US006195651B1

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

(10) **Patent No.: US 6,195,651 B1**
(45) **Date of Patent: Feb. 27, 2001**

(54) **SYSTEM, METHOD AND ARTICLE OF MANUFACTURE FOR A TUNED USER APPLICATION EXPERIENCE**

(75) Inventors: **Sean Handel**, San Francisco; **Brian Day**, Burlingame; **Miya Yuen**, Foster City, all of CA (US)

(73) Assignee: **Andersen Consulting Properties BV**, The Hague (NL)

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

(21) Appl. No.: **09/196,483**

(22) Filed: **Nov. 19, 1998**

(51) **Int. Cl.**[7] **G06F 12/30**
(52) **U.S. Cl.** **707/2**; 707/5
(58) **Field of Search** 707/2, 5

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,388,198	2/1995	Layman et al.	395/155
5,454,106	9/1995	Burns et al. .	
5,465,358	11/1995	Blades et al.	395/700
5,555,169	9/1996	Namba et al. .	
5,652,850	7/1997	Hollander	395/333
5,675,753	10/1997	Hansen et al.	395/333
5,692,194	11/1997	Nguyen et al. .	
5,694,592	12/1997	Driscoll .	
5,717,923	2/1998	Dedrick	395/613
5,724,521	3/1998	Dedrick	395/226
5,732,218	3/1998	Bland et al.	395/200
5,754,938	5/1998	Herz et al.	455/4.2
5,754,939	5/1998	Herz, et al.	455/4.2
5,761,662 *	6/1998	Dasan	707/10
5,778,181	7/1998	Hidary et al. .	
5,812,847	9/1998	Joshi et al.	395/682
5,890,152	3/1999	Rapaport et al.	707/6
5,907,706	5/1999	Brodsky et al.	395/701
5,920,848 *	7/1999	Schutzer et al.	705/42
5,924,090 *	7/1999	Krellenstein	707/5
5,973,686	10/1999	Shimogori	345/333
5,974,253	10/1999	Nahaboo et al.	395/701
5,974,447 *	10/1999	Cannon et al.	709/206
6,014,665 *	1/2000	Culliss	707/5
6,029,045 *	2/2000	Picco et al.	455/5.1
6,029,165 *	2/2000	Gable	707/3
6,035,294 *	3/2000	Fish	707/2

FOREIGN PATENT DOCUMENTS

0367 709 A1	5/1990	(EP) .	
0643359A2	3/1995	(EP)	G06F/17/30
0441798 B1	11/1997	(EP) .	
0 854421A1	7/1998	(EP) .	
0855660A2	7/1998	(EP)	G06F/17/30
0537098 B1	1/1999	(EP) .	
WO97/02537	1/1997	(WO) .	
WO97/26729	7/1997	(WO) .	
WO 97/41673	11/1997	(WO) .	
WO 98/03907	1/1998	(WO) .	
WO 98/07103	2/1998	(WO) .	
WO98/35297	8/1998	(WO)	G06F/15/18
WO98/35469	8/1998	(WO) .	
WO 98/41913	9/1998	(WO) .	
WO98/40832	9/1998	(WO) .	
WO 98/56188	12/1998	(WO) .	
WO 99/01018	1/1999	(WO) .	
WO 99/01829	1/1999	(WO) .	

OTHER PUBLICATIONS

Firefly's Intelligent Agents Could End Up In Explorer; Computergram International, 1998.

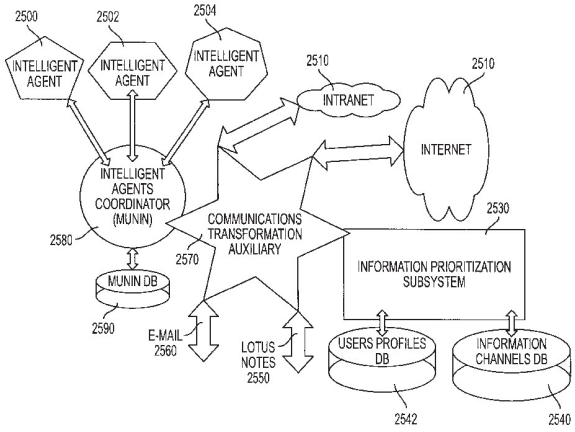
(List continued on next page.)

Primary Examiner—Wayne Amsbury
(74) *Attorney, Agent, or Firm*—Hickman Coleman & Hughes, LLP

(57) **ABSTRACT**

A system is disclosed that facilitates a web-based user interface to a particular application program that is enabled by obtaining user profile information, parsing the content of the particular application, matching the parsed content to user profile information and presenting the parsed content matches in a format based on information in the user's profile on a display. An innovative pattern matching system is integrated into the match processing to provide improved matching capability.

20 Claims, 27 Drawing Sheets



OTHER PUBLICATIONS

Mehran Sahami, Salim Yusufali, Michelle Q. W. Baldonaldo; SONIA: A Service for Organizing Networked Information Autonomously; Proceedings of the third ACM Conference on Digital libraries, pp. 200–209, 1998.

Keith L. Clark, Vasilios S. Lazarou; A Multi–Agent System for Distributed Information Retrieval on the World Wide Web; Proceedings, Sixth IEEE Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises; Jun. 18–20, 1997.

Terry R. Payne, Peter Edwards, Claire L. Green; Experience with Rule Induction and k–Nearest Neighbor Methods for Interface Agents that Learn; IEEE Transactions in Knowledge and Data Engineering, vol. 9, No. 2, Mar.–Apr. 1997.

Niel Robertson; A Personalized Web: New technologies enable Web sited to better serve their visitors by gauging users' unique needs and customizing the delivery of content; Internet World, Apr. 1997.

Bum Ryeol Yoon, Jae Jeong You, Soo Dong Kim; COPEN: A CORBA–Based Intelligent Push–engine; Proceedings 1998 Asia Pacific Software Engineering Conference; IEEE Computer Society; Dec. 2–4, 1998.

M. Montebello, W. A. Gray, S. Hurley; An Evolvable Personal Advisor to optimize Internet Search Technologies; Proceedings, Database and Expert Systems Applications, 9th International Conference, 1998.

Don Crabb; Active assistance taken to its logical potential; MacWEEK, vol. 8, No. 38; Sep. 26, 1994.

L. J. Bannon; Problems in Human–Machine Interaction and Communication; Design of Computing Systems: Social and Ergonomic Considerations; Proceedings of the Seventh International Conference on Human–Computer Interaction; vol. 2, Aug. 24–29, 1997.

Scott E. Sampson; Employing Internet Technologies to Gather Customer Quality Perceptions; Proceedings, Annual meeting of the Decision Sciences Institute, vol. 3; 1998.

Jac–Hyun Lim, Hyon–Woo Seung, Jun Hwang, Young–Chan Kim, Heung–Nam Kim; Query Expansion for Intelligent Information Retrieval on Internet; Proceedings of the 1997 International Conference on Parallel and Distributed Systems; Dec. 10–13, 1997; IEEE Computer Society.

Katashi Nagao, Jun Rekimoto; Agent Augmented Reality: A Software Agent Meets the Real World; ICMAS–96.

E. S. Lee, R. Okada, I. G. Jeon; Agent–based Support for Personalized Information with Web Search Engines; Design of Computing Systems: Cognitive Considerations; Proceedings of the Seventh International Conference on Human–Computer Interaction; vol. 1, Aug. 24–29, 1997.

Richard V. Dragan; Future Agent Software: Intelligent software will personalize the Web and make it work for you; PC Magazine; Mar. 25, 1997.

Scott M. Brown, Eugene Santos Jr., Sheila B. Banks; Utility Theory–Based User Models for Intelligent Interface Agents; Advances in Artificial Intelligence; 12th Biennial Conference of the Canadian Society for Computational Studies of Intelligence; Jun. 18–20, 1998.

Chungnan Lee, Yao–tsung Chen; An Embedded Visual Programming Interface for Intelligent Information Retrieval on the Web; Proceedings, 1997 IEEE Knowledge and Data Engineering Exchange Workshop; IEEE Computer Society, Nov. 4, 1997.

Z. Chen, Q. Zhu; Exploring agent–based data mining refinement cycle; Proceedings of the Second International Conference on the Practical Application of Knowledge Discovery and Data Mining, 1998.

Scott E. Sampson; Gathering customer feedback via the Internet: Instruments and prospects; Industrial Management & Data Systems, vol. 98, No. 2, 1998.

"Clean: Using Syntactic Information in Document Filtering", R. Chandrasekar & B. Srinivas, Institute for Research in Cognitive Science and Center for the Advanced Study of India, University of Pennsylvania, Jul. 7, 1998, pp. 623–640.

"Information Filtering Based on User Behavior Analysis and Best match Text Retrieval", Masahiro & Yoichi Shinda, School of Information Science Japan Advanced Institute of Science and Technology, Mar. 7, 1994, pp. 272–281.

"Knowledge Discovery and Delivery", John Davies and Ralph Cochrane, British Telecommunications Engineering, vol. 17, Apr. 1998, pp. 25–35.

"Stereotypes in Information Filtering Systems", Bracha Shapira, Peretz Shoval and Uri Hanami, Department of Industrial Engineering and Management, Bar–Ilan University of the Negev, Beer–Sheva, Isreal, Information Processing & Management, vol. 33, No. 3, pp. 273–287, Jan. 10, 1997.

* cited by examiner

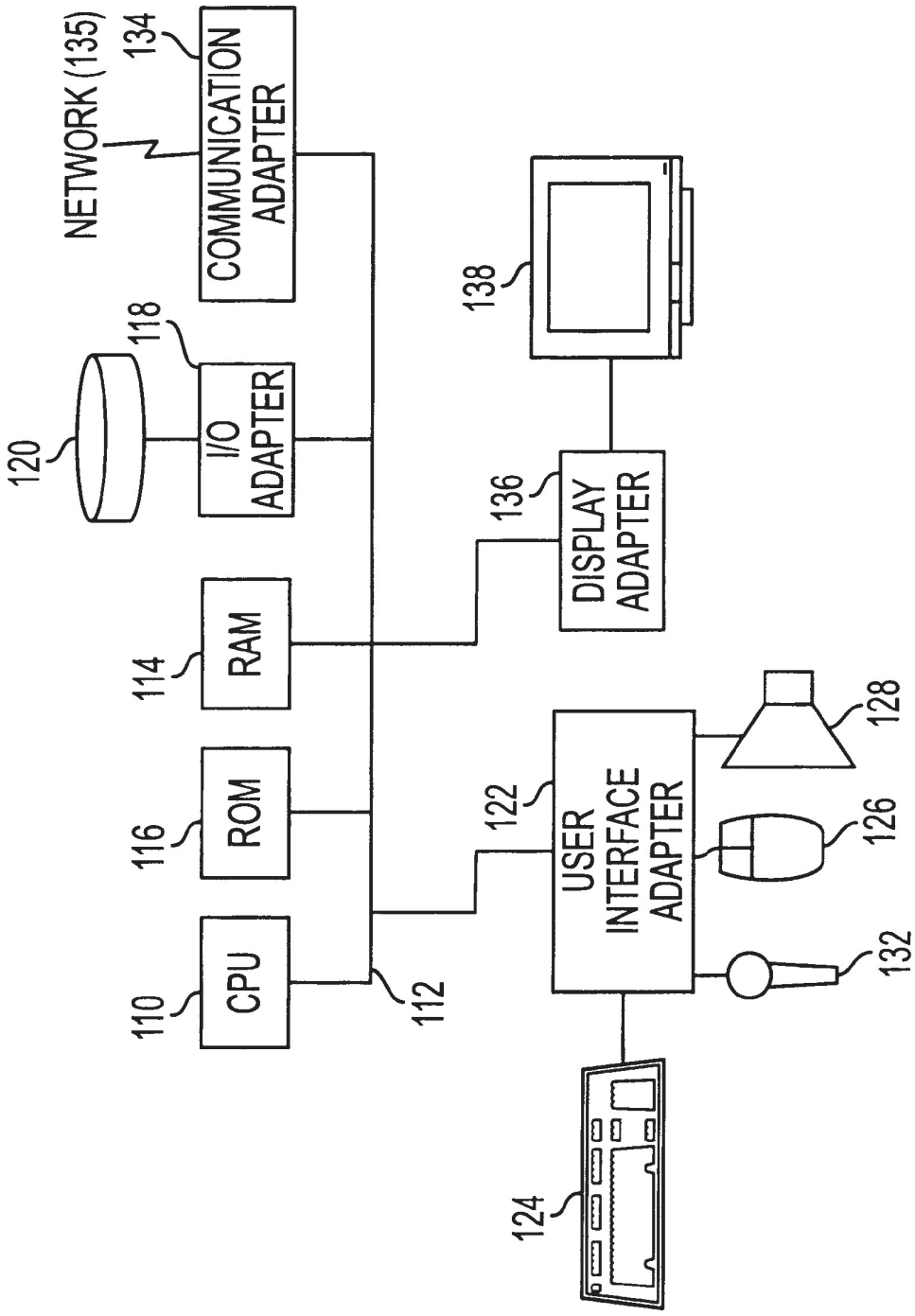

NETWORK (135)
134
COMMUNICATION ADAPTER
118
I/O ADAPTER
120
138
136
DISPLAY ADAPTER
114
RAM
116
ROM
110
CPU
112
122
USER INTERFACE ADAPTER
128
126
132
124

FIG. 1

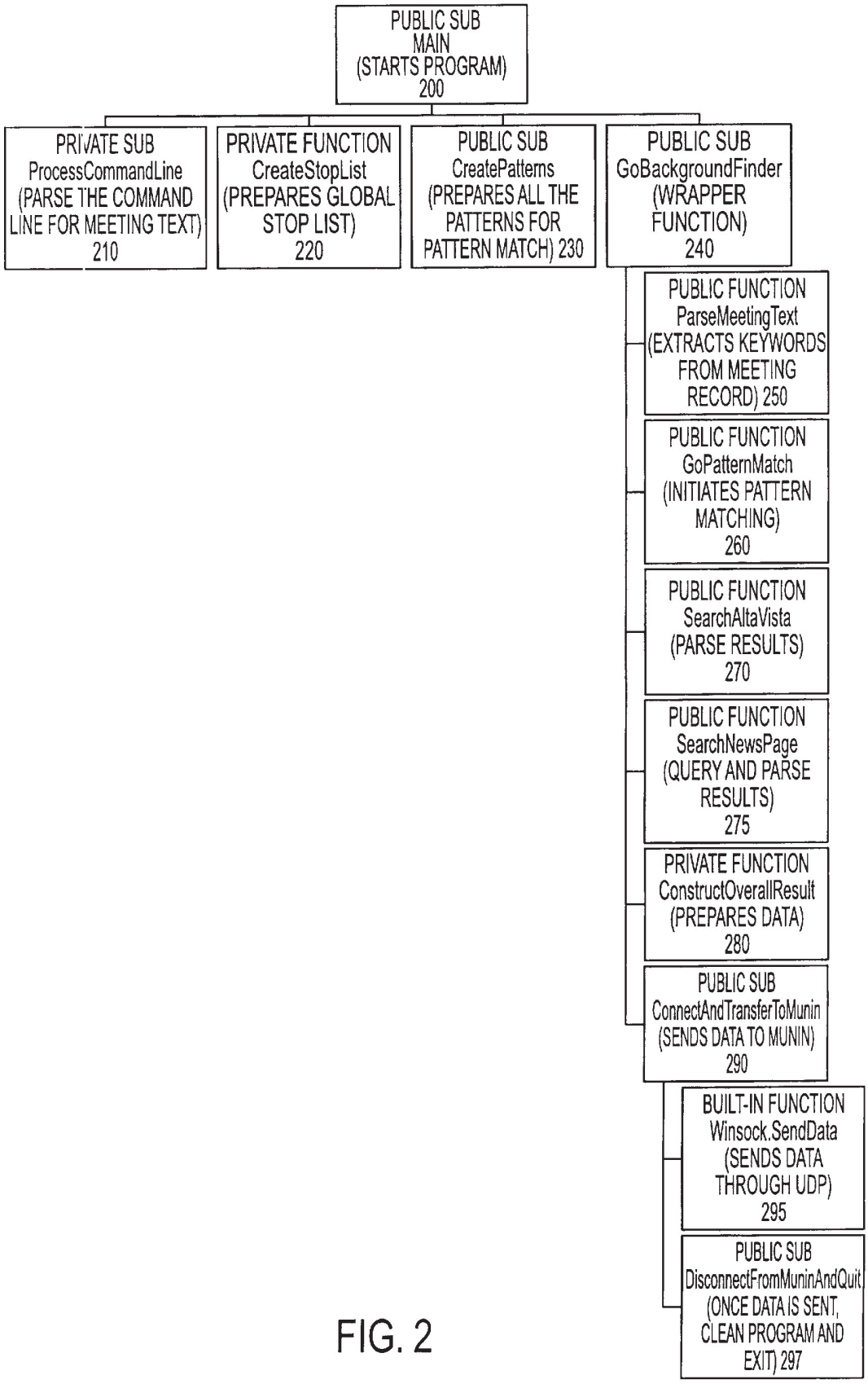

PUBLIC SUB
MAIN
(STARTS PROGRAM)
200
PRIVATE SUB
ProcessCommandLine
(PARSE THE COMMAND
LINE FOR MEETING TEXT)
210
PRIVATE FUNCTION
CreateStopList
(PREPARES GLOBAL
STOP LIST)
220
PUBLIC SUB
CreatePatterns
(PREPARES ALL THE
PATTERNS FOR
PATTERN MATCH) 230
PUBLIC SUB
GoBackgroundFinder
(WRAPPER
FUNCTION)
240
PUBLIC FUNCTION
ParseMeetingText
(EXTRACTS KEYWORDS
FROM MEETING
RECORD) 250
PUBLIC FUNCTION
GoPatternMatch
(INITIATES PATTERN
MATCHING)
260
PUBLIC FUNCTION
SearchAltaVista
(PARSE RESULTS)
270
PUBLIC FUNCTION
SearchNewsPage
(QUERY AND PARSE
RESULTS)
275
PRIVATE FUNCTION
ConstructOverallResult
(PREPARES DATA)
280
PUBLIC SUB
ConnectAndTransferToMunin
(SENDS DATA TO MUNIN)
290
BUILT-IN FUNCTION
Winsock.SendData
(SENDS DATA
THROUGH UDP)
295
PUBLIC SUB
DisconnectFromMuninAndQuit
(ONCE DATA IS SENT,
CLEAN PROGRAM AND
EXIT) 297

