XXXXXXXXX</p>
PART CODE: 999999999</p>
UNIT PRICE: $9999.99</p>
SHIP METHOD:XXXXXXX CHARGE: $9999.99</p>
TOTAL AMOUNT: $9999.99</p>
</FORM>
<FORM ACTION=<file_name> METHOD=PAYMENT>
<INPUT TYPE=AUTOSUBMIT>
</FORM>
</BODY>
</HTML>
```

3.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.16B

EXHIBIT E
PAGE 24 OF 43

4.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=SCR1 METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
SLIDE CARD:
<INPUT TYPE="MSRT2" SIZE=40 NAME=track2>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="LOCAL" NAME=work_key>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

4.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99                                TIME:9999A</P>
ACCOUNT NUMBER  9999999999999999999</p>
EXP DATE:                                    99/99</p>
AMOUNT:                                      $9999.99</p>
AUTH NUMBER                                  999999999</p>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

4.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.17A

EXHIBIT E
PAGE 25 OF 43

5.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=SCR2 METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
ENTER PIN:
<INPUT TYPE="PASSWORD" SIZE=4 NAME=pin>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="LOCAL" NAME=work_key>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

5.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99 TIME:9999A</P>
ACCOUNT NUMBER 9999999999999999999</P>
EXP DATE: 99/99</P>
AMOUNT $9999.99</P>
AUTH NUMBER 999999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

5.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.17B

EXHIBIT E
PAGE 26 OF 43

6.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=host_URL METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
SLIDE CARD
<INPUT TYPE="MSRT2" SIZE=40 NAME=track2>
<INPUT TYPE="PIN" SIZE=4 NAME=pin>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

6.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99 TIME:9999A</P>
TERMINAL ID: 999999999</P>
ACCOUNT NUMBER 9999999999999999999</P>
EXP DATE: 99/99</P>
AMOUNT $9999.99</P>
AUTH NUMBER 999999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

6.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
<INPUT TYPE="LOCAL" NAME=work_key VALUE="9999999999999999">
</BODY>
</HTML>
```

FIG.18

EXHIBIT E
PAGE 27 OF 43

7.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM  ACTION=dbase_URL
        METHOD=SQL
                "IF EXISTS ( SELECT account FROM check_table)
                        SELECT * FROM check_table WHERE account = DDAaccount
                 ELSE SELECT * FROM error_table WHERE error_no=1">
<INPUT TYPE="LOCAL" NAME=tid>
SCAN CHECK:
<INPUT TYPE="MICR" SIZE=20 NAME=account>
 ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

7.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99                          TIME:9999</P>
TERMINAL ID:                           99999999</P>
ACCOUNT NUMBER   999999999999999999999</P>
AMOUNT                                  $9999.99</P>
AUTH NUMBER                            999999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

7.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.19

EXHIBIT E
PAGE 29 OF 43

8.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM  ACTION=dbase_URL
       METHOD=SQL
              * BEGIN TRAN
               IF NOT EXISTS ( SELECT substring(account,51 ,20) FROM auth_table)
                     BEGIN
                     SELECT cur_bal FROM cust_tbl WHERE substring(account,51 ,20)=account
                     SELECT amount = amount - ( points / .01 )
                     SELECT cur_bal = cur_bal + ( amount * .01 )
                     UPDATE TABLE=cust_tbl VALUES=( getdate(), account, cur_bal - points )
                     SELECT * FROM log_table WHERE trandate = getdate()
                     INSERT TABLE=log_table VALUES=( getdate(), tid, substring(account,51 ,20),
                     substring( account, 72, 4), amount)
                     END
       ELSE SELECT * FROM error_table WHERE error_no=1"
       COMMIT TRAN">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER ACCOUNT NUMBER:
<INPUT TYPE="MSRT1" SIZE=90 NAME=account>
ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
REDEEM POINTS?
<INPUT TYPE="INT" SIZE=6 NAME=points>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

8.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
JUNE 1 1995        10:30AM            PURCHASE</P>
TERMINAL ID:                          99999999</P>
ACCOUNT NUMBER     9999999999999999999</P>
EXP DATE                                  99/99</P>
AMOUNT                                $9999.99</P>
AUTH NUMBER                          999999999</P>
</p>
- - - - - - - - - - - - - - - - - - - - - - - </p>
           CUSTOMER SIGNATURE</p>
</p>
THANK YOU!</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
POINTS REDEEMED:                     999999</p>
POINTS EARNED:                       999999</p>
CURRENT POINT BALANCE:               999999</p>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

8.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <MESSAGE>
</BODY>
</HTML>
```

FIG.20

EXHIBIT C
PAGE 29 OF 43

9.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
      METHOD=SQL
            "SELECT fields FROM table WHERE condition">
<INPUT TYPE="LOCAL" NAME=tic>
ENTER SEARCH TABLE NAME:
<INPUT TYPE="TEXT" SIZE=10 NAME=table>
ENTER SEARCH FIELD NAMES:
<INPUT TYPE="TEXT" SIZE=100 NAME=fields>
ENTER SEARCH CONDITION:
<INPUT TYPE="TEXT" SIZE=50 NAME=condition>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

9.b. Transaction Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
FIELD1  FIELD2  FIELD3  ----  FIELDN  </p>
_____  _____  _____        _____  </p>
XXXXX  XXXXX  XXXXX        XXXXX  </p>
XXXXX  XXXXX  XXXXX        XXXXX  </p>
•
•
XXXXX  XXXXX  XXXXX        XXXXX  </p>
</FORM>
</BODY>
</HTML>
```

FIG.21

EXHIBIT E
PAGE 30 OF 43

10.a Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM  ACTION=dbase_URL
        METHOD=SQL
        * INSERT TABLE=log_table VALUES=( getdate(), tid, gross_sales, opn_chks, voids,
        emp_disc, mgr_disc, vip_card, man_over, coupons, sales_tax, c_dep1, c_dep2,
        c_dep3, c_dep4, chg_fund,cc_dep, batch_no, chrg_sales, paid_outs, co_sales,
        cc_sales, te_sales, gross_sales - opn_chks - voids - emp_disc - mgr_disc - vip_card-
        man_over - coupons - sales_tax, gross_sales - opn_chks - voids -
        emp_disc - mgr_disc - vip_card - man_over - coupons - c_dep1 - c_dep2 -
        c_dep3 - c_dep4 - chg_fund - cc_dep - batch_no - chrg_sales - paid_outs)
        SELECT * FROM log_table WHERE trandate = getdate()">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER GROSS SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=gross_sales>
ENTER  OPEN CHECKS:
<INPUT TYPE="INT" SIZE=7 NAME=opn_chks>
ENTER VOIDS:
<INPUT TYPE="INT" SIZE=7 NAME=voids>
ENTER EMP DISCOUNTS:
<INPUT TYPE="INT" SIZE=7 NAME=emp_disc>
ENTER MGR DISCOUNT:
<INPUT TYPE="INT" SIZE=7 NAME=mgr_disc>
ENTER VIP CARD:
<INPUT TYPE="INT" SIZE=7 NAME=vip_card>
ENTER MANUAL OVERRINGS:
<INPUT TYPE="INT" SIZE=7 NAME=man_over>
ENTER COUPONS:
<INPUT TYPE="INT" SIZE=7 NAME=coupons>
ENTER SALES TAX:
<INPUT TYPE="AMT" SIZE=8 NAME=sales_tax>
ENTER CASH DEPOSIT 1:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep1>
ENTER CASH DEPOSIT 2:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep2>
ENTER CASH DEPOSIT 3:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep3>
ENTER CASH DEPOSIT 4:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep4>
ENTER CHANGE FUND:
<INPUT TYPE="AMT" SIZE=8 NAME=chg_fund>
```

FIG.22A

EXHIBIT E
PAGE 31 OF 43

```
ENTER CC DEPOSIT:
<INPUT TYPE="AMT" SIZE=8 NAME=cc_dep>
ENTER BATCH #:
<INPUT TYPE="INT" SIZE=3 NAME=batch_no>
ENTER CHARGE SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=chrg_sales>
ENTER PAID OUTS:
<INPUT TYPE="INT" SIZE=8 NAME=paid_outs>
ENTER CARRY OUT SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=co_sales>
ENTER CREDIT CARD SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=cc_sales>
ENTER TAX EXEMPT SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=te_sales>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

10.b. Transaction Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
JUNE 1 1995        10:30AM        DAILY REPORT</P>
TERMINAL ID:                          999999999</P>
GROSS SALES                           999999.99</P>
 VOIDS                        99        99999.99</P>
 EMP DISCOUNTS                99        99999.99</P>
 MANAGER DISCOUNTS            99        99999.99</P>
 VIP CARD                     99        99999.99</P>
 COUPONS                      99        99999.99</P>
 MANUAL OVERRINGS             99        99999.99</P>
 SALES TAX                              999999.99</P>
 CASH DEPOSIT 1                         999999.99</P>
 CASH DEPOSIT 2                         999999.99</P>
 CASH DEPOSIT 3                         999999.99</P>
 CASH DEPOSIT 4                         999999.99</P>
 CASH DEPOSIT 5                         999999.99</P>
 CHANGE FUND                            999999.99</P>
 CC DEPOSIT                   999       999999.99</P>
 CHARGE SALES                           999999.99</P>
 PAID OUTS                     99        99999.99</P>
 CARRY OUT SALES                        999999.99</P>
 CREDIT CARD SALES                      999999.99</P>
 TAX EXEMPT SALES                       999999.99</P>
 ----------------------------------------</P>

 NET SALES                               9999999</P>
 OVER/SHORT                              9999999</P>
</FORM>
</BODY>
</HTML>
```

FIG.22B

EXHIBIT E
PAGE 32 OF 43

11.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM  ACTION=MAIL TO: mail_to>
 ENTER MAIL ADDRESS:
<INPUT TYPE="TEXT" SIZE=20 NAME=mail_to>
ENTER MESSAGE:
<INPUT TYPE="TEXT" SIZE=100>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

FIG.23

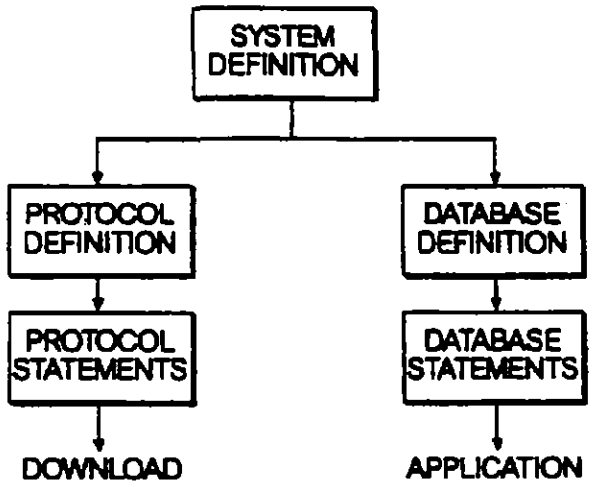

FIG.24A

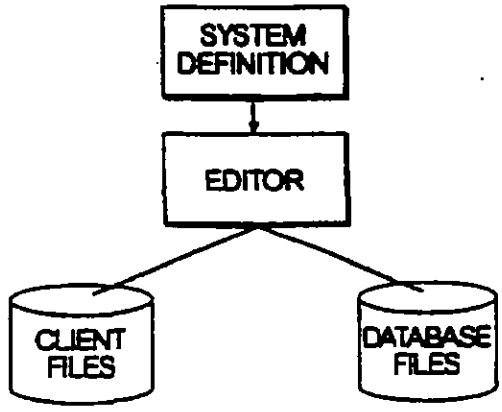

FIG.24B

EXHIBIT E
PAGE 33 OF 43

SYSTEM FOR ENABLING SMART CARD TRANSACTIONS TO OCCUR OVER THE INTERNET AND ASSOCIATED METHOD

This application is a continuation of co-pending application Ser. No. 09/907,076, filed on Jul. 17, 2001, which is a continuation of application Ser. No. 09/314,266, filed on May 18, 1999, now U.S. Pat. No. 6,366,967 which is a continuation of application Ser. No. 08/995,123 filed Dec. 19, 1997 (now U.S. Pat. No. 5,905,908), which is a continuation of application Ser. No. 08/493,772 filed Jun. 22, 1995 (now U.S. Pat. No. 5,742,845).

FIELD OF THE INVENTION

This invention relates to data transaction systems, and more particularly, to data transaction systems using non-standard input/output devices.

BACKGROUND OF THE INVENTION

Data transaction systems which communicate with a plurality of remote terminals to transfer information used to complete a transaction or compile a database are well known. Typically, such systems include a central transaction processing system which may maintain a database of information such as customer or consumer data. Exemplary information in such a database may include customer identification, customer account numbers, credit limits and/or account balances from which a customer may draw. The central transaction processing system is typically coupled to a plurality of remote transaction or data input terminals. Transaction computers may include special purpose devices such as automatic teller machines (ATMs), point of sale (POS) terminals, credit card terminals, and screen phone terminals. Screen phone terminals are devices which integrate a telephone with an ATM-like device and possibly a magnetic card swipe reader. Data input terminals may include personal computers (PCs) interfaced to data collection devices or special purpose data collection terminals or monitors.

In these known data transaction systems, a user usually initiates a transaction by requesting access to funds in an account or from a credit line maintained by the central processing system. The request is transmitted to the central processing system which performs a verification to determine whether the user is a valid user of the system, has an account within the system, and that the amount of the transaction is within the limits of the consumer's credit line or that the user has the requested funds available in an existing account monitored by the central processing system. The central processing system then transmits authorization for or denial of the transaction to the remote terminal. In response to the message from the central processing system, the remote terminal dispenses cash (for an ATM) or the merchant provides the goods being purchased to the user if the authorization message indicates that the consumer's funds will be transferred to the merchant's account. Similar communication exchanges occur in data systems where electronic documents and other information are provided to a central site for compilation or processing. Consequently, this background discussion applies to all such transaction and data systems. Though the remainder of the discussion is directed to transaction systems, the reader should appreciate that the comments also apply to data systems as well.

The remote terminals may be coupled to the central processing system in several ways. For example, in some ATM systems, the ATMs are coupled to the central processing system through dedicated telephone or other data communication lines. These systems are preferred because they provide a relatively high degree of security since the dedicated data line coupling the central processing system to the ATM is not generally accessible by members of the public. The physical security of the dedicated data line is, however, expensive because no other traffic may utilize the line. Thus, the cost of leasing the dedicated line to an ATM with relatively low volumes of transactions may yield a high communication cost per transaction.

In an effort to reduce the communication cost per transaction, some transaction or data systems utilize telephone lines through a publicly-switched telephone network (PSTN) which may be accessed by other members of the public. Specifically, devices such as credit card terminals and screen phone terminals typically include a modem which converts the digital messages of the remote terminal into frequency modulated analog signals which may be transmitted over telephone lines to a modem at the central processing system. In other systems, the terminal may communicate digital data directly over ISDN lines of the PSTN to the central processing system. This line of communication between a remote terminal and the central processing system is performed by having the remote terminal dial a telephone number associated with the central processing system to establish communication with the central processing system. This type of communication path is relatively secure because the switching networks for the communication traffic through the PSTN are not readily accessible by the public and during the course of the financial transaction, only the central processing system and remote terminal are on the line.

Regardless of the communication method used to couple the central processing system to the remote terminals, the protocol and data formats used between the devices is typically proprietary. That is, the operator of each financial transaction system designs its own protocol and data message format for communication with the processor at the central site or generates a variant within a standard such as those established by the ANSI committee or the like for such communication. As a result, the remote terminals must include software that supports each operator's protocol and message formats in order to be compatible with an operator's central site. For example, application software in a credit terminal such as the TRANZ330, TRANZ380, or OMNI390 manufactured by VeriFone implement one or more of the communication protocols and formats for National Data Corporation (NDC), VISAINET, MASTERCARD, BUYPASS, and National Bancard Corporation (NaBANCO) system processors in order to support transactions with the most popular transaction centers. Thus, the communication software absorbs a significant amount of terminal resources which could be used to support other terminal operations.

A related problem arises from the expanding home banking market. A customer of home banking system typically uses a screen phone terminal or a personal computer (PC) having a modem to establish communication through a PSTN to a central transaction processing system. Again, the operator of the central processing system must provide information regarding the data message formats for communicating with the central processing system to a vendor of software for the home banking terminals or must provide that software to its customers. As a result, home banking customers must purchase software to communicate with each banking system of which the customer wants to be a member. This cost and the need to install additional com-

munication programs may make some consumers reluctant to be a member of more than one banking system or to change banking systems.

A communication system becoming increasingly popular and which provides standardized communication is the Internet. The Internet is an open network of networks which communicate through a variety of physical communication devices such as telephone lines, direct communication lines, and the like. Each network is coupled to the main Internet network for communication through a host computer supporting a TCP/IP router or bridger. The host computer typically includes a program, frequently called a Web server, which acts as a gateway to resources at the host computer which may be resident on the host computer or a network coupled to the host computer. Each server has an address identifying the location of the resources available through the Web server. The router recognizes communication for the server and directs the message to the server or it recognizes that the communication should be forwarded to another server. As a result, communication within the Internet may be point-to-point, but more likely, the communication path is a somewhat circuitous one with the information passing through the routers of multiple servers before reaching its final destination.

A number of message protocols and formats have been developed for the Internet. The physical communication protocol and data message format is the Transport Control Protocol/Internet Protocol (TCP/IP). The TCP/IP protocol involves multiple layers of encapsulating headers containing communication information which are used to provide byte streams or datagram communications to computers on the networks coupled to the Internet. Encapsulated within TCP/IP headers are protocols which are used to format the data messages or transfer data from one computer to another computer coupled to the Internet. These protocols include File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), Telnet, and Hyper Text Transport Protocol (HTTP). The advantage of these protocols is that each provides a standardized communication format for transferring information between computers on the Internet. These protocols are typically called open system protocols as they are publicly known and may be utilized by any programmer to develop programs for communicating with another computer coupled to the Internet. These non-proprietary protocols have contributed to the acceptance of using the Internet as an open network for coupling computer networks together.

While the Internet provides an open network for computer communication with publicly accessible protocols and formats, the Internet suffers from a number of limitations which preclude its effective use as a transaction or data system which uses non-standard I/O terminals and devices. First, circuitous communication presents a number of security issues for such a system. For example, a Web server could incorporate a router which examines the address of each message coming through it and upon recognizing an address associated with a central transaction processing system, copy the data message for the unauthorized retrieval of customer-sensitive information such as account numbers and personal identification numbers (PINs) which may be contained in the message.

A second limitation of open networks such as the Internet is that communication on such networks is only supported for computers acting as servers or clients. Specifically, all of the protocols and formats are constructed for standard input/output (I/O) operations for a PC terminal. That is, text information is directed to a standard monitor screen, user input is expected from a standard keyboard, and files are transferred to standard peripherals such as a hard disk or diskette drive. Especially absent is the ability in open network protocols for communication with devices that only use communication interfaces such as RS-232C. As a result, communication over the Internet is primarily performed with standard PCs through network communication methods and interfaces.

This presents a number of problems for home banking or for interfacing non-standard I/O terminals such as credit card terminals or screen phones to open networks such as the Internet either directly or through a PC. Generally, non-standard I/O devices are devices which interface to a PC through a port not normally used for networks, such as a RS-232C port, or are devices which have limited input and output capabilities such as small screen displays or ten keypads. These devices are not supported on the Internet because servers use protocols that communicate with PCs supporting standard QWERTY keyboards and standard monitors. Consequently, users are limited to entering account numbers and the like through a keyboard of a PC-like device for processing at a central transaction processing system. To request a transaction, one need only have a person's credit card account number. If the credit card number had to be input through a magnetic card reader, unauthorized access to a customer's account would be less likely since physical possession of the credit card would be required to initiate the transaction.

Another limitation of the standard I/O devices currently supported by the open network protocols is the lack of encryption. For example, PIN pads, which are typically incorporated in ATMs, automatically encrypt in hardware a PIN entered by a user. Such devices typically encrypt the number by implementing a data encryption standard (DES) algorithm in hardware before the PIN is transmitted or stored. When a standard keyboard is used to input the PIN, no hardware encryption is performed and, as a result, an unencrypted copy of the PIN is provided to the memory of the PC. Storage of unencrypted PINs is in contravention of current banking regulations. If PIN pads could be read via Internet protocols, then such a lapse in PIN security would be less likely to occur.

Another I/O device not supported on open networks are smart cards which are increasing in use. Smart cards include a processor and memory in which information regarding the amount of funds in a particular account, a transaction history, account numbers, and customer data may be stored. The card may be read through a smart card reader which is a computer having a processor and memory but usually provided with non-QWERTY keypads and limited displays. A transaction processor may validate a card owner through a PIN provided through a keypad, determine the amount of money remaining on the card and debit the card itself for a transaction amount by communicating with the smart card reader with one of the proprietary protocols discussed above. Such information is not readily obtainable by the owner of the card and so cannot be entered through a keyboard or the like. Smart card readers are non-standard devices which may be coupled to a PC through a COMM1 or COMM2 port. However, none of the standard protocols and message formats for open network communications currently provide I/O operations for such devices.

All systems which attempt to provide three party communication to execute an electronic transaction suffer from a number of limitations which present risks greater than those in a normal transaction performed at the point of sale. In a typical point of sale (POS) transaction, the consumer

hands a debit or credit card to a merchant's agent who may examine the card for security markings such as holograms, watermarks, or a cardholder signature. The agent then places the card into a reader for acquiring information from the card and, in some cases, have the consumer enter a PIN into a PIN entry device which encrypts the PIN in a hardware implemented scheme. If the PIN is entered, it is transmitted with the information from the card to a processing center, typically in one of the formats discussed above, under a X.25 protocol or the like. The processing center returns an authorization granted or denied message. The reader typically has a printer coupled to it through an RS-232C port or the like and a purchase agreement is printed. The consumer signs the agreement, the merchant's agent may verify the signature, and the merchant retains an original of the agreement and the consumer a copy. In this scenario, the merchant has initiated the communication to the processing center. The safeguards noted above permit the processing center to charge a merchant a lower processing fee than when a consumer initiates a transaction. Consumer initiated transactions present a greater risk because the consumer provides an agent an account number in a telephone conversation or non-encrypted DTMF transmission. Thus, there is no card inspection, signature verification, or PIN verification. As a result, such transactions are limited to credit cards because debit cards require that the cardholder be present to enter a PIN into an appropriate PIN entry device.

What is needed is a system that permits consumers remote from a merchant to order goods and present payment in a secured manner so the merchant's risk and processing costs, as well as a cardholder's exposure to fraud, is reduced. What is needed is a way for a processing center to communicate through an open network with non-standard I/O devices such as credit card terminals, personal digital assistants, and screen phone terminals or with non-standard I/O devices coupled to the open network through a PC or the like. What is needed is a transaction or data system which utilizes an open network such as the Internet to support electronic transactions or data compilation in a secure manner without undue limitation as to the devices with which communication may be made.

SUMMARY OF THE INVENTION

The present invention provides transaction and data systems which may be implemented on an open network such as the Internet. The system comprises a server for communicating in an open network protocol and a plurality of input/output (I/O) devices coupled to the server through an open network, the I/O devices communicating with the server in the extended open network protocol that supports communication with non-standard I/O devices over the open network. The system of the present invention provides a server with the capability of communicating with a number of I/O devices useful in transaction and data systems which heretofore have been unsupported on an open network system such as the Internet.

The system of the present invention is implemented by extending present open network communication protocols and data message formats to communicate with non-standard I/O devices either coupled to an open network as a client or coupled to an open network through a client, such as a PC, credit card terminal, screen phone, or PDA. That is, commands which are compatible with the communication schema of a presently-implemented protocol for the Internet are used and additions are made to commands implemented within the control structure of that existing protocol to support non-standard I/O device communication. At the server, the extended protocol is further supported by a common gateway interface (CGI) which converts the communication from a non-standard I/O device to a format which is compatible with a transaction or data application program which may be executed on the server or a computer coupled to the server. In this manner, the CGI permits the processing of the extended capability commands to be segregated from the communication functions performed by the server.

Preferably, the server and the I/O devices communicate through an Internet protocol and most preferably, the Hyper Text Transport Protocol (HTTP), to exchange data between an application program and non-standard I/O devices over an open network. Although HTTP is the preferred protocol used to implement the present invention, other protocols such as Telnet or SMTP, for example, may also be extended in a similar manner. Specifically, the HTTP protocol is expanded to communicate with printers, magnetic card readers, credit card terminals, smart card readers, check readers, PIN pads, bar-code readers, PDAs, or the like, and includes a command which instructs a non-standard I/O device to disconnect from the open network and re-couple to a transaction processing system to transfer funds from a consumer account to a merchant account through a PSTN or dedicated data line. By using these extended capability commands within HTTP, a processing system may operate on an open network such as the Internet and communicate with transaction or other data I/O devices which have not previously been able to couple to such open networks. Such a system may be used to execute a transaction between a consumer and a merchant so the merchant receives remittance information in a timely manner. The system permits the consumer to initiate a transaction and order from a merchant and then use a more secure link supported by PIN entry devices or the like to reduce the risk of fraud for the transaction.

Because the server may communicate through such open networks with non-standard I/O devices, the transaction or data processing system is available for the ever-expanding market available through the Internet. Such a system is able to communicate with non-standard I/O devices in myriad locations such as retail establishments or in consumers' homes. For example, a consumer may utilize the standard capability of an Internet protocol to communicate with a server that provides information regarding services or goods for sale over the Internet and then consummate a sales transaction by using the extended capability of the Internet protocol. Such a home consumer could provide transaction data through a smart card reader coupled to a COMM1 or COMM2 port of a PC. A database program executing at the server for the central processing site may accept product ordering information from a non-standard keypad or touch screen associated with a screen phone terminal at the remote site and then communicate with the smart card reader to consummate the transaction. Such a transaction system requires that the consumer have physical possession of the smart or credit card and not simply knowledge of the account number. Likewise, the server would be able to communicate with a PIN pad or the like to ensure the hardware encryption of PINs and other data before it is transmitted to the server site. Such a system is less susceptible to consumer fraud.

Another feature of the present invention is a PAYMENT command implemented in the extended Internet protocol that directs a non-standard I/O device or a PC interfaced with such devices to communicate with a transaction processor through an alternative communication link. In one

form, the PAYMENT command is used by a merchant terminal to submit a consumer's account number with a merchant deposit account number through a PSTN network or the like to the processing center. In another form of the PAYMENT command, a client program in a consumer's terminal receives an account number for a merchant account from a merchant's server with the PAYMENT command. On receipt of this command, the client program suspends its operation and passes the account number to a conventional bank processing program co-resident in memory. The bank processing program establishes a standard communication link with a transaction processing system through a dedicated data line or a PSTN network. Using that communication link, the bank processing program executes a commercial transaction using a standard VISA protocol or the like. The consumer may use a magnetic stripe reader and a PIN entry device to improve the security of the data transmission. The transaction center may transmit remittance data over the open network to the merchant so the merchant is apprised of payment and ships the ordered product. Once this consumer initiated transaction is complete, the bank processing program terminates and returns control to the client program which may terminate communication with the open network or retrieve information from another server on the open network for another transaction. In this way, the user may use the open network for non-confidential communication such as collecting product information, pricing, and product availability. This information may be collected quickly and efficiently using the extended Internet protocol. The conventional bank processing program and more secure communication links may then be used for the confidential information required for the transaction. Thus, the present invention is able to combine the features and advantages of the Internet with the more secure communication link and data security enhancing devices of systems presently known.

Preferably, an editor is provided which permits a user to define an application database table with data fields, define client application data fields, and define the integrated forms for communicating data between the defined database tables and a client application. The editor verifies the syntax of the user generated integrated forms containing extended Internet protocol statements and client application statements. The editor ensures that the variable names for the client application and the data fields for the database application correspond. Following the generation of the integrated form, the editor parses the integrated form to segregate the database language statements from the extended Internet protocol statements. A database language identifier is substituted in the Internet protocol statements for the database statements contained in the integrated form. The Internet protocol statements are downloaded as a file which is interpreted by the client program for the collection and submission of data from non-standard I/O devices to the database application. The database language statements segregated from the extended Internet protocol statements are placed in a second file which is named to correspond to the database table defined by the user. The CGI application recognizes the database language identifier contained in the returned forms of the Internet protocol statements. The CGI application correlates the database identifier with the file previously generated by the editor which contains the database command statements. The application then inserts the data from the returned form into the database command statements and provides the re-integrated database command statements to the database application. In this manner, the database may be queried by or retrieve data from the non-standard I/O device. In the most preferred embodiment, the editor permits a user to develop integrated forms comprised of the extended HTML language and standard query language (SQL) database application statements. In this manner, the user does not have to manually generate the SQL commands, the HTML commands, and carefully correlate the data fields of the two commands in order to implement a transaction between a client and a database.

These and other advantages and features of the present invention may be discerned from reviewing the accompanying drawings and the detailed description of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention may take form in various components and arrangement of components and in various steps and arrangement of steps. The drawings are only for purposes of illustrating a preferred embodiment and are not to be construed as limiting the invention.

FIG. 1 is a diagram of an open network system in which the present invention is utilized;

FIG. 2 is a diagram of the format of the FORM and INPUT tags implemented in the preferred embodiment of the present invention;

FIG. 3 is a diagram of the preferred SQL commands supported in the preferred embodiment of the present invention;

FIG. 4 is a flowchart of the high level processing of the client program which interprets the HTML files of the preferred embodiment of the present invention;

FIG. 5 is a flowchart of the HTML file processing performed by the client program of the preferred embodiment of the present invention;

FIG. 6 is a flowchart of the attribute processing for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. 7 is a flowchart of the processing of the ACTION attribute for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. 8 is a flowchart of the processing for the METHOD attribute for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. 9 is a flowchart of the attribute processing for the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. 10 is a flowchart of the processing for the TYPE attribute for the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. 11 is a flowchart of the processing for the NAME attribute of the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. 12 is a diagram of the format for the ACTION attribute for the FORM tag performed by the common gateway interface between the Web server and an application program;

FIG. 13A is a diagram of the possible communication paths which may be used by an I/O device according to the principles of the present invention;

FIG. 13B shows an exemplary FORM tag and INPUT tag for the PAYMENT method implemented in a merchant's terminal according to the principles of the present invention;

FIG. 13C shows an exemplary FORM tag and INPUT tag for the PAYMENT method implemented in a consumer's terminal according to the principles of the present invention;

FIG. 14 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to

generate the HTML files for the client program and the SQL files for the application program for a card initiated payment authorization and capture transaction;

FIGS. **15A** and B show exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a bar code reader input with card-initiated payment authorization transaction;

FIGS. **16A** and B show exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a key input order with secure payment transaction;

FIG. **17A** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a smart card debit (Type 1) transaction;

FIG. **17B** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a smart card debit (Type 2) transaction;

FIG. **18** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a debit card transaction;

FIG. **19** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a check verification transaction;

FIG. **20** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a customer frequency transaction;

FIG. **21** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for an item search transaction;

FIGS. **22A** and B show exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for retail store end of day reporting;

FIG. **23** shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a store reporting an e-mail transaction;

FIG. **24A** is a diagram of a manual development process for the files interpreted by the client program and the files interpreted by the application program in accordance with the principles of the present invention; and

FIG. **24B** is a diagram of the generation of the files interpreted by the client program and the files interpreted by application program performed by an editor constructed in accordance with the principles of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

A transaction or data system constructed in accordance with the principles of the present invention is shown is FIG. **1**. The system **10** includes a Web server **12** which is coupled to an open network **14** such as the Internet for communication with various I/O devices and terminals. For example, the I/O devices which may be coupled directly to network **14** include standard I/O devices already supported by Internet protocols such as PCs **30** and non-standard I/O devices such as a screen phone terminal **16**, a personal digital assistant (PDA) **18**, and a credit card terminal **20**. Other exemplary non-standard I/O devices such as smart card reader **32**, personal identification number (PIN) pad **34**, magnetic card swipe reader **36**, printer **38**, or the like, may be coupled to PCs through non-standard I/O ports such as COMM1 and COMM2 ports or to other non-standard I/O devices such as phone terminal **16**, PDA **18**, or credit card terminal **20**. Typically, these devices are coupled to PCs or devices **16**,**18**, or **20** through an interface such as a RS-232C interface. Merchants or other vendors may use a Web server **2** to couple to network **14** to communicate with the devices and processing system **40**.

The Web server **12** is preferably coupled to a Common Gateway Interface (CGI) application **28** which converts and communicates the data and commands between the devices on network **14** and the processing system **40** so the I/O devices do not have to use the database command language to interact with the database. System **40** and the devices may communicate directly if they are implemented in the same language or if a user implements a communication interface such as CGI **28** that correlates data fields in the client with those in system **40**. Server **12**, CGI **28**, and the applications supporting system **40** may all reside on a single host computer or they may reside on separate computers coupled together by a local area network (LAN) or a wide area network (WAN). Preferably, the application interfaces with a database which supports Open Data Base Connectivity (ODBC) and Structured Query Language (SQL).

The communication sessions between the I/O devices coupled to the open network **14** and the Web server **12** are generally conducted in the same fashion as Internet protocol communication sessions are currently performed. That is, the I/O device establishes a communication connection with Web server **12**, sends a request to the Web server, the Web server responds to the request and the I/O device or server closes the connection. Preferably, the non-standard I/O devices or PCs interfaced to such devices selectively couple to a local access port on the open network **14** through a local modem/ISDN connection. In this manner, the device is only coupled to the open network **14** when a transaction or a data operation is to be performed. While connected to the open network **14**, a device may access a number of servers to accomplish a purpose. For example, a device may couple to a local access port and communicate with a first server to check inventory levels at a site, communicate with a second server to order stock for the inventory, and communicate with a third server to settle payment for the ordered goods. When all aspects of the transaction are complete, the connection with the local access port is terminated. In the preferred embodiment of the present invention, the protocol used to transport data messages between Web server **12** and the I/O devices coupled to the open network **14** is the Hyper Text Transport Protocol (HTTP), although other open system protocols utilized on the Internet may be used.

In standard HTTP protocol, a client program executing in one of the I/O devices may initiate communication with a server by sending a query message of the format:

```
http://<host>:<port>/<path>?<search part>
```

The message identifies the client as seeking communication with a HTTP server at the host address on the specified

EXHIBIT E
PAGE 39 OF 43

port. In the HTTP protocol, the default value for the port is 80 and the host address is the Internet protocol (IP) address of the type well-known in the art. The path value selects the file in the HTTP server which is activated in response to the message and the search part specifies a query for the selected file. In the initial communication, the query may be omitted so that the selected host file responds to the client program before a query is processed.

In the present invention, the client program uses a similar message to initiate a transaction or data operation, except that database commands are preferably embedded in a file at the server 12 and not in the "search part" of the command, although search parts may be constructed in accordance with the principles of the present invention that support non-standard I/O devices. Preferably, the client program interprets Hyper Text Markup Language (HTML) files containing HTML commands for communicating data between non-standard I/O devices and server 12. Most preferably, the HTML commands contain identifiers which are used by the CGI to place data returned in the forms of the HTML commands into database commands for queries or data insertions for the database. HTML is a command language well known for the retrieval and display of electronic documents for standard I/O devices such as PCs supported by full screen monitors, QWERTY keyboards, and standard peripherals such as hard disk drives and diskette drives. Standard HTML commands use text and previously known commands that reference Universal Resource Locators (URLs) to support the communication of electronic documents. These documents are files which may contain HTML commands, text, audio, video, or image data. The present invention extends HTML with commands that support communication between the server and the non-standard I/O devices.

In the HTTP protocol, data may be obtained during a communication session by using a tag called a FORM as part of the file defined by <path> in the command discussed above. The FORM format for standard HTTP is:

```
<FORM ACTION="URL"
    METHOD=GET|POST
>
Command
</FORM>
```

where "|" is an "OR" operator. The commands supported by standard HTTP are INPUT, SELECT, and TEXTAREA. Additionally, standard HTTP permits the inclusion of text data in the command area. In the present invention, HTML has been extended to support new ACTIONs, METHODs, and INPUTs.

In accordance with the principles of the present invention, tags are preferably used to identify device transfers and input operations. Preferably, the FORM tag is used to identify device transfers and ACTION and METHOD attributes further identify the device operation. As shown in FIG. 2, the extended ACTION field may include a FROM and TO attribute for accessing a local terminal file or smart card reader or a TO PRINTER attribute for directing output data to a printer local to the I/O device. The FROM and TO attributes for accessing local files and smart card readers and for directing output data to a local printer have previously been unsupported in any Internet protocol. As a result, the server 12 may access non-standard I/O peripherals for any of the I/O devices used in the transaction or data system 10. The ACTION="URL" is a part of standard HTTP and is well known.

The METHOD attributes may include the GET, POST, PAYMENT, or SQL methods. The GET and POST methods are currently supported in standard HTTP and are well known. The PAYMENT attribute is a directive to deliver data retrieved by an INPUT command to a private payment network for authorization and settlement and is not available in current Internet protocols. This directive is used by the client program to activate a conventional financial transaction application which communicates with the transaction system over a dedicated data line or PSTN in a known protocol such as VISA. Such an attribute is used where the more secure physical connection between remote site and transaction system and data encryption devices or the like are preferred. The SQL method preferably identifies a database language file which CGI 28 uses to correlate data in the HTML FORM to an insertion or query command contained in the file.

The preferred format for the INPUT tag which is used to identify input operations is also shown in FIG. 2. The TYPE and NAME attributes are used to define a non-standard I/O device or local storage variable for the input of data. The TYPE field values "text," "password," "checkbox," "radio," "submit," and "reset" are previously known, as are the attributes NAME, VALUE, CHECKED, SIZE, and MAXLENGTH. To support the extended capability of the present invention, the TYPE attribute preferably includes attributes MSRT1 for reading track 1 of a magnetic swipe reader, MSRT2 for reading a magnetic swipe reader track 2, KEY for reading input from a terminal command keypad, PIN for reading a personal identification number pad, BCW for reading a bar code wand, MICR for reading a check magnetic code reader, AIM for reading a dollar amount via a key input mask, INT for reading an integer via a key input mask, database with the INSERT attribute or update data already existing in a database with the UPDATE attribute. The values for the INSERT attribute may be identified with the VALUES attribute, and the SET and WHERE attributes may be used to define and conditionally update values in the identified database. Preferably, the present invention implements two DELETE and CREATE attributes. The DELETE attribute deletes all items in an identified column of a database table which may satisfy a condition defined by a WHERE attribute. The CREATE attribute creates a database table having a primary key identified by the PRIMARY KEY attribute.

Preferably, the server program executes on a computer system having at least an Intel 80386 or better processor with at least 4 megabytes of RAM and at least 3 megabytes of hard disk space available. The computer system running the server may operate any known server platform operating system such as WINDOWS 3.1, WINDOWS 95, or WINDOWS NT, UNIX, AIX, and others. The non-standard I/O devices require a processor of a Z80A type or better, at least 32K bytes of RAM, and at least 32K bytes of ROM. The device includes a modem capable of at least 1200 bits-per-second (bps) but other modem speeds may be used for communication between client and server. Alternatively, the device may be coupled to a LAN which in turn is coupled to the Internet for communication with server 12. A typical non-standard device which executes the client program is a VeriFone OMNI390, OMNI395, or VuFone terminal. OMNI390, OMNI395, and VuFone are trademarks of VeriFone, Inc., of Redwood City, Calif. Other exemplary devices include Phillips Screen phone, Hypercomm T7 terminal, and Apple Computer Newton MessagePad.

To build the preferred HTML files which CGI 28 preferably uses to implement the client program and database

EXHIBIT E
PAGE 39 OF 43

application, the user preferably uses an off-line editor. The files generated by the editor are preferably comprised of an integrated statements formed from HTML statements and database statements for retrieving and writing data with the database. Exemplary files showing such integrated statements for performing transactions are depicted in FIGS. 14–23B. After such a file is generated, the editor parses the integrated statements into HTML statements and into database statements such as SQL commands. The HTML files required by the client program to support communication with a transaction or data processing center may be downloaded to a device or PC for execution. The files containing the database application statements used by the CGI interface to communicate data with the database application program preferably reside on server **12**. Preferably, the database files used by the CGI interface include SQL commands for the application program interfaced to an ODBC compliant database.

The general format of the HTML commands in the HTML files used for communication with a client program and server are of the general format: TAG ATTRIBUTE. Preferably, the TAG field may be one of FORM, INPUT, SQL, or TEXTAREA. The ATTRIBUTE field value depends upon the TAG value. Preferably, the FORM tag may include the ACTION or METHOD attributes where the ACTION attributes include the FROM<file>, TO PRINTER, TO<file>, and TO SCR values noted above, as well as the standard HTML ACTION value of URL=<file>. The METHOD attributes include the PAYMENT and SQL attributes noted above, as well as the standard HTML METHOD values of GET and POST. Also in accordance with the principles of the present invention, the INPUT tag may include TYPE, NAME, VALUE, CHECKED, SIZE, and MAXLENGTH attributes. These attributes are previously supported for the INPUT tag in HTML, however, the present invention further includes TYPE values of MSRT1, MSRT2, KEY, PIN, BCW, MICR, AMT, INT, LOCAL, and AUTOSUB, as well as the standard HTML TYPE values of TEXT, PASSWORD, CHECKBOX, RADIO BUTTON, SUBMIT, and RESET. The present invention also supports NAME attributes of IP_ADDRESS, HOST_PHONE, TID, WORK_KEY, DATETIME, and DEPOSIT_ACCT to identify local storage areas as well as standard HTML NAME attribute <Field_NM> to identify a FORM variable.

The preferred high level processing of the client program is shown in FIG. 4. That processing includes an idle step (Block **100**) in which the program performs general housekeeping tasks such as maintaining internal time, scanning for input which may activate the device, or other known functions. Further processing is activated by some operator action at the device or PC which causes the device to either open a remote URL (Block **102**) or open a local URL (Block **104**). If a remote URL is required, the device transmits a message of the format discussed previously which is routed through the open network and delivered to a server **12** for a transaction or data processing system (Block **106**). The HTML file selected at the server **12** is identified by the remote URL in the initial communication between the device and server **12** and that URL is used to return the selected HTML file to the device for processing (Blocks **108**, **110**).

FIG. **4** also shows that an operator may initiate an open local URL function by typing in a command or by pushing a hot key which is associated with a local URL. The I/O device reads the HTML file identified by the URL from local memory (Block **112**) and passes the HTML file to the function for processing HTML files (Block **110**). After a file is processed (Block **110**), the client program determines whether the HTML file is to be stored (Block **114**). If it is not, the process returns to the idle processing (Block **100**). Otherwise, the process determines whether the HTML file is to be associated with a hot key (Block **116**) and, if it is, it stores the file and generates the link between a hot key and the stored file (Blocks **118**, **120**). If the HTML file is only to be stored, no association is made with a hot key and the file is simply stored in local memory (Block **20**). The client program then returns to idle processing (Block **100**).

The high-level processing for the HTML file (Block **110**, FIG. **4**) is shown in further detail in FIG. **5**. The process begins by scanning the HTML file for a TAG (Block **140**). If no TAG is found, the file is not in proper format for processing and processing returns to Block **114** discussed in FIG. **4** above. If a TAG is found (Block **142**), the process determines whether the TAG is a FORM TAG (Block **144**) or an INPUT TAG (Block **146**). If it is a FORM TAG, then the FORM TAG is processed and the program continues by looking for other TAGS to process (Block **140**). If the TAG is an INPUT TAG, the INPUT TAG is processed (Block **150**) and the program continues by looking for other TAGS to process (Block **140**). If the TAG is one of the standard HTML TAGS, the program implements the TAG in standard known ways (Block **152**) and then scans for other TAGs to process (Block **140**).

Processing the ATTRIBUTES used to implement a FORM TAG is shown in FIG. **6**. That process continues by scanning the HTML file for an attribute (Block **160**). If an attribute is not found (Block **162**), the program returns to scan for other TAGS (Block **140**, FIG. **5**). If an attribute is found, the program determines whether it is an ACTION attribute (Block **164**) or a METHOD attribute (Block **166**). Depending on the type of attribute, the appropriate function for processing the attribute is executed (Blocks **168** or **170**) and scanning for additional attributes continues (Block **160**). If the attribute is not an ACTION or METHOD attribute, there is an error in the file and processing returns to scan for other TAGs.

The processing for the ACTION attribute is shown in FIG. **7**. There, the ACTION attribute is examined to determine whether it is a FROM<file> (Block **180**), TO PRINTER (Block **182**), TO<file> (Block **184**), TO SCR (Block **186**), FROM SCR (Block **188**) or a URL=<file> (Block **192**). The URL=<file> ACTION is a standard HTML action which is processed in a known way (Block **194**). The FROM <file> action is processed by reading data from a file associated with the I/O device or PC interfaced to the I/O device (Block **196**). The TO PRINTER action results in data in the FORM being sent to the printer (Block **198**) while the TO <file> action results in data in the FORM being written to a local file (Block **200**). The TO SCR action causes data to be written to the smart card via a smart card reader (Block **202**) and FROM SCR reads data from a smart card through a smart card reader (Block **204**). After the appropriate action processing takes place, the HTML file is scanned for additional ACTION values to perform (Block **206**), and if one is found, the process continues. If no attribute is located (Block **208**), the process returns to scan for other attributes (Block **160**, FIG. **6**).

The processing for the METHOD attributes for FORM tags are shown in FIG. **8**. The process determines which type of METHOD is present in the FORM and then properly processes the attribute. For the GET and POST methods (Blocks **210**, **212**) the processing is the same as that performed in standard HTML (Blocks **226**, **228**). That is, for the GET method, the identified URL<file> is queried for data

EXHIBIT E
PAGE 40 OF 43

while the POST attribute causes data to be transferred to the URL<file>. The preferred METHOD attributes extending the HTML implementation of the present invention are SQL (Block 214), and PAYMENT (Block 224) attributes. The SQL attribute is preferably not expanded into a SQL command at the client, but rather is expanded by the CGI 28 at server 12 by correlating the data or variable field names in a returned form with the SQL commands stored at the server. This processing is done in a manner described in more detail below. The client program passes the SQL file identifier to the server 12 (Block 230). The processing of the PAYMENT command (Block 232) is discussed in more detail below. The HTML file is scanned for other METHODs (Blocks 242, 244), and, if one is found, the processing continues by identifying the METHOD (Blocks 210–224). Otherwise (Block 244), the process returns to scan the HTML file for other ACTION or METHOD attributes (Block 160, FIG. 6).

Processing for the INPUT tag is shown in FIG. 9. The process scans the HTML file following the INPUT tag for attributes (Block 250). If no attributes are found (Block 252), the process continues by scanning the HTML file for other tags to process (Block 140, FIG. 5). If an attribute is found and it is a TYPE attribute (Block 254), it is processed (Block 256), and if the attribute is a NAME attribute (Block 258), it is processed (Block 260). Both the TYPE and NAME processing is shown in more detail in FIGS. 10 and 11, respectively. If the attribute is neither a NAME or TYPE attribute, it is a standard attribute for an INPUT tag supported by standard HTML and is processed in a known manner (Block 262). Following processing of the INPUT attribute, the HTML file is scanned for other attributes to process (Block 250).

Processing for the TYPE attribute is shown in FIG. 10. The process first identifies the TYPE attribute for the INPUT tag and then performs the appropriate processing. The new TYPE attributes of the preferred embodiment of the present invention are MSRT1 (Block 270), MSRT2 (Block 272), KEY (Block 274), PIN (Block 276), BCW (Block 278), MICR (Block 280), AMT (Block 282), INT (Block 284), LOCAL (Block 286), and AUTOSUB (Block 288). If the TYPE attribute is not one of these, it is a standard HTML type attribute that is processed in a known manner (Block 310). Each of the new HTML TYPES supported by the present invention causes an I/O operation with a non-standard device. Specifically, these operations are the reading of Track 1 of the magnetic stripe reader (Block 290), the reading of the second track of the magnetic stripe reader (Block 292), the reading of a keypad (Block 294), the reading of an encrypted PIN through a PIN entry device (Block 296), the reading of a bar code through a bar code reader (Block 298), the reading of encoded data on a check through a magnetic check reader (Block 300), the reading of a dollar amount from a keypad through a key input mask (Block 302), the reading of a number from a keypad through a key input mask (Block 304), the reading of data from a local variable (Block 306), and the submission of the data read from one of these devices in a FORM returned to the server 12 (Block 308). The data mask for AMT constrains the dollar amount read to a predetermined number of characters with only two characters following the decimal point. The data mask for INT ensures the number is an integer value within a predetermined range. Processing continues by scanning the HTML file for other TYPE attributes (Block 312) and, if another TYPE attribute is found (Block 314), processing continues by determining the TYPE attribute and performing the appropriate processing. Otherwise, the process returns to scan the HTML file for other attributes (Block 250, FIG. 9).

The NAME attribute processing is performed in accordance with the process shown in FIG. 11. That process examines the NAME attribute to determine if the variable name identified by the attribute is IP_ADDRESS, HOST_PHONE, TID, WORK_KEY, DATETIME, or DEPOSIT_ACCT (Blocks 320, 322, 324, 326, 328, 330). If they are, the INPUT value resulting from one of the INPUTS in a FORM of the HTML file is stored in a local variable identified by the NAME attribute. Following storage (Block 332), the file is scanned for other NAME attributes (Block 328) and, if there are none (Block 332), processing continues by scanning for other attributes for the INPUT tag (Block 250, FIG. 9). If the NAME attribute is a standard HTML INPUT NAME, it is processed by known methods (Block 336). Processing then continues by scanning for other NAME attributes to process (Block 338, 340). Otherwise, the process returns to scan the HTML file for other attributes (Block 250, FIG. 9).

CGI 28 receives Internet protocol statements in a file transmitted from a client program and provides data from those statements to the application(s) implementing system 40 and receives the output of system 40 and provides them to the client program in a file. CGI 28 may be implemented by a program developed by a user using a manual development method as shown in FIG. 24A. That method requires a user to generate a system definition from which a file statement definition for the client and application are developed to implement the transactional or data system. Using the file statement definitions, the user generates the files for the client and database programs which are interpreted by the respective programs to implement transactions or data processing. This process requires the user to not only have knowledge regarding the transaction or data process but specific details of the interaction between the client and database. The user is further required to resolve and correlate all data identifiers in the statements for the client and database environments.

Preferably, CGI 28 is developed with an editor that only requires the user to define the system with statements which are an integration of the protocol statements and the database language. The process implemented by this editor is shown in FIG. 24B. Examples of such integrated statements for files which implement a specific transaction are shown in FIGS. 14 to 23B. The editor verifies the syntax of the integrated statements and correlates the data variables of the protocol statements with the data fields of the database. Following the generation of the integrated statements, the editor segregates the protocol statements from the database language statements. The protocol statements are stored in files which are identified as being for a particular transaction or data process and the database statements are stored in files which are identified as being for a particular transaction or data process on an identified database table. The editor places a database file identifier in the protocol statements which contained embedded database statements. The database file identifiers are used by CGI 28 to select the file for the appropriate transaction so CGI 28 may correlate data variables in the protocol statements with data fields in the database files. The files containing statements to be interpreted by the client program are then downloaded to the appropriate terminals, and the database files containing database language statements are stored on the system executing the CGI 28.

Alternatively, the editor of the present invention may parse integrated statements which are segregated into source code statements for first and second processors, such an editor further includes a compiler to generate executable

code for each processor and, if the processors execute differing source code, a compiler for each source code language. The executable code may then be downloaded to the respective processors for execution.

More specifically, the editor preferably places the database statements for one of the transactions of the preferred embodiment in a file identified by the database name following SQL in FIG. 12. The attributes and tags forming the HTML statements for one of the transactions of the preferred embodiment are placed in a file generally denoted as <html_file>.HTM. The name <html_file> is a name which identifies one of the transactions. Where SQL statements are in the fields of the integrated statements shown in FIGS. 14 to 23B, the string "<html_file>.SQL" is substituted as the database name in the statements of the <html_file>.HTM file. When the CGI executable file is initiated and parses the returning forms, the returned data is placed in the corresponding "<html_file>.SQL" file which is passed to the application program as a command line argument. In this manner, an abbreviated form for the SQL commands may be communicated over the open network between the client and CGI and the CGI may be able to expand those abbreviated SQL commands into the appropriate SQL commands which the application program requires to manipulate the ODBC database.

To effectuate a transaction, for example, an operation at a terminal with non-standard I/O devices may activate a terminal file with a hot key or other action. In processing the activated file, the client program may acquire data which is stored in a local variable or accessible through a non-standard I/O device. This data may then be stored in a FORM and submitted to a server file at a processing system address. The server file activates CGI **28** which retrieves data from the FORM and incorporates it into database statements in the database file for the appropriate transaction and database. If the database statement is a query, the requested data is returned to the CGI in the database file and the CGI places it in the corresponding FORM variables so the server may return the data to the terminal. If the database statement provides data to a database to obtain an authorization, for example, the action performed by the database application in response to the data is placed in the corresponding FORM and returned to the terminal. In this way, data is exchanged between the terminal and the database application. This exchange is supported by CGI **28** even though the server/client communication is performed in an open system protocol, such as HTTP, and the database application is performed in another language, such as SQL. CGI **28** is able to convert and exchange the data between the client and database without the user having to specifically design and implement a conversion program.

The communication paths available for a device implementing the present invention are shown in FIG. 13A. As shown there, an I/O device **420** is coupled through the World Wide Web open network **426** to an Internet Web server **12**. This connection may be implemented with the preferred extended capability HTML described above. Although HTML files may be encrypted to enhance the security of the document as it is communicated across the Internet, the operator of the system may choose to utilize a more secure physical connection between the device **420** and the Web server **12**. To obtain this alternative connection, the PAYMENT command for the METHOD attribute is preferably used. One form of the PAYMENT command is for a merchant's terminal and the other is for a consumer's terminal. In either terminal, the client program which supports the extended capability HTML operates independently but co-resident in memory with a certified bank card authorization and capture application, which may be provided by a financial institution or a bank card processor.

For the form of the command shown in FIG. 13B, the client program in the merchant terminal suspends its execution and passes the terminal identifier, stored locally, which identifies the merchant's account and the consumer account information read via a magnetic stripe reader or the like, to the bank card application. The bank card application communicates this information via a PSTN **424** or the like to a transaction processor **422**. The processor **422** authorizes or denies the transaction and, if authorized, a printer at the merchant terminal prints a purchase agreement which the consumer may execute to complete the transaction.

In response to a HTML file having a FORM with an ACTION attribute equal to an executable file name for a bank card application program or the like, a METHOD attribute with a field value of PAYMENT, and an INPUT tag with a TYPE attribute of LOCAL_NAME which identifies a deposit only account supplied by a merchant (as shown in FIG. 13C), the client program is suspended and control is transferred to the bank processing application. The bank processing application then uses a modem or ISDN D channel using T3 POS protocol or the like to connect to a secure packet network **424** to connect in a virtual point-to-point manner with a payment processor through a PSTN network or the like. This physical connection provides an additional security element to the encrypted data for the transaction of account information, PIN numbers encrypted by PIN pads provided at the consumer site, and other sensitive information. The bank processor **422** may submit remittance data to the merchant, via the Web or otherwise. After receiving the remittance data, the merchant may ship the product to the consumer. Thus, in this manner, the I/O device may communicate with a plurality of Web servers to "shop" for a best price, delivery date, or other relevant information for selecting a preferred transaction, and then execute the PAYMENT method to utilize a more secure physical communication connection and data security devices to consummate the financial elements of the transaction with less risk and costs for the merchant, consumer, and bank processor.

The preferred integrated HTML/SQL statements which support a card initiated payment authorization and capture transaction are shown in FIG. 14. A first file **500** includes statements which identify the URL database from which the non-standard I/O device seeks authorization for a transaction. The prompts to the operator to enter the account number and amount of the transaction are supported by the INPUT tags which read the second track of the magnetic stripe reader to accept a number of up to 40 characters and assign that information from that track to a variable, and to input the up to 8 characters from the keyboard or the like into a variable called AMOUNT. The INPUT tag with the TYPE attribute of AUTOSUBMIT returns the form to the server for processing in accordance with the method defined in the returned form. As shown in FIG. 14, that METHOD statement causes CGI **28** to incorporate returned data into SQL commands which query the database as to whether the subfield of the track 2 data representing the account number is present in the authorization table of the database. If the data is not present, then a new record is inserted into a table labeled "log_table". The new record consists of the account number and the amount returned in the FORM. Based upon the results of this processing, the application program supplies the data fields to the FORM which will be returned to the client program for printing the transaction record. That

EXHIBIT E
PAGE 42 OF 43

file **510** is shown in FIG. **14**. The ACTION attribute TO PRINTER and the POST METHOD causes the data in the next eight lines to be directed to the printer coupled to the non-standard I/O device for printing the transaction form. The customer may then execute the printed form to complete the transaction. If the transaction is declined or an error is otherwise encountered, the file **520** is used to return a denial to the client program.

In a similar manner, the preferred integrated statements for a bar code order input with card-initiated payment authorization is shown in FIG. **15**. The file **550**, supported by the present invention which implements the transaction request, is again directed to the proper database by the ACTION attribute. The necessary customer information such as name and address may be input through a standard keyboard. The HTML command in the present invention also permits the form to receive the bar code, unit price, and credit card information in a manner similar to that discussed above for the magnetic card reader. Once this information is returned to the server and CGI interface, it is processed by the application program in accordance with the METHOD identified in the returned form. The method of HTML file **550** also creates a database order_table having the information shown in the method. Again, if the transaction is approved, the data for the order and customer acceptance of the order is provided in HMTL file **555**, which is directed by the ACTION attribute to the printer at the non-standard I/O device. If the account number is not in the authorization database, the authorization declined or error response is provided in correspondence with the statements in file **560**.

In a similar manner, FIGS. **16–22** show the integrated statements for a transaction request, authorization response, or authorization declined response files for key input order with secure payment transaction (FIG. **16**), a smart card-debit (Type 1) transaction (FIG. **17A**), a smart card debit (Type 2) transaction (FIG. **17B**), a debit card transaction (FIG. **18**), a check verification transaction (FIG. **19**), a customer frequency transaction (FIG. **20**), an item search transaction for which there is no denial (FIG. **21**), retail store end of day reporting (FIG. **22**) and a store reporting an e-mail transaction (FIG. **23**).

While the present invention has been illustrated by the description of a preferred and alternative embodiments and processes, and while the preferred and alternative embodiments and processes have been described in considerable detail, it is not the intention of the applicant to restrict or in any way limit the scope of the appended claims to such detail. Additional advantages and modifications will readily appear to those skilled in the art. For example, rather than expanding HTTP to support non-standard I/O devices, the FTP, POP, SMTP, TELNET or other protocols may be expanded in like manner to couple non-standard I/O devices to the Internet. Similarly, the preferred implementation of the present invention supports a variety of non-standard I/O devices and I/O operations. An Internet protocol may be constructed in accordance with the principles of the present invention to support only selected I/O devices or operations disclosed in the present application. The invention in its broadest aspects is therefore not limited to the specific details, preferred embodiment, and illustrative examples shown and described. Accordingly, departures may be made from such details without departing from the spirit or scope of applicant's general inventive concept.

What is claimed is:

1. A system for enabling smart card reader transactions to occur over the Internet, comprising:

a client computer communicatively coupled to an open network for communicating files having open network protocol statements over the open network;

a smart card reader communicatively coupled to the client computer through a port on the client computer, the smart card reader communicating with the client computer through the port on the client computer in accordance with a non-open network protocol; and

a client program executing in the client computer to *determine an input/output (I/O) operation to be performed* by the smart card reader that is identified by an extended open network protocol statement contained in at least one of the files communicated by the client computer over the open network, the client program causing the smart card to perform the determined I/O operation by communicating through the port on the client computer in the non-open network protocol.

2. The system of claim **1** wherein the communication through the port corresponds to a standard protocol.

3. The system of claim **2** wherein the standard protocol complies with RS-232C specification.

4. The system of *claim* **2** *wherein the extended* open network protocol statement is an HTML statement.

5. The system of claim **1** wherein the determined I/O operation is one of a data write to the smart card reader and a *data read from the smart card reader.*

6. A method for enabling smart card transactions to occur over the Internet, comprising:

communicating files having open network protocol statements over an open network with a client computer communicatively coupled to a smart card reader through a port on the client computer;

determining within the client computer an input/output (I/O) operation to be performed by the smart card reader, the I/O operation being identified by an extended open network protocol statement contained in at least one of the communicated files; and

causing the smart card reader to perform the determined I/O operation by communicating with the smart card reader in accordance with a non-open network protocol through the port on the client computer.

7. The method of claim **6** wherein the I/O operation determination is made from processing an HTML statement.

8. The method of claim **6** wherein the I/O operation determination determines the I/O operation is one of a data write to the smart card reader and a data read from the smart card reader.

* * * * *

EXHIBIT 9
PAGE 43 OF 43

EXHIBIT / ATTACHMENT

F

(To be scanned in place of tab)

US006745259B2

(12) **United States Patent**
Wagner

(10) **Patent No.: US 6,745,259 B2**
(45) **Date of Patent: *Jun. 1, 2004**

(54) **OPEN NETWORK SYSTEM FOR I/O OPERATION INCLUDING A COMMON GATEWAY INTERFACE AND AN EXTENDED OPEN NETWORK PROTOCOL WITH NON-STANDARD I/O DEVICES UTILIZING DEVICE AND IDENTIFIER FOR OPERATION TO BE PERFORMED WITH DEVICE**

(75) Inventor: **Richard Hiers Wagner**, Dunwoody, GA (US)

(73) Assignee: **Datascape, Inc.**, Atlanta, GA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 322 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **09/907,076**

(22) Filed: **Jul. 17, 2001**

(65) **Prior Publication Data**

US 2002/0032810 A1 Mar. 14, 2002

Related U.S. Application Data

(63) Continuation of application No. 09/314,266, filed on May 18, 1999, now Pat. No. 6,366,967, which is a continuation of application No. 08/995,123, filed on Dec. 19, 1997, now Pat. No. 5,905,908, which is a continuation of application No. 08/493,772, filed on Jun. 22, 1995, now Pat. No. 5,742,845.

(51) **Int. Cl.**[7] **G06F 13/00**

(52) **U.S. Cl.** **710/33**; 710/11; 710/20; 370/401; 709/203; 709/227; 709/228

(58) **Field of Search** 710/11, 20, 33; 370/401; 709/203, 227, 228

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,668,653	A	6/1972	Fair et al.	
4,410,962	A	10/1983	Daniels et al.	
4,438,511	A	3/1984	Baran	
4,481,574	A	* 11/1984	DeFino et al.	379/93.01
4,577,152	A	3/1986	Macovski	
4,620,153	A	10/1986	Hino	

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

WO	WO 95/16971	6/1995
WO	WO 96/41286	12/1996

OTHER PUBLICATIONS

Salsman, James P., "Form–based Device Input and Upload in HTML", available at http://www.w3.org/1999/07/NOTE–device–upload–19990706, Cisco Systems, © 1999, Jul. 6, 1999 (9 pages).
Memo entitled: "Form–based File Upload in HTML", E. Nebel and L. Masinter, Network Working Group of Xerox Corporation, Nov. 1995 (12 pages).
HTML file entitled: "HTML 3.0 Draft (Expired!) Materials", available at http://www.w3.org/MarkUp/html3, Dec. 21, 1995 (4 pages).

(List continued on next page.)

Primary Examiner—Rehana Perveen
(74) *Attorney, Agent, or Firm*—Maginot Moore & Beck

(57) **ABSTRACT**

An open network system for supporting input/output (I/O) operations for non-standard I/O devices are disclosed. The system includes a server coupled to a plurality of I/O devices through an open network and an extended open system protocol that supports communication with devices that are not personal computers (PCs). These devices include magnetic stripe readers, check readers, smart card readers, credit card terminals, screen phone terminals, PIN pads, printers, and the like. The extended open network protocol includes tags which identify device and input operations and attributes which identify the location, data exchange method, and data variable names for the retrieval, acquisition, and submission of data between the server and I/O devices. Preferably, the open network protocol is implemented in a Hyper Text Transport Protocol (HTTP).

104 Claims, 25 Drawing Sheets

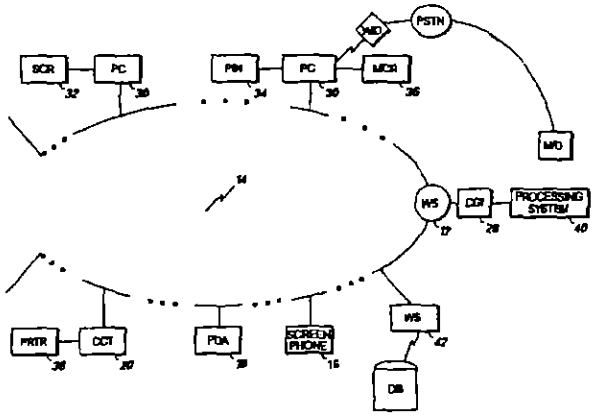

EXHIBIT F
PAGE 1 OF 49

U.S. PATENT DOCUMENTS

4,625,171	A		11/1986	Sekihara et al.
4,625,276	A		11/1986	Benton et al.
4,707,592	A		11/1987	Ware
4,724,521	A		2/1988	Carron et al.
RE32,701	E		6/1988	Moran
4,755,940	A		7/1988	Brachtl et al.
4,774,655	A		9/1988	Kollin et al.
4,787,035	A		11/1988	Bourne
4,799,156	A		1/1989	Shavit et al.
4,815,029	A		3/1989	Barker et al.
4,847,604	A		7/1989	Doyle
4,885,777	A		12/1989	Takaragi et al.
4,901,223	A		2/1990	Rhyne
4,912,309	A		3/1990	Danielson et al.
4,926,325	A		5/1990	Benton et al.
4,942,532	A		7/1990	Mori
4,942,552	A		7/1990	Merrill et al.
4,947,028	A		8/1990	Gorog
4,949,248	A		8/1990	Caro
4,962,531	A		10/1990	Sipman et al.
4,965,742	A		10/1990	Skeirik
4,972,463	A		11/1990	Danielson et al.
4,999,806	A	*	3/1991	Chernow et al. 717/177
5,014,221	A		5/1991	Mogul
5,018,196	A		5/1991	Takaragi et al.
5,021,949	A		6/1991	Morten et al.
5,025,373	A		6/1991	Keyser, Jr. et al.
5,036,461	A		7/1991	Elliott et al.
5,036,484	A		7/1991	McCoy et al.
5,046,094	A		9/1991	Kawamura et al.
5,047,923	A		9/1991	Elstner et al.
5,083,262	A		1/1992	Haff, Jr.
5,103,478	A		4/1992	Matyas et al.
5,113,517	A		5/1992	Beard et al.
5,119,465	A		6/1992	Jack et al.
5,146,499	A		9/1992	Geffrotin
5,146,553	A		9/1992	Noguchi et al.
5,148,481	A		9/1992	Abraham et al.
5,149,945	A		9/1992	Johnson et al.
5,161,222	A		11/1992	Montejo et al.
5,179,690	A		1/1993	Ishikawa
5,195,130	A		3/1993	Weiss et al.
5,202,828	A		4/1993	Vertelney et al.
5,204,947	A		4/1993	Bernstein et al.
5,206,951	A		4/1993	Khoyi et al.
5,212,369	A		5/1993	Karlisch et al.
5,220,380	A		6/1993	Hirata et al.
5,220,501	A		6/1993	Lawlor et al.
5,239,662	A		8/1993	Danielson et al.
5,243,284	A		9/1993	Noll
5,256,863	A		10/1993	Ferguson et al.
5,272,754	A		12/1993	Boerbert
5,274,821	A		12/1993	Rouquie
5,276,312	A		1/1994	McCarthy
5,277,192	A		1/1994	Cumoulin
5,278,955	A		1/1994	Forte et al.
5,278,972	A		1/1994	Baker et al.
5,294,782	A		3/1994	Kumar
5,297,249	A		3/1994	Bernstein et al.
5,307,411	A		4/1994	Anvret et al.
5,307,499	A		4/1994	Yin
5,309,351	A		5/1994	McCain et al.
5,321,806	A		6/1994	Meinerth et al.
5,321,808	A		6/1994	Rupp
5,347,580	A		9/1994	Molva et al.
5,347,632	A		9/1994	Filepp et al.
5,349,678	A		9/1994	Morris et al.
5,351,296	A		9/1994	Sullivan
5,363,489	A	*	11/1994	Snyder 710/105
5,367,572	A		11/1994	Weiss
5,367,635	A		11/1994	Bauer et al.
5,373,561	A		12/1994	Haber et al.
5,377,191	A		12/1994	Farrell et al.
5,379,344	A		1/1995	Larsson et al.
5,386,517	A		1/1995	Sheth et al.
5,390,314	A		2/1995	Swanson
5,418,908	A		5/1995	Keller et al.
5,461,217	A		10/1995	Claus
5,475,585	A		12/1995	Bush
5,475,858	A		12/1995	Gupta et al.
5,497,487	A		3/1996	Fortier
5,499,343	A		3/1996	Pettus
5,510,710	A		4/1996	Nauerth
5,511,122	A		4/1996	Atkinson
5,517,569	A		5/1996	Clark
5,521,980	A		5/1996	Brands
5,526,037	A		6/1996	Cortjens et al.
5,526,409	A		6/1996	Conrow et al.
5,530,844	A		6/1996	Phillips et al.
5,544,161	A		8/1996	Bigham et al.
5,544,246	A		8/1996	Mandelbaum et al.
5,544,320	A		8/1996	Konrad
5,546,582	A		8/1996	Brockmeyer et al.
5,548,721	A		8/1996	Denslow
5,550,561	A		8/1996	Ziarno
5,550,984	A		8/1996	Gelb
5,552,586	A		9/1996	Kalman
5,557,203	A		9/1996	Nauerth
5,563,878	A		10/1996	Blakeley et al.
5,570,465	A		10/1996	Tsakanikas
5,577,233	A		11/1996	Goettelmann et al.
5,581,478	A		12/1996	Cruse et al.
5,581,686	A		12/1996	Koppolu et al.
5,581,708	A		12/1996	Iijima
5,583,940	A		12/1996	Vidrascu et al.
5,586,036	A		12/1996	Pinstov
5,586,175	A		12/1996	Hogan et al.
5,590,038	A		12/1996	Pitroda
5,590,181	A		12/1996	Hogan et al.
5,590,196	A		12/1996	Moreau
5,590,197	A		12/1996	Chen et al.
5,590,199	A		12/1996	Krajewski, Jr. et al.
5,594,910	A		1/1997	Filepp et al.
5,602,905	A		2/1997	Mettke
5,602,915	A		2/1997	Campana et al.
5,602,918	A		2/1997	Chen et al.
5,604,802	A		2/1997	Holloway
5,606,493	A		2/1997	Duscher et al.
5,613,012	A		3/1997	Hoffman et al.
5,613,096	A		3/1997	Danknick
5,615,251	A		3/1997	Hogan et al.
5,623,547	A		4/1997	Jones et al.
5,630,101	A		5/1997	Sieffert
5,633,919	A		5/1997	Hogan et al.
5,638,430	A		6/1997	Hogan et al.
5,640,193	A		6/1997	Wellner
5,642,485	A		6/1997	Deaton et al.
5,649,114	A		7/1997	Deaton et al.
5,649,118	A		7/1997	Carlisle et al.
5,652,876	A		7/1997	Ashe et al.
5,655,008	A		8/1997	Futch et al.
5,664,110	A		9/1997	Green et al.
5,666,412	A		9/1997	Handelman et al.
5,668,878	A		9/1997	Brands
5,677,953	A		10/1997	Dolphin
5,677,955	A		10/1997	Doggett et al.
5,678,002	A		10/1997	Fawcett et al.
5,679,471	A		10/1997	Cheng et al.

EXHIBIT 1F
PAGE 2 OF 49

5,679,945	A		10/1997	Renner et al.
5,687,322	A		11/1997	Deaton et al.
5,689,799	A		11/1997	Dougherty et al.
5,694,546	A		12/1997	Reisman
5,696,903	A		12/1997	Mahany
5,703,795	A		12/1997	Mankovitz
5,706,273	A		1/1998	Guerreri
5,708,780	A		1/1998	Levergood et al.
5,708,782	A		1/1998	Larson et al.
5,708,825	A		1/1998	Sotomayor
5,710,884	A		1/1998	Dedrick
5,715,314	A		2/1998	Payne et al.
5,722,754	A		3/1998	Langner
5,724,506	A		3/1998	Cleron et al.
5,729,549	A		3/1998	Kostreski et al.
5,732,219	A		3/1998	Blumer et al.
5,734,720	A		3/1998	Salganicoff
5,742,845	A		4/1998	Wagner
5,742,931	A		4/1998	Spiegelhoff et al.
5,748,908	A		5/1998	Yu
5,754,655	A		5/1998	Hughes et al.
5,756,978	A		5/1998	Soltesz et al.
5,758,257	A		5/1998	Herz et al.
5,778,067	A		7/1998	Jones et al.
5,781,189	A		7/1998	Holleran et al.
5,787,305	A	*	7/1998	Chen 715/540
5,799,156	A		8/1998	Hogan et al.
5,805,719	A		9/1998	Pare, Jr. et al.
5,805,807	A		9/1998	Hanson et al.
5,809,415	A		9/1998	Rossmann
5,815,577	A		9/1998	Clark
5,815,657	A		9/1998	Williams et al.
5,819,092	A		10/1998	Ferguson et al.
5,826,245	A		10/1998	Sandberg-Diment
5,838,906	A		11/1998	Doyle et al.
5,850,442	A		12/1998	Muftic
5,851,149	A		12/1998	Xidos et al.
5,854,833	A		12/1998	Hogan et al.
5,867,821	A		2/1999	Ballantyne et al.
5,870,552	A		2/1999	Dozier et al.
5,870,724	A		2/1999	Lawlor et al.
5,873,099	A		2/1999	Hogan et al.
5,884,292	A		3/1999	Baker et al.
5,889,863	A		3/1999	Weber
5,931,917	A		8/1999	Nguyen et al.
5,933,816	A		8/1999	Zeanah et al.
RE36,310	E		9/1999	Bjerrum et al.
5,953,731	A		9/1999	Glaser
5,960,216	A		9/1999	Vishlitzky et al.
5,963,205	A		10/1999	Sotomayor
5,973,731	A		10/1999	Schwab
5,978,569	A		11/1999	Traeger
5,978,773	A		11/1999	Hudetz et al.
5,982,891	A		11/1999	Ginter et al.
5,983,004	A		11/1999	Shaw et al.
5,987,312	A		11/1999	Dekker
5,996,076	A		11/1999	Rowney et al.
6,085,224	A		7/2000	Wagner
6,112,191	A		8/2000	Burke
6,175,922	B1		1/2001	Wang
6,182,052	B1		1/2001	Fulton et al.
6,205,437	B1		3/2001	Gifford
6,208,904	B1		3/2001	Mullen, Jr.
6,212,575	B1		4/2001	Cleron et al.
6,226,744	B1		5/2001	Murphy et al.
6,226,752	B1		5/2001	Gupta et al.
6,253,027	B1		6/2001	Weber et al.
6,275,869	B1		8/2001	Sieffert et al.
6,279,112	B1		8/2001	O'Toole, Jr. et al.
6,282,656	B1		8/2001	Wang
6,283,375	B1		9/2001	Wilz, Sr. et al.
6,314,456	B1		11/2001	Van Andel et al.

OTHER PUBLICATIONS

Berners–Lee, T. and Connolly, D., HTML file entitled: "HyperText Markup Language Specification—2.0", available available at http://www.w3.org/MarkUp/html–spec., Nov. 1995 (10 pages).

Raggett, David, HTML file entitled: "A Review of the HTML + Document Format", Hewlett Packard Laboratories, date unknown (13 pages).

Raggett, David, HTML file entitled: "10–Fill–out Forms and Input Fields", available at http://www.w3.org/MarkUp HTMLPlus/htmlplus_41.html, Nov. 8, 1993 (5 pages).

Raggett, David, HTML File entitled: "ISO 8879:1986", available at http://www.w3.org/MarkUp/htmlplus_paper/htmlplus.did, Apr. 5, 1994 (17 pages).

"Viola in a Nutshell: The Viola World Wide Web Toolkit", http://www.xcf.berkeley.edu/~wei/viola/book/, Apr. 1995 (176 pages).

O'Reilly & Associates, Inc., ViolaWWW, Version 3.3 Beta, Features List, http://www.xcf.berkely.edu/~wei/viola/README.txt, Apr. 10, 1995 (2 pages).

Wei, Pei, "A Brief Overview of the VIOLA Engine, and its Applications",http://www.xcf.berkeley.edu/~wei/viola/violaIntro.html, Aug. 16, 1994, (13 pages).

Wei, Pei, "WWW Browsers: Extensibility Issues", O'Reilly & Associates, Sep. 20, 1994, (13 pages).

"Basic HTTP as defined in 1992", W3C, http://www.w3.org/Protocols/HTTP/HTTP2.html, 1992, (125 pages).

McCool, Rob, "CGI/1.0: last call", Dec. 4, 1993 (1 page).

"Olde Viola Montage", http://www.xcf.berkeley.edu/~wei/viola/vintage/montage.html, 1991, (5 pages).

Masinter, Larry, "Naming and resource location", Stanford Computer Forum WWW Workshop, Sep. 20, 1994, (1 page).

Cousins, Steve and Shoham Yoav, "Software Agents on the WWW", Stanford Computer Forum WWW Workshop, Sep. 20, 1994, (1 page).

Winograd, Terry, "Web Structure and Meta–Information", Stanford Computer Forum WWW Workshop, Sep. 20, 1994, (1 page).

Shafrir, Eviatar, "Interacting with Information", Stanford Computer Forum WWW Workshop, http://www–pcd.stanford.edu/workshop/abstracts/shafrir.html, Sep. 20, 1994 (2 pages).

Gruber, Tom, "Virtual Documents", Stanford Computer Forum WWW Workshop, Sep. 20, 1994, (21 pages).

Keller, Arthur M., "CIT's Work on CommerceNet", Center for Information Technology, Stanford University, http://cit.stanford.edu/cit/comercenet.html, Oct. 1994, (6 pages).

Shapiro,Ehud, "Enhancing the WWW with co–presence", Stanford Computer Forum WWW Workshop, Sep. 20, 1994, (9 pages).

"Readings for Stanford WWW Workshop", Stanford Computer Forum, http://www–pcd.stanford.edu/workshop/readings.html, Sep. 20, 1994, (3 pages).

"WorldWideWeb–Summary", http://www.w3.org/Summary.html, 1992, (2 pages).

"The Original HTTP as defined in 1991", http://www.w3.org/Protocols/HTTP/AsImplemented.html, 1991, (2 pages).

"WorldWide Web Seminar", http://www.w3.org/Talks/General/About.html, 1993, (15 pages).

EXHIBIT F
PAGE 3 OF 49

White, Sean, "Online Communities on the WWW", Stanford Computer Forum WWW Workshop, Sep. 20, 1994, (1 page).

"The Common Gateway Interface", http://hoohoo.ncsa.uiuc.edu/cgi/, 1994, (23 pages).

"W3C: Object MetaInformation", http://www.w3.org/Protocols/HTTP/Object_Headers.html, 1992, (9 pages).

"Hypertext Markup Language (HTML)", http://www.w3.org/MarkUp/1995-archive/1, 1995, (66 pages).

McCool, Rob, "Re: CGP/1.0 specification", Nov. 19, 1993 (2 pages).

McCool, Rob, "Revised CG1/1.0 specification", Nov. 19, 1993, http://www.webhistory.org/www-talk.1993q4/0540.html, (1 page).

Citation List of References noting "The Common Gateway Interface", CiteSeer, 2002, (6 pages).

Robinson, David and Coar, Ken, "The Common Gateway Interface (CGI) Version 1.1", Apr. 16, 2003, (30 pages).

Varela, Carlos and Hayes, Caroline, "Providing Data on the Web: From Examples to Programs", Dept. Computer Science, University of Illinois at Urbana-Champaign, http://archive.ncsa.uiuc.edu/SDG/IT94/Proceedings/DDay/varela/, Oct. 17, 1994, (26 pages).

Dynamic Information Systems Corporation, "Developing JDBC Servlet Applications", http://www.disc.com/jdbcserv.html, May 8, 2002, (12 pages).

ProSolutions, Inc. Active Server Pages (ASP) Technology, http://www.prosolutionsinc.com/asp.htm, 2002, (2 pages).

Rasmussen, B.F., "WDB—A Web Interface to Sybase", http://www.stsci.edu/stsci/meetings/adassIV/rasmussenb.html, 1995, (4 pages).

Rescorla et al., "The Secure HyperText Transfer Protocol", Network Working Group, © Aug. 1999, (42 pages).

Berners-Lee et al., "HyterText Markup Language Specification—2.0", Nov. 29, 1994, Available online at http://marc.merlins.org/~merlin/htmlearn/html2.html., Aug. 21, 2002, (69 pages).

Raggett, Dave, "HyperText Markup Language Specification Version 3.0", Mar. 28, 1995, Available online at http://www.w3.or./MarkUp/html3/html3.txt., Aug. 21, 2002, (181 pages).

Merle et al., "Integration of heterogeneous environments: World Wide Web, Smart Card and Corba," Date Unknown, (16 pages).

"ProSolutions, Inc. ASP vs. CGI 'Active Server pages (ASP) Technology'", E Business Solutions, © 1995-2002, Available online at http://www.prosolutionsinc.com/asp.htm., May 8, 2002, (2 pages).

Kirby et al., "Developing JDBC Servlet Applications", Dynamic Information Systems Corporation, Available online at http://www.disc.com/jdbcserv.html., May 8, 2002, (12 pages).

Pasian et al., "A Generalized Mosaic-to-SQL Interface with Extensions to Distributed Archives", Astronomical Data Analysis Software and Systems IV ASP Conference Series, vol. 77, 1995, Available online at http://www.stsci.edu/stsci/meetings/adassIV/pasianf.html., May 7, 2002, (5 pages).

Ng, Jason, "GSQL—a Mosaic-SQL gateway", Dec. 1993, Available online at http://archive.ncsa.uiuc.edu/SDG/People/jason/pub/gsql/starthere.html., May 7, 2002, (16 pages).

Rasmussen, B. F., "WDB—A Web Interface to Sybase", Astronomical Data Analysis Software and Systems IV ASP Conference Series, vol. 77, 1995, Available online at http://www.stsci.edu/stsci/meetings/addasIV/rasmussenb.html., May 7, 2002, (7 pages).

Connolly, Daniel W., "Paul Burchard on HTML 2.0 FORMs", Jul. 11, 1994, Available online at http://lists.w3.org/Archives/Public/www-html/1994Jul/0014.html., Dec. 3, 2001, (6 pages).

Hackborn, Dianne, "Re: Java Applet element proposal", Jun. 8, 1995, Available online at http://ksi.cpsc.ucalagary.ca/archives/HTML-WG/html-wg-95q2.messages/1228.html., Nov. 1, 2001, (2 pages).

Kaye, Walter Ian, "Re: Protocol registry? (fwd. Java Applet element proposal)", Jun. 5, 1995, Available online at http://lists.w3.org/Archives/Public/www-html/1995Jun/0069.html., Aug. 14, 2001, (3 pages).

"The APP HTML Tag", Hot Java, Sun Microsystems, Inc., © 1995, Available online at http://www-math.uni-paderborn.de/dictionaire/Dictionaries/HTML-Dictionary/j.../index.html., Apr. 24, 2002, (2 pages).

Yourdon, Edward, "Java and the new Internet programming paradigm", © 2002, Available online at http://www.javaworld.com/javaworld/jw-08-1996/jw-08-yourdonbook_p.html., Apr. 29, 2002, (20 pages).

Merckling et al., "Smart Card Introduction", Mar. 1994, Available online at http://www.opengroup.org/rfc/mirror-rfc/rfc57.0.txt., Mar. 1, 2001, (16 pages).

Bellovin et al., "Limitations of the Kerberos Authentication System", USENIX, 1991, (16 pages).

"Guideline for the Use of Advanced Authentication Technology Alternatives", Federal Information Processing Standards Publicaiton 190, Sep. 28, 1994, Available online at http://www.itl.nist.gov/fipspubs/fip190.htm., Mar. 21, 2002, (49pages).

"HTML + (Hypertext markup format)", Nov. 8, 1993, Available online at http://www.w3.org/MarkUp/HTMLPlus/htmlplus_2.html., Mar. 28, 2002, (89 pages).

Everhart, Craig, "Re: Interesting uses of networking", Jun. 11, 1990, Available online at http://www.cs.ucsd.edu/users/bsy/coke.history.txt., Sep. 17, 2002 (4 pages).

Kotanchik, J., Kerberos and Two-Factor Authentication, Security Dynamics, Mar. 1994, (11 pages).

McLaughlin, L. III, ed., "Line Printer Daemon Protocol", Network Printing Working Group, Aug. 1990, Available online at http://www.ietf.org/rfc/rfc1179.txt., Sep. 17, 2002, (14 pages).

Gutfreund, Yechezkal-Shimon, "Mosaic Accessories", Jan. 26, 1994, Available online at file://C\WINDOWS/TEMP/WWW-Talk%20Jan-Mar%201994%20Mosaic%20Accessories.htm., Apr. 1, 2002, (8 pages).

Gessler et al., "PDAs as mobile WWW browsers", Date Unknown, Telecooperation Office, University of Karlsruhe, Available online at file://C:\WINDOWS/TEMP/gwpring/PDAs%20as%20mobile%20WWW%20browsers.htm., Mar. 28, 2002, (9 pages).

Rose, M., "Principles of Operation for the TPC.INT Subdomain: Radio Paging—Technical Procedures", Dover Beach Consulting, Inc., Jan. 1994, (6 pages).

Workshop on Network and Distributed System Security, Privacy and Security Research Group, Feb. 11-12, 1993, (6 pages).

EXHIBIT F
PAGE 4 OF 49

Malamud et al., "Principles of Operation for the TPC.INT Subdomain: Remote Printing—Technical Procedures", Dover Beach Consulting, Inc., Oct. 1993, (11 pages).
Louis, Tristan, "Java on the Menu; Causing Jitters?," Jun. 1995, Available online at http://www.tnl.net/who/bibliography/javajitters.asp., Apr. 29, 2002, (2 pages).
Allen, Terry, "Java Applet element proposal," Jun. 1, 1995, Available online at http://ksi.cpsc.ucalgary.ca/archives/HTML-WG/html-wg-95q2.messages/1119.html., Nov. 16, 2001, (4 pages).
Hume, Tom, "WAP: Where we've come from, where we are, where we're going," Available online at http://www.tomhume.org/stories/2002/02/04/wapWhereWeveComeFromWhereWeAreW..., (18 pages).
Goldman, Chris, "Browsing for Perfection," ARC Group in the Press, Feb. 1, 2001, Available online at http://www.arcgroup.com/homepage.nsf/PRSARC/ar010201, (4 pages).
"Handheld Devices Markup Language (HDML) specification," Apr. 11, 1997, Available online at http//www.w3.org/TR/hdml20-3.html through http://www.w3.org/TR/hdml20-9 html, (33 pages).
Hinden, R., "Simple Internet Protocol Plus White Paper," Network Working Group, Oct. 1994, Available online at ftp://ftp.rfc-editor.org/in-notes/rfc1710.txt, (22 pages).
McCarthy, Vance, "Novell to link wireless laptops, PDAs to LANs (portable computers, personal digital assistants, local area networks)," InfoWorld, Sep. 13, 1993 v15 n37 p1, (2 pages).
Schwartz, Barry K. and Stephen B. Weinstein, "Dual-Media Messaging Using Screen Telephones on the Telephone Network," Bell Communications Research, IEEE 1993, pp. 1138-1188, (6 pages).
Verjinski, Richard D., "PHASE, A Portable Host Access System Environment," Unisys Defense Systems, IEEE 1989, pp. 806-809, (4 pages).
Klemets, Anders, Gerald Q. Maguire, Frank Reichert, and Mark T. Smith, "MINT—A Mobile Internet Router," IEEE 1993, pp. 70-74, (5 pages).
Perkins, Charles E. and Pravin Bhagwat, "A Mobile Networking System based on Internet Protocol," IEEE Personal Communications, First Quarter 1994, pp. 32-41, (10 pages).
Connolly, T., P. Amer, and P. Conrad, "An Extension to TCP: Partial Order Service," Network Working Group, Nov. 1994, Available online at ftp://ftp.rfc-editor.org/in-notes/rfc1693.txt, (14 pages).
Furniss, P., "Octet Sequences for Upper-Layer OSI to Support Basic Communications Applications," Network Working Group, Oct. 1994, Available online at ftp://ftp.rfc-editor.org/in-notes/rfc1698.txt. (28 pages).
Graves, C., T. Butts, and M. Angel, "TN3270 Extensions for LUname and Printer Selection," Network Working Group, Jul. 1994, Available online at ftp://ftp.rfc-editor.org/in-notes/rfc1646.txt, (13 pages).
Klensin, J., N. Freed, M. Rose, E. Stefferud, and D. Crocker, "SMTP Service Extensions," Network Working Group, Feb. 1993, Available online at ftp://ftp.rfc-editor.org/in-notes/rfc1425.txt, (10 pages).
Wang, Z., "EIP: The Extended Internet Protocol A Framework for Maintaining Backward Compatibility," Network Working Group, Nov. 1992, Available online at ftp://ftp.rfc-editor.org/in-notes/rfc1385.txt. (16 pages).
Gwinn, A., "Simple Network Paging Protocol—Version 2." Network Working Group, Jul. 1994. Available online at ftp://ftp.rfc-editor.org/in-notes/rfc1645.txt, (15 pages).
Braden, B., J. Postel, and Y. Rekhter, "Internet Architecture Extensions for Shared Media," Network Working Group, May 1994, Available online at ftp://ftp.rfc-editor.org/in-notes/rfc1620.txt, (18 pages).
Delgrossi, Luca, Christian Halstrick, Ralf Guido Herrtwich, and Heinrich Stüttgen, "HeiTP—A Transport Protocol for ST-II," IBM European Networking Center, IEEE 1992, pp. 1369-1373, (5 pages).
Tsang, P. W. M. and R. W. C. Wang, "Development of a Distributive Lighting Control System Using Local Operation Network," IEEE Transactions on Consumer Electronics, vol. 40, No. 4, Nov. 1994, (11 pages).
Adomeit, Reinhard and Bernhard Holtkamp, "A Reference Framework for Integrated POI/POS Systems," Fraunhofer Institute for Software Engineering and Systems Engineering, (4 pages).
Caneve, Maurizio, and Roberto Gagliardi, "A WAIS-Based Hypermedia Document System," (4 pages).
Schmidt, Claudia, Burkhard Stiller, and Martina Zitterbart, "Towards Flexible Service-integrated Communication Subsystems," Jul. 1, 1994, (4 pages).
Plagemann, Thomas, Janusz Waclawczyk, and Bernhard Plattner, "Management of Configurable Protocols for Multimedia Applications," (4 pages).
"Frame relay attracts chains; networks at Kash N' Karry, Office Max (network communications protocol)," Chain Store Age Executive with Shopping Center Age, Dec. 1994 v70, (4 pages).
Cummings, Joanne, "What's Hot What's Not '95," Network World, Dec. 26, 1994/Jan. 2, 1995, (5 pages).
Aronin, Mindy, "XcelleNet launches fifth-generation RemoteWare 2.0 for remote and mobile applications," Business Wire, Nov. 28, 1994, (4 pages).
Baum, David, "Designing mobile applications (case studies of companies developing mobile programs)," InfoWorld, Nov. 28, 1994 v16 n48 p73, (7 pages).
Moeller, Michael, "Wireless standard to ease application portability; extensions will provide a common layer for CDPD, RAM, Ardis software," PC Week, Nov. 21, 1994 v11 n46 p1, (3 pages).
Petreley, Nicholas, Nancy Durlester, and Laura Wonnacott, "The interoperability headache: linking disparate clients and servers," InfoWorld, Nov. 15, 1993 v15 n46 p124, (12 pages).
Miscellaneous List of Article Abstracts, (50 pages).
Riordan, T., "A Toll Collector on the Information Highway", *The New York Times*, Sunday, Aug. 22, 1993, (3 pages).
Burchard et al., "Compound Documents in HTML", Internet Draft Memo, Date Unknown, (14 pages).
Press Release, "Netscape Introduces Netscape Navigator 2.0", Netscape Company Press Relations, Sep. 18, 1995, (3 pages).
Stabell, B., The EMBED Tag by Netscape (v2.0), Sep. 19, 1995, (9 pages).
Article entitled, "Eolas Sues Microsoft for Infringement of Patent for Fundamental Web Browser Technology That Makes "Plug-Ins" and "Applets" Possible", Chicago, Illinois, Feb. 2, 1999, (2 pages).
Sottong, S., "Quick HTML Basics", Date Unknown, (12 pages).
Hansen, Wilfred J., "Enhancing Documents with Embedded Programs: How Ness Extends Insets in the Andrew Toolkit", IEEE 1990, pp. 23-32, (10 pages).

EXHIBIT F
PAGE 5 OF 49

Tani, M., Yamaashi, K., Tanikoshi, K., Futakawa, M., and Tanufuji, S., "Object–Oriented Video: Interaction with Real–World Objects Through Live Video", ACM 1992, pp. 593–598, (6 pages).

Crowley, T., Milazzo, P., Baker, E., Forsdick, H., and Tomlinson, R., "MMConf: An Infrastructure for Building Shared Multimedia Applications", ACM 1990, pp. 329–342, (14 pages).

Davis, H., Hall, W., Heath, I., Hill, G., and Wilkins, R., "Towards an Intergrated Information Environment with Open Hypermedia Systems", ACM 1992, pp. 181–191, (10 pages).

Ferrara, Fabrizio M., "The KIM Query System: An Iconic Interface for the Unified Access to Distributed Multimedia Databases", SIGGHI Bulletin, 1994, vol. 26, No. 3, pp. 30–39, (10 pages).

Gibbs, Simon, "Composite Multimedia and Active Objects", OOPSLA, 1991, pp. 97–112, (16 pages).

Davis, H., Hall, W., Pickering, A., and Wilkins, R., "Microcosm: An Open Hypermedia System" ACM, 1993, (1 page).

Cullen, J., Day, B. and Planes, M., "The Use of FTAM to Access Graphical Pictures Across Wide Area Networks", Elsevier Science Publischers B.V., 1992, pp. 377–383, (7 pages).

Vaziri, Arsi, "Scientific Visualization in High–Speed Network Environments", Elsevier Science Publishers B.V., 1991, pp. 111–129, (19 pages).

Lashkari, Y., Rao, S., Mhaskar, V. and Shelat, A., "PLX: A Proposal to Implement a General Broadcasting Facility in a Distributed Environment Running X Windows", Comput. And Graphics, vol. 16, No. 2, 1992, pp. 143–149, (7 pages).

Kirste, Thomas, "SpacePicture– An Interactive Hypermedia Satellite Image Archival System", Comput. & Graphics, vol. 17, No. 3, 1993, pp. 251–260, (10 pages).

Coulson, G., Blair, G., Davies, N. and Williams, N., "Extensions to ANSA for Multimedia Computing", Elsevier Science Publishers B.V., 1992, pp. 305–322, (18 pages).

Le Huynh, D., Jensen, M., Larsen, R., Southard, J. and Wang, F., "PIX: An Object–Oriented Network Graphics Environment", Comput. & Graphics, vol. 17, No. 3, 1993, pp. 295–304, (10 pages).

Berners–Lee, T.J., Cailliau, R., and Groff, J., "The World–Wide Web", Elsevier Science Publishers B.V. 1992, pp. 454–459, (6 pages).

Shackelford, D., Smith, J., and Smith, F., "The Architecture and Implementation of a Distributed Hypermedia Storage System", Nov., 1993, pp. 1–13, (13pages).

Labriola Don, "Whiteboard Software: Remote Possibilities", PC Magazine, Jun. 14, 1994, pp. 223–224, (2 pages).

"How We Evaluate the Whiteboard Software", PC Magazine, Jun. 14, 1994, pp. 225–228, (4 pages).

Udell, John, "Visual Basic Custom Controls Meet OLE", BYTE, Mar., 1994, pp. 197–198 and 200, (3 pages).

Sarna, D. and Febish, G., "OLE Gains Without (Much) Pain", DATAMATION, Jun. 15, 1994, pp. 31, (1 page).

Scrupski, Susan, "Outsourcing: It Could be Good", DATAMATION, Jun. 15, 1994, pp. 32–33 and 114, (3 pages).

Rizzo, John, "What's OpenDoc?", MacUser, Apr. 1994, pp. 119–123, (3pages).

Cello Browser Release 1.01a, ftp.law.cornell.edu/pub/L11/cello , Mar. 16, 1994, (8 pages).

"Why Jim Clark Loves Mosaic", Wired, Oct. 1994, pp. 118–121, 150–151 and 157, (7 pages).

Marovac, N., and Osburn, L., "Hypernet: A Tool to Choreograph Worldwide Distributed Hypermedia Documents", Comput. & Graphics, vol. 16, No. 2, 1992, pp. 197–202, (6 pages).

Moeller, M., and Leach, N., "Microsoft Maps New OCX Plan; ActiveX seen as Web Content Platform", PC Week, Mar. 11, 1996, vol. 13, No. 10, pp. 1–2, (2 pages).

"OLE 2.0: Death to Monoliths?", BYTE, Mar. 1994, pp. 121–122, (2 pages).

"Netscape Communications Offers New Network Navigator Free on the Internet", http://home.netscape.com/newsref/pr/newsrelease1.html , Jan. 29, 1998.

Sackman, Gleason, "Remote Tele–Excavation via the Web", Tele–robotics, vol. 11, Issue 9, (1 page).

Wolf, Gary, "The (Second Phase of) the Revolution has Begun", WIRED, Oct. 1994, pp. 16–17, (2 pages).

Wynne, B. and Marovac, N., "Lean Management, Group Support Systems, and Hypermedia: A Combination Whose Time Has Come", IEEE, 1993, pp. 112–121, (10 pages).

Moran, Patrick J. "Tele–Nicer–Slicer–Dicer: A New Tool for the Visualization of Large Volumetric Data", Biomedical Imaging Group, pp. 1–7, (7 pages).

Hansen, W., Consortium, A, "Andrew as a Multiple Environment for Visual Languages", IEEE, 1993, pp. 256–260, (5 pages).

Doyle, M., "The Virtual Embryo: VR Applications in Human Development Anatomy", Jan. 27, 1994, pp. 1–9, 9 pages).

Doyle, M., "A Knowledge Management Environment through the World Wide Web", Jan. 7, 1994.

Oliver, D. et al., "Netscape™ Unleashed".

Ang, C., Martin, D., and Doyle, M., "Intergrated Control of Distributed Volume Visulazation Through the World–Wide–Web", 1994, pp. 1–10 and 1, (11 pages).

Ang et al., "Integrated Control of Distributed Volume Visualization Through the World–Wide–Web", Oct. 30, 2001, (8 pages).

Krajewski, Jr., et al., "Applicability of Smart Cards to Network User Authentication", *Computing Systems*, vol. 7, No. 1, Winter 1994, (15 pages).

"Financial Institution Sign–On Authentication for Wholesale Financial Transactions", American National Standards Institute, X9.26, Feb. 28, 1990, (35 pages).

Kohl et al., "The Evoluation of *Kerberos* Authentication Service", Date Unknown, (15 pages).

Cordonnier et al., "Time as an Aid to Improving Security in Smart Cards", Date Unknown, (14 pages).

Cordonnier et al., "The Concept of Suspicion: A New Security Model for Identification Management in Smart Cards", Date Unknown, (14 pages).

Haykin et al., "Computer Science and Technology—Smart Card Technology: New Methods for Computer Access Control", U.S. Department of Commerce, National Technical Information Service, Sep. 1998, (55 pages).

Rescorla et al., "The Secure HyperText Transfer Protocol", Internet Draft Memo, Dec. 1994, (29 pages).

Takei, D., "Videotex Information System and Credit System Connecting with MARS–301 of JNR", *Japanese Railway Engineering*, No. 95, Sep. 1985, (5 pages).

Pongratz et al., "IC Cards in Videotex Systems", *Smart Card 2000: The Future of IC Cards*, International Conference, Oct. 19–20, 1987, (9 pages).

EXHIBIT F
PAGE 6 OF 49

Neuman, B.C., "Security, Payment, and Privacy for Network Commerce", *IEEE Journal on Selected Areas in Communication*, vol. 13, No. 8, Oct. 1995, (8 pages).
Zilles, S.N., "Catalog–Based Order Entry System", IBM Technical Disclosure Bulletin, vol. 25, No. 11B, Apr. 1983, (3 pages).
Article entitled, "PKCS #11: New Member of Public Key Cryptography", available online at http://www.rsasecurity-.com/news/pr/940112–1.html., Jan. 12, 1994, (12 pages).
Kessler, J., "The French Minitel: Is There Life Outside of the 'US ASCII' Internet? A Challenge of Convergence", *D–Lib Magazine*, Dec. 1995, (10 pages).
Schöter et al., "Digital Money Online, A Review of Some Existing Technologies", Intertrader Ltd., ©Feb. 1997, (60 pages).
Sutherland, E., "The Quickening Pace of Technology", ©1995, (3 pages).
Article entitled "NetBill: An Internet Commerce System Optimized for Network Delivered Services", Date Unknown, (9 pages).
Reagle, Jr., et al., expanded bibliography for "Papers Describing the Operation of Electronic Commerce Protocols", Date Unknown, (3 pages).
Article entitled "ISO/IEC 7816 Part 4: Interindustry Command for Interchange", Nov. 26, 1998, (77 pages).
1991 Statement on Role of NIST, available online at http://csrc.nist.gov/publications/nistbul/cs191–02.txt, Jun. 17, 1992, (38 pages).
Kirkman, P., "Electronic Funds Transfer Systems: The Revolution in Cashless Banking and Payment Methods", ©1987, (14 pages).
Fancher, C., "Smart Cards: As Potential Applications Grow, Computers in the Wallet are Making Unobtrusive Inroads", *Scientific American*, Date Unknown, (10 pages).
Walsh, E., "Smart Card Standards", Datapro Information Services, Aug. 25, 2000, (10 pages).
Article entitled "The Birth of Smart Cards: 1980", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1981", Le Courier de la Monétique, Date Unknown, (3 pages).
Article entitled "The Birth of Smart Cards: 1982", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1983", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1984", Le Courier de la Monétique, Date Unknown, (3 pages).
Article entitled "The Birth of Smart Cards: 1985", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1986", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1987", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1988", Le Courier de la Monétique, Date Unknown, (2 pages).
Article entitled "The Birth of Smart Cards: 1989", Le Courier de la Monétique, Date Unknown, (2 pages).
Doyle, M., University of California, San Diego, Office of Continuing Medical Education, Disclosure Statement, Nov. 30, 1993, (10 pages).
Article entitled "Workshop on Network and Distributed System Security", Privacy and Research Group, Feb. 11–12, 1993, (5 pages).
Krajewski, Jr., M., "Concept for a Smart Card Kerberos", Jan. 22, 2002, (8 pages).
Kohl et al., "The Kerberos Network Authentication Service (V5)", Sep. 1993, (103 pages).
Nebel, E., "File Upload in HTML Forms", Sep. 23, 1994, (4 pages).
Raggett, D., E–mail memo entitled "Re: Request for New Forms Submission Consensus", Oct. 12, 1993, (4 pages).
Gaskell et al., "Improved Security for Smart Card Use in DCE", Feb. 1995, (17 pages).
Pato, J., "Using Pre–Authentication to Avoid Password Guessing Attacks", Jun. 1993, (5 pages).
Shipplett, M., E–mail memo entitled "Re: Insecure WWW Access Authorization Protocol?", Mar. 7, 1994, (3 pages).
"Financial Transaction Card Originated Messages—Interchange Message Specifications", International Standard, Second Edition, Dec. 15, 1993, (128 pages).
Various articles on the subject "1994 NIST Guideline for Advanced Authentication (Smart Cards)", Date Unknow, (73 pages).
Allen et al., "Smart Cards: Seizing Strategic Business Opportunities", The Smart Card Forum, ©1997 The McGraw–Hill Companies, (323 pages).
Fielding et al., "Hypertext Transfer Protocol—HTTP/1.1", Jan. 1997, (162 pages).
Branwyn, G., "M•O•S•A•I•C Quick Tour for Windows: Accessing & Navigating the Internet's World Wide Web", ©1994 Ventana Press, Inc., (215 pages).
Shelly et al., "Netscape Navigator: An Introduction", ©1996 Boyd & Fraser Publishing Company, (138 pages).
James, P., "Official Netscape Navigator 2.0 Book: Windows Edition", ©1996, Ventana Press, Inc., (692 pages).
Article entitled "A Smart Card for On–Line Home Services", ChipCord News, Issue No. 32, Jul. 1990, (2 pages).
Bosak et al., "XML and the Second Generation Web", *Scientific American*, May 1999, (5 pages).
Berners–Lee et al., "RFC1866–Hypertex Markup Language–2.0", Nov. 1995, (58 pages).
"Europay Members Forum", Europay International, May 8, 2001, (3 pages).
"NCR 2170 Hospitality Workstation Designed for Bars, Restaurants", AT&T News Release, Feb. 2, 1993, (2 pages).
"AT&T to Increase Smart Card Reliability, Expand Memory to 8KB", AT&T News Release, Apr. 8, 1994, (2 pages).
"AT&T and NCR Offer Smart Card Upgrade Kits for Bank ATMs", AT&T News Release, Nov. 17, 1993, (2 pages).
"The Smart Card Cashes In", The Economist, Jan. 29, 1994, (2 pages).
"NCR 7450 Terminal Brings ATM–style Interface to Point–of–Sale", AT&T News Release, Oct. 11, 1993, (4 pages).
"Mastercard Smart Card Milestones", MasterCard International Press Office, Mar. 5, 2001, (3 pages).
Svigals, Jerome, "Microchip Smart Cards Will Save Money", American Banker, Feb. 2, 1994, (1 page).
Barthel, Matt, "Fears of Crime at ATMs are on the Rise", American Banker, Mar. 30, 1994, (1 page).
"Smart Card Publication List", Smart Card Forum, Mar. 2, 2001, (7 pages).
"Appendix: Major Electronic Purse Projects, By Country", pp. 35–50 and 52–64, Date Unknown (29 pages).
O'Kelly, Robin, "Mondex 'Global Electronic Cash' Wins Helsinki Award", Sep. 8, 1996, (2 pages).
"White Paper—The Euro in the Electronic Purse", Apr. 2000, (5 pages).
"History of Gemplus", Gemplus, Mar. 2, 2001, (3 pages).

EXHIBIT F
PAGE 7 OF 49

Pauli, Ralf and Koponen, Risto, "Toward Electronic Money", Bank of Finland Bulletin, Apr. 1997, (3 pages).
"A Frequently Asked Questions List (FAQ) for alt.technology.smartcards", Estonian Institute of Cybernetics, Jul. 4, 1995, (12 pages).
Husemann, Dirk, "The Smart Card: Don't Leave Home Without It", Institute of Electrical and Electronics Engineers, Inc., Sep. 18, 2000, (9 pages).
"Report to the Council of The European Monetary Institute on Prepaid Cards", The Working Group on EU Payment Systems, May 1994, (12 pages).
Banerjee, Ram, "Smart Card Standards and Electronic Purse, A Review Paper", Aston University, May 1997, (20 pages).
Townend, Robin C., "Finance: History, Development & Market Overview", The International Smart Card Industry Guide, 1995–1996, (23 pages).
Monod, Elsbeth Marikje, "Case Study: The Multi-Functional Smart Carte Sante", The International Smart Card Industry Guide, 1995–1996, (8 pages).
Smith, Jack, "Smart Card Industry Review 1995", Smart Card Monthly Newsletter, Jan. 1996, (6 pages).
Crotch–Harvey, Trevor, "Smart Cards in Telecoms", The International Smart Card Industry Guide, 1995–1996, (14 pages).
Farmer, Donna and Tonnac, Remy de, "What's So Smart About Smart Cards", Gemplus, Oct. 10, 2000, (1 page).
"P401 Definition of a pan–European IC Card", Eurescom, Mar. 5, 2001, (1 page).
"Innovonics, Inc. Selects AMP as Manufacturer for the PC Pay Module", Innovonics News Release, May 19, 1995, (3 pages).
Levy, Steven, "E–Money (That's What I Want)", Wired, Dec., 1994, (2 pages).
Fickenscher, Lisa, "Banc One Gears Up to Go National With Its Microchip Smart Cards", American Banker, Jan. 27, 1994 (2 pages).
Strachman, Daniel and Kutler, Jeffrey, "Convention Goers See Bank Future in Smart Cards", American Banker, Nov. 30, 1993, (1 page).
Meece, Mickey, "First Data Strengthens Position in Card Processing", American Banker, Dec. 21, 1993, (1 page).
Kutler, Jeffrey, "Prepaid Cards, Even in Concept Stage, Sparking Interest Among Consumers", American Banker, Apr. 27, 1994, (1 page).
Kutler, Jeffrey, "Visa Starts Push for Smart Card to Replace Cash", American Banker, Mar. 22, 1994, (2 pages).
"AT&T Introduces Enhanced Multimedia Interface for ATM Networks", AT&T News Release, May 4, 1994, (2 pages).
"Trendar and AT&T Test Contactless Smart Cards at Truck Stops", AT&T News Release, Feb. 26, 1993, (3 pages).
"NCR Announces 5980 Signature Capture System for Retailers", AT&T News Release, Sep. 30, 1991, (2 pages).
Merckling, Roger and Anderson, Anne, "Smart Card Introduction", Mar. 1994,(16 pages).
Merckling, Roger and Anderson, Anne, "DCE Smart Card Integration", Mar. 1994, (16 pages).
"Chapter 2—Technology", Smart Card Primer, The Schuler Consultancy, ®1993, pp. 13–24 and 26–29, (16 pages).
"CP8© Products, PASCAM, Network Security Processor", Jan. 1994, (4 pages).
"Notes on Open Networks and Device Addressing", IBM Corporation, Apr. 25, 2001, (19 pages).
Chakrabarti, Samdih, Knight, James and Thies, Bill, "A Centralized Smart Card System Utilizing Distributed Proxies", 2000, University of Leeds, (16 pages).
"Single–Issuer Multiple Uses", Smart Card Primer, The Schuler Consultancy, ©1993, (3 pages).
Sazegar, Hossein, "The Impact of Smart Cards in the Global Payphone Business", Schlumberger Technologies, ©1992, (11 pages).
Derksen, Eduard, "Money–Bit: The Open Off–Host Prepaid Card System", Ascom Autelca, Ltd., Date Unknown, (10 pages).
Corzo, Gerardo, "Credisa's Elite Card System", Date Unknown, (5 pages).
Cordonnier, Vincent M., "Assessing the Future of Smart Cards", Cardtech, Date Unknown, (7 pages).
"NatWest Launch Mondex Global Electronic Cash", Smart Card News, Ltd., Dec. 1993, (7 pages).
Merkert, Sr., Robert J., "Stored Value Cards and The Electronic Purse: Developments in the Americas", Danyl Corporation, ©1995, (8 pages).
"CP8® Products, PinPadLINK, Readers/Encoders", Dactyl France, Jun. 1993, (2 pages).
"Smart Security", Byte Talk, Feb., 1994, (2 pages).
Verschuren, Ton, "Smart Access: Strong Authentication on the Web", Apr. 24, 2001, (10 pages).
"PKCS #11: New Member of Public Key Cyrptography", RSA Data Security, Inc., Jan. 12, 1994, (2 pages).
Petri, Steve, "An Introduction to Smarcards", Litronic, Inc., ©1998, (2 pages).
Violino, Bob, "The Cashless Society", Information Week, Oct. 11, 1993, (5 pages).
Merkert, Sr., Robert J., 'Keynote Presentation for the AIC Conference, Auckland, New Zealand', Danyl Corporation, ©1995, (4 pages).
"Videoway's Experience Shows a New Approach to the Interactive Opportunity", Inside Cable and Telecom Services, Jun. 1995, (3 pages).
Bull CP8® News, Products and Technologies, No. 22, Oct. 1994, (2 pages).
Bull CP8® Systems Product Sheet, Sep. 1994, (2 pages).
"One Store's Comprehensive Loyalty Program—The Takashimaya Experience", Verifone, Inc., ©1994, (4 pages).
"Mondex Payments Scheme", Smart Card News Ltd., Dec. 1993, (2 pages).
"Why Consumers Are Still Carrying Cash Everywhere", Philadelphia Inquirer, Jun. 11, 1995, (2 pages).
"Electronic Payment System Architecture for Point of Sale Application", IBM Technical Disclosure Bulletin, Jan. 1992, (1 page).
"Portable Self Checkout Retail System", IBM Technical Disclosure Bulletin, Jun. 1992, (3 pages).
"Transaction Completion Code Based on Digital Signatures", IBM Technical Disclosure Bulletin, Aug. 1985, (6 pages).
"Chapter 4—Uses", Smart Card Primer, The Schuler Consultancy, ©1993, pp. 57–66, 80–83, and 87–91, (19 pages).
Figure 8.5 from p. 151 of Zoreda & Oton "Smart Cards", Artech House, ©1994, (1 page).
"Internet Music Library Process", Date Unknown, (1 page).
Bibliography for "CITM Working Paper WP–95–1005: Designing Electronic Catalogs for Business Value: Results from the CommerceNet Pilot", Date Unknown, (3 pages).
Van Hove, Leo, Bibliography on Electronic Purses, Date Unknown, (4 pages).

EXHIBIT F
PAGE 8 OF 49

Bibliography for "CITM Working Paper WP–95–1006: Financial EDI Over the Internet: A Case Study", Date Unknown, (5 pages).
Bibliography for Hawkes, P.L., Davies, Donald W., and Price, W.L., "Integrated Circuit Cards, Tags, and Tokens: New Technology and Applications", Oxford; Boston: BSP Professional, ©1990, (1 page).
Bibliography for Devargas, M., "Smart Cards and Memory Cards", Oxford; Blackwell NCC, ©1992, (1 page).
Bibliography for Svigals, Jerome, "Smart Cards: The New Bank Cards", New York: Macmillan; London: Collier Macmillan Publishers, ©1987, (1 page).
Bibliography for Bright, Roy, "Smart Cards: Principles, Practice, Applications", Chichester [England]: Ellis Horwood; New York: Halsted Press, ©1988, (1 page).
Bibliography for Svigals, Jerome, "Smart Cards: The Ultimate Personal Computer", New York: Macmillan Publishing Co.; London: Collier Macmillan, ©1985, (1 page).
Bibliography for "Applications of Computer Card Technology", Washington, D.C., U.S. Dept. of Labor, Office of the Inspector General, [1990], (1 page).
German article entitled "Erfahrungen Mit Einer Hochsicheren, Multifunktionalen Chipkarte fur Bankenanwendungen" Date Unknown, (21 pages).
Document entitled: "Dialog Search for U.S. patent No. 5,905,908", Date Unknown, (59 pages).
UBI Consortium Inc. Document, Mar. 26, 2001 (2 pages).
HTML file entitled: "The Common Gateway Interface", available at httpd@ncsa.uiuc.edu, circa, Jun. 1995.
HTML file entitled: "Critique of Secure–HTTP", available at shttp://www.spyglass.com/secure, Apr. 19, 1995.
Directory of abstracts of HTML files available from IEFT Secretariat—Corporation for National Research Initiatives; circa, Jun. 1995.
HTML file entitled: "A Beginner's Guide to HTML" available from National Center for Supercomputing Applications/pubs@ncsa.uiuc.edu; circa, Jun. 1995.
Memo entitled: "Uniform Resource Locators (URL)", by URI working group of the Internet Engineering Task Force; Dec. 1994.
Press release entitled: "VeriFone to acquire Enterprise Integration Technologies (EIT), Internet Commerce Software and Services Leader", VeriFone, Inc. and Enterprise Integration Technologies; Redwood, California; Aug. 21, 1995.
Spyglass white paper entitled: "Electronic Commerce Standards for the WWW"; Spyglass, Inc. http://spyglass.com; Aug. 21, 1995.
Memo entitled: "The Secure Hypertext Transfer Protocol"; E. Rescorla and A. Schiffman; Jul. 1995.
Article entitled: "Cash on the Wirehead", Andrew Singleton, BYTE Magazine, Jun. 1995.
"Beyond the Web: Excavating the Real World Via Mosaic", Goldberg et al., 2nd International WWW Conference.
"Magnetic Resonance Angiography", Radiology 1986; 161, p. 717–720, Dumoulin et al.
"Velocity and Diffusion Imaging in Dynamic NMR Microscopy", JMR 91, p. 326–352, 1991, Callaghan et al.
"Reduced Circular Field–of–View Imaging", MRM 40:474–480, 1998, Scheffler et al.
"Measurement of Flow Using Back Projection Reconstruction: Display of Limited Regions of the Body", Young et al.

* cited by examiner

EXHIBIT F
PAGE 9 OF 49

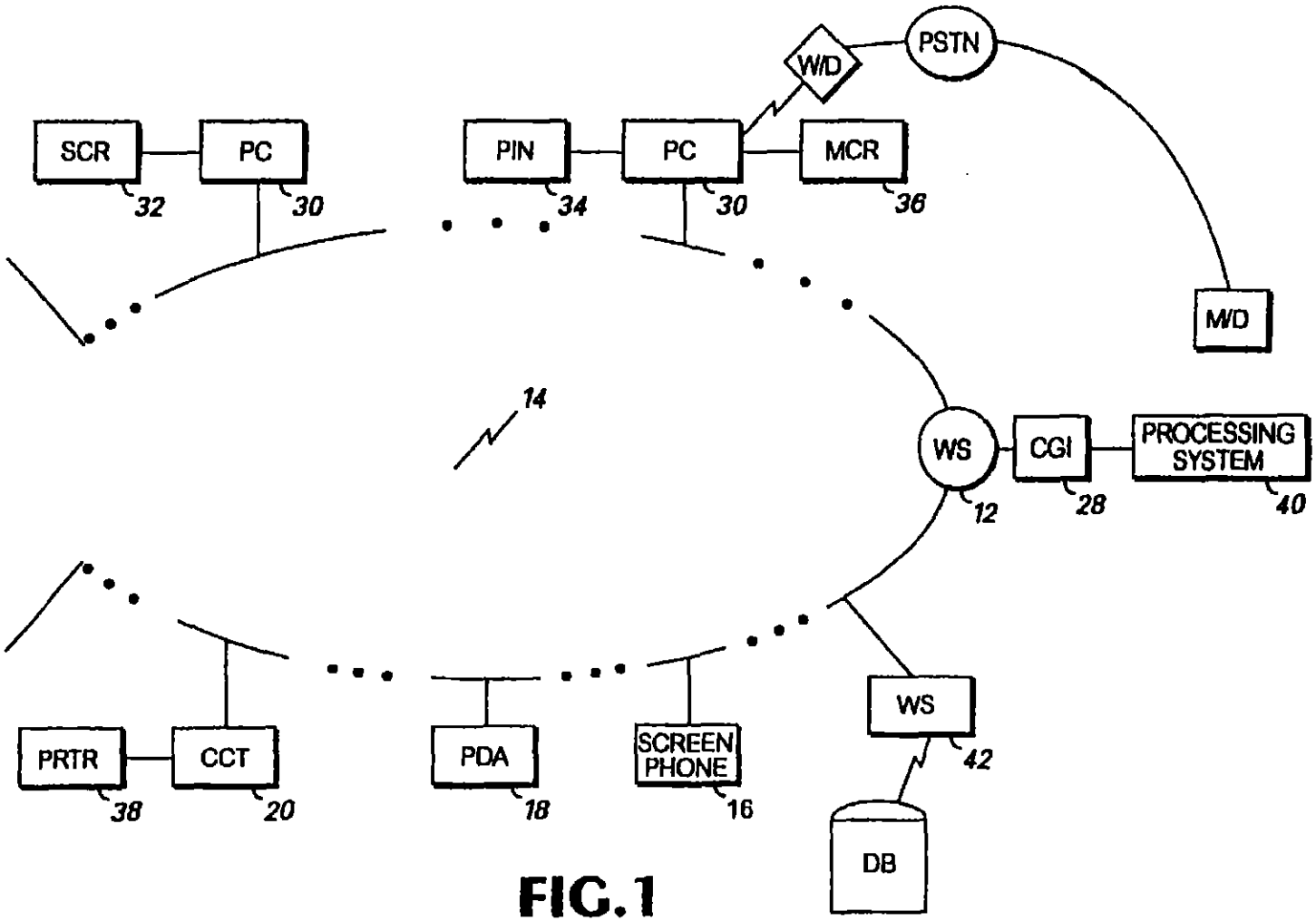

FIG.1

HTML+D Attributes		Description
<FORM ACTION =	"url"	To/From Web Server URL
	FROM "file name"	From Terminal Local File
	TO PRINTER	To Local Printer
	TO "file name"	To Terminal Local File
	FROM SCR	From Smart Card Reader
	TO SCR	To Smart Card Reader
METHOD =	"GET"	Retrieve Data
	"POST"	Store Data
	"PAYMENT"	Directive to deliver INPUT data to a private Payment Network for authorization and settlement.
	SQL <database name>	SQL statement database table

Attribute	Value	Description
<INPUT TYPE =	"text"	
	"password"	
	"checkbox"	
	"radio"	
	"submit"	
	"reset"	
NAME =	<field name>	
VALUE =	<initial value>	
CHECKED =		
SIZE =		
MAXLENGTH =		
>		

Attribute	HTML+D Value	Terminal Device
TYPE =	*"MSRT1"*	Mag Stripe Reader - Track 1
	"MSRT2"	Mag Stripe Reader - Track 2
	"KEY"	Terminal Command Keypad
	"PIN"	PIN Pad
	"BCW"	Bar Code Wand
	"MICR"	Check MICR Reader
	"AMT"	Dollar amount key input mask
	"INT"	Integer key input mask
	"LOCAL"	Input from Local Variable
	"AUTOSUBMIT"	Submit FORM to ACTION URL
NAME =	*ip_address*	Local Variable - Terminal's IP Address
	host_phone	Local Variable - Local Internet Access Phone Number
	tid	Local Variable - Terminal ID
	work_key	Local Variable - PIN encryption working key
	datetime	Local Variable - Date and time
	deposit_acct	Local Variable - Merchant Deposit Account

FIG.2

EXHIBIT F
PAGE 11 OF 49

SQL Statements

The following SQL commands represent a subset of the entire command set that varies by database vendor.

HTML+D Attributes	Description
SELECT, field_name,-*	Request field_name (one or many) from a database table
FROM=<table name>,-	Database table name
WHERE=<condition>	Conditional selection of data
name = "constant"	
name LIKE "constant"	
name IN "constant"	
AND	
OR	
ORDER=ASC	Request in ascending order
DESC	_descending
2	_by2's
GROUP=<name>	
INSERT TABLE=<table name>	Insert new data in database table
VALUES="constants"	
UPDATE FROM	
<table name>	Update field_name in database table
SET=field_name="constant"	
[WHERE=<condition>]	Update if WHERE clause is satisfied
DELETE FROM <table name>	Delete all columns that satisfy WHERE clause
[WHERE=<condition>]	
CREATE TABLE<table_name>	Create database table
PRIMARY KEY<name>	

FIG.3

EXHIBIT F
PAGE 12 OF 49

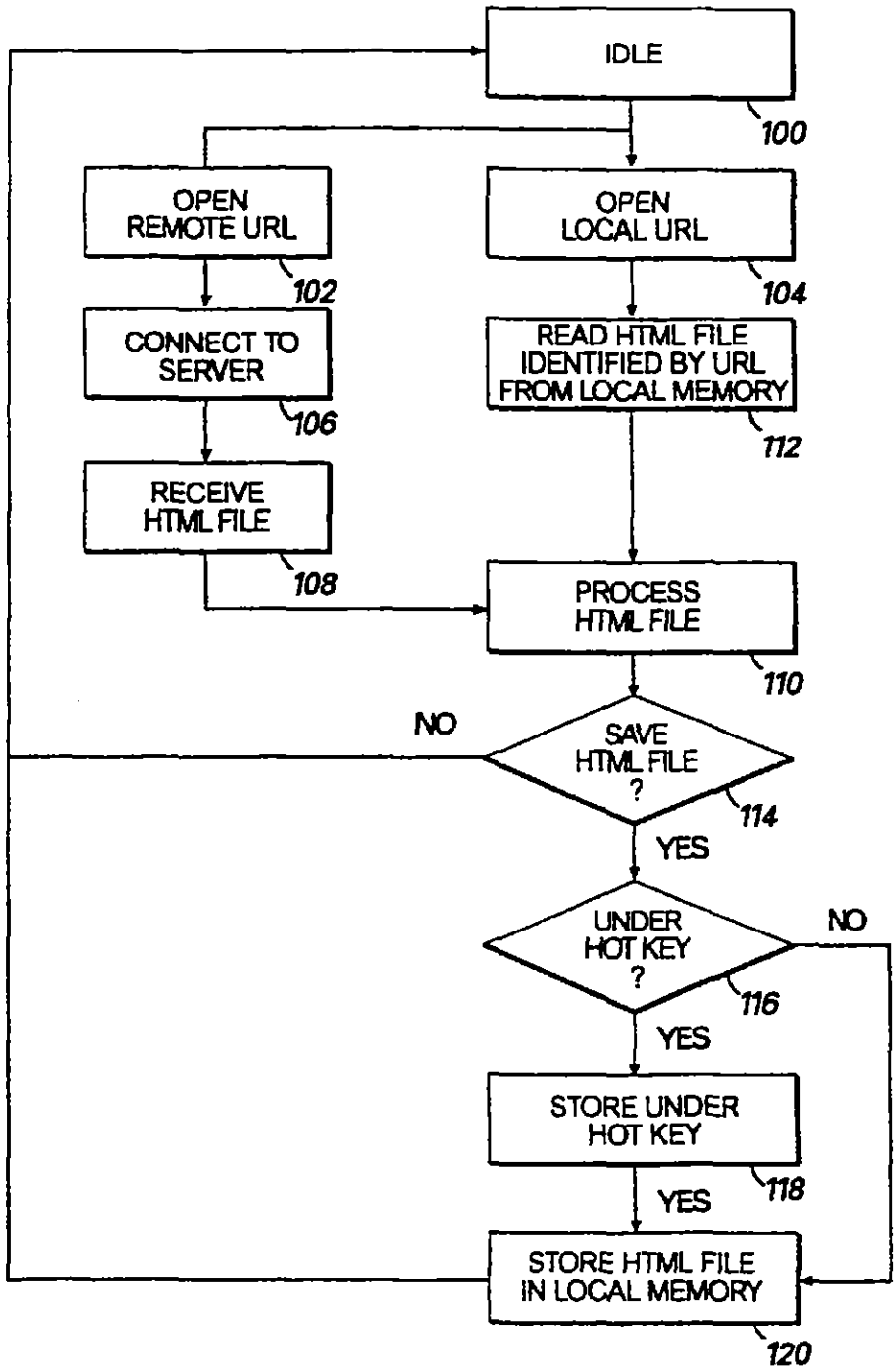

FIG.4

EXHIBIT F
PAGE 13 OF 49

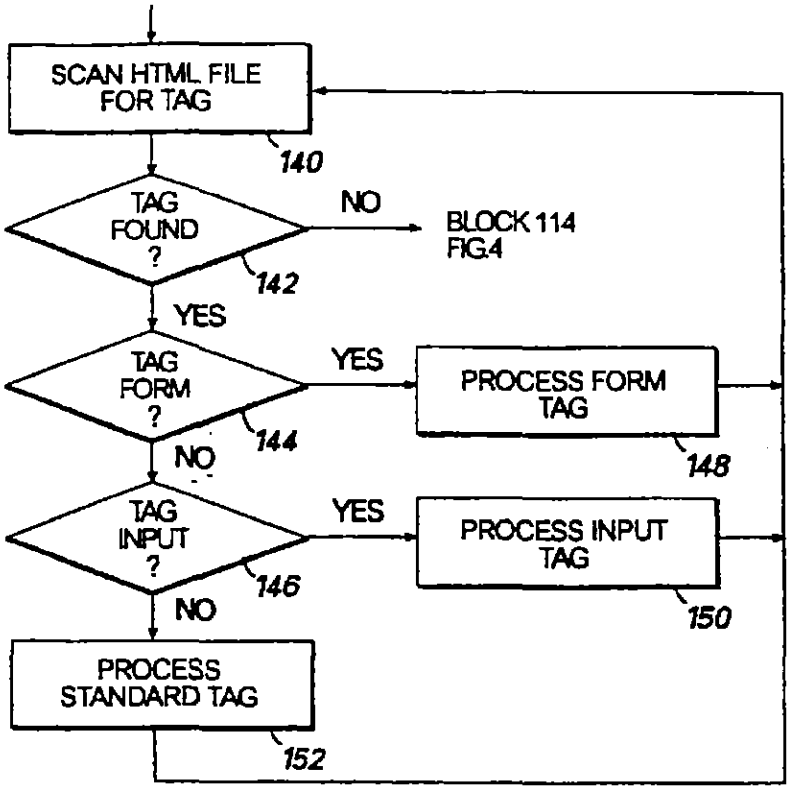

FIG.5

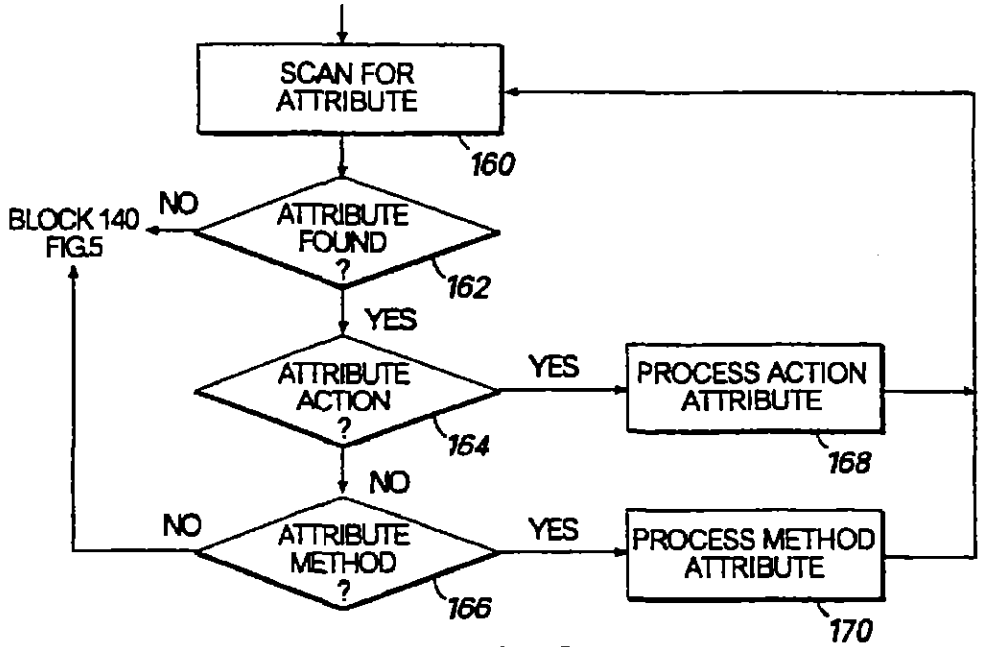

FIG.6

EXHIBIT F
PAGE 14 OF 49

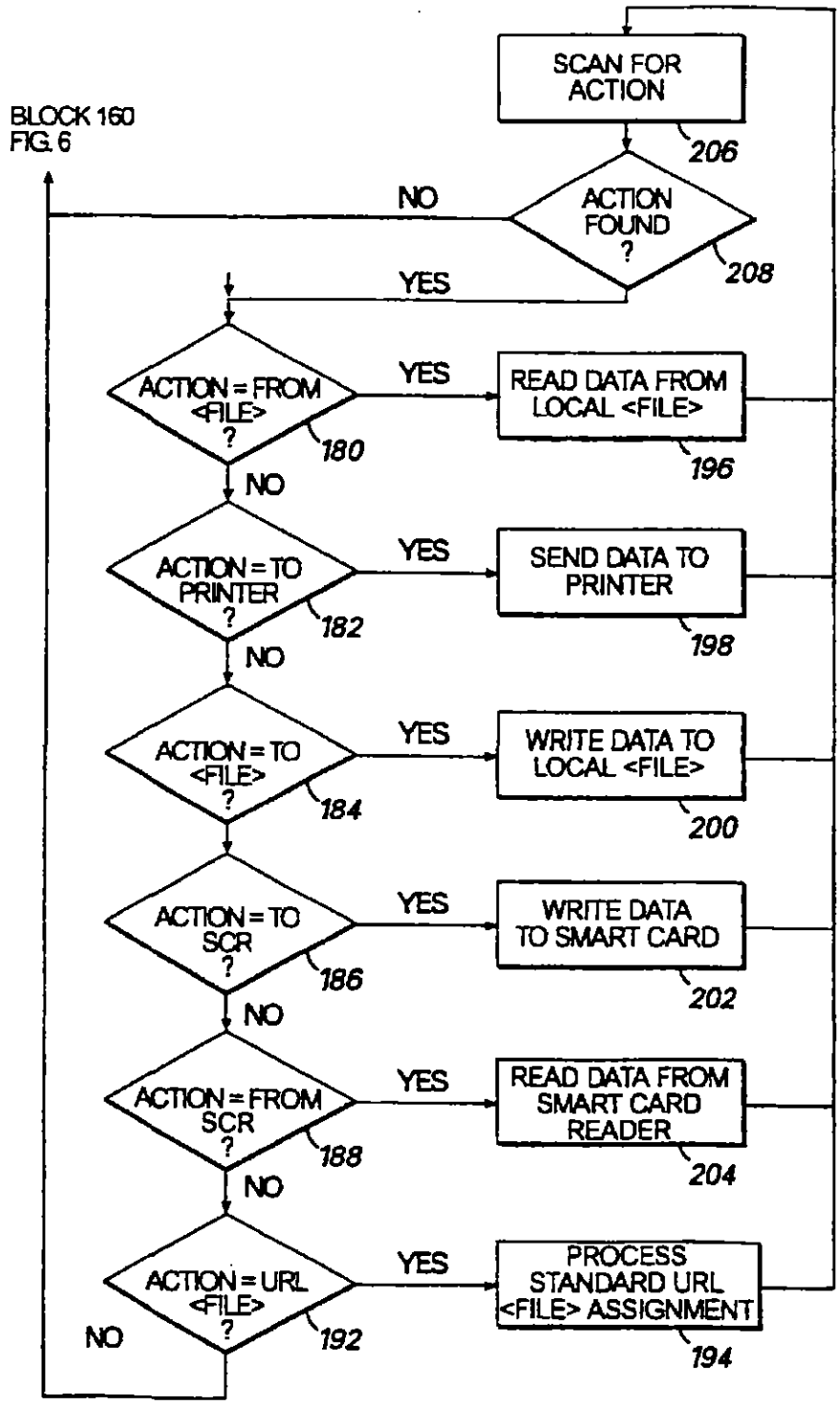

FIG. 7

EXHIBIT F
PAGE 15 OF 49

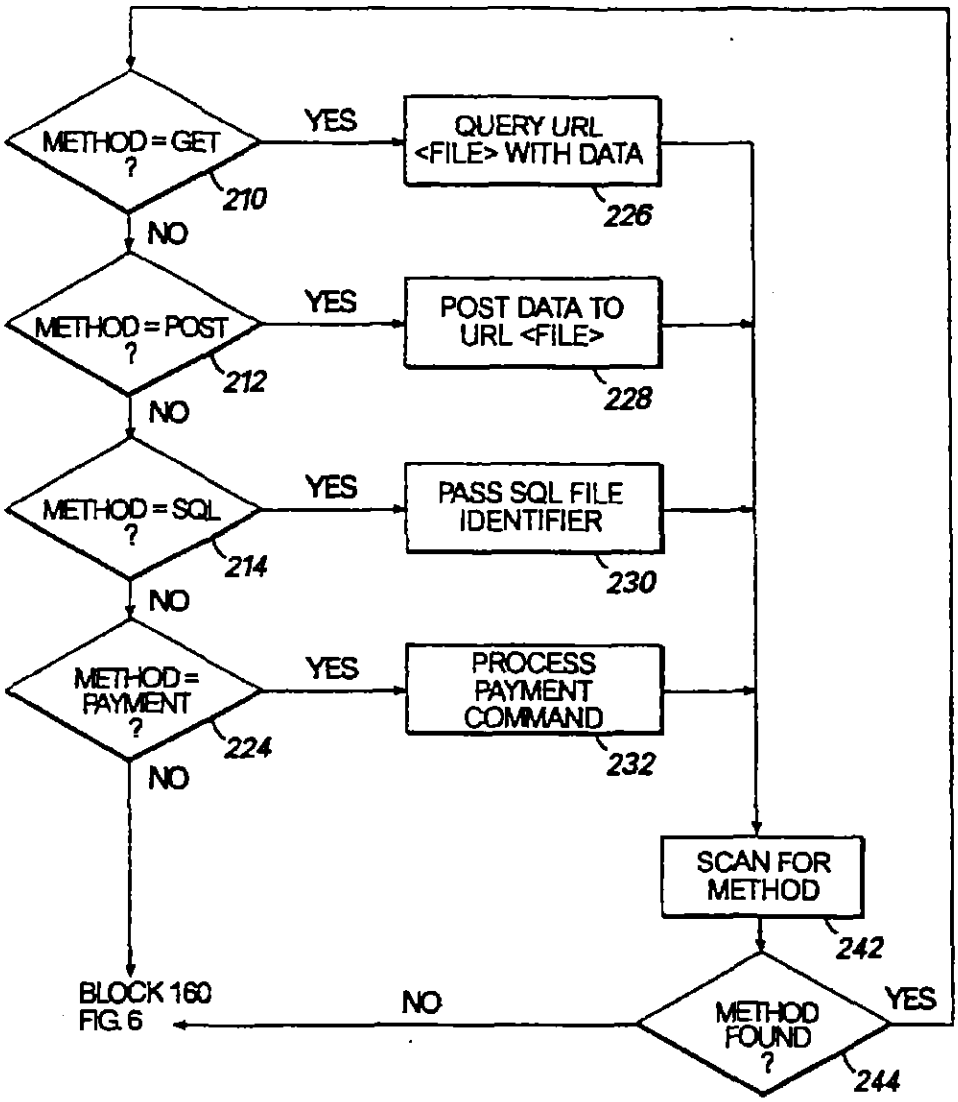

FIG.8

EXHIBIT F
PAGE 16 OF 49

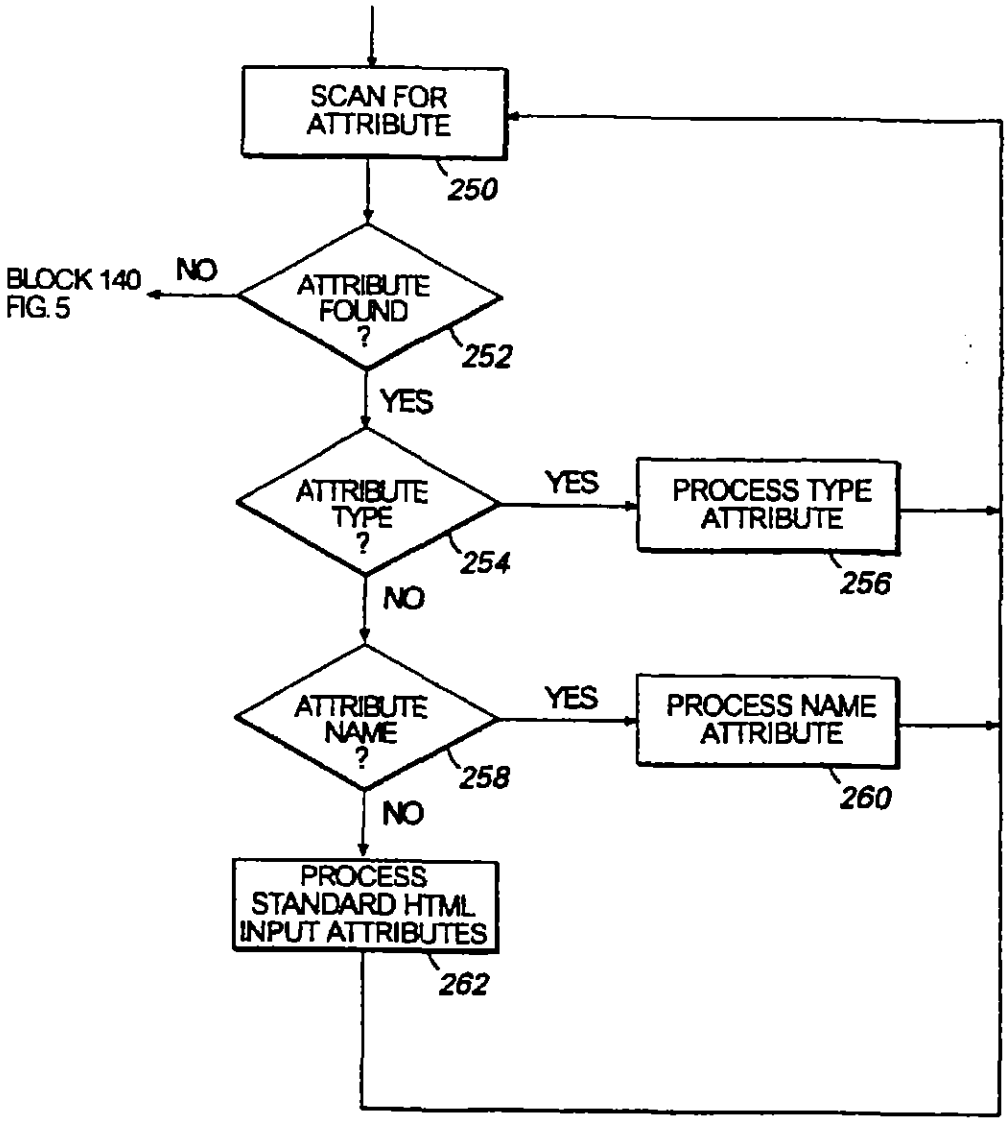

FIG. 9

EXHIBIT E
PAGE 17 OF 49

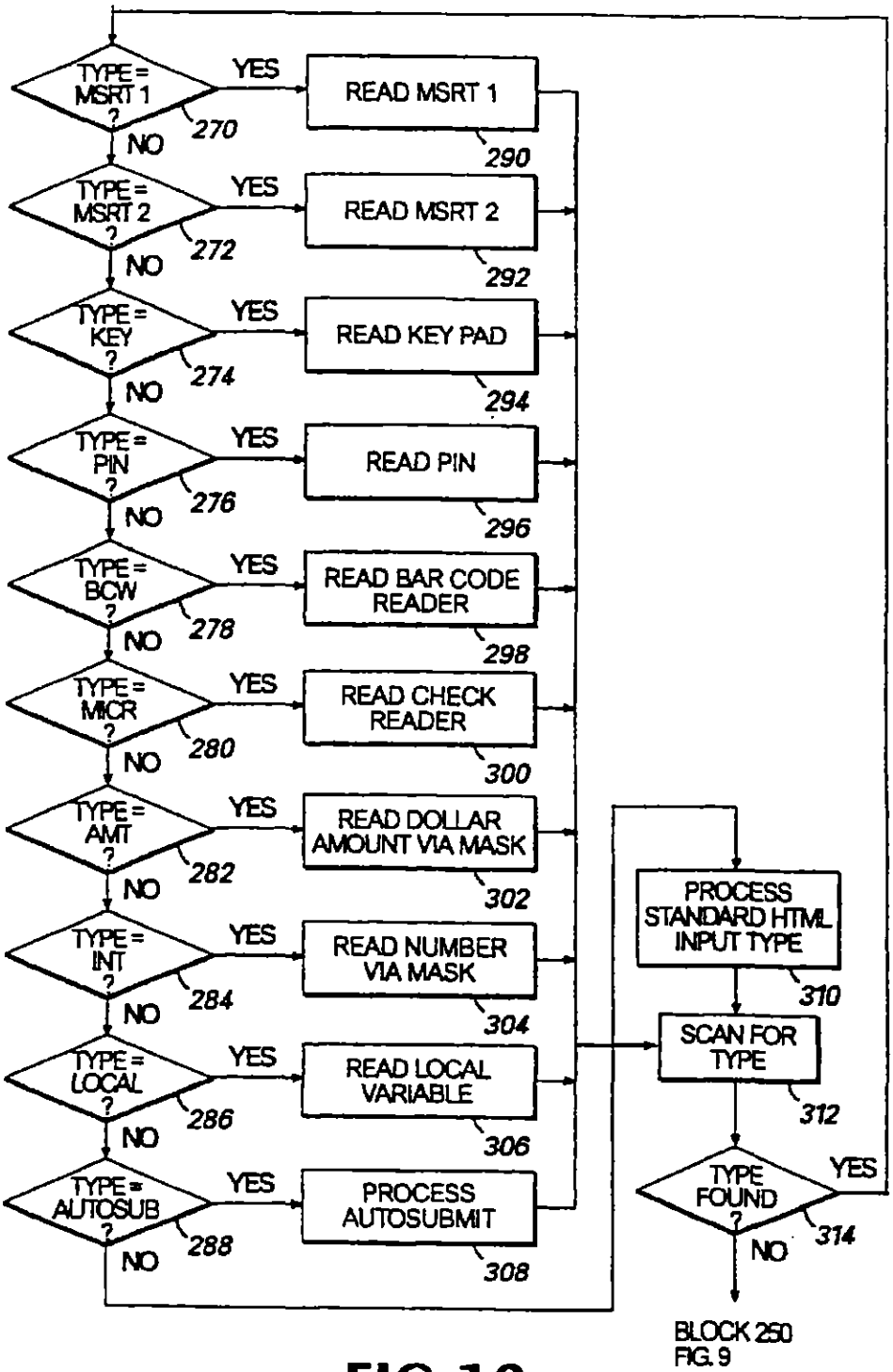

FIG.10

EXHIBIT F
PAGE 18 OF 49

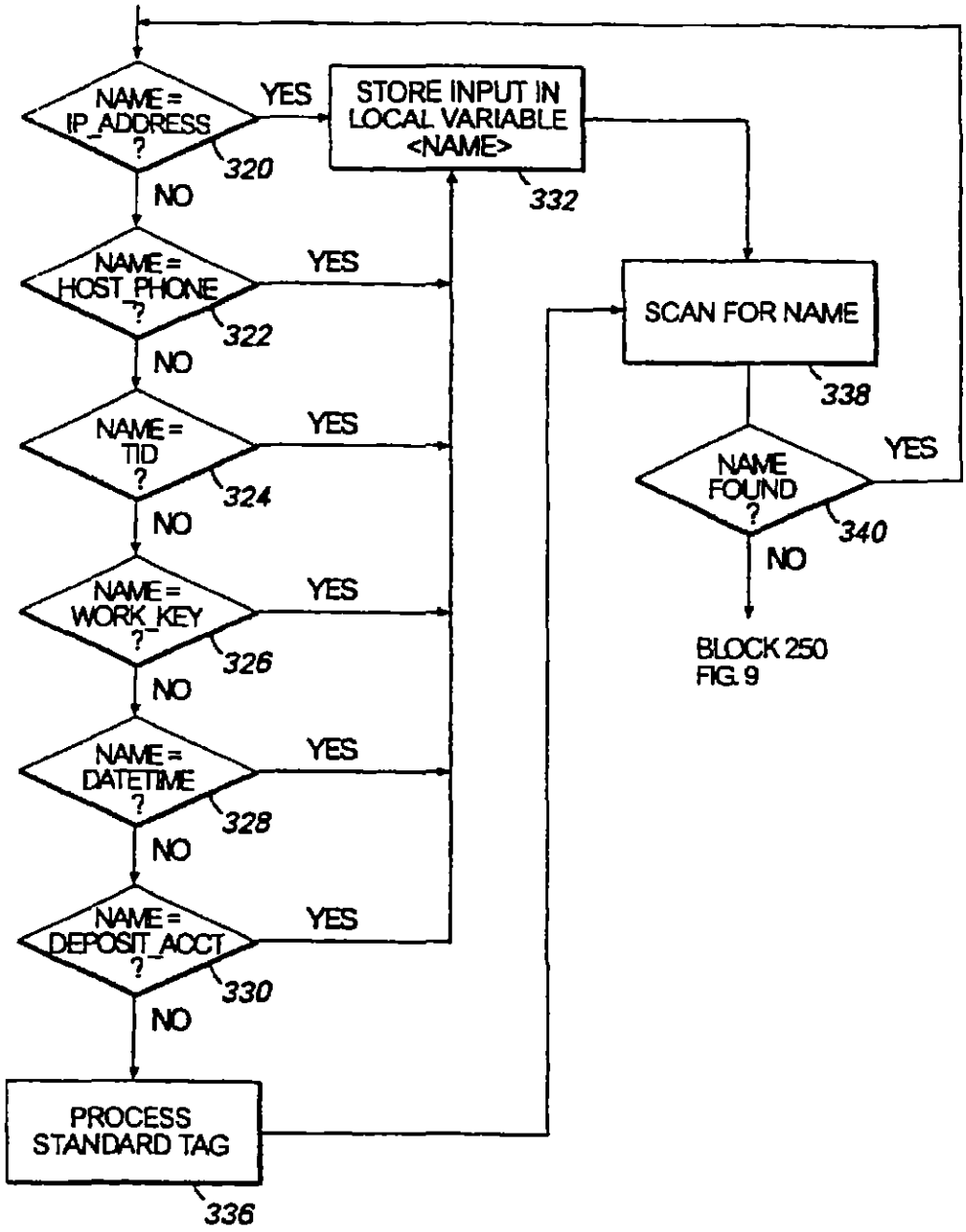

FIG.11

EXHIBIT F
PAGE 19 OF 49

1. <FORM ACTION=URL METHOD= GET>
2. <FORM ACTION=URL METHOD= POST>
3. <FORM ACTION=URL METHOD= SQL <database_name>

FIG.12

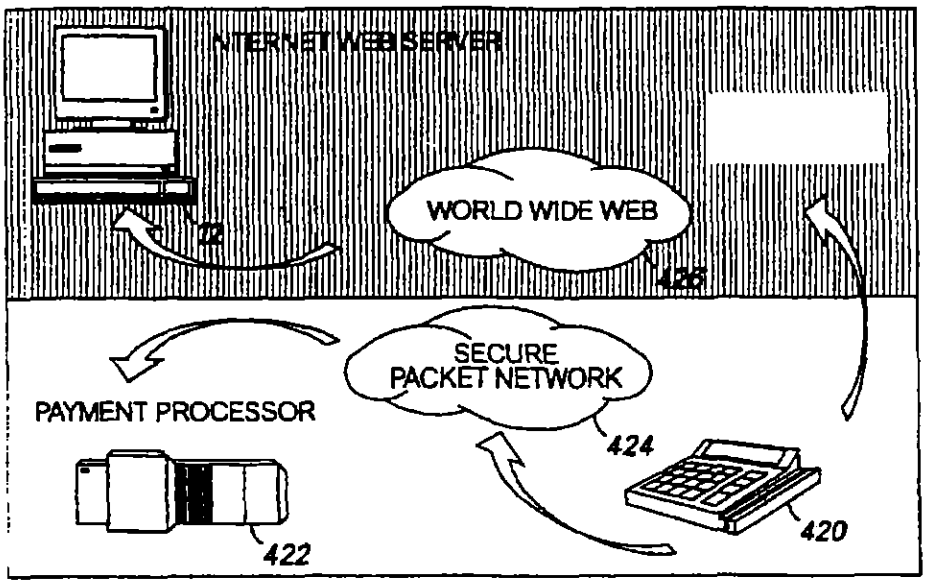

FIG.13A

<FORM ACTION=<filename> METHOD=PAYMENT>
<INPUT TYPE=AUTOSUBMIT>
</FORM>

FIG.13B

<FORM ACTION=dsinet METHOD=PAYMENT>
<INPUT TYPE=LOCAL NAME=DEPOSIT_ACCT VALUE=1234568902345678 90>
<INPUT TYPE=AUTOSUBMIT>
</FORM>

FIG.13C

EXHIBIT C
PAGE 20 OF 49

1.a. Transaction Request HTML+D

500

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
    METHOD=SQL
        "BEGIN TRAN
        IF NOT EXISTS ( SELECT substring(account, 1, 20) FROM auth_table)
            BEGIN
            INSERT TABLE=log_table VALUES=(getdate(),tid, substring (account, 1,20),
            substring( account, 22, 4), amount)
            SELECT * FROM log_table WHERE trandate = getdate()
            END
        ELSE SELECT * FROM error_table WHERE error_no=1
        COMMIT TRAN">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER ACCOUNT NUMBER:
<INPUT TYPE="MSRT2" SIZE=40 NAME=account>
ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

1.b. Transaction Accepted Response

510

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
JUNE 1 1995          10:30AM       PURCHASE</P>
TERMINAL ID:                      999999999</P>
ACCOUNT NUMBER         9999999999999999999</P>
EXP DATE:                             99/99</P>
AMOUNT                             $9999.99</P>
AUTH NUMBER                       999999999</P>
</p>
----------------------------------------</p>
        CUSTOMER SIGNATURE </p>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

1.c. Transaction Declined or Submit Error Response

520