FIG. 2

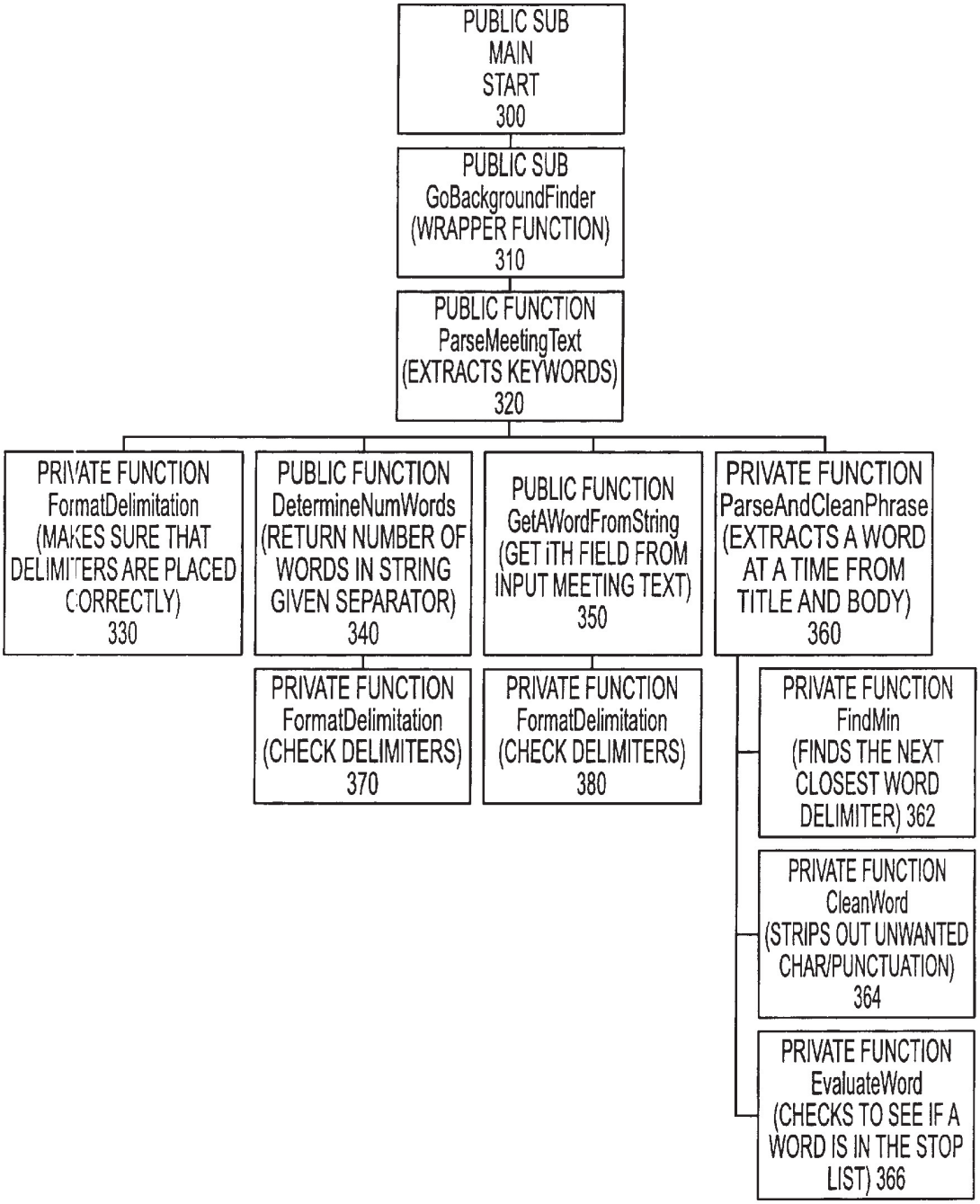

PUBLIC SUB
MAIN
START
300
PUBLIC SUB
GoBackgroundFinder
(WRAPPER FUNCTION)
310
PUBLIC FUNCTION
ParseMeetingText
(EXTRACTS KEYWORDS)
320
PRIVATE FUNCTION
FormatDelimitation
(MAKES SURE THAT
DELIMITERS ARE PLACED
CORRECTLY)
330
PUBLIC FUNCTION
DetermineNumWords
(RETURN NUMBER OF
WORDS IN STRING
GIVEN SEPARATOR)
340
PUBLIC FUNCTION
GetAWordFromString
(GET iTH FIELD FROM
INPUT MEETING TEXT)
350
PRIVATE FUNCTION
ParseAndCleanPhrase
(EXTRACTS A WORD
AT A TIME FROM
TITLE AND BODY)
360
PRIVATE FUNCTION
FormatDelimitation
(CHECK DELIMITERS)
370
PRIVATE FUNCTION
FormatDelimitation
(CHECK DELIMITERS)
380
PRIVATE FUNCTION
FindMin
(FINDS THE NEXT
CLOSEST WORD
DELIMITER) 362
PRIVATE FUNCTION
CleanWord
(STRIPS OUT UNWANTED
CHAR/PUNCTUATION)
364
PRIVATE FUNCTION
EvaluateWord
(CHECKS TO SEE IF A
WORD IS IN THE STOP
LIST) 366

FIG. 3

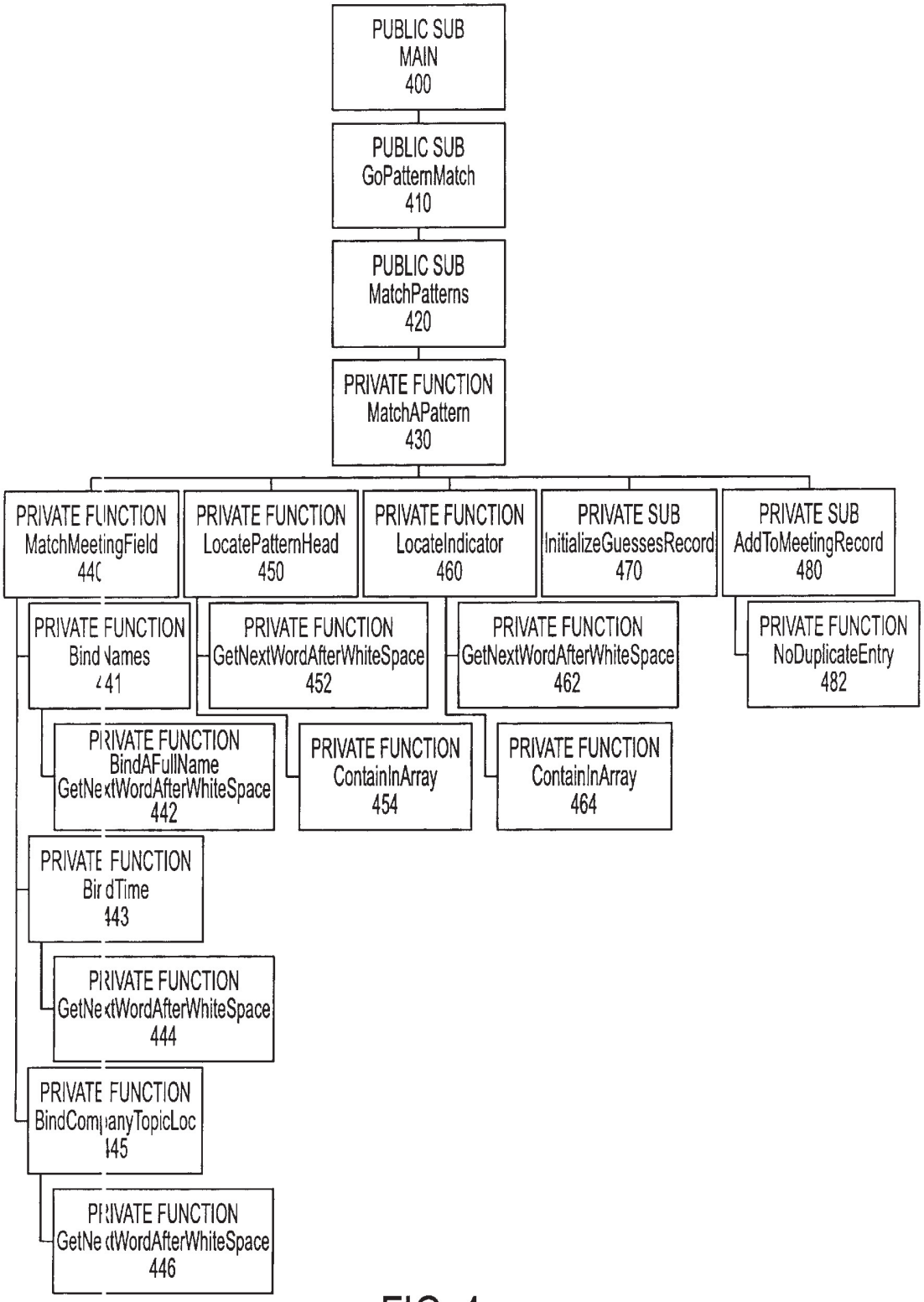

PUBLIC SUB
MAIN
400
PUBLIC SUB
GoPatternMatch
410
PUBLIC SUB
MatchPatterns
420
PRIVATE FUNCTION
MatchAPattern
430
PRIVATE FUNCTION
MatchMeetingField
440
PRIVATE FUNCTION
LocatePatternHead
450
PRIVATE FUNCTION
LocateIndicator
460
PRIVATE SUB
InitializeGuessesRecord
470
PRIVATE SUB
AddToMeetingRecord
480
PRIVATE FUNCTION
BindNames
441
PRIVATE FUNCTION
GetNextWordAfterWhiteSpace
452
PRIVATE FUNCTION
GetNextWordAfterWhiteSpace
462
PRIVATE FUNCTION
NoDuplicateEntry
482
PRIVATE FUNCTION
BindAFullName
GetNextWordAfterWhiteSpace
442
PRIVATE FUNCTION
ContainInArray
454
PRIVATE FUNCTION
ContainInArray
464
PRIVATE FUNCTION
BindTime
443
PRIVATE FUNCTION
GetNextWordAfterWhiteSpace
444
PRIVATE FUNCTION
BindCompanyTopicLoc
445
PRIVATE FUNCTION
GetNextWordAfterWhiteSpace
446

FIG. 4

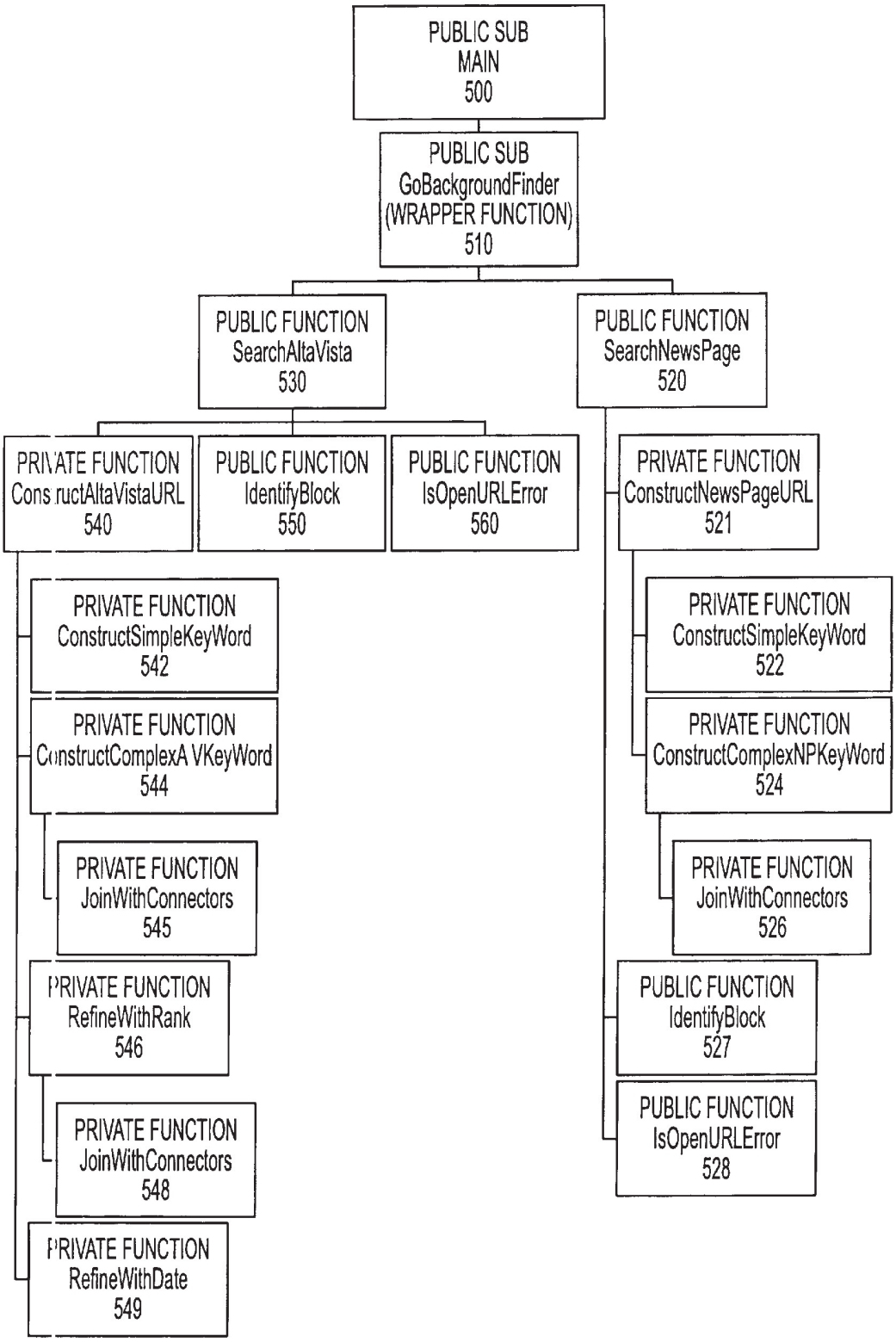

PUBLIC SUB
MAIN
500
PUBLIC SUB
GoBackgroundFinder
(WRAPPER FUNCTION)
510
PUBLIC FUNCTION
SearchAltaVista
530
PUBLIC FUNCTION
SearchNewsPage
520
PRIVATE FUNCTION
ConstructAltaVistaURL
540
PUBLIC FUNCTION
IdentifyBlock
550
PUBLIC FUNCTION
IsOpenURLError
560
PRIVATE FUNCTION
ConstructNewsPageURL
521
PRIVATE FUNCTION
ConstructSimpleKeyWord
542
PRIVATE FUNCTION
ConstructComplexAVKeyWord
544
PRIVATE FUNCTION
JoinWithConnectors
545
PRIVATE FUNCTION
RefineWithRank
546
PRIVATE FUNCTION
JoinWithConnectors
548
PRIVATE FUNCTION
RefineWithDate
549
PRIVATE FUNCTION
ConstructSimpleKeyWord
522
PRIVATE FUNCTION
ConstructComplexNPKeyWord
524
PRIVATE FUNCTION
JoinWithConnectors
526
PUBLIC FUNCTION
IdentifyBlock
527
PUBLIC FUNCTION
IsOpenURLError
528