```
<HTML>
<BODY>
DECLINED<MESSAGE>
</BODY>
</HTML>
```

FIG.14

2.a. Transaction Request HTML+D

550

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
METHOD=SQL"BEGIN TRAN
        IF NOT EXISTS ( SELECT substring(account, 1, 20) FROM auth_table)
        BEGIN
                INSERT TABLE=log_table VALUES=(getdate(),tid, substring( account, 1,20),
                substring( account, 22, 4), amount)
        END
        ELSE  BEGIN
                        SELECT * FROM error_table WHERE error_no=1
                        RETURN
                END
        INSERT TABLE=order_table VALUES=( getdate(), cust_name,address,city, state, zip,
                part_code, unit_price, tax, ship_method, ship_chrg, unit_price + tax +
                ship_chrg, substring( account, 1, 20) ,substring( account, 22, 4))
        SELECT * FROM order_table WHERE trandate = getdate()
        COMMIT TRAN">
CUSTOMER NAME:
<INPUT TYPE="TEXT"SIZE=30 NAME=cust_name></p>
ADDRESS:
<INPUT TYPE="TEXT" SIZE=40 NAME=address></p>
CITY:
<INPUT TYPE="TEXT" SIZE=20 NAME=city><p>
STATE:
<INPUT TYPE="TEXT" SIZE=2 NAME=state>
ZIP:
<INPUT TYPE="TEXT" SIZE=10 NAME=address></p>
SCAN PART CODE:
<INPUT TYPE="BCW" SIZE=9 NAME=part_code></p>
ENTER UNIT PRICE:
<INPUT TYPE="AMT" SIZE=8 NAME=unit_price></p>
TAX:
<INPUT TYPE="AMT" SIZE=5 NAME=tax></p>
SHIPPING METHOD:
<INPUT TYPE="TEXT" SIZE=10 NAME=ship_method></p>
SHIPPING AMOUNT:
<INPUT TYPE="AMT" SIZE=5 NAME=ship_chrg></p>
SLIDE CARD:
<INPUT TYPE="MSRT2" SIZE=40 NAME=account></p>
<INPUT TYPE="SUBMIT" >
</FORM>
</BODY>
</HTML>
```

FIG.15A

EXHIBIT F
PAGE 22 OF 49

2.b. Transaction Accepted Response 555

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
ORDER # 9999999999 APPROVED</p>
JUNE 1 1995        10:30AM        PURCHASE</P>
TERMINAL ID:                      999999999</P>
NAME: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
ADDRESS:</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
CITY: XXXXXXXXXXXXXXXXXXXXXXXXX</p>
STATE: XX                ZIP: XXXXXXXXXX</p>
ACCOUNT NUMBER:          9999999999999999999</p>
EXP DATE:                            99/99</p>
PART CODE:                       999999999</p>
UNIT PRICE:                       $9999.99</p>
SHIP METHOD:XXXXXXX CHARGE        $9999.99</p>
TOTAL AMOUNT:                     $9999.99</p>
AUTH NUMBER:                     999999999</p>
</p>

- - - - - - - - - - - - - - - - - - - - - - - </p>
         CUSTOMER SIGNATURE</p>

</FORM>
</BODY>
</HTML>
```

2.c. Transaction Declined or Submit Error Response 560

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.15B

3.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
METHOD SQL
    "INSERT TABLE=order_table VALUES=( getdate(), cust_name,address,city, state, zip,
        part_code, unit_price, tax, ship_method, ship_chrg,unit_price + tax +
        ship_chrg, substring( account, 1, 20) ,substring( account, 22, 4))
    SELECT * FROM order_table WHERE trandate = getdate()">
<INPUT TYPE="LOCAL" NAME=tid>
CUSTOMER NAME:
<INPUT TYPE="TEXT" SIZE=30 NAME=cust_name></p>
ADDRESS:
<INPUT TYPE="TEXT" SIZE=40 NAME=address></p>
CITY:
<INPUT TYPE="TEXT" SIZE=20 NAME=city><p>
STATE:
<INPUT TYPE="TEXT" SIZE=2 NAME=state>
ZIP:
<INPUT TYPE="TEXT" SIZE=10 NAME=address></p>
ENTER PART CODE:
<INPUT TYPE="TEXT" SIZE=10 NAME=part_code></p>
ENTER UNIT PRICE:
<INPUT TYPE="AMT" SIZE=8 NAME=unit_price></p>
TAX:
<INPUT TYPE="AMT" SIZE=5 NAME=tax></p>
SHIPPING METHOD:
<INPUT TYPE="TEXT" SIZE=10 NAME=ship_method></p>
SHIPPING AMOUNT:
<INPUT TYPE="AMT" SIZE=5 NAME=ship_chrg></p>
<INPUT TYPE="SUBMIT" >
</FORM>
</BODY>
</HTML>
```