FIG. 5

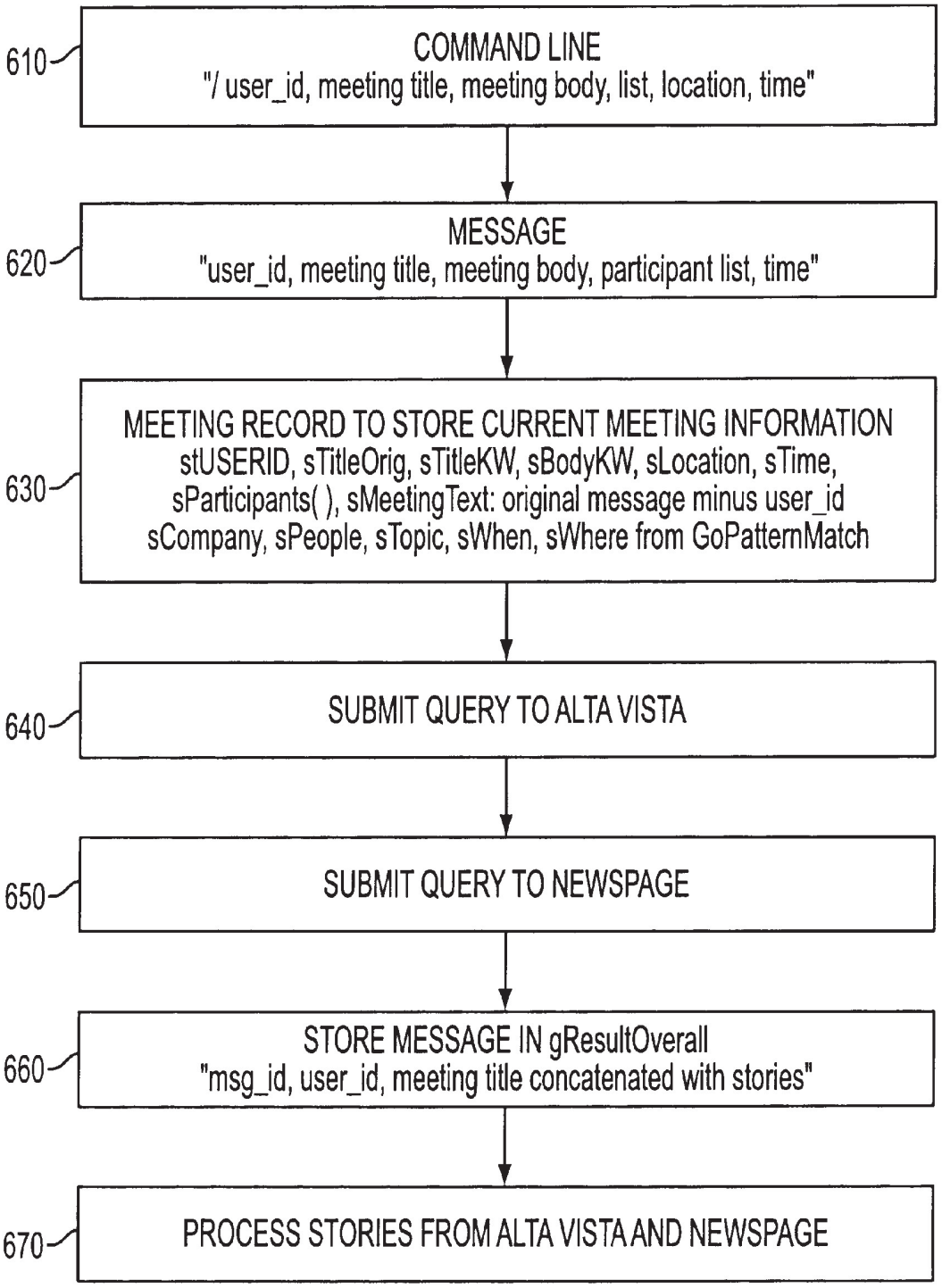

610
COMMAND LINE
"/ user_id, meeting title, meeting body, list, location, time"
620
MESSAGE
"user_id, meeting title, meeting body, participant list, time"
630
MEETING RECORD TO STORE CURRENT MEETING INFORMATION
stUSERID, sTitleOrig, sTitleKW, sBodyKW, sLocation, sTime,
sParticipants( ), sMeetingText: original message minus user_id
sCompany, sPeople, sTopic, sWhen, sWhere from GoPatternMatch
640
SUBMIT QUERY TO ALTA VISTA
650
SUBMIT QUERY TO NEWSPAGE
660
STORE MESSAGE IN gResultOverall
"msg_id, user_id, meeting title concatenated with stories"
670
PROCESS STORIES FROM ALTA VISTA AND NEWSPAGE

FIG. 6

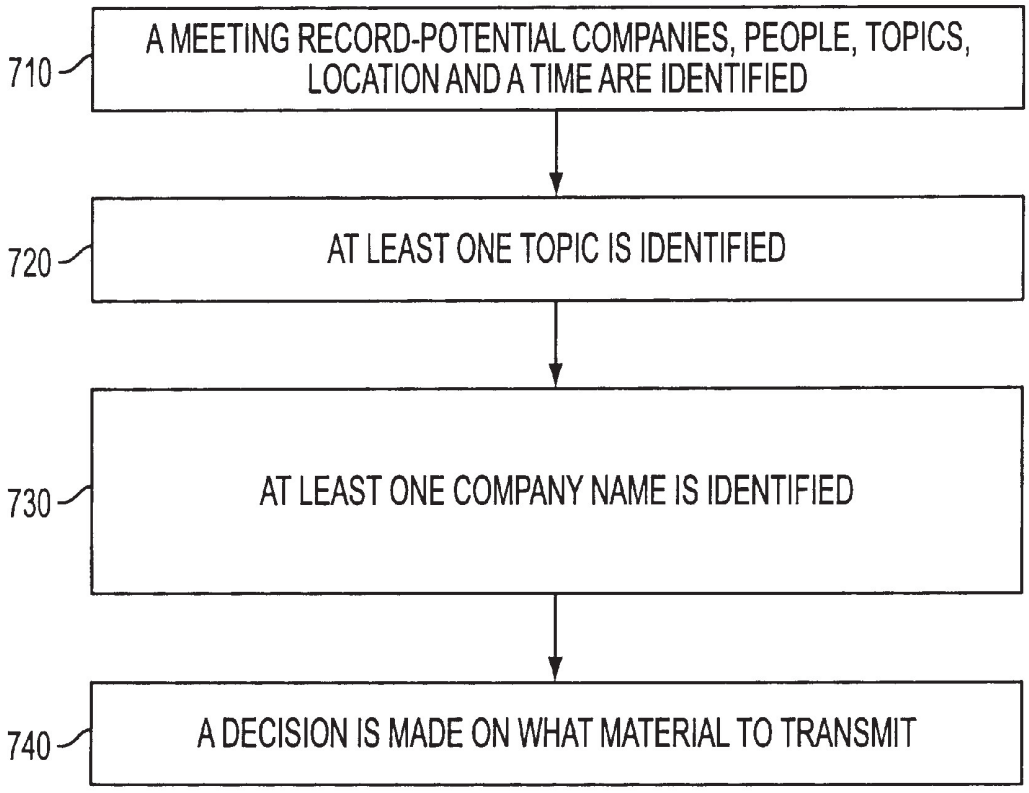

710
A MEETING RECORD-POTENTIAL COMPANIES, PEOPLE, TOPICS, LOCATION AND A TIME ARE IDENTIFIED
720
AT LEAST ONE TOPIC IS IDENTIFIED
730
AT LEAST ONE COMPANY NAME IS IDENTIFIED
740
A DECISION IS MADE ON WHAT MATERIAL TO TRANSMIT

FIG. 7

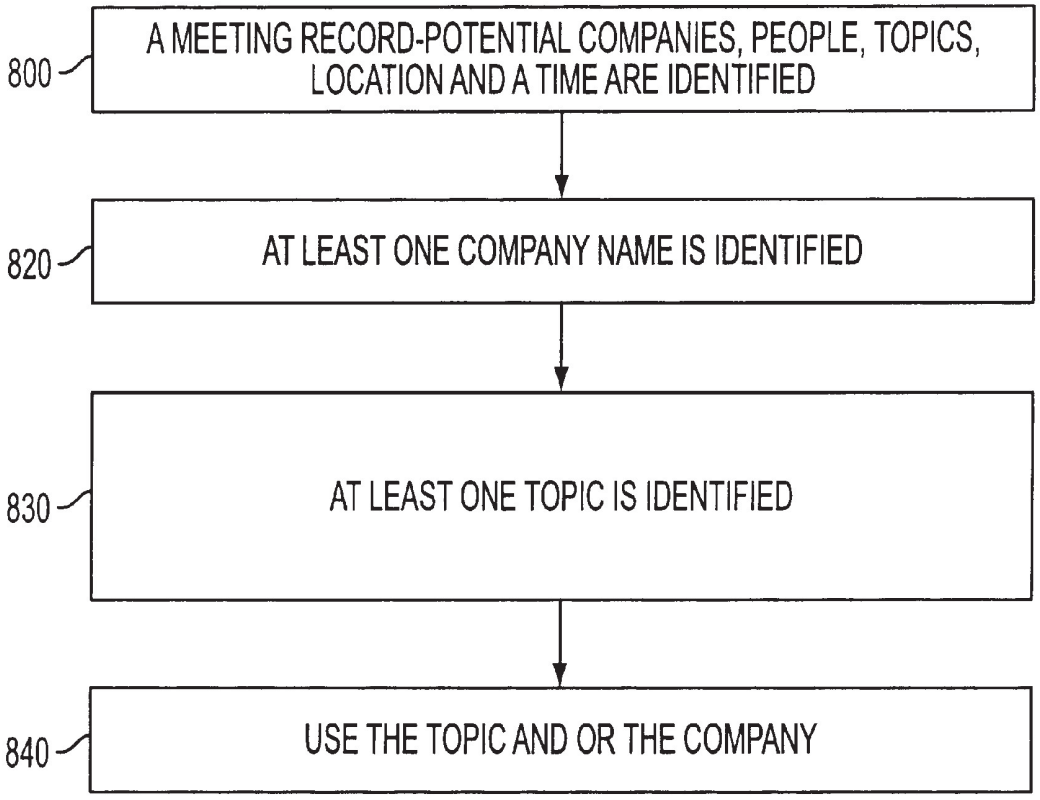

800
A MEETING RECORD-POTENTIAL COMPANIES, PEOPLE, TOPICS, LOCATION AND A TIME ARE IDENTIFIED
820
AT LEAST ONE COMPANY NAME IS IDENTIFIED
830
AT LEAST ONE TOPIC IS IDENTIFIED
840
USE THE TOPIC AND OR THE COMPANY

FIG. 8

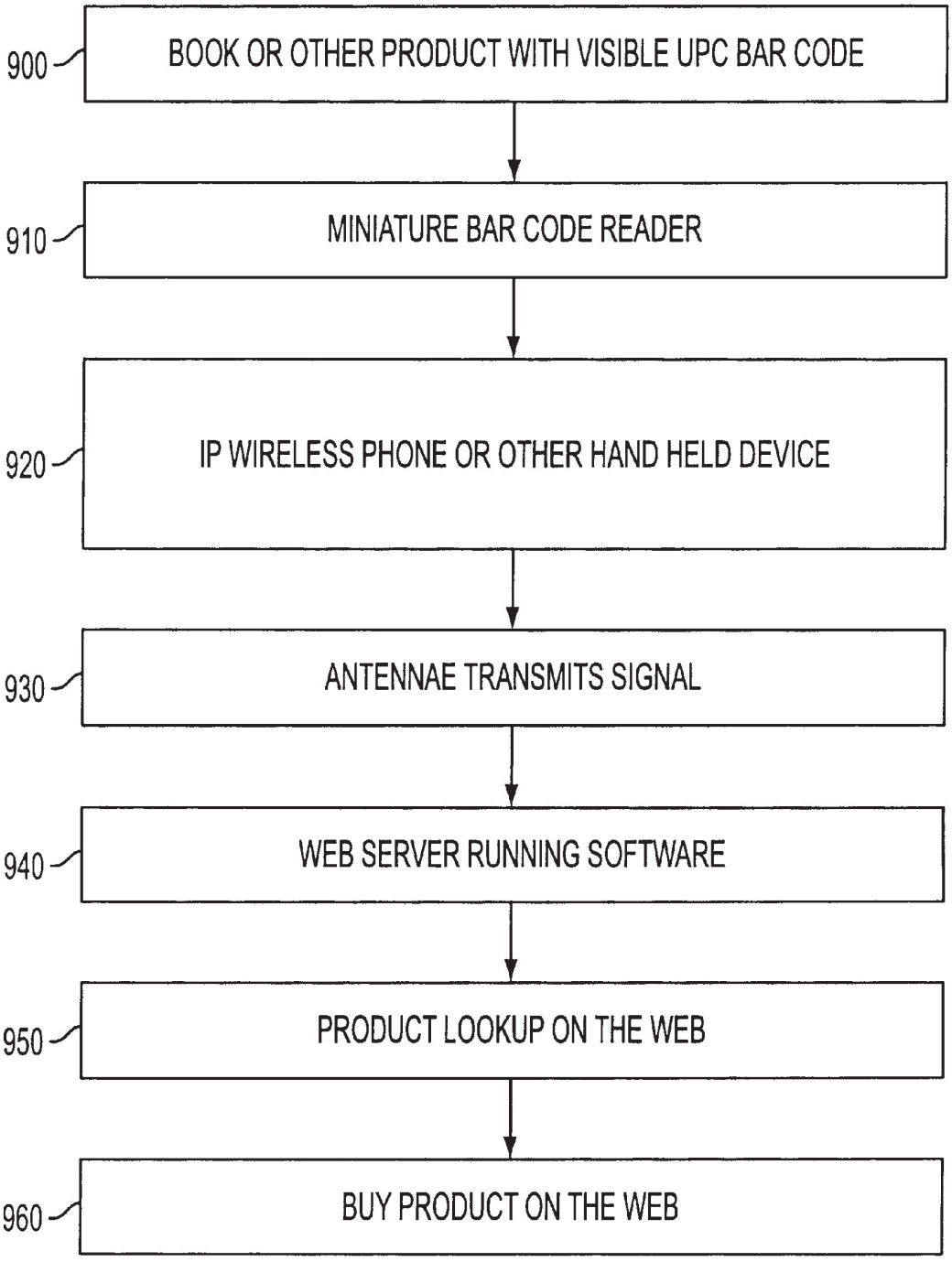

900
BOOK OR OTHER PRODUCT WITH VISIBLE UPC BAR CODE
910
MINIATURE BAR CODE READER
920
IP WIRELESS PHONE OR OTHER HAND HELD DEVICE
930
ANTENNAE TRANSMITS SIGNAL
940
WEB SERVER RUNNING SOFTWARE
950
PRODUCT LOOKUP ON THE WEB
960
BUY PRODUCT ON THE WEB

FIG. 9

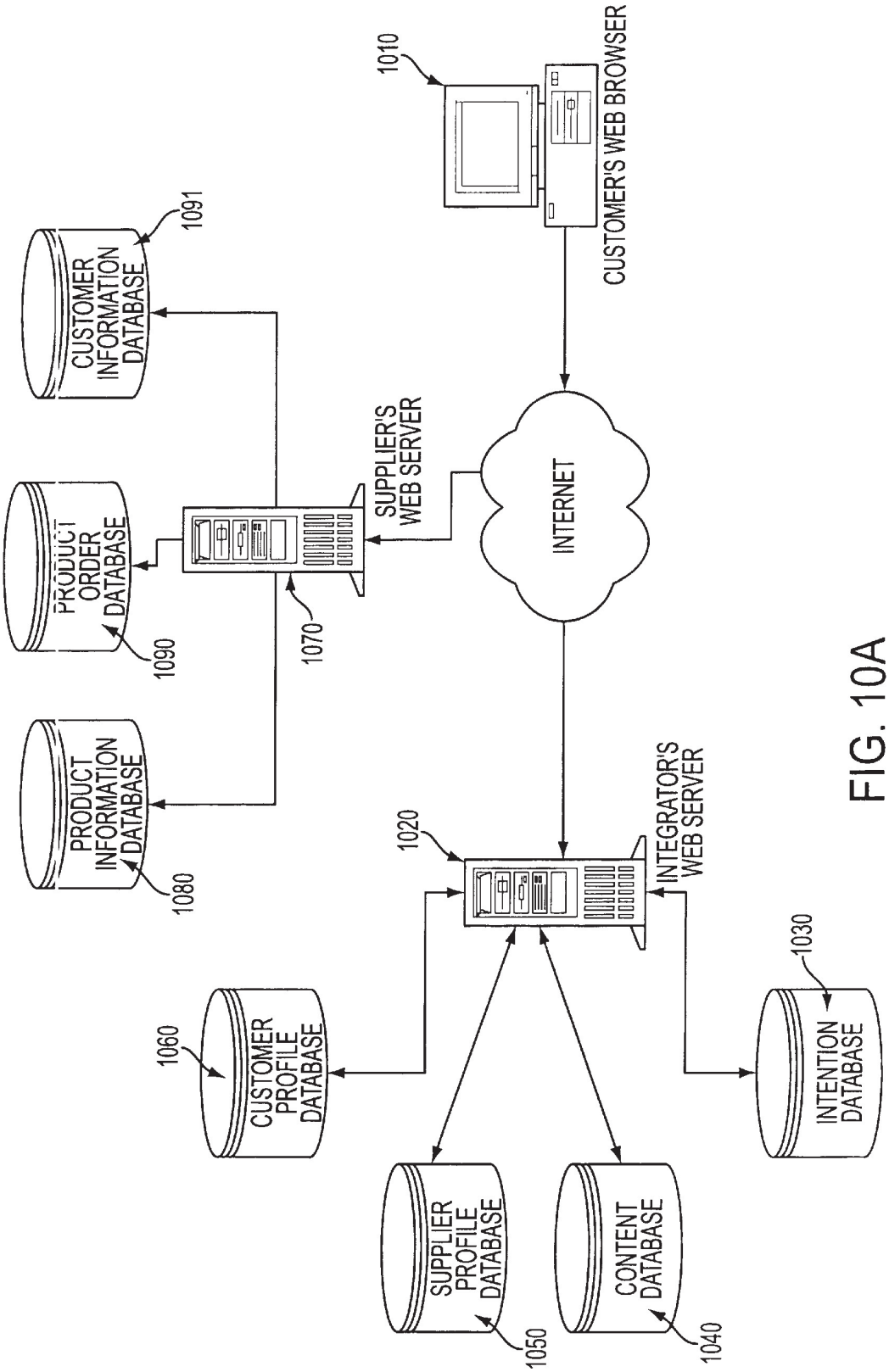

1010
CUSTOMER'S WEB BROWSER
1091
CUSTOMER INFORMATION DATABASE
SUPPLIER'S WEB SERVER
INTERNET
PRODUCT ORDER DATABASE
1090
1070
PRODUCT INFORMATION DATABASE
1080
1020
INTEGRATOR'S WEB SERVER
1030
1060
CUSTOMER PROFILE DATABASE
INTENTION DATABASE
SUPPLIER PROFILE DATABASE
CONTENT DATABASE
1050
1040

FIG. 10A

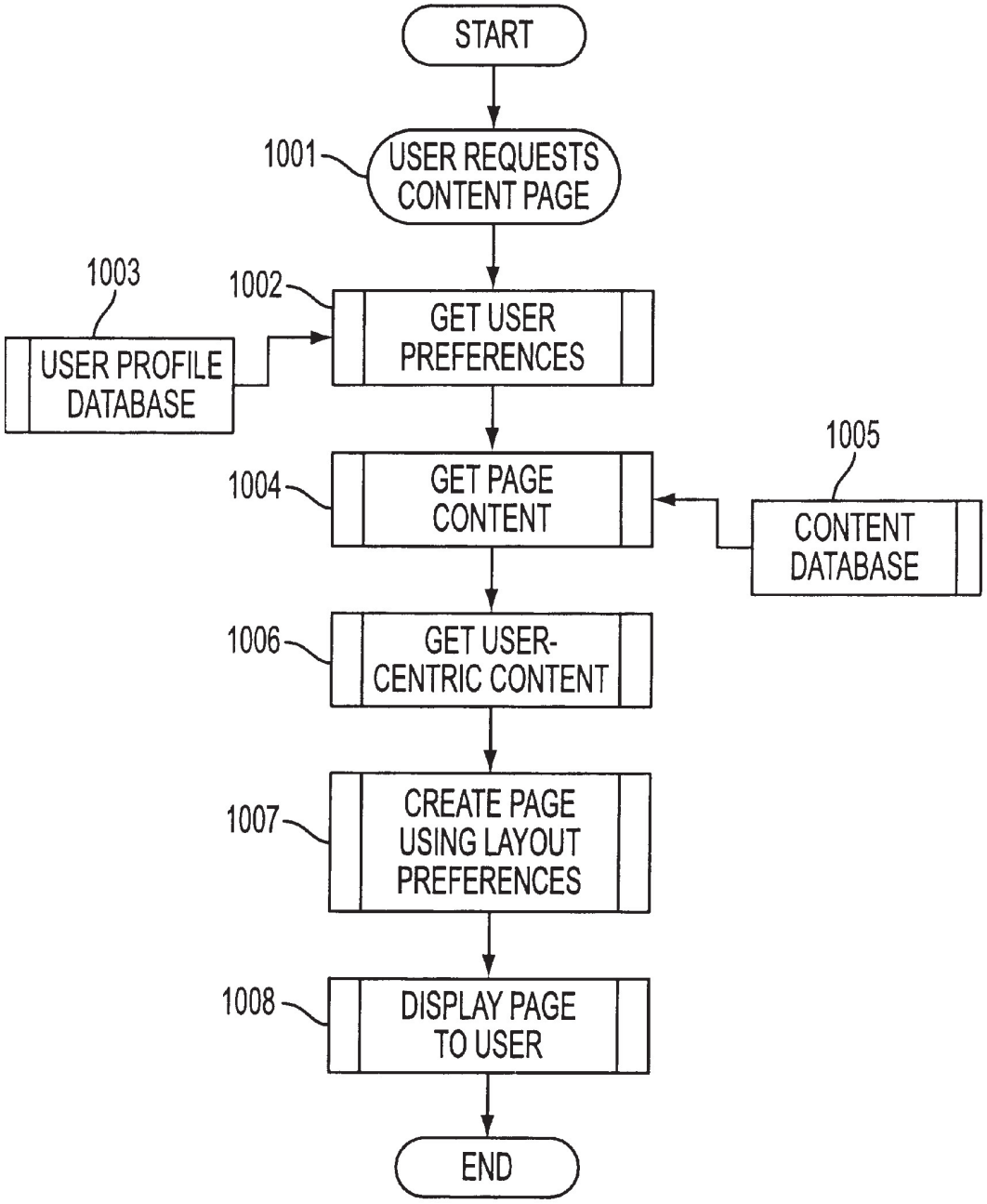

START
1001
USER REQUESTS CONTENT PAGE
1003
1002
USER PROFILE DATABASE
GET USER PREFERENCES
1005
1004
GET PAGE CONTENT
CONTENT DATABASE
1006
GET USER-CENTRIC CONTENT
1007
CREATE PAGE USING LAYOUT PREFERENCES
1008
DISPLAY PAGE TO USER
END

FIG. 10B

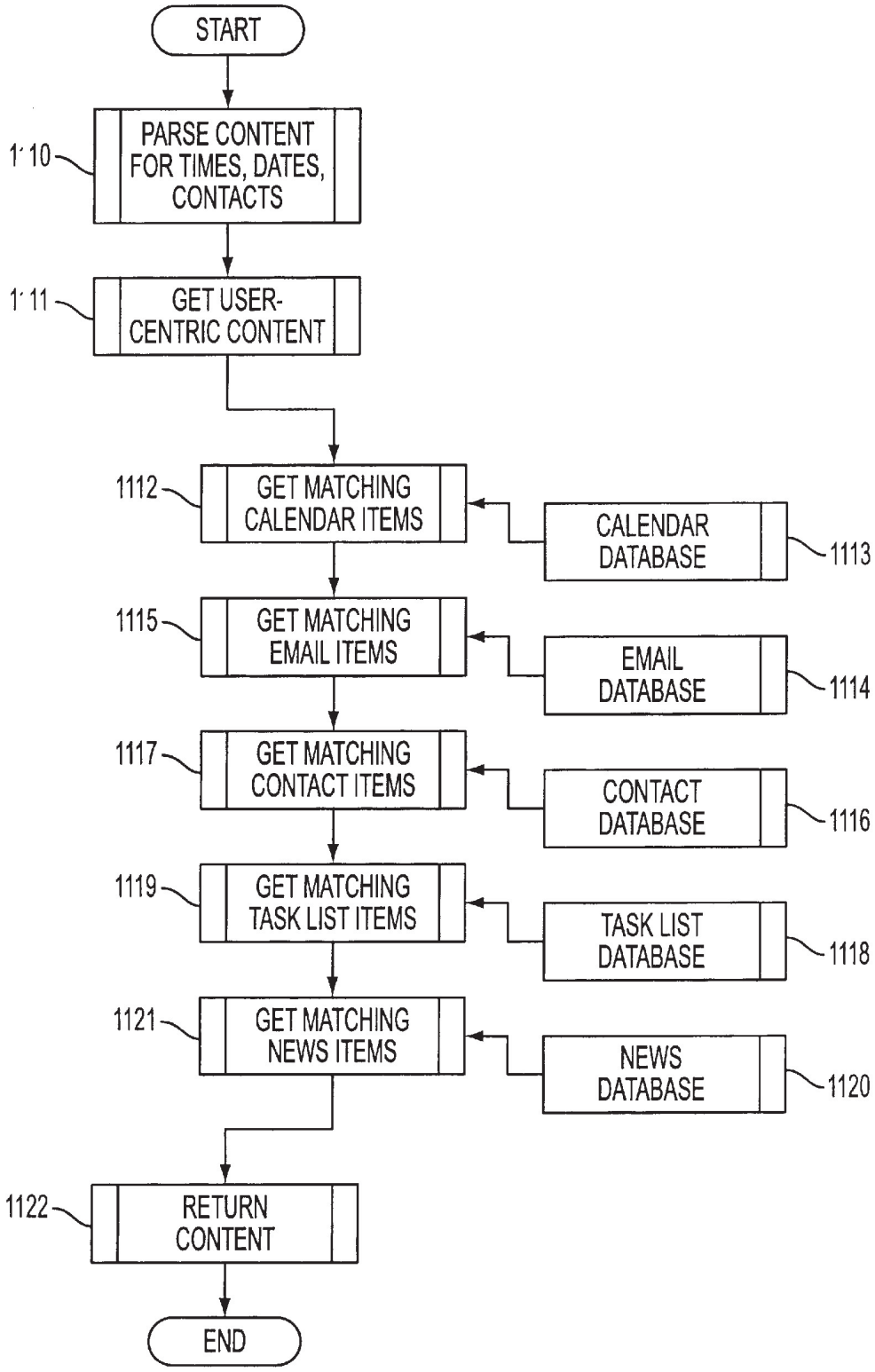

START
1110
PARSE CONTENT FOR TIMES, DATES, CONTACTS
1111
GET USER-CENTRIC CONTENT
1112
GET MATCHING CALENDAR ITEMS
CALENDAR DATABASE
1113
1115
GET MATCHING EMAIL ITEMS
EMAIL DATABASE
1114
1117
GET MATCHING CONTACT ITEMS
CONTACT DATABASE
1116
1119
GET MATCHING TASK LIST ITEMS
TASK LIST DATABASE
1118
1121
GET MATCHING NEWS ITEMS
NEWS DATABASE
1120
1122
RETURN CONTENT
END

FIG. 11

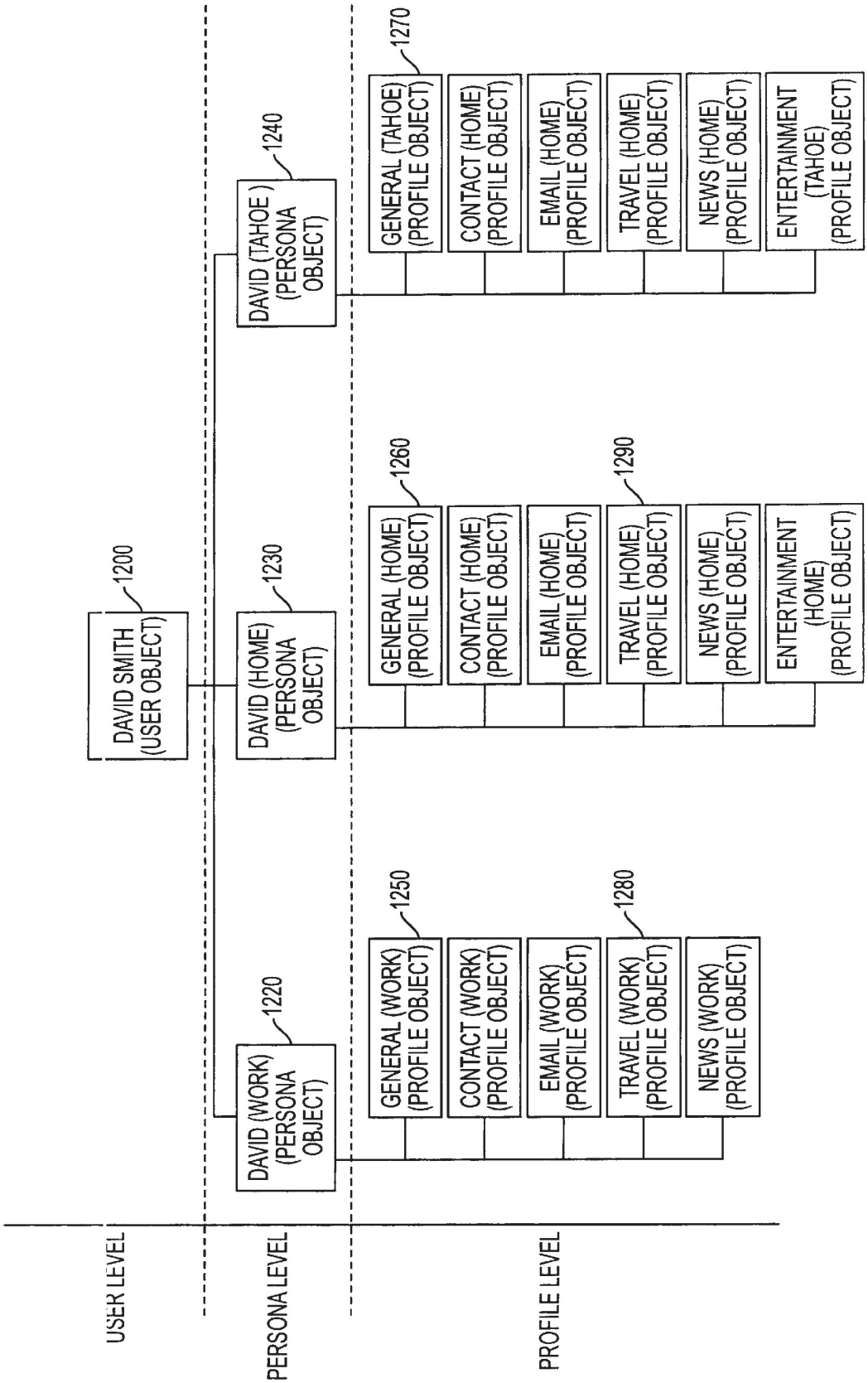

USER LEVEL
PERSONA LEVEL
PROFILE LEVEL
DAVID SMITH (USER OBJECT)
1200
DAVID (WORK) (PERSONA OBJECT)
1220
DAVID (HOME) (PERSONA OBJECT)
1230
DAVID (TAHOE) (PERSONA OBJECT)
1240
GENERAL (WORK) (PROFILE OBJECT)
1250
CONTACT (WORK) (PROFILE OBJECT)
EMAIL (WORK) (PROFILE OBJECT)
TRAVEL (WORK) (PROFILE OBJECT)
1280
NEWS (WORK) (PROFILE OBJECT)
GENERAL (HOME) (PROFILE OBJECT)
1260
CONTACT (HOME) (PROFILE OBJECT)
EMAIL (HOME) (PROFILE OBJECT)
TRAVEL (HOME) (PROFILE OBJECT)
1290
NEWS (HOME) (PROFILE OBJECT)
ENTERTAINMENT (HOME) (PROFILE OBJECT)
GENERAL (TAHOE) (PROFILE OBJECT)
1270
CONTACT (HOME) (PROFILE OBJECT)
EMAIL (HOME) (PROFILE OBJECT)
TRAVEL (HOME) (PROFILE OBJECT)
NEWS (HOME) (PROFILE OBJECT)
ENTERTAINMENT (TAHOE) (PROFILE OBJECT)