FIG.16A

3.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
ORDER # 9999999999 APPROVED</p>
JUNE 1 1995 10:30AM PURCHASE</P>
TERMINAL ID: 999999999</P>
NAME: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
ADDRESS:</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
CITY: XXXXXXXXXXXXXXXXXXXXXXXX</p>
STATE: XX ZIP:XXXXXXXXX</p>
PART CODE: 999999999</p>
UNIT PRICE: $9999.99</p>
SHIP METHOD:XXXXXXX CHARGE: $9999.99</p>
TOTAL AMOUNT: $9999.99</p>
</FORM>
<FORM ACTION=<file_name> METHOD=PAYMENT>
<INPUT TYPE=AUTOSUBMIT>
</FORM>
</BODY>
</HTML>
```

3.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.16B

EXHIBIT E
PAGE 25 OF 49

4.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=SCR1 METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
SLIDE CARD:
<INPUT TYPE="MSRT2" SIZE=40 NAME=track2>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="LOCAL" NAME=work_key>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

4.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99                              TIME:9999A</P>
ACCOUNT NUMBER   9999999999999999999</p>
EXP DATE:                                  99/99</p>
AMOUNT:                                    $9999.99</p>
AUTH NUMBER                                999999999</p>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

4.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.17A

EXHIBIT F
PAGE 26 OF 49

5.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=SCR2 METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
ENTER PIN:
<INPUT TYPE="PASSWORD" SIZE=4 NAME=pin>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="LOCAL" NAME=work_key>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

5.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99 TIME:9999A</P>
ACCOUNT NUMBER 9999999999999999999</P>
EXP DATE: 99/99</P>
AMOUNT $9999.99</P>
AUTH NUMBER 999999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

5.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.17B

EXHIBIT F
PAGE 27 OF 49

6.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=host_URL METHOD=POST>
<INPUT TYPE="LOCAL" NAME=tid>
SLIDE CARD
<INPUT TYPE="MSRT2" SIZE=40 NAME=track2>
<INPUT TYPE="PIN" SIZE=4 NAME=pin>
ENTER AMOUNT:
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

6.b. Transaction Accepted HTML+D

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99 TIME:9999A</P>
TERMINAL ID: 99999999</P>
ACCOUNT NUMBER 9999999999999999999</P>
EXP DATE: 99/99</P>
AMOUNT $9999.99</P>
AUTH NUMBER 999999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

6.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
<INPUT TYPE="LOCAL" NAME=work_key VALUE="9999999999999999">
</BODY>
</HTML>
```

FIG.18

EXHIBIT F
PAGE 28 OF 49

7.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
        METHOD=SQL
                "IF EXISTS ( SELECT account FROM check_table)
                        SELECT * FROM check_table WHERE account = DDAaccount
                ELSE SELECT * FROM error_table WHERE error_no=1">
<INPUT TYPE="LOCAL" NAME=tid>
SCAN CHECK:
<INPUT TYPE="MICR" SIZE=20 NAME=account>
ENTER AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

7.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
DATE:99/99/99                          TIME:9999</P>
TERMINAL ID:                           999999999</P>
ACCOUNT NUMBER   99999999999999999999</P>
AMOUNT                                   $9999.99</P>
AUTH NUMBER                            999999999</P>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

7.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <error code>
</BODY>
</HTML>
```

FIG.19

EXHIBIT F
PAGE 29 OF 49

8.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM  ACTION=dbase_URL
        METHOD=SQL
               * BEGIN TRAN
                 IF NOT EXISTS ( SELECT substring(account,51 ,20) FROM auth_table)
                        BEGIN
                        SELECT cur_bal FROM cust_tbl WHERE substring(account,51 ,20)=account
                        SELECT amount = amount - ( points / .01 )
                        SELECT cur_bal = cur_bal + ( amount * .01 )
                        UPDATE TABLE=cust_tbl VALUES=( getdate(), account, cur_bal - points )
                        SELECT * FROM log_table WHERE trandate = getdate()
                        INSERT TABLE=log_table VALUES=( getdate(), tid, substring(account,51 ,20),
                        substring( account, 72, 4), amount)
                        END
        ELSE  SELECT * FROM error_table WHERE error_no=1"
        COMMIT TRAN">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER ACCOUNT NUMBER:
<INPUT TYPE="MSRT1" SIZE=90 NAME=account>
ENTER  AMOUNT
<INPUT TYPE="AMT" SIZE=8 NAME=amount>
REDEEM POINTS?
<INPUT TYPE="INT" SIZE=6 NAME=points>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

8.b. Transaction Accepted Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
JUNE 1 1995        10:30AM            PURCHASE</P>
TERMINAL ID:                         999999999</P>
ACCOUNT NUMBER      99999999999999999999</P>
EXP DATE:                                99/99</P>
AMOUNT                               $9999.99</P>
AUTH NUMBER                         999999999</P>
</p>
- - - - - - - - - - - - - - - - - - - - - - </p>
          CUSTOMER SIGNATURE</p>
</p>
THANK YOU!</p>
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX</p>
POINTS REDEEMED:                    999999</p>
POINTS EARNED:                      999999</p>
CURRENT POINT BALANCE:              999999</p>
</FORM>
APPROVED:999999999</P>
</BODY>
</HTML>
```

8.c. Transaction Declined or Submit Error Response

```
<HTML>
<BODY>
DECLINED <MESSAGE>
</BODY>
</HTML>
```

FIG.20

EXHIBIT F
PAGE 30 OF 49

9.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
      METHOD=SQL
            "SELECT fields FROM table WHERE condition">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER SEARCH TABLE NAME:
<INPUT TYPE="TEXT" SIZE=10 NAME=table>
ENTER SEARCH FIELD NAMES:
<INPUT TYPE="TEXT" SIZE=100 NAME=fields>
ENTER SEARCH CONDITION:
<INPUT TYPE="TEXT" SIZE=50 NAME=condition>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

9.b. Transaction Response

```
<HTML>
<BODY>
<FORM ACTION=TO PRINTER METHOD=POST>
FIELD1  FIELD2  FIELD3  ••••••  FIELDN  </p>
_____  _____  _____        _____  </p>
XXXXX  XXXXX  XXXXX        XXXXX  </p>
XXXXX  XXXXX  XXXXX        XXXXX  </p>
•
•
XXXXX  XXXXX  XXXXX        XXXXX  </p>
</FORM>
</BODY>
</HTML>
```

FIG.21

EXHIBIT F
PAGE 31 OF 49

10.a Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM ACTION=dbase_URL
      METHOD=SQL
      * INSERT TABLE=log_table VALUES=( getdate(), tid, gross_sales, opn_chks, voids,
      emp_disc, mgr_disc, vip_card, man_over, coupons, sales_tax, c_dep1, c_dep2,
      c_dep3, c_dep4, chg_fund,cc_dep, batch_no, chrg_sales, paid_outs, co_sales,
      cc_sales, te_sales, gross_sales - opn_chks - voids - emp_disc - mgr_disc - vip_card-
      man_over - coupons - sales_tax, gross_sales - opn_chks - voids -
      emp_disc - mgr_disc - vip_card - man_over - coupons - c_dep1 - c_dep2 -
      c_dep3 - c_dep4 - chg_fund - cc_dep - batch_no - chrg_sales - paid_outs)
      SELECT * FROM log_table WHERE trandate = getdate()">
<INPUT TYPE="LOCAL" NAME=tid>
ENTER GROSS SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=gross_sales>
ENTER OPEN CHECKS:
<INPUT TYPE="INT" SIZE=7 NAME=opn_chks>
ENTER VOIDS:
<INPUT TYPE="INT" SIZE=7 NAME=voids>
ENTER EMP DISCOUNTS:
<INPUT TYPE="INT" SIZE=7 NAME=emp_disc>
ENTER MGR DISCOUNT:
<INPUT TYPE="INT" SIZE=7 NAME=mgr_disc>
ENTER VIP CARD:
<INPUT TYPE="INT" SIZE=7 NAME=vip_card>
ENTER MANUAL OVERRINGS:
<INPUT TYPE="INT" SIZE=7 NAME=man_over>
ENTER COUPONS:
<INPUT TYPE="INT" SIZE=7 NAME=coupons>
ENTER SALES TAX:
<INPUT TYPE="AMT" SIZE=8 NAME=sales_tax>
ENTER CASH DEPOSIT 1:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep1>
ENTER CASH DEPOSIT 2:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep2>
ENTER CASH DEPOSIT 3:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep3>
ENTER CASH DEPOSIT 4:
<INPUT TYPE="AMT" SIZE=8 NAME=c_dep4>
ENTER CHANGE FUND:
<INPUT TYPE="AMT" SIZE=8 NAME=chg_fund>
```

FIG.22A

EXHIBIT F
PAGE 32 OF 49

```
ENTER  CC DEPOSIT:
<INPUT TYPE="AMT" SIZE=8 NAME=cc_dep>
ENTER BATCH #:
<INPUT TYPE="INT" SIZE=3 NAME=batch_no>
ENTER CHARGE SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=chrg_sales>
ENTER PAID OUTS:
<INPUT TYPE="INT" SIZE=8 NAME=paid_outs>
ENTER CARRY OUT SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=co_sales>
ENTER CREDIT CARD SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=cc_sales>
ENTER TAX EXEMPT SALES:
<INPUT TYPE="AMT" SIZE=8 NAME=te_sales>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

10.b. Transaction Response

```
<HTML>
<BODY>
<FORM  ACTION=TO PRINTER METHOD=POST>
JUNE 1 1995        10:30AM        DAILY REPORT</P>
TERMINAL ID:                          999999999</P>
GROSS SALES                           999999.99</P>
 VOIDS                        99        99999.99</P>
 EMP DISCOUNTS                99        99999.99</P>
 MANAGER DISCOUNTS            99        99999.99</P>
 VIP CARD                     99        99999.99</P>
 COUPONS                      99        99999.99</P>
 MANUAL OVERRINGS             99        99999.99</P>
 SALES TAX                              999999.99</P>
 CASH DEPOSIT 1                         999999.99</P>
 CASH DEPOSIT 2                         999999.99</P>
 CASH DEPOSIT 3                         999999.99</P>
 CASH DEPOSIT 4                         999999.99</P>
 CASH DEPOSIT 5                         999999.99</P>
 CHANGE FUND                            999999.99</P>
 CC DEPOSIT                   999       999999.99</P>
 CHARGE SALES                           999999.99</P>
 PAID OUTS                     99        99999.99</P>
 CARRY OUT SALES                        999999.99</P>
 CREDIT CARD SALES                      999999.99</P>
 TAX EXEMPT SALES                       999999.99</P>
 ----------------------------------------</P>

 NET SALES                               9999999</P>
 OVER/SHORT                              9999999</P>
</FORM>
</BODY>
</HTML>
```

FIG.22B

EXHIBIT F
PAGE 33 OF 49

11.a. Transaction Request HTML+D

```
<HTML>
<BODY>
<FORM  ACTION=MAIL TO: mail_to>
 ENTER MAIL ADDRESS:
<INPUT TYPE="TEXT" SIZE=20 NAME=mail_to>
ENTER MESSAGE:
<INPUT TYPE="TEXT" SIZE=100>
<INPUT TYPE="AUTOSUBMIT">
</FORM>
</BODY>
</HTML>
```

FIG.23

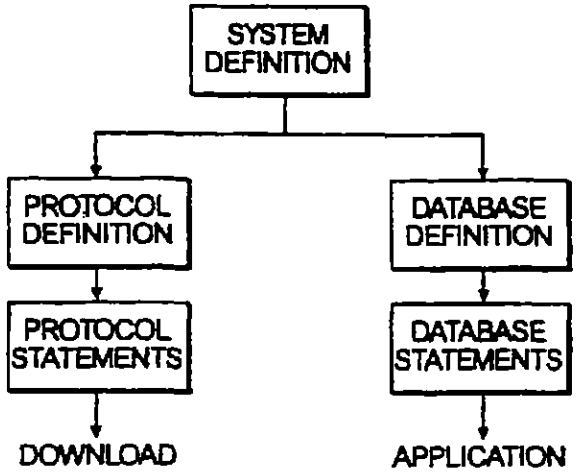

FIG.24A

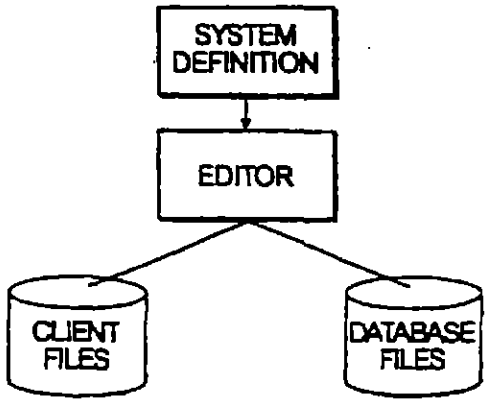

FIG.24B

EXHIBIT F
PAGE 34 OF 49

OPEN NETWORK SYSTEM FOR I/O OPERATION INCLUDING A COMMON GATEWAY INTERFACE AND AN EXTENDED OPEN NETWORK PROTOCOL WITH NON-STANDARD I/O DEVICES UTILIZING DEVICE AND IDENTIFIER FOR OPERATION TO BE PERFORMED WITH DEVICE

This application is a continuation of application Ser. No. 09/314,266, filed on May 18, 1999 now U.S. Pat. No. 6,366,967, which is a continuation of application Ser. No. 08/995,123 filed Dec. 19, 1997 (now U.S. Pat. No. 5,905,908), which is a continuation of application Ser. No. 08/493,772 filed Jun. 22, 1995 (now U.S. Pat. No. 5,742,845).

FIELD OF THE INVENTION

This invention relates to data transaction systems, and more particularly, to data transaction systems using non-standard input/output devices.

BACKGROUND OF THE INVENTION

Data transaction systems which communicate with a plurality of remote terminals to transfer information used to complete a transaction or compile a database are well known. Typically, such systems include a central transaction processing system which may maintain a database of information such as customer or consumer data. Exemplary information in such a database may include customer identification, customer account numbers, credit limits and/or account balances from which a customer may draw. The central transaction processing system is typically coupled to a plurality of remote transaction or data input terminals. Transaction computers may include special purpose devices such as automatic teller machines (ATMs), point of sale (POS) terminals, credit card terminals, and screen phone terminals. Screen phone terminals are devices which integrate a telephone with an ATM-like device and possibly a magnetic card swipe reader. Data input terminals may include personal computers (PCs) interfaced to data collection devices or special purpose data collection terminals or monitors.

In these known data transaction systems, a user usually initiates a transaction by requesting access to finds in an account or from a credit line maintained by the central processing system. The request is transmitted to the central processing system which performs a verification to determine whether the user is a valid user of the system, has an account within the system, and that the amount of the transaction is within the limits of the consumer's credit line or that the user has the requested funds available in an existing account monitored by the central processing system. The central processing system then transmits authorization for or denial of the transaction to the remote terminal. In response to the message from the central processing system, the remote terminal dispenses cash (for an ATM) or the merchant provides the goods being purchased to the user if the authorization message indicates that the consumer's funds will be transferred to the merchant's account. Similar communication exchanges occur in data systems where electronic documents and other information are provided to a central site for compilation or processing. Consequently, this background discussion applies to all such transaction and data systems. Though the remainder of the discussion is directed to transaction systems, the reader should appreciate that the comments also apply to data systems as well.

The remote terminals may be coupled to the central processing system in several ways. For example, in some ATM systems, the ATMs are coupled to the central processing system through dedicated telephone or other data communication lines. These systems are preferred because they provide a relatively high degree of security since the dedicated data line coupling the central processing system to the ATM is not generally accessible by members of the public. The physical security of the dedicated data line is, however, expensive because no other traffic may utilize the line. Thus, the cost of leasing the dedicated line to an ATM with relatively low volumes of transactions may yield a high communication cost per transaction.

In an effort to reduce the communication cost per transaction, some transaction or data systems utilize telephone lines through a publicly-switched telephone network (PSTN) which may be accessed by other members of the public. Specifically, devices such as credit card terminals and screen phone terminals typically include a modem which converts the digital messages of the remote terminal into frequency modulated analog signals which may be transmitted over telephone lines to a modem at the central processing system. In other systems, the terminal may communicate digital data directly over ISDN lines of the PSTN to the central processing system. This line of communication between a remote terminal and the central processing system is performed by having the remote terminal dial a telephone number associated with the central processing system to establish communication with the central processing system. This type of communication path is relatively secure because the switching networks for the communication traffic through the PSTN are not readily accessible by the public and during the course of the financial transaction, only the central processing system and remote terminal are on the line.

Regardless of the communication method used to couple the central processing system to the remote terminals, the protocol and data formats used between the devices is typically proprietary. That is, the operator of each financial transaction system designs its own protocol and data message format for communication with the processor at the central site or generates a variant within a standard such as those established by the ANSI committee or the like for such communication. As a result, the remote terminals must include software that supports each operator's protocol and message formats in order to be compatible with an operator's central site. For example, application software in a credit terminal such as the TRANZ330, TRANZ380, or OMNI390 manufactured by VeriFone implement one or more of the communication protocols and formats for National Data Corporation (NDC), VISANET, MASTERCARD, BUYPASS, and National Bancard Corporation (NaBANCO) system processors in order to support transactions with the most popular transaction centers. Thus, the communication software absorbs a significant amount of terminal resources which could be used to support other terminal operations.

A related problem arises from the expanding home banking market. A customer of home banking system typically uses a screen phone terminal or a personal computer (PC) having a modem to establish communication through a PSTN to a central transaction processing system. Again, the operator of the central processing system must provide information regarding the data message formats for communicating with the central processing system to a vendor of software for the home banking terminals or must provide that software to its customers. As a result, home banking

customers must purchase software to communicate with each banking system of which the customer wants to be a member. This cost and the need to install additional communication programs may make some consumers reluctant to be a member of more than one banking system or to change banking systems.

A communication system becoming increasingly popular and which provides standardized communication is the Internet. The Internet is an open network of networks which communicate through a variety of physical communication devices such as telephone lines, direct communication lines, and the like. Each network is coupled to the main Internet network for communication through a host computer supporting a TCP/IP router or bridger. The host computer typically includes a program, frequently called a Web server, which acts as a gateway to resources at the host computer which may be resident on the host computer or a network coupled to the host computer. Each server has an address identifying the location of the resources available through the Web server. The router recognizes communication for the server and directs the message to the server or it recognizes that the communication should be forwarded to another server. As a result, communication within the Internet may be point-to-point, but more likely, the communication path is a somewhat circuitous one with the information passing through the routers of multiple servers before reaching its final destination.

A number of message protocols and formats have been developed for the Internet. The physical communication protocol and data message format is the Transport Control Protocol/Internet Protocol (TCP/IP). The TCP/IP protocol involves multiple layers of encapsulating headers containing communication information which are used to provide byte streams or datagram communications to computers on the networks coupled to the Internet. Encapsulated within TCP/IP headers are protocols which are used to format the data messages or transfer data from one computer to another computer coupled to the Internet. These protocols include File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP), Post Office Protocol (POP), Telnet, and Hyper Text Transport Protocol (HTTP). The advantage of these protocols is that each provides a standardized communication format for transferring information between computers on the Internet. These protocols are typically called open system protocols as they are publicly known and may be utilized by any programmer to develop programs for communicating with another computer coupled to the Internet. These non-proprietary protocols have contributed to the acceptance of using the Internet as an open network for coupling computer networks together.

While the Internet provides an open network for computer communication with publicly accessible protocols and formats, the Internet suffers from a number of limitations which preclude its effective use as a transaction or data system which uses non-standard I/O terminals and devices. First, circuitous communication presents a number of security issues for such a system. For example, a Web server could incorporate a router which examines the address of each message coming through it and upon recognizing an address associated with a central transaction processing system, copy the data message for the unauthorized retrieval of customer-sensitive information such as account numbers and personal identification numbers (PINs) which may be contained in the message.

A second limitation of open networks such as the Internet is that communication on such networks is only supported for computers acting as servers or clients. Specifically, all of the protocols and formats are constructed for standard input/output (I/O) operations for a PC terminal. That is, text information is directed to a standard monitor screen, user input is expected from a standard keyboard, and files are transferred to standard peripherals such as a hard disk or diskette drive. Especially absent is the ability in open network protocols for communication with devices that only use communication interfaces such as RS-232C. As a result, communication over the Internet is primarily performed with standard PCs through network communication methods and interfaces.

This presents a number of problems for home banking or for interfacing non-standard I/O terminals such as credit card terminals or screen phones to open networks such as the Internet either directly or through a PC. Generally, non-standard I/O devices are devices which interface to a PC through a port not normally used for networks, such as a RS-232C port, or are devices which have limited input and output capabilities such as small screen displays or ten keypads. These devices are not supported on the Internet because servers use protocols that communicate with PCs supporting standard QWERTY keyboards and standard monitors. Consequently, users are limited to entering account numbers and the like through a keyboard of a PC-like device for processing at a central transaction processing system. To request a transaction, one need only have a person's credit card account number. If the credit card number had to be input through a magnetic card reader, unauthorized access to a customer's account would be less likely since physical possession of the credit card would be required to initiate the transaction.

Another limitation of the standard I/O devices currently supported by the open network protocols is the lack of encryption. For example, PIN pads, which are typically incorporated in ATMs, automatically encrypt in hardware a PIN entered by a user. Such devices typically encrypt the number by implementing a data encryption standard (DES) algorithm in hardware before the PIN is transmitted or stored. When a standard keyboard is used to input the PIN, no hardware encryption is performed and, as a result, an unencrypted copy of the PIN is provided to the memory of the PC. Storage of unencrypted PINs is in contravention of current banking regulations. If PIN pads could be read via Internet protocols, then such a lapse in PIN security would be less likely to occur.

Another I/O device not supported on open networks are smart cards which are increasing in use. Smart cards include a processor and memory in which information regarding the amount of funds in a particular account, a transaction history, account numbers, and customer data may be stored. The card may be read through a smart card reader which is a computer having a processor and memory but usually provided with non-QWERTY keypads and limited displays. A transaction processor may validate a card owner through a PIN provided through a keypad, determine the amount of money remaining on the card and debit the card itself for a transaction amount by communicating with the smart card reader with one of the proprietary protocols discussed above. Such information is not readily obtainable by the owner of the card and so cannot be entered through a keyboard or the like. Smart card readers are non-standard devices which may be coupled to a PC through a COMM1 or COMM2 port. However, none of the standard protocols and message formats for open network communications currently provide I/O operations for such devices.

All systems which attempt to provide three party communication to execute an electronic transaction suffer from

EXHIBIT F
PAGE 36 OF 49

a number of limitations which present risks greater than those in a normal transaction performed at the point of sale. In a typical point of sale (POS) transaction, the consumer hands a debit or credit card to a merchant's agent who may examine the card for security markings such as holograms, watermarks, or a cardholder signature. The agent then places the card into a reader for acquiring information from the card and, in some cases, have the consumer enter a PIN into a PIN entry device which encrypts the PIN in a hardware implemented scheme. If the PIN is entered, it is transmitted with the information from the card to a processing center, typically in one of the formats discussed above, under a X.25 protocol or the like. The processing center returns an authorization granted or denied message. The reader typically has a printer coupled to it through an RS-232C port or the like and a purchase agreement is printed. The consumer signs the agreement, the merchant's agent may verify the signature, and the merchant retains an original of the agreement and the consumer a copy. In this scenario, the merchant has initiated the communication to the processing center. The safeguards noted above permit the processing center to charge a merchant a lower processing fee than when a consumer initiates a transaction. Consumer initiated transactions present a greater risk because the consumer provides an agent an account number in a telephone conversation or non-encrypted DTMF transmission. Thus, there is no card inspection, signature verification, or PIN verification. As a result, such transactions are limited to credit cards because debit cards require that the cardholder be present to enter a PIN into an appropriate PIN entry device.

What is needed is a system that permits consumers remote from a merchant to order goods and present payment in a secured manner so the merchant's risk and processing costs, as well as a cardholder's exposure to fraud, is reduced. What is needed is a way for a processing center to communicate through an open network with non-standard I/O devices such as credit card terminals, personal digital assistants, and screen phone terminals or with non-standard I/O devices coupled to the open network through a PC or the like. What is needed is a transaction or data system which utilizes an open network such as the Internet to support electronic transactions or data compilation in a secure manner without undue limitation as to the devices with which communication may be made.

SUMMARY OF THE INVENTION

The present invention provides transaction and data systems which may be implemented on an open network such as the Internet. The system comprises a server for communicating in an open network protocol and a plurality of input/output (I/O) devices coupled to the server through an open network, the I/O devices communicating with the server in the extended open network protocol that supports communication with non-standard I/O devices over the open network. The system of the present invention provides a server with the capability of communicating with a number of I/O devices useful in transaction and data systems which heretofore have been unsupported on an open network system such as the Internet.

The system of the present invention is implemented by extending present open network communication protocols and data message formats to communicate with non-standard I/O devices either coupled to an open network as a client or coupled to an open network through a client, such as a PC, credit card terminal, screen phone, or PDA. That is, commands which are compatible with the communication schema of a presently-implemented protocol for the Internet are used and additions are made to commands implemented within the control structure of that existing protocol to support non-standard I/O device communication. At the server, the extended protocol is further supported by a common gateway interface (CGI) which converts the communication from a non-standard I/O device to a format which is compatible with a transaction or data application program which may be executed on the server or a computer coupled to the server. In this manner, the CGI permits the processing of the extended capability commands to be segregated from the communication functions performed by the server.

Preferably, the server and the I/O devices communicate through an Internet protocol and most preferably, the Hyper Text Transport Protocol (HTTP), to exchange data between an application program and non-standard I/O devices over an open network. Although HTTP is the preferred protocol used to implement the present invention, other protocols such as Telnet or SMTP, for example, may also be extended in a similar manner. Specifically, the HTTP protocol is expanded to communicate with printers, magnetic card readers, credit card terminals, smart card readers, check readers, PIN pads, bar-code readers, PDAs, or the like, and includes a command which instructs a non-standard I/O device to disconnect from the open network and re-couple to a transaction processing system to transfer funds from a consumer account to a merchant account through a PSTN or dedicated data line. By using these extended capability commands within HTTP, a processing system may operate on an open network such as the Internet and communicate with transaction or other data I/O devices which have not previously been able to couple to such open networks. Such a system may be used to execute a transaction between a consumer and a merchant so the merchant receives remittance information in a timely manner. The system permits the consumer to initiate a transaction and order from a merchant and then use a more secure link supported by PIN entry devices or the like to reduce the risk of fraud for the transaction.

Because the server may communicate through such open networks with non-standard I/O devices, the transaction or data processing system is available for the ever-expanding market available through the Internet. Such a system is able to communicate with non-standard I/O devices in myriad locations such as retail establishments or in consumers' homes. For example, a consumer may utilize the standard capability of an Internet protocol to communicate with a server that provides information regarding services or goods for sale over the Internet and then consummate a sales transaction by using the extended capability of the Internet protocol. Such a home consumer could provide transaction data through a smart card reader coupled to a COMM1 or COMM2 port of a PC. A database program executing at the server for the central processing site may accept product ordering information from a non-standard keypad or touch screen associated with a screen phone terminal at the remote site and then communicate with the smart card reader to consummate the transaction. Such a transaction system requires that the consumer have physical possession of the smart or credit card and not simply knowledge of the account number. Likewise, the server would be able to communicate with a PIN pad or the like to ensure the hardware encryption of PINs and other data before it is transmitted to the server site. Such a system is less susceptible to consumer fraud.

Another feature of the present invention is a PAYMENT command implemented in the extended Internet protocol

that directs a non-standard I/O device or a PC interfaced with such devices to communicate with a transaction processor through an alternative communication link. In one form, the PAYMENT command is used by a merchant terminal to submit a consumer's account number with a merchant deposit account number through a PSTN network or the like to the processing center. In another form of the PAYMENT command, a client program in a consumer's terminal receives an account number for a merchant account from a merchant's server with the PAYMENT command. On receipt of this command, the client program suspends its operation and passes the account number to a conventional bank processing program co-resident in memory. The bank processing program establishes a standard communication link with a transaction processing system through a dedicated data line or a PSTN network. Using that communication link, the bank processing program executes a commercial transaction using a standard VISA protocol or the like. The consumer may use a magnetic stripe reader and a PIN entry device to improve the security of the data transmission. The transaction center may transmit remittance data over the open network to the merchant so the merchant is apprised of payment and ships the ordered product. Once this consumer initiated transaction is complete, the bank processing program terminates and returns control to the client program which may terminate communication with the open network or retrieve information from another server on the open network for another transaction. In this way, the user may use the open network for non-confidential communication such as collecting product information, pricing, and product availability. This information may be collected quickly and efficiently using the extended Internet protocol. The conventional bank processing program and more secure communication links may then be used for the confidential information required for the transaction. Thus, the present invention is able to combine the features and advantages of the Internet with the more secure communication link and data security enhancing devices of systems presently known.

Preferably, an editor is provided which permits a user to define an application database table with data fields, define client application data fields, and define the integrated forms for communicating data between the defined database tables and a client application. The editor verifies the syntax of the user generated integrated forms containing extended Internet protocol statements and client application statements. The editor ensures that the variable names for the client application and the data fields for the database application correspond. Following the generation of the integrated form, the editor parses the integrated form to segregate the database language statements from the extended Internet protocol statements. A database language identifier is substituted in the Internet protocol statements for the database statements contained in the integrated form. The Internet protocol statements are downloaded as a file which is interpreted by the client program for the collection and submission of data from non-standard I/O devices to the database application. The database language statements segregated from the extended Internet protocol statements are placed in a second file which is named to correspond to the database table defined by the user. The CGI application recognizes the database language identifier contained in the returned forms of the Internet protocol statements. The CGI application correlates the database identifier with the file previously generated by the editor which contains the database command statements. The application then inserts the data from the returned form into the database command statements and provides the re-integrated database command statements to the database application. In this manner, the database may be queried by or retrieve data from the non-standard I/O device. In the most preferred embodiment, the editor permits a user to develop integrated forms comprised of the extended HTML language and standard query language (SQL) database application statements. In this manner, the user does not have to manually generate the SQL commands, the HTML commands, and carefully correlate the data fields of the two commands in order to implement a transaction between a client and a database.

These and other advantages and features of the present invention may be discerned from reviewing the accompanying drawings and the detailed description of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention may take form in various components and arrangement of components and in various steps and arrangement of steps. The drawings are only for purposes of illustrating a preferred embodiment and are not to be construed as limiting the invention.