FIG. 12

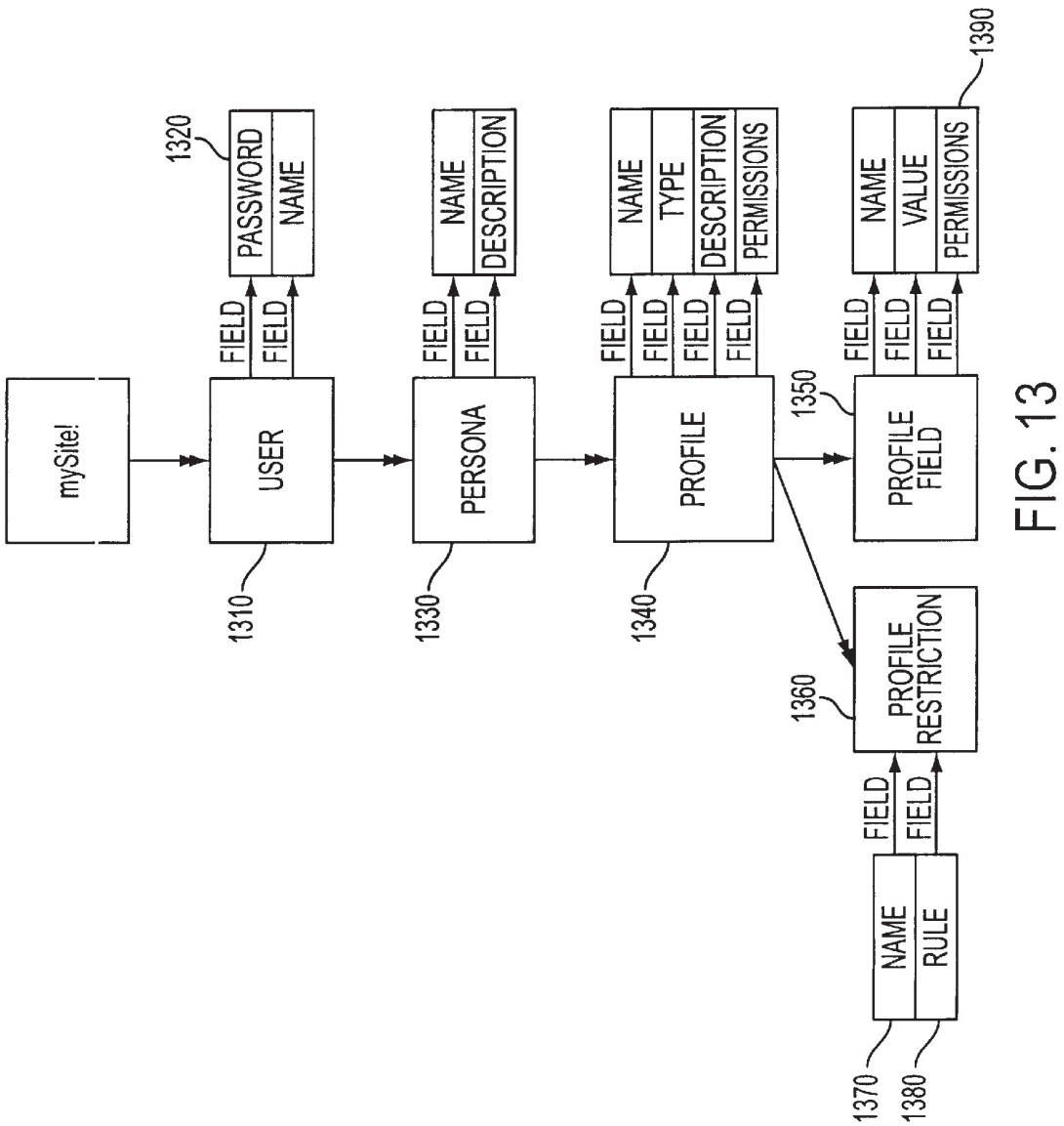

mySite!
USER
1310
FIELD
FIELD
1320
PASSWORD
NAME
PERSONA
1330
FIELD
FIELD
NAME
DESCRIPTION
PROFILE
1340
FIELD
FIELD
FIELD
FIELD
NAME
TYPE
DESCRIPTION
PERMISSIONS
PROFILE FIELD
1350
FIELD
FIELD
FIELD
1390
NAME
VALUE
PERMISSIONS
PROFILE RESTRICTION
1360
FIELD
FIELD
NAME
RULE
1370
1380

FIG. 13

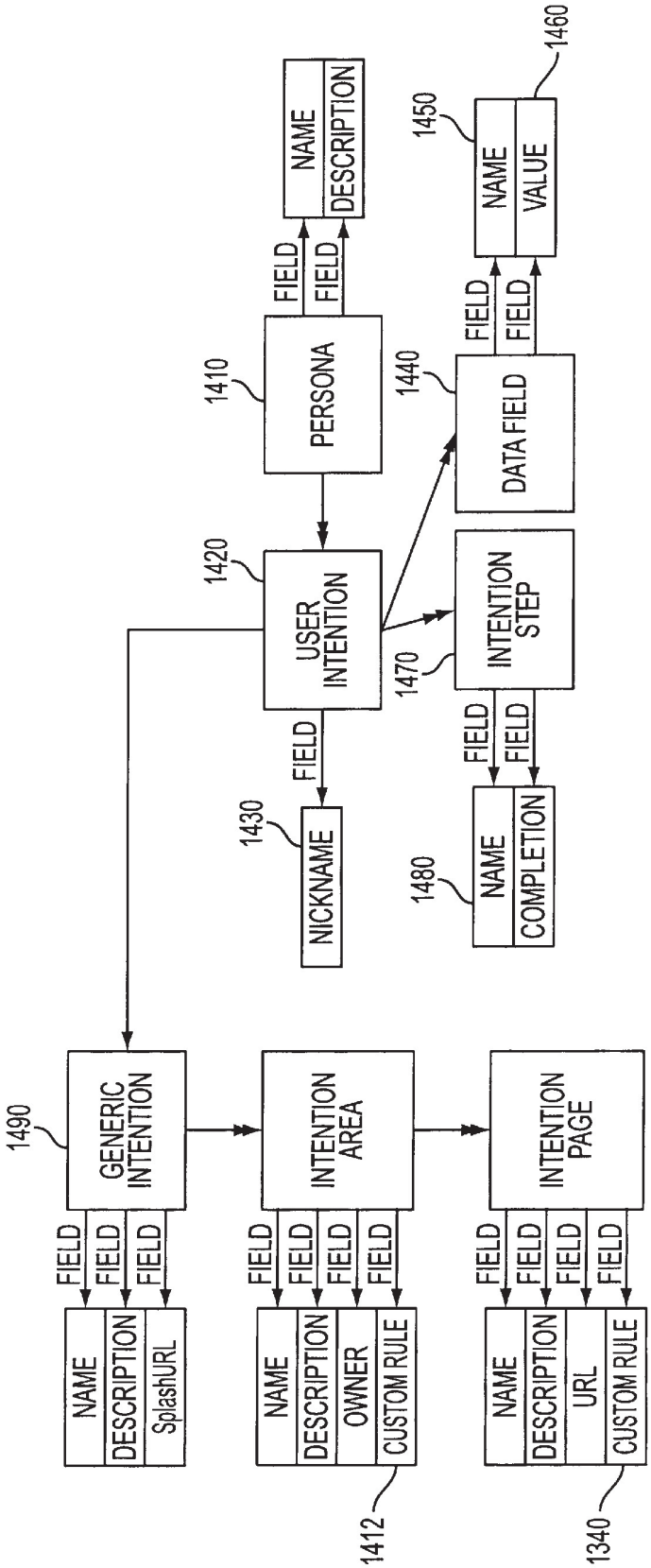

1460
1450
NAME
DESCRIPTION
NAME
VALUE
FIELD
FIELD
FIELD
FIELD
1410
PERSONA
1440
DATA FIELD
1420
USER INTENTION
1470
INTENTION STEP
FIELD
FIELD
FIELD
FIELD
1430
NICKNAME
1480
NAME
COMPLETION
1490
GENERIC INTENTION
INTENTION AREA
INTENTION PAGE
FIELD
FIELD
FIELD
FIELD
FIELD
FIELD
FIELD
FIELD
FIELD
FIELD
FIELD
FIELD
NAME
DESCRIPTION
SplashURL
NAME
DESCRIPTION
OWNER
CUSTOM RULE
NAME
DESCRIPTION
URL
CUSTOM RULE
1412
1340

FIG. 14

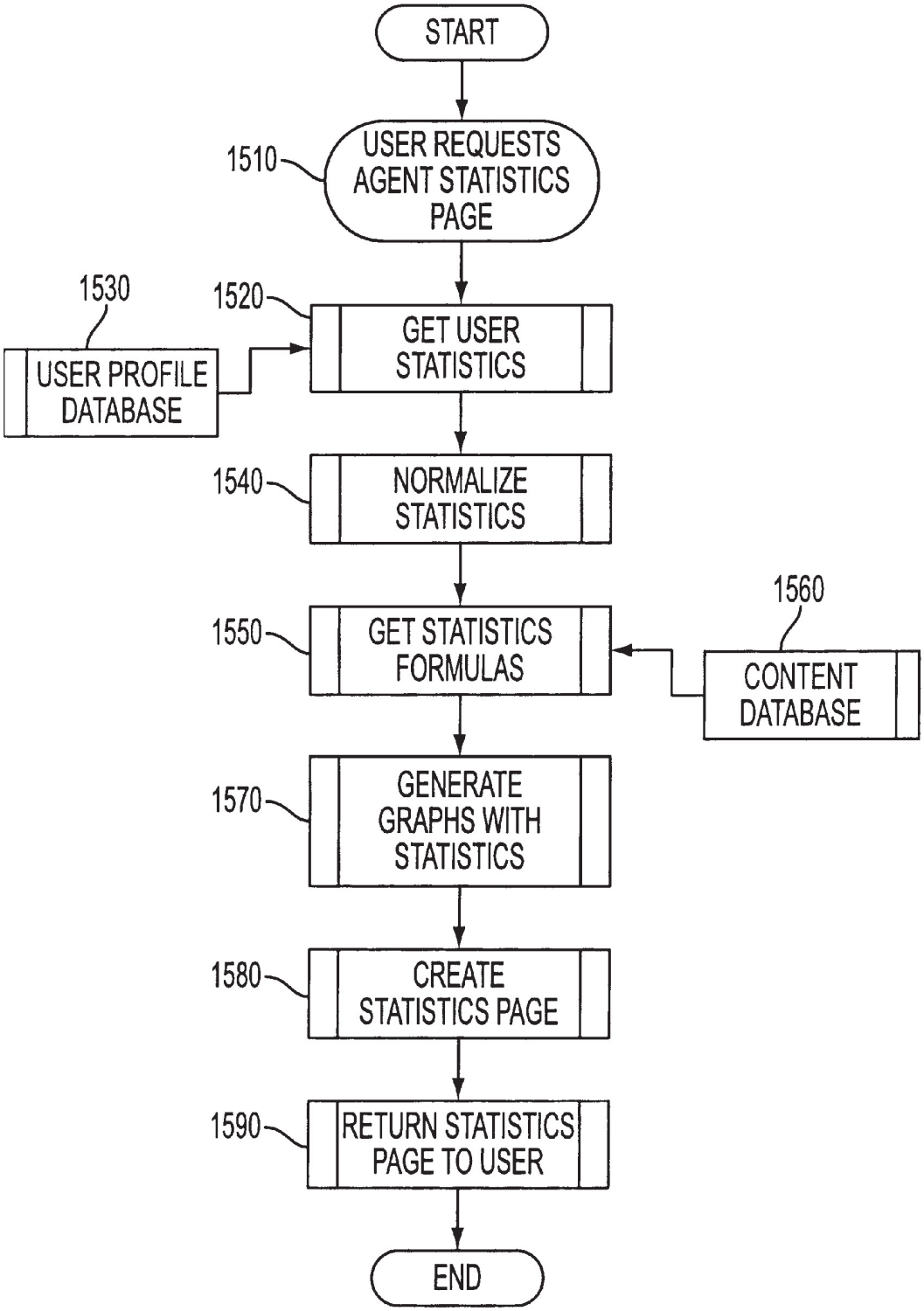

START
1510
USER REQUESTS AGENT STATISTICS PAGE
1530
1520
USER PROFILE DATABASE
GET USER STATISTICS
1540
NORMALIZE STATISTICS
1560
1550
GET STATISTICS FORMULAS
CONTENT DATABASE
1570
GENERATE GRAPHS WITH STATISTICS
1580
CREATE STATISTICS PAGE
1590
RETURN STATISTICS PAGE TO USER
END

FIG. 15

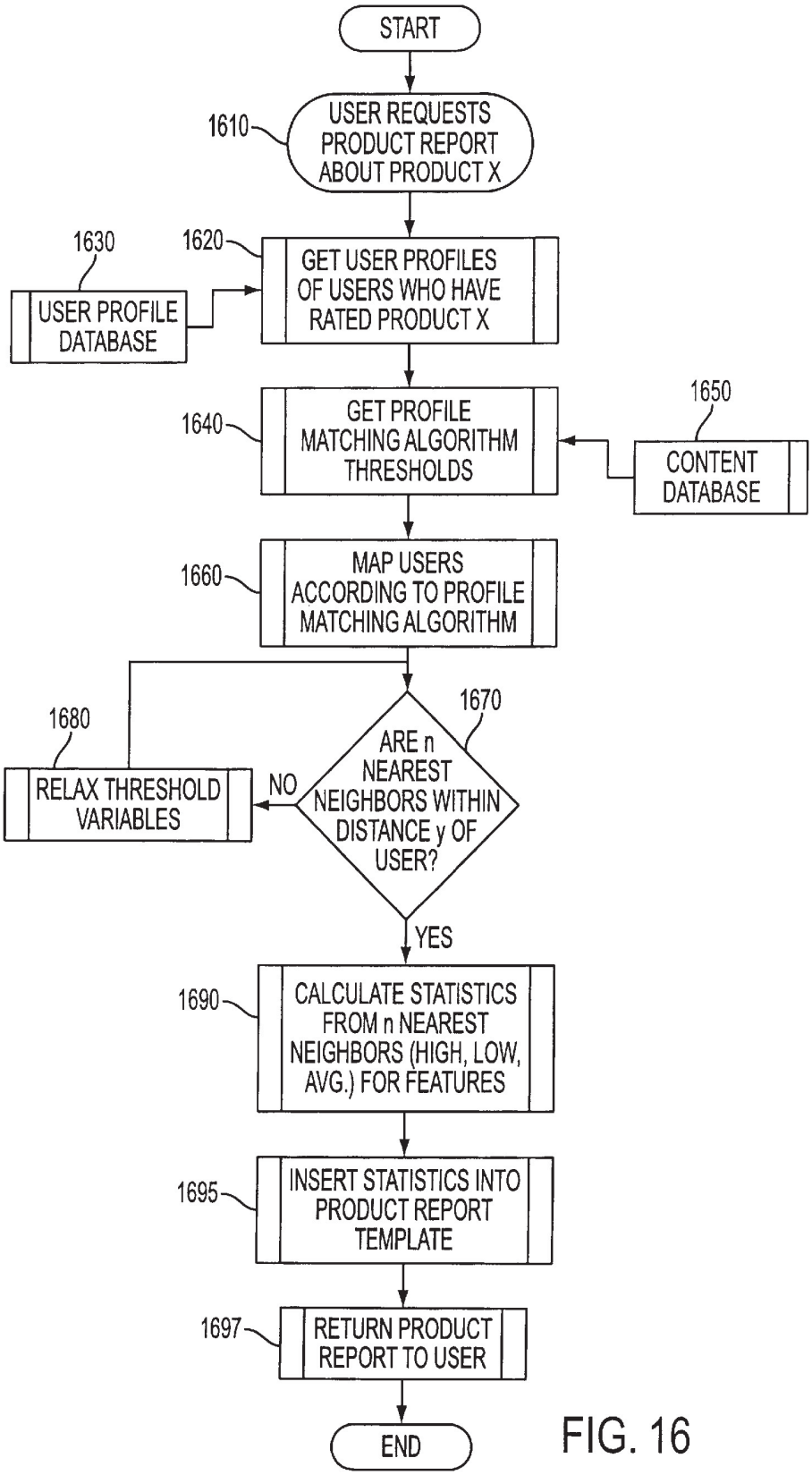

START
1610
USER REQUESTS PRODUCT REPORT ABOUT PRODUCT X
1630
1620
USER PROFILE DATABASE
GET USER PROFILES OF USERS WHO HAVE RATED PRODUCT X
1650
1640
GET PROFILE MATCHING ALGORITHM THRESHOLDS
CONTENT DATABASE
1660
MAP USERS ACCORDING TO PROFILE MATCHING ALGORITHM
1680
1670
RELAX THRESHOLD VARIABLES
NO
ARE n NEAREST NEIGHBORS WITHIN DISTANCE y OF USER?
YES
1690
CALCULATE STATISTICS FROM n NEAREST NEIGHBORS (HIGH, LOW, AVG.) FOR FEATURES
1695
INSERT STATISTICS INTO PRODUCT REPORT TEMPLATE
1697
RETURN PRODUCT REPORT TO USER
END

FIG. 16

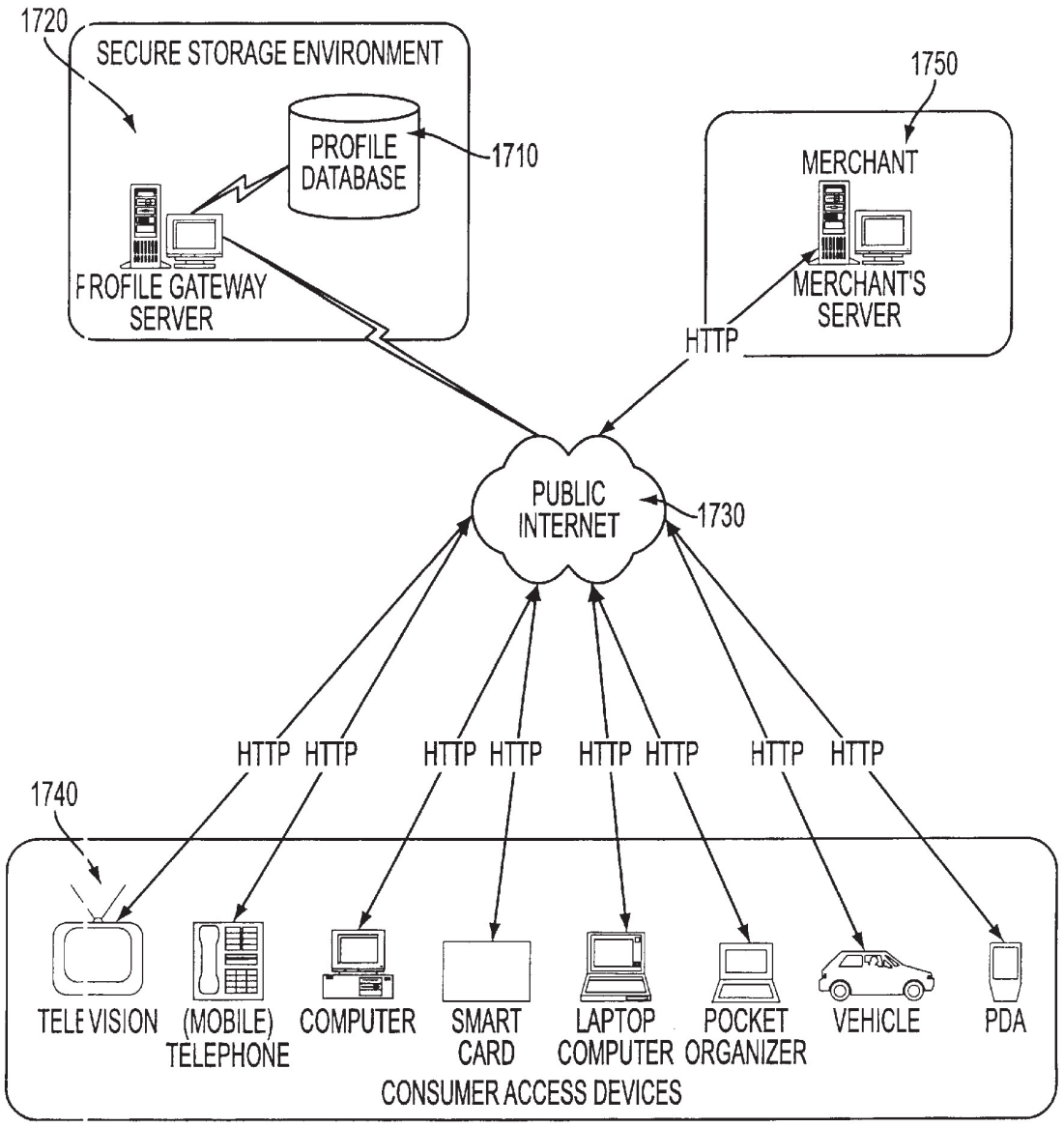

1720
SECURE STORAGE ENVIRONMENT
PROFILE DATABASE
1710
PROFILE GATEWAY SERVER
1750
MERCHANT
MERCHANT'S SERVER
HTTP
PUBLIC INTERNET
1730
HTTP HTTP
HTTP HTTP
HTTP HTTP
HTTP HTTP
1740
TELEVISION
(MOBILE) TELEPHONE
COMPUTER
SMART CARD
LAPTOP COMPUTER
POCKET ORGANIZER
VEHICLE
PDA
CONSUMER ACCESS DEVICES

FIG. 17

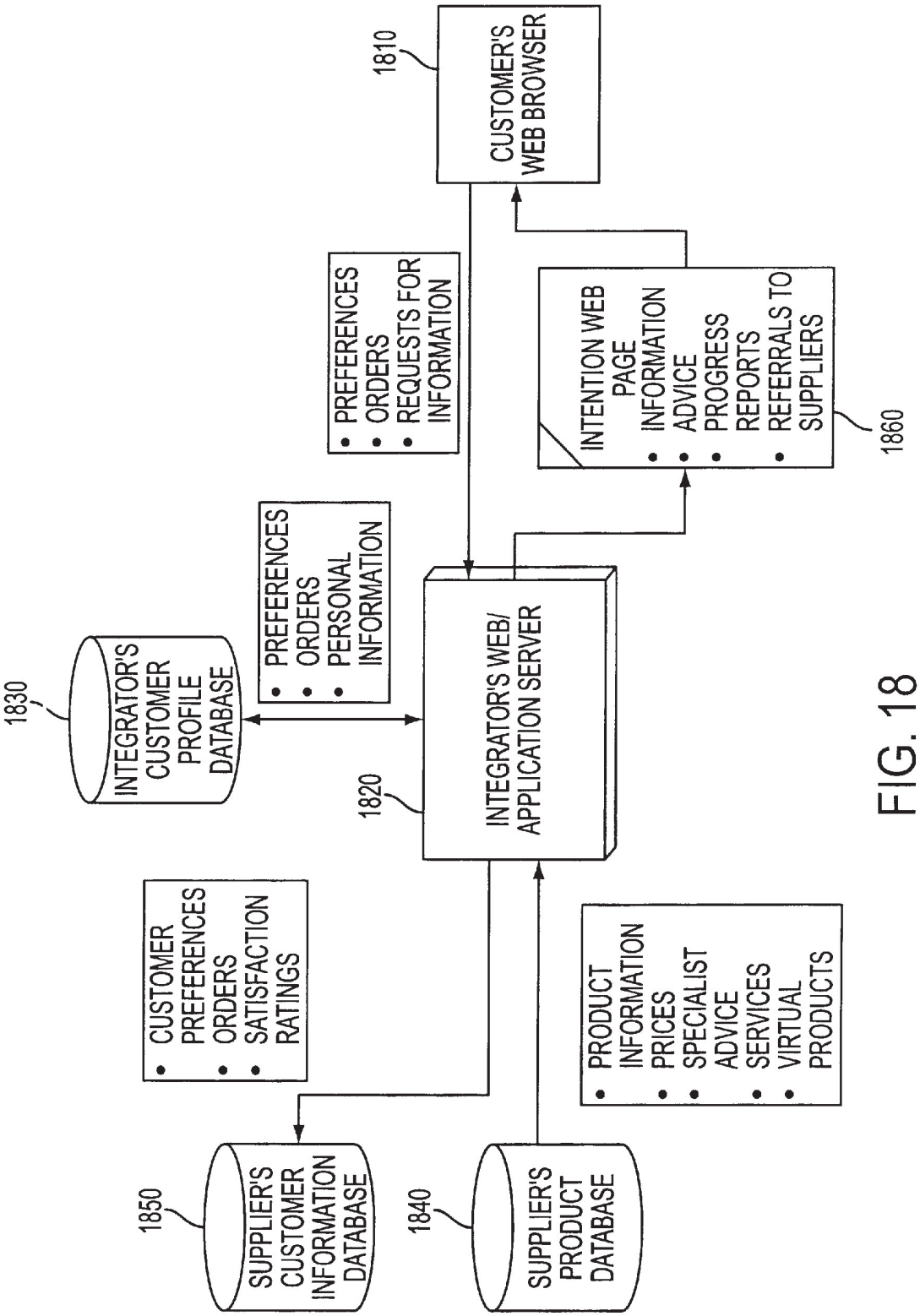

1810
CUSTOMER'S WEB BROWSER
• PREFERENCES
• ORDERS
• REQUESTS FOR INFORMATION
INTENTION WEB PAGE
• INFORMATION
• ADVICE
• PROGRESS REPORTS
• REFERRALS TO SUPPLIERS
1860
1830
INTEGRATOR'S CUSTOMER PROFILE DATABASE
• PREFERENCES
• ORDERS
• PERSONAL INFORMATION
1820
INTEGRATOR'S WEB/ APPLICATION SERVER
• CUSTOMER PREFERENCES
• ORDERS
• SATISFACTION RATINGS
• PRODUCT INFORMATION
• PRICES
• SPECIALIST ADVICE
• SERVICES
• VIRTUAL PRODUCTS
1850
SUPPLIER'S CUSTOMER INFORMATION DATABASE
1840
SUPPLIER'S PRODUCT DATABASE

FIG. 18

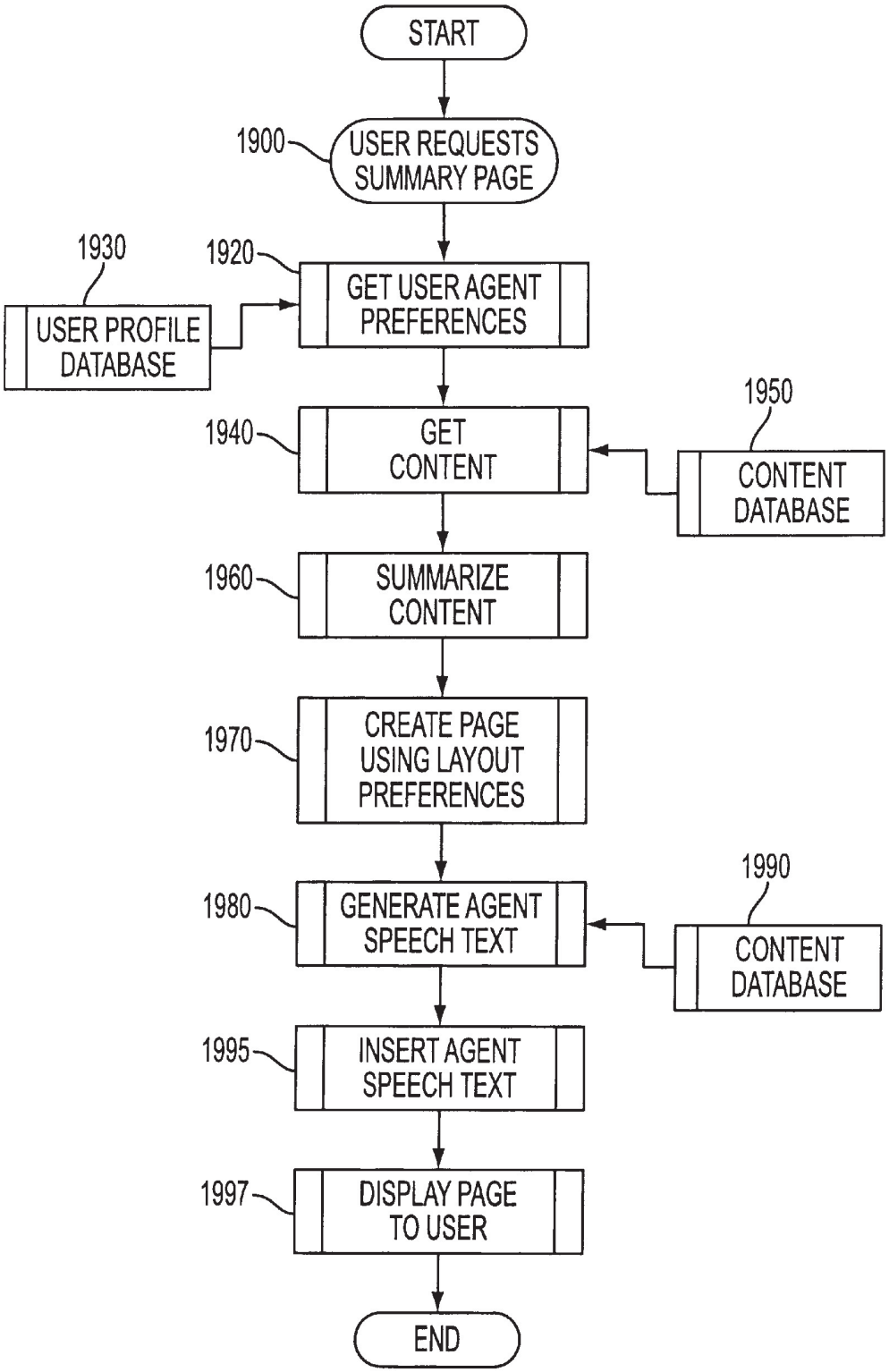

START
1900
USER REQUESTS SUMMARY PAGE
1930
1920
USER PROFILE DATABASE
GET USER AGENT PREFERENCES
1950
1940
GET CONTENT
CONTENT DATABASE
1960
SUMMARIZE CONTENT
1970
CREATE PAGE USING LAYOUT PREFERENCES
1990
1980
GENERATE AGENT SPEECH TEXT
CONTENT DATABASE
1995
INSERT AGENT SPEECH TEXT
1997
DISPLAY PAGE TO USER
END