FIG. **1** is a diagram of an open network system in which the present invention is utilized;

FIG. **2** is a diagram of the format of the FORM and INPUT tags implemented in the preferred embodiment of the present invention;

FIG. **3** is a diagram of the preferred SQL commands supported in the preferred embodiment of the present invention;

FIG. **4** is a flowchart of the high level processing of the client program which interprets the HTML files of the preferred embodiment of the present invention;

FIG. **5** is a flowchart of the HTML file processing performed by the client program of the preferred embodiment of the present invention;

FIG. **6** is a flowchart of the attribute processing for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. **7** is a flowchart of the processing of the ACTION attribute for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. **8** is a flowchart of the processing for the METHOD attribute for the FORM tag performed by the client program of the preferred embodiment of the present invention;

FIG. **9** is a flowchart of the attribute processing for the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. **10** is a flowchart of the processing for the TYPE attribute for the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. **11** is a flowchart of the processing for the NAME attribute of the INPUT tag performed by the client program of the preferred embodiment of the present invention;

FIG. **12** is a diagram of the format for the ACTION attribute for the FORM tag performed by the common gateway interface between the Web server and an application program;

FIG. **13A** is a diagram of the possible communication paths which may be used by an I/O device according to the principles of the present invention;

FIG. **13B** shows an exemplary FORM tag and INPUT tag for the PAYMENT method implemented in a merchant's terminal according to the principles of the present invention;

FIG. **13C** shows an exemplary FORM tag and INPUT tag for the PAYMENT method implemented in a consumer's terminal according to the principles of the present invention;

FIG. 14 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a card initiated payment authorization and capture transaction;

FIGS. 15A and B shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a bar code reader input with card-initiated payment authorization transaction;

FIGS. 16A and B shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a key input order with secure payment transaction;

FIG. 17A shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a smart card debit (Type 1) transaction;

FIG. 17B shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a smart card debit (Type 2) transaction;

FIG. 18 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a debit card transaction;

FIG. 19 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a check verification transaction;

FIG. 20 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a customer frequency transaction;

FIG. 21 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for an item search transaction;

FIGS. 22A and B shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for retail store end of day reporting;

FIG. 23 shows exemplary integrated statements for a file used in the preferred embodiment of the present invention to generate the HTML files for the client program and the SQL files for the application program for a store reporting an e-mail transaction;

FIG. 24A is a diagram of a manual development process for the files interpreted by the client program and the files interpreted by the application program in accordance with the principles of the present invention; and

FIG. 24B is a diagram of the generation of the files interpreted by the client program and the files interpreted by application program performed by an editor constructed in accordance with the principles of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

A transaction or data system constructed in accordance with the principles of the present invention is shown is FIG. 1. The system **10** includes a Web server **12** which is coupled to an open network **14** such as the Internet for communication with various I/O devices and terminals. For example, the I/O devices which may be coupled directly to network **14** include standard I/O devices already supported by Internet protocols such as PCs **30** and non-standard I/O devices such as a screen phone terminal **16**, a personal digital assistant (PDA) **18**, and a credit card terminal **20**. Other exemplary non-standard I/O devices such as smart card reader **32**, personal identification number (PIN) pad **34**, magnetic card swipe reader **36**, printer **38**, or the like, may be coupled to PCs through non-standard I/O ports such as COMM1 and COMM2 ports or to other non-standard I/O devices such as phone terminal **16**, PDA **18**, or credit card terminal **20**. Typically, these devices are coupled to PCs or devices **16**, **18**, or **20** through an interface such as a RS-232C interface. Merchants or other vendors may use a Web server **2** to couple to network **14** to communicate with the devices and processing system **40**.

The Web server **12** is preferably coupled to a Common Gateway Interface (CGI) application **28** which converts and communicates the data and commands between the devices on network **14** and the processing system **40** so the I/O devices do not have to use the database command language to interact with the database. System **40** and the devices may communicate directly if they are implemented in the same language or if a user implements a communication interface such as CGI **28** that correlates data fields in the client with those in system **40**. Server **12**, CGI **28**, and the applications supporting system **40** may all reside on a single host computer or they may reside on separate computers coupled together by a local area network (LAN) or a wide area network (WAN). Preferably, the application interfaces with a database which supports Open Data Base Connectivity (ODBC) and Structured Query Language (SQL).

The communication sessions between the I/O devices coupled to the open network **14** and the Web server **12** are generally conducted in the same fashion as Internet protocol communication sessions are currently performed. That is, the I/O device establishes a communication connection with Web server **12**, sends a request to the Web server, the Web server responds to the request and the I/O device or server closes the connection. Preferably, the non-standard I/O devices or PCs interfaced to such devices selectively couple to a local access port on the open network **14** through a local modem/ISDN connection. In this manner, the device is only coupled to the open network **14** when a transaction or a data operation is to be performed. While connected to the open network **14**, a device may access a number of servers to accomplish a purpose. For example, a device may couple to a local access port and communicate with a first server to check inventory levels at a site, communicate with a second server to order stock for the inventory, and communicate with a third server to settle payment for the ordered goods. When all aspects of the transaction are complete, the connection with the local access port is terminated. In the preferred embodiment of the present invention, the protocol used to transport data messages between Web server **12** and the I/O devices coupled to the open network **14** is the Hyper Text Transport Protocol (HTTP), although other open system protocols utilized on the Internet may be used.

In standard HTTP protocol, a client program executing in one of the I/O devices may initiate communication with a server by sending a query message of the format:

http://<host>:<port>/<path>?<search part>

The message identifies the client as seeking communication with a HTTP server at the host address on the specified

EXHIBIT F
PAGE 39 OF 49

port. In the HTTP protocol, the default value for the port is 80 and the host address is the Internet protocol (IP) address of the type well-known in the art. The path value selects the file in the HTTP server which is activated in response to the message and the search part specifies a query for the selected file. In the initial communication, the query may be omitted so that the selected host file responds to the client program before a query is processed.

In the present invention, the client program uses a similar message to initiate a transaction or data operation, except that database commands are preferably embedded in a file at the server **12** and not in the "search part" of the command, although search parts may be constructed in accordance with the principles of the present invention that support non-standard I/O devices. Preferably, the client program interprets Hyper Text Markup Language (HTML) files containing HTML commands for communicating data between non-standard I/O devices and server **12**. Most preferably, the HTML commands contain identifiers which are used by the CGI to place data returned in the forms of the HTML commands into database commands for queries or data insertions for the database. HTML is a command language well known for the retrieval and display of electronic documents for standard I/O devices such as PCs supported by full screen monitors, QWERTY keyboards, and standard peripherals such as hard disk drives and diskette drives. Standard HTML commands use text and previously known commands that reference Universal Resource Locators (URLs) to support the communication of electronic documents. These documents are files which may contain HTML commands, text, audio, video, or image data. The present invention extends HTML with commands that support communication between the server and the non-standard I/O devices.

In the HTTP protocol, data may be obtained during a communication session by using a tag called a FORM as part of the file defined by <path> in the command discussed above. The FORM format for standard HTTP is:

```
<FORM ACTION="URL"
    METHOD=GET | POST
>
Command
</FORM>
```

where "|" is an "OR" operator. The commands supported by standard HTTP are INPUT, SELECT, and TEXTAREA. Additionally, standard HTTP permits the inclusion of text data in the command area. In the present invention, HTML has been extended to support new ACTIONs, METHODs, and INPUTs.

In accordance with the principles of the present invention, tags are preferably used to identify device transfers and input operations. Preferably, the FORM tag is used to identify device transfers and ACTION and METHOD attributes further identify the device operation. As shown in FIG. **2**, the extended ACTION field may include a FROM and TO attribute for accessing a local terminal file or smart card reader or a TO PRINTER attribute for directing output data to a printer local to the I/O device. The FROM and TO attributes for accessing local files and smart card readers and for directing output data to a local printer have previously been unsupported in any Internet protocol. As a result, the server **12** may access non-standard I/O peripherals for any of the I/O devices used in the transaction or data system **10**. The ACTION="URL" is a part of standard HTTP and is well known.

The METHOD attributes may include the GET, POST, PAYMENT, or SQL methods. The GET and POST methods are currently supported in standard HTTP and are well known. The PAYMENT attribute is a directive to deliver data retrieved by an INPUT command to a private payment network for authorization and settlement and is not available in current Internet protocols. This directive is used by the client program to activate a conventional financial transaction application which communicates with the transaction system over a dedicated data line or PSTN in a known protocol such as VISA. Such an attribute is used where the more secure physical connection between remote site and transaction system and data encryption devices or the like are preferred. The SQL method preferably identifies a database language file which CGI **28** uses to correlate data in the HTML FORM to an insertion or query command contained in the file.

The preferred format for the INPUT tag which is used to identify input operations is also shown in FIG. **2**. The TYPE and NAME attributes are used to define a non-standard I/O device or local storage variable for the input of data. The TYPE field values "text," "password," "checkbox," "radio," "submit," and "reset" are previously known, as are the attributes NAME, VALUE, CHECKED, SIZE, and MAXLENGTH. To support the extended capability of the present invention, the TYPE attribute preferably includes attributes MSRT1 for reading track 1 of a magnetic swipe reader, MSRT2 for reading a magnetic swipe reader track 2, KEY for reading input from a terminal command keypad, PIN for reading a personal identification number pad, BCW for reading a bar code wand, MICR for reading a check magnetic code reader, AIM for reading a dollar amount via a key input mask, INT for reading an integer via a key input mask, database with the INSERT attribute or update data already existing in a database with the UPDATE attribute. The values for the INSERT attribute may be identified with the VALUES attribute, and the SET and WHERE attributes may be used to define and conditionally update values in the identified database. Preferably, the present invention implements two DELETE and CREATE attributes. The DELETE attribute deletes all items in an identified column of a database table which may satisfy a condition defined by a WHERE attribute. The CREATE attribute creates a database table having a primary key identified by the PRIMARY KEY attribute.

Preferably, the server program executes on a computer system having at least an Intel 80386 or better processor with at least 4 megabytes of RAM and at least 3 megabytes of hard disk space available. The computer system running the server may operate any known server platform operating system such as WINDOWS 3.1, WINDOWS 95, or WINDOWS NT, UNIX, AIX, and others. The non-standard I/O devices require a processor of a Z80A type or better, at least 32 K bytes of RAM, and at least 32 K bytes of ROM. The device includes a modem capable of at least 1200 bits-per-second (bps) but other modem speeds may be used for communication between client and server. Alternatively, the device may be coupled to a LAN which in turn is coupled to the Internet for communication with server **12**. A typical non-standard device which executes the client program is a VeriFone OMNI390, OMNI395, or VuFone terminal. OMNI390, OMNI395, and VuFone are trademarks of VeriFone, Inc., of Redwood City, Calif. Other exemplary devices include Phillips Screen phone, Hypercomm T7 terminal, and Apple Computer Newton MessagePad.

To build the preferred HTML files which CGI **28** preferably uses to implement the client program and database

application, the user preferably uses an off-line editor. The files generated by the editor are preferably comprised of an integrated statements formed from HTML statements and database statements for retrieving and writing data with the database. Exemplary files showing such integrated statements for performing transactions are depicted in FIGS. 14–23B. After such a file is generated, the editor parses the integrated statements into HTML statements and into database statements such as SQL commands. The HTML files required by the client program to support communication with a transaction or data processing center may be downloaded to a device or PC for execution. The files containing the database application statements used by the CGI interface to communicate data with the database application program preferably reside on server 12. Preferably, the database files used by the CGI interface include SQL commands for the application program interfaced to an ODBC compliant database.

The general format of the HTML commands in the HTML files used for communication with a client program and server are of the general format: TAG ATTRIBUTE. Preferably, the TAG field may be one of FORM, INPUT, SQL, or TEXTAREA. The ATTRIBUTE field value depends upon the TAG value. Preferably, the FORM tag may include the ACTION or METHOD attributes where the ACTION attributes include the FROM<file>, TO PRINTER, TO<file>, and TO SCR values noted above, as well as the standard HTML ACTION value of URL=<file>. The METHOD attributes include the PAYMENT and SQL attributes noted above, as well as the standard HTML METHOD values of GET and POST. Also in accordance with the principles of the present invention, the INPUT tag may include TYPE, NAME, VALUE, CHECKED, SIZE, and MAXLENGTH attributes. These attributes are previously supported for the INPUT tag in HTML, however, the present invention further includes TYPE values of MSRT1, MSRT2, KEY, PIN, BCW, MICR, AMT, INT, LOCAL, and AUTOSUB, as well as the standard HTML TYPE values of TEXT, PASSWORD, CHECKBOX, RADIO BUTTON, SUBMIT, and RESET. The present invention also supports NAME attributes of IP_ADDRESS, HOST_PHONE, TID, WORK_KEY, DATETIME, and DEPOSIT_ACCT to identify local storage areas as well as standard HTML NAME attribute <Field_NM> to identify a FORM variable.

The preferred high level processing of the client program is shown in FIG. 4. That processing includes an idle step (Block **100**) in which the program performs general housekeeping tasks such as maintaining internal time, scanning for input which may activate the device, or other known functions. Further processing is activated by some operator action at the device or PC which causes the device to either open a remote URL (Block **102**) or open a local URL (Block **104**). If a remote URL is required, the device transmits a message of the format discussed previously which is routed through the open network and delivered to a server **12** for a transaction or data processing system (Block **106**). The HTML file selected at the server 12 is identified by the remote URL in the initial communication between the device and server 12 and that URL is used to return the selected HTML file to the device for processing (Blocks **108**, **110**).

FIG. 4 also shows that an operator may initiate an open local URL function by typing in a command or by pushing a hot key which is associated with a local URL. The I/O device reads the HTML file identified by the URL from local memory (Block **112**) and passes the HTML file to the function for processing HTML files (Block **110**). After a file is processed (Block **110**), the client program determines whether the HTML file is to be stored (Block **114**). If it is not, the process returns to the idle processing (Block **100**). Otherwise, the process determines whether the HTML file is to be associated with a hot key (Block **116**) and, if it is, it stores the file and generates the link between a hot key and the stored file (Blocks **118**, **120**). If the HTML file is only to be stored, no association is made with a hot key and the file is simply stored in local memory (Block **20**). The client program then returns to idle processing (Block **100**).

The high-level processing for the HTML file (Block **110**, FIG. 4) is shown in further detail in FIG. 5. The process begins by scanning the HTML file for a TAG (Block **140**). If no TAG is found, the file is not in proper format for processing and processing returns to Block **114** discussed in FIG. 4 above. If a TAG is found (Block **142**), the process determines whether the TAG is a FORM TAG (Block **144**) or an INPUT TAG (Block **146**). If it is a FORM TAG, then the FORM TAG is processed and the program continues by looking for other TAGS to process (Block **140**). If the TAG is an INPUT TAG, the INPUT TAG is processed (Block **150**) and the program continues by looking for other TAGS to process (Block **140**). If the TAG is one of the standard HTML TAGS, the program implements the TAG in standard known ways (Block **152**) and then scans for other TAGs to process (Block **140**).

Processing the ATTRIBUTES used to implement a FORM TAG is shown in FIG. 6. That process continues by scanning the HTML file for an attribute (Block **160**). If an attribute is not found (Block **162**), the program returns to scan for other TAGS (Block **140**, FIG. 5). If an attribute is found, the program determines whether it is an ACTION attribute (Block **164**) or a METHOD attribute (Block **166**). Depending on the type of attribute, the appropriate function for processing the attribute is executed (Blocks **168** or **170**) and scanning for additional attributes continues (Block **160**). If the attribute is not an ACTION or METHOD attribute, there is an error in the file and processing returns to scan for other TAGs.

The processing for the ACTION attribute is shown in FIG. 7. There, the ACTION attribute is examined to determine whether it is a FROM<file> (Block **180**), TO PRINTER (Block **182**), TO<file> (Block **184**), TO SCR (Block **186**), FROM SCR (Block **188**) or a URL=<file> (Block **192**). The URL=<file> ACTION is a standard HTML action which is processed in a known way (Block **194**). The FROM <file> action is processed by reading data from a file associated with the I/O device or PC interfaced to the I/O device (Block **196**). The TO PRINTER action results in data in the FORM being sent to the printer (Block **198**) while the TO <file> action results in data in the FORM being written to a local file (Block **200**). The TO SCR action causes data to be written to the smart card via a smart card reader (Block **202**) and the FROM SCR reads data from a smart card through a smart card reader (Block **204**). After the appropriate action processing takes place, the HTML file is scanned for additional ACTION values to perform (Block **206**), and if one is found, the process continues. If no attribute is located (Block **208**), the process returns to scan for other attributes (Block **160**, FIG. 6).

The processing for the METHOD attributes for FORM tags are shown in FIG. 8. The process determines which type of METHOD is present in the FORM and then properly processes the attribute. For the GET and POST methods (Blocks **210**, **212**) the processing is the same as that performed in standard HTML (Blocks **226**, **228**). That is, for the GET method, the identified URL<file> is queried for data

EXHIBIT F
PAGE 41 OF 49

while the POST attribute causes data to be transferred to the URL<file>. The preferred METHOD attributes extending the HTML implementation of the present invention are SQL (Block 214), and PAYMENT (Block 224) attributes. The SQL attribute is preferably not expanded into a SQL command at the client, but rather is expanded by the CGI 28 at server 12 by correlating the data or variable field names in a returned form with the SQL commands stored at the server. This processing is done in a manner described in more detail below. The client program passes the SQL file identifier to the server 12 (Block 230). The processing of the PAYMENT command (Block 232) is discussed in more detail below. The HTML file is scanned for other METHODs (Block 242, 244), and, if one is found, the processing continues by identifying the METHOD (Blocks 210–224). Otherwise (Block 244), the process returns to scan the HTML file for other ACTION or METHOD attributes (Block 160, FIG. 6).

Processing for the INPUT tag is shown in FIG. 9. The process scans the HTML file following the INPUT tag for attributes (Block 250). If no attributes are found (Block 252), the process continues by scanning the HTML file for other tags to process (Block 140, FIG. 5). If an attribute is found and it is a TYPE attribute (Block 254), it is processed (Block 256), and if the attribute is a NAME attribute (Block 258), it is processed (Block 260). Both the TYPE and NAME processing is shown in more detail in FIGS. 10 and 11, respectively. If the attribute is neither a NAME or TYPE attribute, it is a standard attribute for an INPUT tag supported by standard HTML and is processed in a known manner (Block 262). Following processing of the INPUT attribute, the HTML file is scanned for other attributes to process (Block 250).

Processing for the TYPE attribute is shown in FIG. 10. The process first identifies the TYPE attribute for the INPUT tag and then performs the appropriate processing. The new TYPE attributes of the preferred embodiment of the present invention are MSRT1 (Block 270), MSRT2 (Block 272), KEY (Block 274), PIN (Block 276), BCW (Block 278), MICR (Block 280), AMT (Block 282), INT (Block 284), LOCAL (Block 286), and AUTOSUB (Block 288). If the TYPE attribute is not one of these, it is a standard HTML type attribute that is processed in a known manner (Block 310). Each of the new HTML TYPES supported by the present invention causes an I/O operation with a non-standard device. Specifically, these operations are the reading of Track 1 of the magnetic stripe reader (Block 290), the reading of the second track of the magnetic stripe reader (Block 292), the reading of a keypad (Block 294), the reading of an encrypted PIN through a PIN entry device (Block 296), the reading of a bar code through a bar code reader (Block 298), the reading of encoded data on a check through a magnetic check reader (Block 300), the reading of a dollar amount from a keypad through a key input mask (Block 302), the reading of a number from a keypad through a key input mask (Block 304), the reading of data from a local variable (Block 306) and the submission of the data read from one of these devices in a FORM returned to the server 12 (Block 308). The data mask for AMT constrains the dollar amount read to a predetermined number of characters with only two characters following the decimal point. The data mask for INT ensures the number is an integer value within a predetermined range. Processing continues by scanning the HTML file for other TYPE attributes (Block 312) and, if another TYPE attribute is found (Block 314), processing continues by determining the TYPE attribute and performing the appropriate processing. Otherwise, the process returns to scan the HTML file for other attributes (Block 250, FIG. 9).

The NAME attribute processing is performed in accordance with the process shown in FIG. 11. That process examines the NAME attribute to determine if the variable name identified by the attribute is IP_ADDRESS, HOST_PHONE, TID, WORK_KEY, DATETIME, or DEPOSIT_ACCT (Blocks 320, 322, 324, 326, 328, 330). If they are, the INPUT value resulting from one of the INPUTS in a FORM of the HTML file is stored in a local variable identified by the NAME attribute. Following storage (Block 332), the file is scanned for other NAME attributes (Block 328) and, if there are none (Block 332), processing continues by scanning for other attributes for the INPUT tag (Block 250, FIG. 9). If the NAME attribute is a standard HTML INPUT NAME, it is processed by known methods (Block 336). Processing then continues by scanning for other NAME attributes to process (Block 338, 340). Otherwise, the process returns to scan the HTML file for other attributes (Block 250, FIG. 9).

CGI 28 receives Internet protocol statements in a file transmitted from a client program and provides data from those statements to the application(s) implementing system 40 and receives the output of system 40 and provides them to the client program in a file. CGI 28 may be implemented by a program developed by a user using a manual development method as shown in FIG. 24A. That method requires a user to generate a system definition from which a file statement definition for the client and application are developed to implement the transactional or data system. Using the file statement definitions, the user generates the files for the client and database programs which are interpreted by the respective programs to implement transactions or data processing. This process requires the user to not only have knowledge regarding the transaction or data process but specific details of the interaction between the client and database. The user is further required to resolve and correlate all data identifiers in the statements for the client and database environments.

Preferably, CGI 28 is developed with an editor that only requires the user to define the system with statements which are an integration of the protocol statements and the database language. The process implemented by this editor is shown in FIG. 24B. Examples of such integrated statements for files which implement a specific transaction are shown in FIGS. 14 to 23B. The editor verifies the syntax of the integrated statements and correlates the data variables of the protocol statements with the data fields of the database. Following the generation of the integrated statements, the editor segregates the protocol statements from the database language statements. The protocol statements are stored in files which are identified as being for a particular transaction or data process and the database statements are stored in files which are identified as being for a particular transaction or data process on an identified database table. The editor places a database file identifier in the protocol statements which contained embedded database statements. The database file identifiers are used by CGI 28 to select the file for the appropriate transaction so CGI 28 may correlate data variables in the protocol statements with data fields in the database files. The files containing statements to be interpreted by the client program are then downloaded to the appropriate terminals, and the database files containing database language statements are stored on the system executing the CGI 28.

Alternatively, the editor of the present invention may parse integrated statements which are segregated into source code statements for first and second processors, such an editor further includes a compiler to generate executable

EXHIBIT F
PAGE 42 OF 49

code for each processor and, if the processors execute differing source code, a compiler for each source code language. The executable code may then be downloaded to the respective processors for execution.

More specifically, the editor preferably places the database statements for one of the transactions of the preferred embodiment in a file identified by the database name following SQL in FIG. 12. The attributes and tags forming the HTML statements for one of the transactions of the preferred embodiment are placed in a file generally denoted as <html_file>.HTM. The name <html_file> is a name which identifies one of the transactions. Where SQL statements are in the fields of the integrated statements shown in FIGS. 14 to 23B, the string "<html_file>.SQL" is substituted as the database name in the statements of the <html_file>.HTM file. When the CGI executable file is initiated and parses the returning forms, the returned data is placed in the corresponding "<html_file>.SQL" file which is passed to the application program as a command line argument. In this manner, an abbreviated form for the SQL commands may be communicated over the open network between the client and CGI and the CGI may be able to expand those abbreviated SQL commands into the appropriate SQL commands which the application program requires to manipulate the ODBC database.

To effectuate a transaction, for example, an operation at a terminal with non-standard I/O devices may activate a terminal file with a hot key or other action. In processing the activated file, the client program may acquire data which is stored in a local variable or accessible through a non-standard I/O device. This data may then be stored in a FORM and submitted to a server file at a processing system address. The server file activates CGI **28** which retrieves data from the FORM and incorporates it into database statements in the database file for the appropriate transaction and database. If the database statement is a query, the requested data is returned to the CGI in the database file and the CGI places it in the corresponding FORM variables so the server may return the data to the terminal. If the database statement provides data to a database to obtain an authorization, for example, the action performed by the database application in response to the data is placed in the corresponding FORM and returned to the terminal. In this way, data is exchanged between the terminal and the database application. This exchange is supported by CGI **28** even though the server/client communication is performed in an open system protocol, such as HTTP, and the database application is performed in another language, such as SQL. CGI **28** is able to convert and exchange the data between the client and database without the user having to specifically design and implement a conversion program.

The communication paths available for a device implementing the present invention are shown in FIG. 13A. As shown there, an I/O device **420** is coupled through the WorldWide Web open network **426** to an Internet Web server **12**. This connection may be implemented with the preferred extended capability HTML described above. Although HTML files may be encrypted to enhance the security of the document as it is communicated across the Internet, the operator of the system may choose to utilize a more secure physical connection between the device **420** and the Web server **12**. To obtain this alternative connection, the PAYMENT command for the METHOD attribute is preferably used. One form of the PAYMENT command is for a merchant's terminal and the other is for a consumer's terminal. In either terminal, the client program which supports the extended capability HTML operates independently but co-resident in memory with a certified bank card authorization and capture application, which may be provided by a financial institution or a bank card processor.

For the form of the command shown in FIG. 13B, the client program in the merchant terminal suspends its execution and passes the terminal identifier, stored locally, which identifies the merchant's account and the consumer account information read via a magnetic stripe reader or the like, to the bank card application. The bank card application communicates this information via a PSTN **424** or the like to a transaction processor **422**. The processor **422** authorizes or denies the transaction and, if authorized, a printer at the merchant terminal prints a purchase agreement which the consumer may execute to complete the transaction.

In response to a HTML file having a FORM with an ACTION attribute equal to an executable file name for a bank card application program or the like, a METHOD attribute with a field value of PAYMENT, and an INPUT tag with a TYPE attribute of LOCAL_NAME which identifies a deposit only account supplied by a merchant (as shown in FIG. 13C), the client program is suspended and control is transferred to the bank processing application. The bank processing application then uses a modem or ISDN D channel using T3 POS protocol or the like to connect to a secure packet network **424** to connect in a virtual point-to-point manner with a payment processor through a PSTN network or the like. This physical connection provides an additional security element to the encrypted data for the transaction of account information, PIN numbers encrypted by PIN pads provided at the consumer site, and other sensitive information. The bank processor **422** may submit remittance data to the merchant, via the Web or otherwise. After receiving the remittance data, the merchant may ship the product to the consumer. Thus, in this manner, the I/O device may communicate with a plurality of Web servers to "shop" for a best price, delivery date, or other relevant information for selecting a preferred transaction, and then execute the PAYMENT method to utilize a more secure physical communication connection and data security devices to consummate the financial elements of the transaction with less risk and costs for the merchant, consumer, and bank processor.

The preferred integrated HTML/SQL statements which support a card initiated payment authorization and capture transaction are shown in FIG. 14. A first file **500** includes statements which identify the URL database from which the non-standard I/O device seeks authorization for a transaction. The prompts to the operator to enter the account number and amount of the transaction are supported by the INPUT tags which read the second track of the magnetic stripe reader to accept a number of up to 40 characters and assign that information from that track to a variable, and to input the up to 8 characters from the keyboard or the like into a variable called AMOUNT. The INPUT tag with the TYPE attribute of AUTOSUBMIT returns the form to the server for processing in accordance with the method defined in the returned form. As shown in FIG. 14, that METHOD statement causes CGI **28** to incorporate returned data into SQL commands which query the database as to whether the subfield of the track 2 data representing the account number is present in the authorization table of the database. If the data is not present, then a new record is inserted into a table labeled "log_table". The new record consists of the account number and the amount returned in the FORM. Based upon the results of this processing, the application program supplies the data fields to the FORM which will be returned to the client program for printing the transaction record. That

EXHIBIT F
PAGE 43 OF 49

file **510** is shown in FIG. **14**. The ACTION attribute TO PRINTER and the POST METHOD causes the data in the next eight lines to be directed to the printer coupled to the non-standard I/O device for printing the transaction form. The customer may then execute the printed form to complete the transaction. If the transaction is declined or an error is otherwise encountered, the file **520** is used to return a denial to the client program.

In a similar manner, the preferred integrated statements for a bar code order input with card-initiated payment authorization is shown in FIG. **15**. The file **550**, supported by the present invention which implements the transaction request, is again directed to the proper database by the ACTION attribute. The necessary customer information such as name and address may be input through a standard keyboard. The HTML command in the present invention also permits the form to receive the bar code, unit price, and credit card information in a manner similar to that discussed above for the magnetic card reader. Once this information is returned to the server and CGI interface, it is processed by the application program in accordance with the METHOD identified in the returned form. The method of HTML file **550** also creates a database order_table having the information shown in the method. Again, if the transaction is approved, the data for the order and customer acceptance of the order is provided in HTML file **555**, which is directed by the ACTION attribute to the printer at the non-standard I/O device. If the account number is not in the authorization database, the authorization declined or error response is provided in correspondence with the statements in file **560**.

In a similar manner, FIGS. **16–22** show the integrated statements for a transaction request, authorization response, or authorization declined response files for key input order with secure payment transaction (FIG. **16**), a smart card-debit (Type 1) transaction (FIG. **17**A), a smart card debit (Type 2) transaction (FIG. **17**B), a debit card transaction (FIG. **18**), a check verification transaction (FIG. **19**), a customer frequency transaction (FIG. **20**), an item search transaction for which there is no denial (FIG. **21**), retail store end of day reporting (FIG. **22**) and a store reporting an e-mail transaction (FIG. **23**).

While the present invention has been illustrated by the description of a preferred and alternative embodiments and processes, and while the preferred and alternative embodiments and processes have been described in considerable detail, it is not the intention of the applicant to restrict or in any way limit the scope of the appended claims to such detail. Additional advantages and modifications will readily appear to those skilled in the art. For example, rather than expanding HTTP to support non-standard I/O devices, the FTP, POP, SMTP, TELNET or other protocols may be expanded in like manner to couple non-standard I/O devices to the Internet. Similarly, the preferred implementation of the present invention supports a variety of non-standard I/O devices and I/O operations. An Internet protocol may be constructed in accordance with the principles of the present invention to support only selected I/O devices or operations disclosed in the present application. The invention in its broadest aspects is therefore not limited to the specific details, preferred embodiment, and illustrative examples shown and described. Accordingly, departures may be made from such details without departing from the spirit or scope of applicant's general inventive concept.

What is claimed is:

1. A system for supporting communication between processing systems and non-standard I/O devices over an open network comprising:

a server that processes extended open network statements, said server being communicatively coupled to an open network;

a first non-standard I/O device communicatively coupled to said open network; and

a client program executing within said first non-standard I/O device for processing extended open network statements so that said first non-standard I/O device may communicate with said server.

2. A system for supporting communication between processing systems and non-standard I/O devices over an open network comprising:

a server that processes extended open network statements, said server being communicatively coupled to an open network;

a first non-standard I/O device communicatively coupled to said open network;

a client program for processing extended open network statements so that said first non-standard I/O device may communicate with said server; and

a computer communicatively coupled to said open network; and said first non-standard I/O device being coupled to said open network through said computer.

3. The system of claim **2** wherein said client executes in said computer so said non-standard I/O device may communicate with said server over said open network.

4. The system of claim **2** wherein said client executes in said non-standard I/O device so said non-standard I/O device may communicate with said server over said open network through said computer.

5. The system of claim **2** wherein said non-standard I/O device is communicatively coupled to said computer through a non-standard I/O port.

6. The system of claim **5** wherein said non-standard I/O device is communicatively coupled to said computer through one of a COMM1 and a COMM2 port.

7. The system of claim **2** wherein said first non-standard I/O device is one of a screen phone, personal digital assistant, a smart card reader, a personal identification number (PIN) pad, a magnetic swipe reader, a bar code wand, a check magnetic code reader, and a printer.

8. A system for supporting communication between processing systems and non-standard I/O devices over an open network comprising:

a server that processes extended open network statements, said server being communicatively coupled to an open network;

a first non-standard I/O device communicatively coupled to said open network;

a client program for processing extended open network statements so that said first non-standard I/O device may communicate with said server;

a second non-standard I/O device communicatively coupled to said open network; and

said first non-standard I/O device being communicatively coupled to said open network through said second non-standard I/O device.

9. The system of claim **8** wherein said first non-standard I/O device is communicatively coupled to said second non-standard I/O device through a non-standard I/O port.

10. The system of claim **9** wherein said first non-standard I/O device is communicatively coupled to said second non-standard I/O device through a RS-232C port.

11. The system of claim **8** wherein said second non-standard I/O device is one of a screen phone, personal digital

EXHIBIT F
PAGE 114 OF 149

assistant, a smart card reader, a personal identification number (PIN) pad, a magnetic swipe reader, and a printer.

12. A client that provides communication between a processing system and a non-standard I/O device via an open network comprising:

means for processing open network protocol statements;

means for processing extended open network protocol statements to support communication between a non-standard input/output (I/O) device communicatively coupled to an open network and a server communicatively coupled to said open network, said means for processing extended open network protocol statements executing in a computer communicatively coupled to said open network; and

a non-standard I/O device communicatively coupled to said computer.

13. A client that provides communication between a processing system and a non-standard I/O device via an open network comprising:

means for processing open network protocol statements; and

means for processing extended open network protocol statements to support communication between a non-standard input/output (I/O) device communicatively coupled to an open network and a server communicatively coupled to said open network, said means for processing extended open network protocol statements executing in a non-standard I/O device communicatively coupled to said open network.