FIG. 19

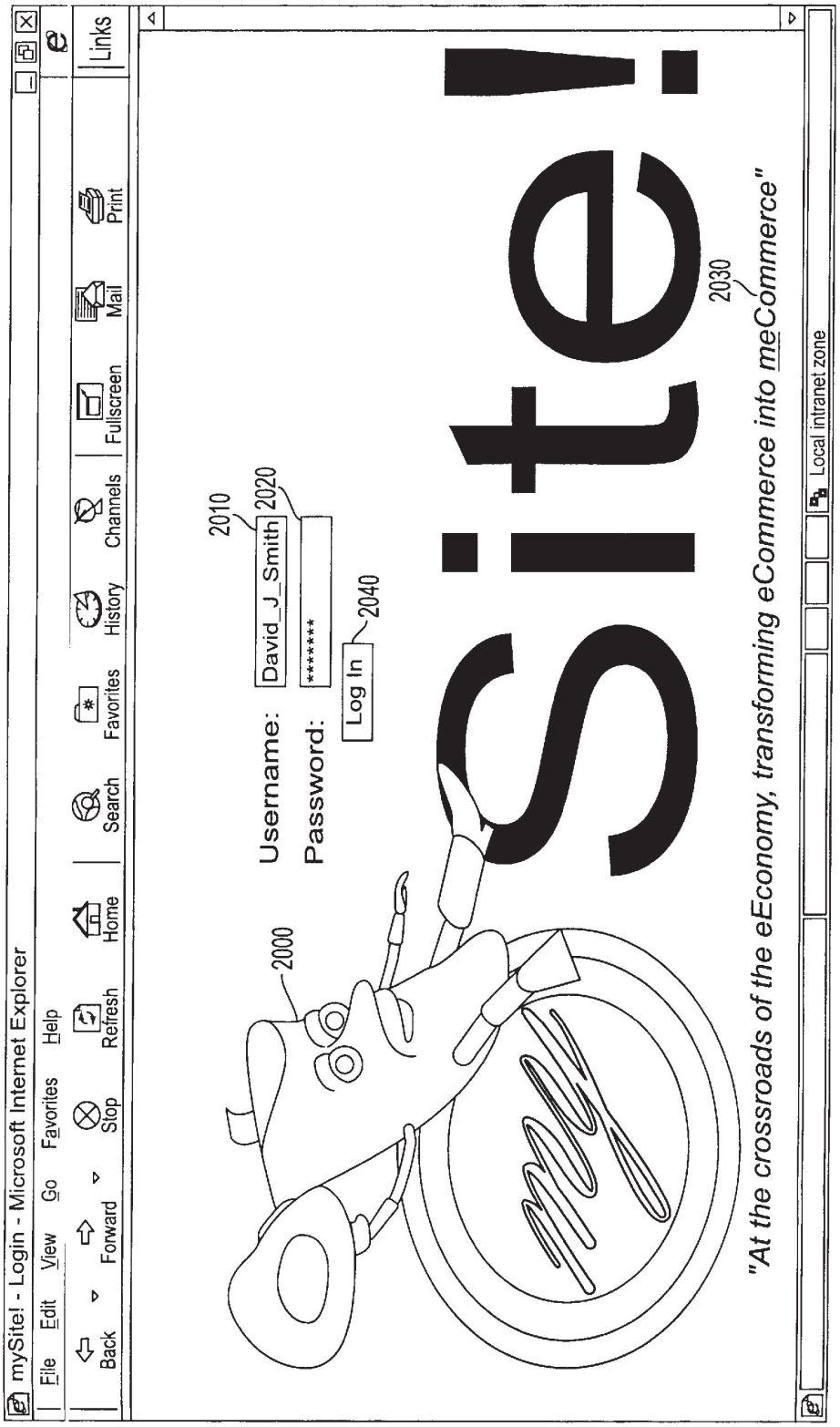

mySite! - Login - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Links
Local intranet zone
mySite!
"At the crossroads of the eEconomy, transforming eCommerce into meCommerce"
Username: David_J_Smith
Password: *******
Log In
2000
2010
2020
2030
2040

FIG. 20

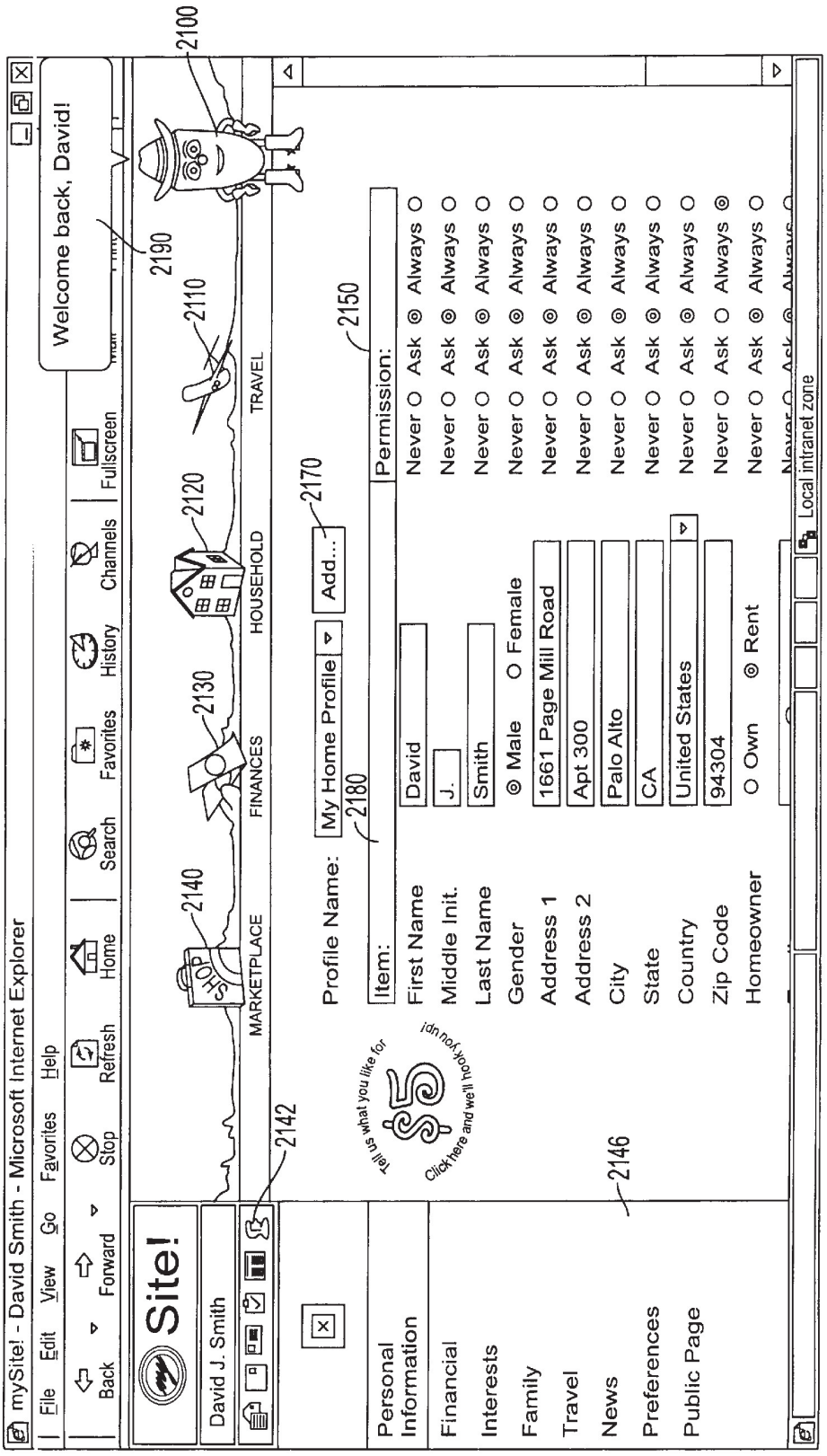

mySite! - David Smith - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen
Welcome back, David!
Site!
David J. Smith
MARKETPLACE FINANCES HOUSEHOLD TRAVEL
Profile Name: My Home Profile Add...
Item:
Permission:
Personal Information
Financial
Interests
Family
Travel
News
Preferences
Public Page
First Name David
Middle Init. J.
Last Name Smith
Gender Male Female
Address 1 1661 Page Mill Road
Address 2 Apt 300
City Palo Alto
State CA
Country United States
Zip Code 94304
Homeowner Own Rent
Never Ask Always
Local intranet zone
2100
2110
2120
2130
2140
2142
2146
2150
2170
2180
2190

FIG. 21

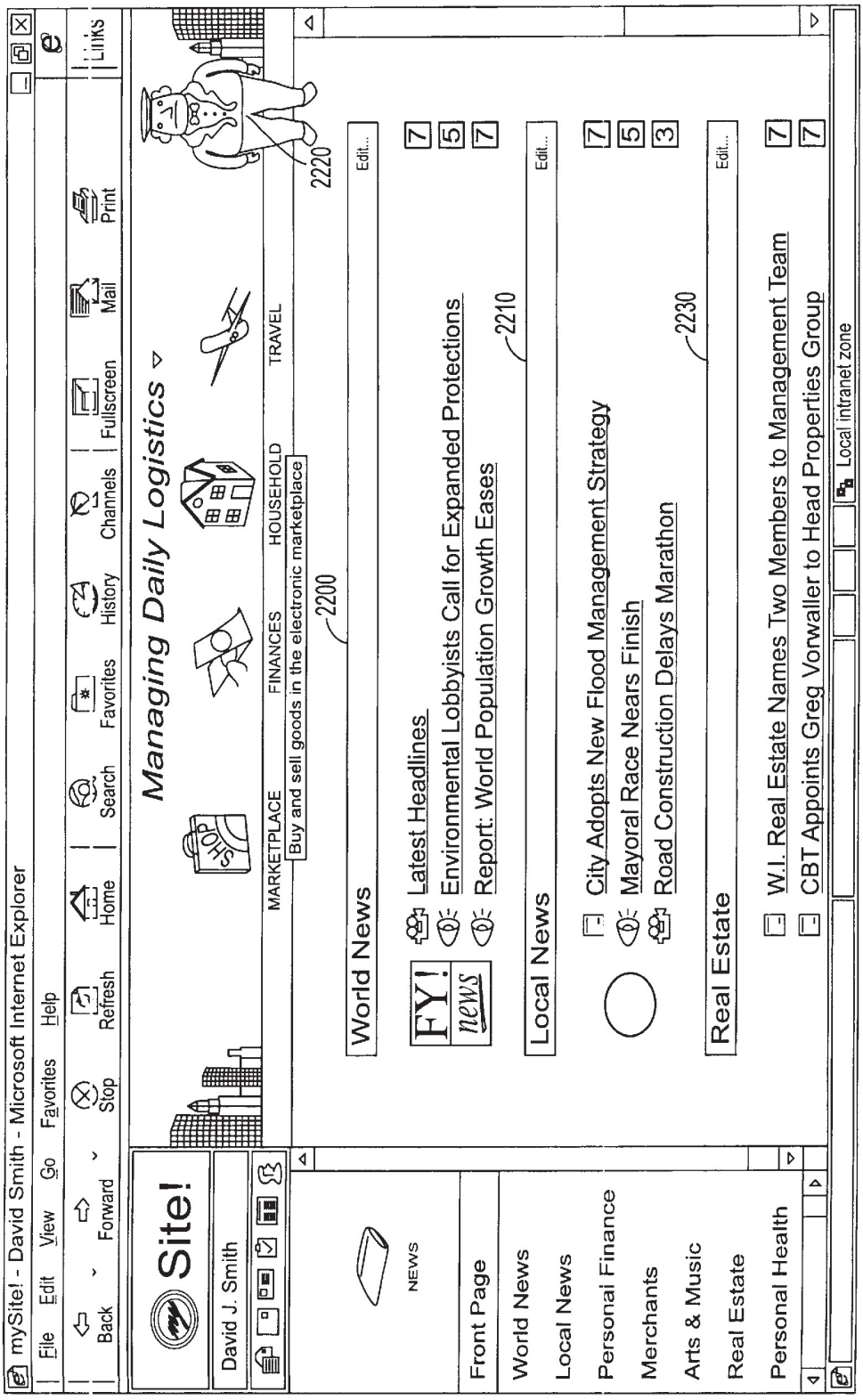

mySite! - David Smith - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen Mail Print Links
Site!
David J. Smith
Managing Daily Logistics
MARKETPLACE FINANCES HOUSEHOLD TRAVEL
Buy and sell goods in the electronic marketplace
2220
2200
NEWS
Front Page
World News
Local News
Personal Finance
Merchants
Arts & Music
Real Estate
Personal Health
World News
Edit...
FY! news
Latest Headlines
Environmental Lobbyists Call for Expanded Protections
Report: World Population Growth Eases
7
5
7
2210
Local News
Edit...
City Adopts New Flood Management Strategy
Mayoral Race Nears Finish
Road Construction Delays Marathon
7
5
3
2230
Real Estate
Edit...
W.I. Real Estate Names Two Members to Management Team
CBT Appoints Greg Vorwaller to Head Properties Group
7
7
Local intranet zone

FIG. 22

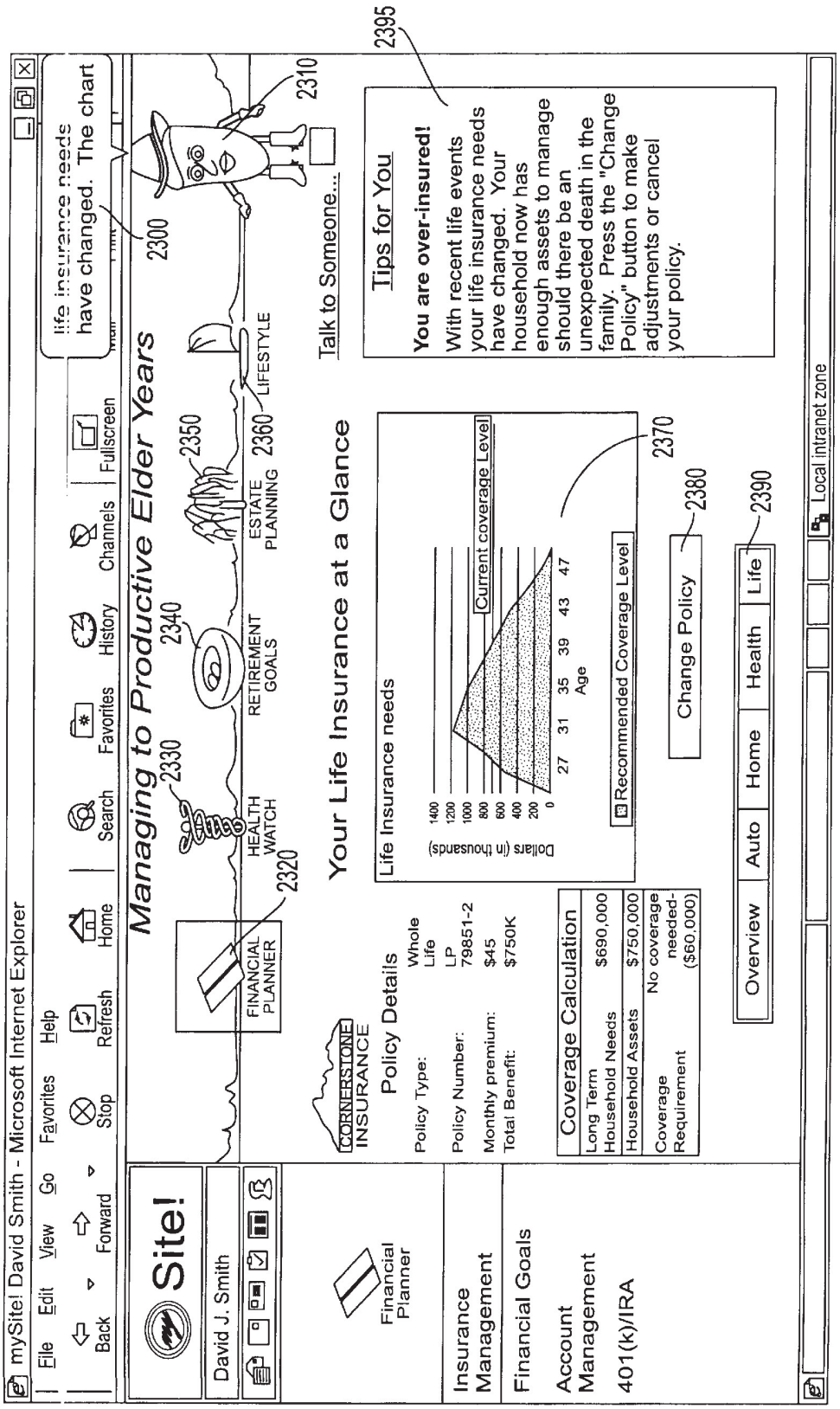

mySite! David Smith - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites History Channels Fullscreen
Site!
David J. Smith
Financial Planner
Insurance Management
Financial Goals
Account Management
401(k)/IRA
Managing to Productive Elder Years
2300
2310
2320
2330
2340
2350
2360
2370
2380
2390
2395
FINANCIAL PLANNER
HEALTH WATCH
RETIREMENT GOALS
ESTATE PLANNING
LIFESTYLE
life insurance needs have changed. The chart
Talk to Someone...
Tips for You
You are over-insured!
With recent life events your life insurance needs have changed. Your household now has enough assets to manage should there be an unexpected death in the family. Press the "Change Policy" button to make adjustments or cancel your policy.
CORNERSTONE INSURANCE
Your Life Insurance at a Glance
Policy Details
Policy Type: Whole Life
Policy Number: LP 79851-2
Monthly premium: $45
Total Benefit: $750K
Life Insurance needs
Dollars (in thousands)
1400
1200
1000
800
600
400
200
0
Current coverage Level
Recommended Coverage Level
27 31 35 39 43 47
Age
Coverage Calculation
Long Term $690,000
Household Needs
Household Assets $750,000
Coverage Requirement No coverage needed- ($60,000)
Change Policy
Overview Auto Home Health Life
Local intranet zone