14. A client that provides communication between a processing system and a non-standard I/O device via an open network comprising:

means for processing open network protocol statements;

means for processing extended open network protocol statements to support communication between a non-standard input/output (I/O) device communicatively coupled to an open network through a computer and a server communicatively coupled to said open network; and

said means for processing extended open network protocol statements executes in said non-standard I/O device.

15. The client of claim 12 wherein said non-standard I/O device is communicatively coupled to said computer through a non-standard I/O port.

16. The client of claim 15 wherein said non-standard I/O device is communicatively coupled to said computer through one of a COMM1 and a COMM2 port.

17. The client of claim 12 wherein said computer is a consumer's terminal.

18. The client of claim 12 wherein said computer is a merchant's terminal.

19. The client of claim 12 wherein said means for processing extended open network protocol statements is co-resident in a memory of said computer with a bank card application program.

20. The client of claim 19 wherein said means for processing extended open network protocol statements processes an extended open network command to transfer communication between said bank card application program and a transaction processor to a communication network other than said open network over which said means for processing extended network protocol statements and said server are communicating.

21. The client of claim 20 wherein said means for processing extended open network protocol statements in response to a command within an extended open network protocol statement provides said bank card application program with consumer account information for transaction processing.

22. The client of claim 21 wherein said means for processing extended open network protocol statements obtains said consumer account information from a non-standard I/O device.

23. The client of claim 12 wherein said non-standard I/O device is one of a screen phone, personal digital assistant, a smart card reader, a personal identification number (PIN) pad, a magnetic swipe reader, a bar code wand, a check magnetic code reader, and a printer.

24. The client of claim 22 wherein said non-standard I/O device is one of a screen phone, personal digital assistant, a smart card reader, a personal identification number (PIN) pad, a magnetic swipe reader, a bar code wand, a check magnetic code reader, and a printer.

25. The client of claim 17 wherein said means for processing extended open network protocol statements is co-resident in a memory of said consumer's terminal with a bank card application program.

26. The client of claim 25 wherein said means for processing extended open network protocol statements provides said bank card application program with consumer account information for transaction processing.

27. The client of claim 26 herein said means for processing extended open network protocol statements obtains said consumer account information from a non-standard I/O device.

28. The client of claim 26 wherein said means for processing extended open network protocol statements obtains a portion of said consumer account information from a standard I/O device; and

said means for processing extended protocol statements obtains a second portion of said consumer account information from said non-standard I/O device.

29. The client of claim 27 wherein said non-standard I/O device is one of a screen phone, personal digital assistant, a smart card reader, a personal identification number (PIN) pad, a magnetic swipe reader, a bar code wand, a check magnetic code reader, and a printer.

30. The client of claim 28 wherein said non-standard I/O device is one of a screen phone, personal digital assistant, a smart card reader, a personal identification number (PIN) pad, a magnetic swipe reader, a bar code wand, a check magnetic code reader, and a printer.

31. The client of claim 28 wherein said consumer's terminal is a personal computer (PC).

32. The client of claim 18 wherein said means for processing extended open network protocol statements is co-resident in a memory of said merchant's terminal with a bank card application program.

33. The client of claim 32 wherein said means for processing extended open network protocol statements provides said bank card application program with consumer account information for transaction processing.

34. The client of claim 33 wherein said means for processing extended open network protocol statements obtains said consumer account information from a non-standard I/O device.

35. The client of claim 33 wherein said means for processing extended open network protocol statements obtains a portion of said consumer account information from a standard I/O device; and

said means for processing extended protocol statements obtains a second portion of said consumer account information from said non-standard I/O device.

EXHIBIT F
PAGE 45 OF 49

36. The client of claim 33 wherein said merchant's terminal is a credit card terminal.

37. A server that provides communication between a processing system and a non-standard I/O device via an open network comprising:

means for processing open network protocol statements within forms submitted by a client communicatively coupled to an open network to which said means for processing open network protocol statements within forms is communicatively coupled; and

means for processing extended open network protocol statements within said forms submitted by said client, said means for processing extended open network protocol statements provides data from said extended open network protocol statements to application programs communicatively coupled to said server.

38. The server of claim 37 wherein said means for processing extended open network protocol statements provides said data to said application programs in database commands.

39. The server of claim 37 wherein said means for processing extended open network protocol statements provides data from said application programs to said client in extended open network protocol statements.

40. The server of claim 38 wherein said means for processing extended open network protocol statements returns records to said client containing a response to said database commands.

41. A client program for processing extended open network protocol statements so a non-standard I/O device may communicate with a processing system over an open network comprising:

means for receiving extended open network protocol statements over an open network;

means for processing said received extended open network protocol statements to control operations associated with a non-standard I/O device; means for activating a function associated with a hot key of said non-standard I/O device; and

means for processing extended open network protocol statements contained in a local file associated with said activated function.

42. The client program of claim 41 further comprising:

means for generating a link between a hot key and a file.

43. The client program of claim 42 wherein said file containing extended open network protocol statements is an HTML form.

44. A client program for processing extended open network protocol statements so a non-standard I/O device may communicate with a processing system over an open network comprising:

means for receiving extended open network protocol statements over an open network;

means for processing said received extended open network Protocol statements to control operations associated with a non-standard I/O device; and

means for processing keypad input data received from a keypad of said non-standard I/O device.

45. The client program of claim 44 wherein said means for processing keypad input data includes a data mask for interpreting data received through said keypad.

46. The client program of claim 45 wherein said data mask identifies the received data as a monetary amount having a predetermined number of characters with only two of said characters following a decimal point.

47. The client program of claim 45 wherein said data mask identifies the received data as an integer value within a predetermined range.

48. A client program for processing extended open network protocol statements so a non-standard I/O device may communicate with a Processing system over an open network comprising:

means for receiving extended open network protocol statements over an open network;

means for processing said received extended open network protocol statements to control operations associated with a non-standard I/O device; and

means for processing extended open network protocol printer statements to direct output data to a printer communicatively coupled to said non-standard I/O device.

49. The client program of claim 48 wherein said means for processing printer statements directs a purchase agreement to said printer for a consumer transaction.

50. A client program for processing extended open network Protocol statements so a non-standard I/O device may communicate with a processing system over an open network comprising:

means for receiving extended open network protocol statements over an open network;

means for processing said received extended open network protocol statements to control operations associated with a non-standard I/O device: and

means for processing local variable commands in said extended open network protocol statements so that data is obtained from a local variable.

51. A client program for processing extended open network protocol statements so a non-standard I/O device may communicate with a processing system over an open network comprising:

means for receiving extended open network protocol statements over an open network;

means for processing said received extended open network protocol statements to control operations associated with a non-standard I/O device; and

means for returning a form to a server communicatively coupled to said open network in response to said means for processing extended open network protocol statements processing an extended open network protocol statement to automatically submit the form.

52. A method for providing communication between a processing system and a non-standard I/O device via an open network comprising:

receiving extended open network protocol statements;

processing extended open network protocol statements to support communication between a non-standard input/output (I/O) device communicatively coupled to an open network and a server communicatively coupled to said open network;

performing said extended open network protocol statement processing in a computer communicatively coupled to said open network; and

coupling a non-standard I/O device to said computer.

53. A method for providing communication between a processing system and a non-standard I/O device via an open network comprising:

receiving extended open network protocol statements;

processing extended open network protocol statements to support communication between a non-standard input/output (I/O) device communicatively coupled to an open network and a server communicatively coupled to said open network;and

EXHIBIT F
PAGE 46 OF 49

performing said extended open network protocol statement processing in a non-standard I/O device communicatively coupled to said open network.

54. A method for providing communication between a processing system and a non-standard I/O device via an open network comprising:

receiving extended open network protocol statements;

processing extended open network protocol statements to support communication between a non-standard input/output (I/O) device communicatively coupled to an open network and a server communicatively coupled to said open network;

coupling a non-standard I/O device to a computer; and

performing said extended open network protocol statement processing in said non-standard I/O device communicatively coupled to said computer.

55. The method of claim 52 wherein said coupling of said non-standard I/O device to said computer is through a non-standard I/O port.

56. The method of claim 55 wherein said coupling of said non-standard I/O device to said computer is through one of a COMM1 and a COMM2 port.

57. The method of claim 52 further comprising:

communicating with a bank card application program resident in a memory of said computer.

58. The method of claim 57 further comprising:

activating said bank card application program to communicate with a transaction processor over a communication network other than said open network in response to a command within an extended open network protocol statement.

59. The method of claim 58 further comprising:

providing said bank card application program with consumer account information for transaction processing in response to a command within an extended open network protocol statement.

60. The method of claim 59 further comprising:

obtaining said consumer account information from said non-standard I/O device.

61. The method of claim 56 wherein said consumer account information is obtained from one of a screen phone, personal digital assistant, a smart card reader, a personal identification number (PIN) pad, a magnetic swipe reader, a bar code wand, a check magnetic code reader, and a printer.

62. The method of claim 52 further comprising:

communicating with a bank card application program resident in a memory of said computer.

63. The method of claim 62 further comprising:

providing said bank card application program with consumer account information for transaction processing in response to a command within an extended open network protocol statement.

64. The method of claim 63 wherein said consumer account information is obtained from said non-standard I/O device.

65. The method of claim 57 further comprising:

providing said bank card application program a portion of consumer account information from a standard I/O device; and

providing a second portion of said consumer account information from said non-standard I/O device.

66. The method of claim 64 wherein said consumer account information is obtained from one of a screen phone, personal digital assistant, a smart card reader, a personal identification number (PIN) pad, a magnetic swipe reader, a bar code wand, a check magnetic code reader, and a printer.

67. The method of claim 65 wherein said second portion of said consumer account information is obtained from one of a screen phone, personal digital assistant, a smart card reader, a personal identification number (PIN) pad, a magnetic swipe reader, a bar code wand, a check magnetic code reader, and a printer.

68. A method for providing communication between a processing system and a non-standard I/O device via an open network comprising:

receiving extended open network protocol statements;

processing extended open network protocol statements to support communication between a non-standard input/output (I/O) device communicatively coupled to an open network and a server communicatively coupled to said open network, said extended open network protocol statements being performed in a consumer's terminal; and communicating with a bank card application program resident in a memory of said consumer's terminal with a bank card application program.

69. The method of claim 68 further comprising:

providing said bank card application program with consumer account information for transaction processing in response to a command within an extended open network protocol statement.

70. The method of claim 69 wherein said consumer account information is obtained from said non-standard I/O device.

71. The method of claim 68 further comprising:

providing said bank card application program a portion of consumer account information from a standard I/O device; and

providing a second portion of said consumer account information from said non-standard I/O device.

72. The method of claim 70 wherein said consumer account information is obtained from one of a screen phone, personal digital assistant, a smart card reader, a personal identification number (PIN) pad, a magnetic swipe reader, a bar code wand, a check magnetic code reader, and a printer.

73. The method of claim 71 wherein said second portion of said consumer account information is obtained from one of a screen phone, personal digital assistant, a smart card reader, a personal identification number (PIN) pad, a magnetic swipe reader, a bar code wand, a check magnetic code reader, and a printer.

74. A method for providing communication between a processing system and a non-standard I/O device via an open network comprising:

processing open network protocol statements within forms submitted by a client communicatively coupled to an open network;

processing extended open network protocol statements within said forms submitted by said client; and

providing data from said processed extended open network protocol statements to application programs communicatively coupled to a server in database commands.

75. The method of claim 74 further comprising:

returning records to said client containing a response to said database commands.

76. A method for supporting communication between processing systems and non-standard I/O devices over an open network comprising:

processing extended open network statements at a server communicatively coupled to an open network;

communicatively coupling a first non-standard I/O device to said open network;

EXHIBIT F
PAGE 47 OF 49

processing extended open network statements with a client program so that said first non-standard I/O device may communicate with said server; and

executing said client program within said first non-standard I/O device so that said first non-standard I/O device may communicate with said server.

77. A method for supporting communication between processing systems and non-standard I/O devices over an open network comprising:

processing extended open network statements at a server communicatively coupled to an open network;

communicatively coupling a first non-standard I/O device to said open network;

processing extended open network statements with a client program so that said first non-standard I/O device may communicate with said server;

coupling a computer to said open network; and

coupling said first non-standard I/O device to said open network through said computer.

78. The method of claim 77 further comprising:

executing said client program in said computer so said first non-standard I/O device may communicate with said server over said open network.

79. The method of claim 77 further comprising:

executing said client program in said first non-standard I/O device so said non-standard I/O device may communicate with said server over said open network through said computer.

80. The method of claim 79 further comprising:

coupling said first non-standard I/O device to said computer through a non-standard I/O port.

81. The method of claim 80 wherein said coupling of said first non-standard I/O device to said computer is through one of a COMM1 and a COMM2 port.

82. A method for supporting communication between processing systems and non-standard I/O devices over an open network comprising:

processing extended open network statements at a server communicatively coupled to an open network;

communicatively coupling a second non-standard I/O device to a first non-standard I/O device and to said open network; and

processing extended open network statements with a client program so that said first non-standard I/O device may communicate with said server.

83. The method of claim 82 further comprising:

coupling said first non-standard I/O device to said second non-standard I/O device through a non-standard I/O port.

84. The method of claim 83 further comprising:

coupling said first non-standard I/O device to said second non-standard I/O device through a RS-232C port.

85. The method of claim 83 wherein said coupling is of one of a screen phone, personal digital assistant, a smart card reader, a personal identification number (PIN) pad, a magnetic swipe reader, and a printer to one of screen phone, personal digital assistant, a smart card reader, a personal identification number (PIN) pad, a magnetic swipe reader, a bar code wand, a check magnetic code reader, and a printer.

86. A method for processing extended open network protocol statements so a non-standard I/O device may communicate with a processing system over an open network comprising:

receiving extended open network protocol statements over an open network;

processing said received extended open network protocol statements to control an operation associated with a non-standard I/O device;

activating a function associated with a hot key of said non-standard I/O device; and

implementing said activated function by processing extended open network protocol statements contained in a local file associated with said activated function.

87. The method of claim 86 further comprising:

generating a link between a hot key and a file containing extended open network protocol statements; and

storing said file.

88. The method of claim 87 further comprising storing said extended open network protocol statements contained in said file in an HTML form.

89. A method for processing extended open network protocol statements so a non-standard I/O device may communicate with a processing system over an open network comprising:

receiving extended open network protocol statements over an open network;

processing said received extended open network protocol statements to control an oDeration associated with a non-standard I/O device; and

processing keypad input data received from a keypad of said non-standard I/O device.

90. The method of claim 89 further comprising:

interpreting data received through said keypad with a data mask.

91. The method of claim 90 further comprising:

identifying the received data with the data mask as a monetary amount having a predetermined number of characters with only two of said characters following a decimal point.

92. The method of claim 90 further comprising:

identifying the received data with the data mask as an integer value within a predetermined range.

93. A method for processing extended open network protocol statements so a non-standard I/O device may communicate with a processing system over an open network comprising:

receiving extended open network protocol statements over an open network;

processing said received extended open network protocol statements to control an operation associated with a non-standard I/O device; and

processing extended open network protocol printer statements to direct output data to a printer communicatively coupled to said non-standard I/O device.

94. The method of claim 93 further comprising:

directing a purchase agreement to said printer for a consumer transaction.

95. A method for processing extended open network protocol statements so a non-standard I/O device may communicate with a processing system over an open network comprising:

receiving extended open network protocol statements over an open network;

processing said received extended open network protocol statements to control an operation associated with a non-standard I/O device; and

processing local variable commands in said extended open network protocol statements so that data is obtained from a local variable.

EXHIBIT F
PAGE 48 OF 49

96. A method for processing extended open network protocol statements so a non-standard I/O device may communicate with a processing system over an open network comprising:

receiving extended open network protocol statements over an open network;

processing said received extended open network protocol statements to control an operation associated with a non-standard I/O device; and

returning a form to a server communicatively coupled to said open network in response to said processing of an extended open network protocol command to automatically submit the form.

97. A system for telephone communication over an open network comprising:

a telephone: and

a client program communicatively coupled to the telephone, the client program for communicating data with the telephone and for processing extended Internet protocol statements to support communication with the telephone over an open network, the client program executing within the telephone.

98. A system for telephone communication over an open network comprising:

a telephone; and

a client program communicatively coupled to the telephone, the client program for communicating data with the telephone and for processing extended Internet protocol statements to support communication with the telephone over an open network, the client program executing within a computer communicatively coupled to the telephone.

99. A system for telephone communication over an open network comprising:

a telephone; and

a client program communicatively coupled to the telephone, the client program for communicating data with the telephone and for processing extended Internet protocol statements to support communication with the telephone over an open network, the extended Internet protocol statements being extended Hypertext Transfer Protocol (HTTP) statements.

100. A system for telephone communication over an open network comprising:

a telephone; and

a client program communicatively coupled to the telephone, the client program for communicating data with the telephone and for processing extended Internet protocol statements to support communication with the telephone over an open network, the extended Internet protocol statements being extended Hypertext Markup Language (HTML) command statements.

101. A system for telephone communication over an open network comprising:

a telephone; and

a client program communicatively coupled to the telephone, the client program for communicating data with the telephone and for processing extended Internet protocol statements to support communication with the telephone over an open network, the extended Internet protocol statements being extended File Transfer Protocol (FTP) statements.

102. A system for telephone communication over an open network comprising:

a telephone; and a client program communicatively coupled to the telephone, the client program for communicating data with the telephone and for processing extended Internet protocol statements to support communication with the telephone over an open network, the extended Internet protocol statements being extended Post Office Protocol (POP) statements.

103. A system for telephone communication over an open network comprising:

a telephone; and

a client program communicatively coupled to the telephone, the client program for communicating data with the telephone and for processing extended Internet protocol statements to support communication with the telephone over an open network, the extended Internet protocol statements being extended Simple Mail Transfer Protocol (SMTP) statements.

104. A system for telephone communication over an open network comprising:

a telephone; and

a client program communicatively coupled to the telephone, the client program for communicating data with the telephone and for processing extended Internet protocol statements to support communication with the telephone over an open network, the extended Internet protocol statements being extended Network Virtual Terminal Protocol (TELNET) statements.

* * * * *

EXHIBIT F
PAGE 49 OF 49

EXHIBIT / ATTACHMENT

G

(To be scanned in place of tab)

TKHR

THOMAS, KAYDEN, HORSTEMEYER & RISLEY, L.L.P.

ATTORNEYS AT LAW

100 Galleria Parkway, NW
Suite 1750
Atlanta, GA 30339-5948
Telephone: (770) 933-9500
Facsimile: (770) 951-0933
www.tkhr.com

Stephen R. Risley *steve.risley@tkhr.com*

August 1, 2002

VIA FEDERAL EXPRESS

Mr. Dong-Jin Oh
President & Chief Executive Officer
Samsung Electronics America, Inc.
105 Challenger Road
Ridgefield Park, NJ 07660

Re: Datascape, Inc.'s U.S. Patent Nos. 5,742,845; 5,905,908; and 6,366,967
Our File No.: 90405-7050

Dear Mr. Oh:

We represent Datascape, Inc. with respect to the licensing and enforcement of its United States patents. In particular, Datascape is the owner of:

(1) U.S. Patent No. 5,742,845 entitled "System For Extending Present Open Network Communication Protocols To Communicate With Non-Standard I/O Devices Directly Coupled To An Open Network;"

(2) U.S. Patent No. 5,905,908 entitled "Open Network System For I/O Operations With Non-Standard I/O Devices Utilizing Extended Protocol Including Device Identifier And Identifier For Operation To Be Performed With Device;" and

(3) U.S. Patent No. 6,366,967 entitled "Open Network System For I/O Operation Including A Common Gateway Interface And An Extended Open Network Protocol With Non-Standard I/O Devices Utilizing Device And Identifier For Operation To Be Performed With Device."

Copies of these three Datascape patents are enclosed for your convenience.

EXHIBIT G
PAGE 1 OF 3

In addition to these three patents, Datascape has other pending patent applications that claim priority back to the original filing date of June 22, 1995.

The Patented Technology

Generally speaking, the Datascape patents cover apparatuses, methods, and associated technologies for connecting non-standard input/output ("I/O") devices to an open network, such as the Internet, using an extended open network protocol. The requisite connection may be either directly to the Internet, or through an intermediary such as a personal computer ("PC"). Examples of such non-standard I/O devices that may be of particular interest to your company include Internet-enabled wireless telephones and Internet-enabled personal digital assistants ("PDAs").

We believe that certain of Datascape's patent claims read directly on products and systems that use the Wireless Application Protocol ("WAP"), such as Internet-enabled wireless telephones and Internet-enabled PDAs, with or without a Subscriber Identity Module ("SIM") card.

Recent Developments In The Licensing Program

As indicated above, Datascape has embarked on a program to license these patents to users of the patented technologies.

During the summer of 2001, the Datascape patents were the subject matter of two now dismissed lawsuits involving the American Express Company and its On-Line Wallet merchants. These lawsuits were dismissed after American Express non-exclusively licensed the Datascape patent portfolio for such products as the AmEx "Blue" smart card. A smart card is one type of Internet-enabled non-standard I/O device covered by Datascape's patents.

In addition, Datascape is pleased to inform you that a patent infringement lawsuit against Rand McNally & Company has been dismissed as a result of Rand McNally's decision to non-exclusively license the Datascape patent portfolio for its TripLink product line, which are PDA-type devices. The TripLink products are another type of Internet-enabled non-standard I/O device covered by the Datascape patents.

Further, Datascape is currently in active licensing negotiations with a number of players in the wireless and handheld arena.

The Current Licensing Offer

For a limited period of time, your company may obtain a field of use license covering wireless telephone and PDA products under the subject patents, as well as any follow-on patents that may issue in the future, at a discounted rate. The benefit to your firm is an attractive license cost, and the benefit to Datascape is another licensee in one of your technical areas of endeavor.

Any such license negotiated between your company and Datascape will cover any back damages, plus any future usage of the '845 patent, the '908 patent, and the '967 patent, as well as any subsequent patents that may issue in this same chain of patents for the wireless telephone and PDA field of use. Please note, however, that the license offer contained in this letter specifically excludes stand alone "smart card" technologies, including semiconductor "chips" for use in smart cards.

I have been authorized by Datascape to send you a draft agreement covering such a field of use patent license whenever you deem that productive and appropriate.

Alternatively, if you have any questions about either the patented technology or the current licensing offer, you are very welcome to give me a call at your convenience.

I appreciate the opportunity to share this opportunity with you and will be pleased to assist you with consummating a license with Datascape

Sincerely,

THOMAS, KAYDEN, HORSTEMEYER & RISLEY, LLP

Stephen R. Risley

/smg
Enclosures: U.S. Patent Nos. 5,742,845; 5,905,908; and 6,366,967
cc: Datascape, Inc. (w/out enclosures)

EXHIBIT G
PAGE 3 OF 3

ORIGINAL

AO 440 (rev. 10/93) Summons in a Civil Action

United States District Court

NORTHERN DISTRICT OF GEORGIA

IN RE: Datascape, Inc.

V.

Samsung Electronics America, Inc., et al

SUMMONS IN A CIVIL CASE

CASE NUMBER: 1:04-CV-1642

TO: (Name and address of defendant)

Staples, Inc.
c/o Corporation Process Co.
180 Cherokee St., N.E.
Marietta, GA 30060

YOU ARE HEREBY SUMMONED and required to serve upon PLAINTIFF'S ATTORNEY (name and address)

PLAINTIFF'S ATTORNEY (name and address)

Stephen R. Risley
J.Scott Culpepper
N. Andrew Crain
Thomas, Kayden, Horstemeyer & Risley, L.L.P.
100 Galleria Parkway, N.W
Suite 1750
Atlanta, Georgia 30339-5948
(770) 933-9500

an answer to the complaint which is herewith served upon you, within **20** days after service of this summons upon you, exclusive of the day of service. If you fail to do so, judgment by default will be taken against you for the relief demanded in the complaint. You must also file your answer with the Clerk of this Court within a reasonable period of time after service.

LUTHER D. THOMAS

JUN 8 2004

CLERK

DATE

BY DEPUTY CLERK

00138302

AO 440 (rev. 10/93) Summons in a Civil Action ORIGINAL

United States District Court

NORTHERN DISTRICT OF GEORGIA

IN RE: Datascape, Inc.

V.

Samsung Electronics America, Inc., et al

SUMMONS IN A CIVIL CASE

CASE NUMBER:

1:04-CV-1642

TO: (Name and address of defendant)

Sprint, Corporation
c/o American Incorporators Ltd.
1220 N. Market Street, Ste 606
Wilmington DE 19801

YOU ARE HEREBY SUMMONED and required to serve upon PLAINTIFF'S ATTORNEY (name and address)

PLAINTIFF'S ATTORNEY (name and address)

Stephen R. Risley
J.Scott Culpepper
N. Andrew Crain
Thomas, Kayden, Horstemeyer & Risley, L.L.P.
100 Galleria Parkway, N.W
Suite 1750
Atlanta, Georgia 30339-5948
(770) 933-9500

an answer to the complaint which is herewith served upon you, within **20** days after service of this summons upon you, exclusive of the day of service. If you fail to do so, judgment by default will be taken against you for the relief demanded in the complaint. You must also file your answer with the Clerk of this Court within a reasonable period of time after service.

LUTHER D. THOMAS

CLERK

JUN 8 2004

DATE

BY DEPUTY CLERK

00138309

AO 440 (rev. 10/93) Summons in a Civil Action ORIGINAL

United States District Court

NORTHERN DISTRICT OF GEORGIA

IN RE: Datascape, Inc.

V.

Samsung Electronics America, Inc., et al

SUMMONS IN A CIVIL CASE

CASE NUMBER: 1:04-CV-1642

TO: (Name and address of defendant)

Office Depot, Inc.
c/o Corporate Creations Network Inc.
2985 Gordy Parkway #421
Marietta, GA 30066

YOU ARE HEREBY SUMMONED and required to serve upon PLAINTIFF'S ATTORNEY (name and address)

PLAINTIFF'S ATTORNEY (name and address)

Stephen R. Risley
J.Scott Culpepper
N. Andrew Crain
Thomas, Kayden, Horstemeyer & Risley, L.L.P.
100 Galleria Parkway, N.W
Suite 1750
Atlanta, Georgia 30339-5948
(770) 933-9500

an answer to the complaint which is herewith served upon you, within **20** days after service of this summons upon you, exclusive of the day of service. If you fail to do so, judgment by default will be taken against you for the relief demanded in the complaint. You must also file your answer with the Clerk of this Court within a reasonable period of time after service.

LUTHER D. THOMAS

CLERK

JUN 8 2004

DATE

BY DEPUTY CLERK

00138301

ORIGINAL

AO 440 (rev. 10/93) Summons in a Civil Action

United States District Court

NORTHERN DISTRICT OF GEORGIA

IN RE: Datascape, Inc.

V.

Samsung Electronics America, Inc., et al

SUMMONS IN A CIVIL CASE

CASE NUMBER:

1:04-CV-1642

TO: (Name and address of defendant)

Circuit City Stores, Inc.
c/o Prentice-Hall Corp. System
40 Technology Pkwy South #300
Norcross, GA 30092

YOU ARE HEREBY SUMMONED and required to serve upon PLAINTIFF'S ATTORNEY (name and address)

PLAINTIFF'S ATTORNEY (name and address)

Stephen R. Risley
J.Scott Culpepper
N. Andrew Crain
Thomas, Kayden, Horstemeyer & Risley, L.L.P.
100 Galleria Parkway, N.W
Suite 1750
Atlanta, Georgia 30339-5948
(770) 933-9500

an answer to the complaint which is herewith served upon you, within **20** days after service of this summons upon you, exclusive of the day of service. If you fail to do so, judgment by default will be taken against you for the relief demanded in the complaint. You must also file your answer with the Clerk of this Court within a reasonable period of time after service.

LUTHER D. THOMAS

CLERK

JUN 8 2004

DATE

BY DEPUTY CLERK

00138294

ORIGINAL

AO 440 (rev. 10/93) Summons in a Civil Action

United States District Court

NORTHERN DISTRICT OF GEORGIA

IN RE: Datascape, Inc.

V.

Samsung Electronics America, Inc., et al

SUMMONS IN A CIVIL CASE

CASE NUMBER:

1:04-CV-1642

TO: (Name and address of defendant)

Best Buy Company, Inc.
c/o National Registered Agents, Inc.
3761 Venture Drive
Duluth, Georgia 30096

YOU ARE HEREBY SUMMONED and required to serve upon PLAINTIFF'S ATTORNEY (name and address)

PLAINTIFF'S ATTORNEY (name and address)

Stephen R. Risley
J.Scott Culpepper
N. Andrew Crain
Thomas, Kayden, Horstemeyer & Risley, L.L.P.
100 Galleria Parkway, N.W
Suite 1750
Atlanta, Georgia 30339-5948
(770) 933-9500

an answer to the complaint which is herewith served upon you, within **20** days after service of this summons upon you, exclusive of the day of service. If you fail to do so, judgment by default will be taken against you for the relief demanded in the complaint. You must also file your answer with the Clerk of this Court within a reasonable period of time after service.

LUTHER D. THOMAS

CLERK

JUN 8 2004

DATE

BY DEPUTY CLERK

00138292

AO 440 (rev. 10/93) Summons in a Civil Action ORIGINAL

United States District Court

NORTHERN DISTRICT OF GEORGIA

IN RE: Datascape, Inc.

V.

Samsung Electronics America, Inc., et al

SUMMONS IN A CIVIL CASE

CASE NUMBER: 1:04-CV-1642

TO: (Name and address of defendant)

Amazon.Com
c/o Corporation Service Company
40 Technology Pkway South #300
Norcross, Georgia 30092

YOU ARE HEREBY SUMMONED and required to serve upon PLAINTIFF'S ATTORNEY (name and address)

PLAINTIFF'S ATTORNEY (name and address)

Stephen R. Risley
J.Scott Culpepper
N. Andrew Crain
Thomas, Kayden, Horstemeyer & Risley, L.L.P.
100 Galleria Parkway, N.W
Suite 1750
Atlanta, Georgia 30339-5948
(770) 933-9500

an answer to the complaint which is herewith served upon you, within **20** days after service of this summons upon you, exclusive of the day of service. If you fail to do so, judgment by default will be taken against you for the relief demanded in the complaint. You must also file your answer with the Clerk of this Court within a reasonable period of time after service.

LUTHER D. THOMAS

CLERK

JUN 8 2004

DATE

BY DEPUTY CLERK

00138290

ORIGINAL

AO 440 (rev. 10/93) Summons in a Civil Action

United States District Court

NORTHERN DISTRICT OF GEORGIA

IN RE: Datascape, Inc.

V.

Samsung Electronics America, Inc., et al

SUMMONS IN A CIVIL CASE

CASE NUMBER:

1:04-CV-1642

TO: (Name and address of defendant)

Samsung Electronics America, Inc.
c/o C.T. Corporation
1201 Peachtree Street, NE
Atlanta, Georgia 30361

YOU ARE HEREBY SUMMONED and required to serve upon PLAINTIFF'S ATTORNEY (name and address)

PLAINTIFF'S ATTORNEY (name and address)

Stephen R. Risley
J.Scott Culpepper
N. Andrew Crain
Thomas, Kayden, Horstemeyer & Risley, L.L.P.
100 Galleria Parkway, N.W.
Suite 1750
Atlanta, Georgia 30339-5948
(770) 933-9500

an answer to the complaint which is herewith served upon you, within **20** days after service of this summons upon you, exclusive of the day of service. If you fail to do so, judgment by default will be taken against you for the relief demanded in the complaint. You must also file your answer with the Clerk of this Court within a reasonable period of time after service.

LUTHER D. THOMAS

CLERK

JUN 8 2004
DATE

BY DEPUTY CLERK