FIG. 23

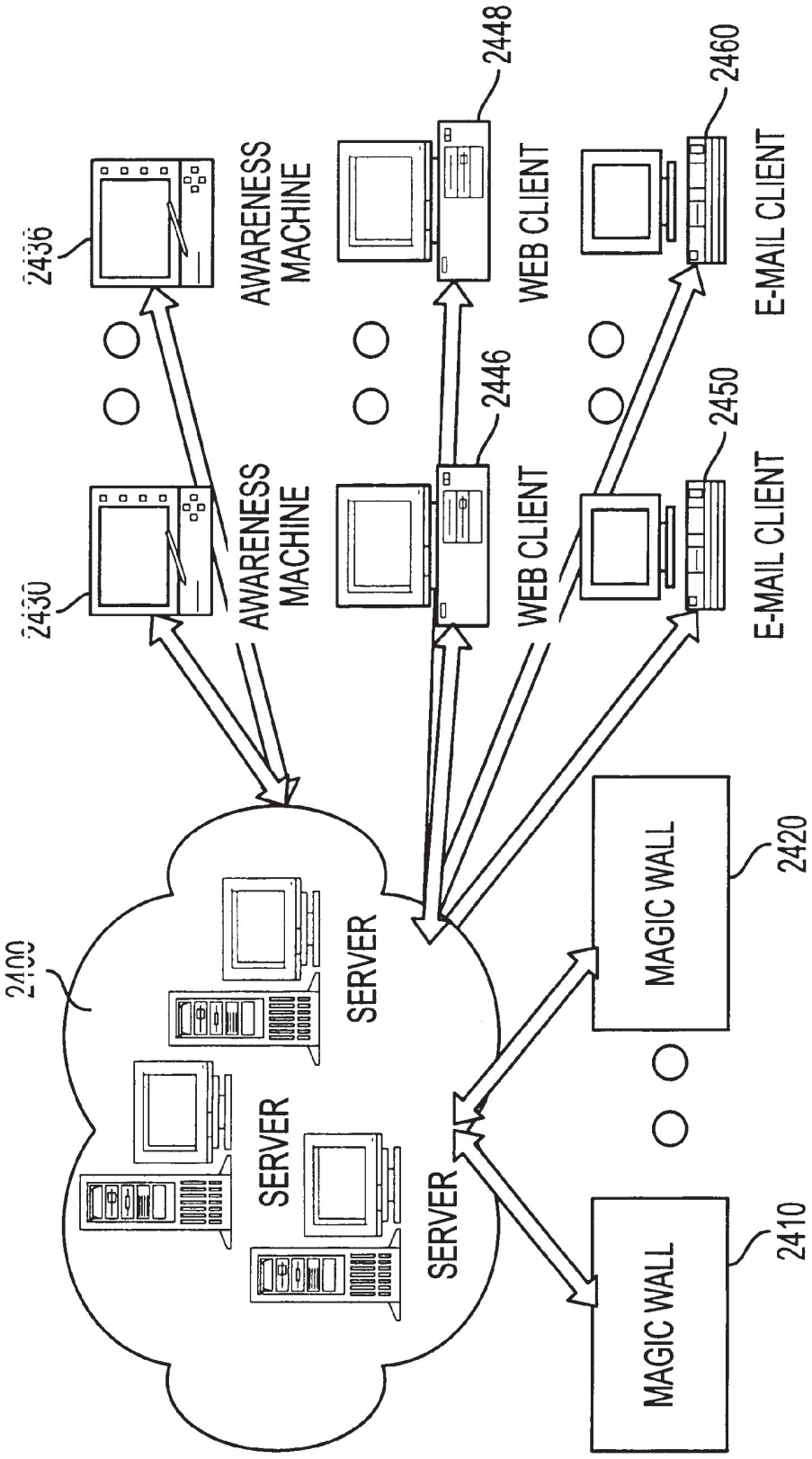

2436
AWARENESS MACHINE
2448
WEB CLIENT
2460
E-MAIL CLIENT
2430
AWARENESS MACHINE
2446
WEB CLIENT
2450
E-MAIL CLIENT
2400
SERVER
SERVER
SERVER
MAGIC WALL
2420
MAGIC WALL
2410

FIG. 24

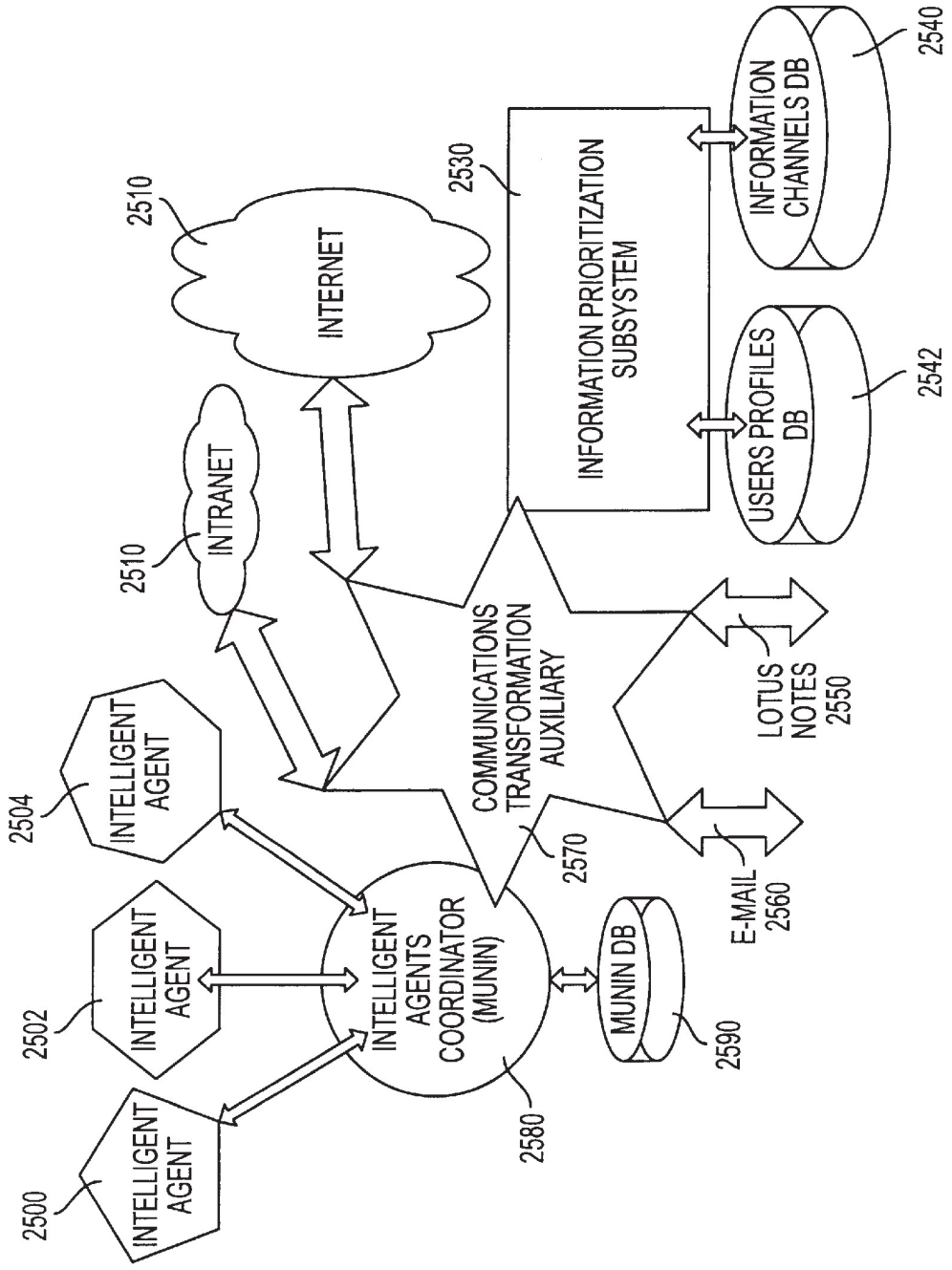

INTERNET
2510
INTRANET
2510
INTELLIGENT AGENT
2504
INTELLIGENT AGENT
2502
INTELLIGENT AGENT
2500
INTELLIGENT AGENTS COORDINATOR (MUNIN)
2580
MUNIN DB
2590
COMMUNICATIONS TRANSFORMATION AUXILIARY
2570
INFORMATION PRIORITIZATION SUBSYSTEM
2530
INFORMATION CHANNELS DB
2540
USERS PROFILES DB
2542
LOTUS NOTES
2550
E-MAIL
2560

FIG. 25

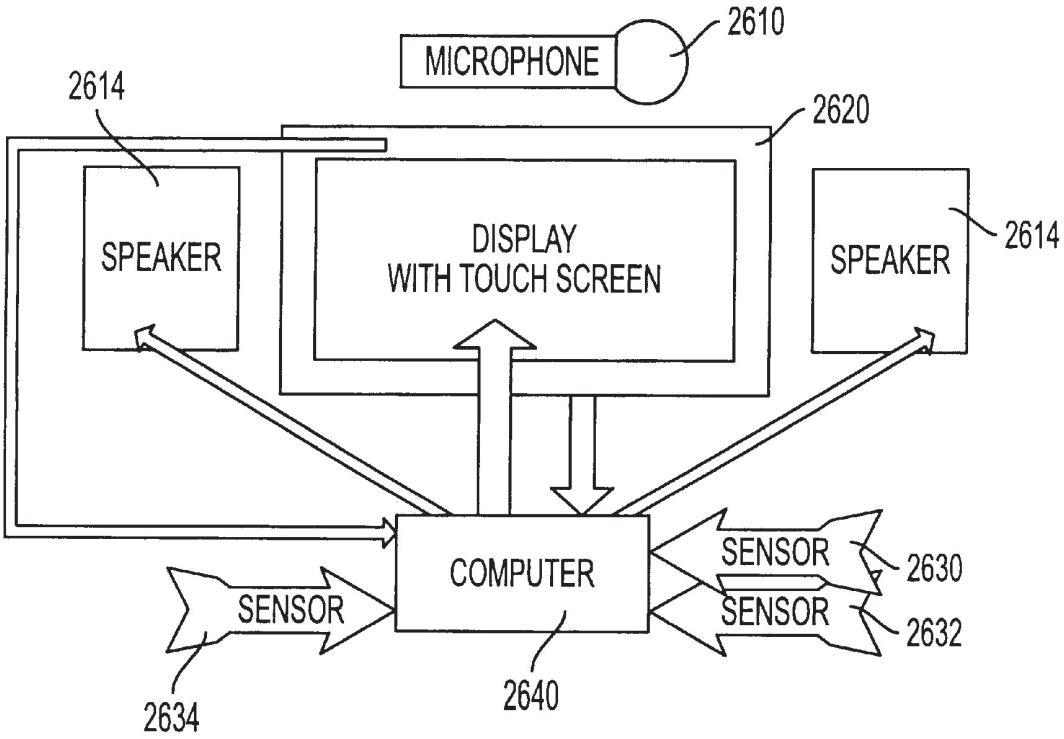

MICROPHONE
2610
2614
2620
SPEAKER
DISPLAY
WITH TOUCH SCREEN
SPEAKER
2614
COMPUTER
SENSOR
2630
SENSOR
2632
SENSOR
2634
2640

FIG. 26

SYSTEM, METHOD AND ARTICLE OF MANUFACTURE FOR A TUNED USER APPLICATION EXPERIENCE

FIELD OF THE INVENTION

The present invention relates to agent based systems and more particularly to an agent based system for providing a user interface that facilitates tuning of a particular application experience to the personal intentions of a particular user. Agent based technology has become increasingly important for use with applications designed to interact with a user for performing various computer based tasks in foreground and background modes. Agent software comprises computer programs that are set on behalf of users to perform routine, tedious and time-consuming tasks. To be useful to an individual user, an agent must be personalized to the individual user's goals, habits and preferences. Thus, there exists a substantial requirement for the agent to efficiently and effectively acquire user-specific knowledge from the user and utilize it to perform tasks on behalf of the user.

The concept of agency, or the user of agents, is well established. An agent is a person authorized by another person, typically referred to as a principal, to act on behalf of the principal. In this manner the principal empowers the agent to perform any of the tasks that the principal is unwilling or unable to perform. For example, an insurance agent may handle all of the insurance requirements for a principal, or a talent agent may act on behalf of a performer to arrange concert dates.

With the advent of the computer, a new domain for employing agents has arrived. Significant advances in the realm of expert systems enable computer programs to act on behalf of computer users to perform routine, tedious and other time-consuming tasks. These computer programs are referred to as "software agents."

Moreover, there has been a recent proliferation of computer and communication networks. These networks permit a user to access vast amounts of information and services without, essentially, any geographical boundaries. Thus, a software agent has a rich environment to perform a large number of tasks on behalf of a user. For example, it is now possible for an agent to make an airline reservation, purchase the ticket, and have the ticket delivered directly to a user. Similarly, an agent could scan the Internet and obtain information ranging from the latest sports or news to a particular graduate thesis in applied physics. Current solutions fail to apply agent technology to existing calendar technology to provide targeted acquisition of background information for a user's upcoming events.

SUMMARY OF THE INVENTION

According to a broad aspect of a preferred embodiment of the invention, a user's interface to a particular application program is modified by obtaining user profile information, parsing the content of the particular application, matching the parsed content to user profile information and presenting the parsed content matches in a format based on information in the user's profile on a display. An innovative pattern matching system is integrated into the match processing to provide improved matching capability.

DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, aspects and advantages are better understood from the following detailed description of a preferred embodiment of the invention with reference to the drawings, in which:

FIG. **1** is a block diagram of a representative hardware environment in accordance with a preferred embodiment;

FIG. **2** is a flowchart of the system in accordance with a preferred embodiment;

FIG. **3** is a flowchart of a parsing unit of the system in accordance with a preferred embodiment;

FIG. **4** is a flowchart for pattern matching in accordance with a preferred embodiment;

FIG. **5** is a flowchart for a search unit in accordance with a preferred embodiment;

FIG. **6** is a flowchart for overall system processing in accordance with a preferred embodiment;

FIG. **7** is a flowchart of topic processing in accordance with a preferred embodiment;

FIG. **8** is a flowchart of meeting record processing in accordance with a preferred embodiment;

FIG. **9** is a block diagram of process flow of a pocket bargain finder in accordance with a preferred embodiment;

FIGS. **10**A and **10**B are a block diagram and flowchart depicting the logic associated with creating a customized content web page in accordance with a preferred embodiment;

FIG. **11** is a flowchart depicting the detailed logic associated with retrieving user-centric content in accordance with a preferred embodiment;

FIG. **12** is a data model of a user profile in accordance with a preferred embodiment;

FIG. **13** is a persona data model in accordance with a preferred embodiment;

FIG. **14** is an intention data model in accordance with a preferred embodiment;

FIG. **15** is a flowchart of the processing for generating an agent's current statistics in accordance with a preferred embodiment;

FIG. **16** is a flowchart of the logic that determines the personalized product rating for a user in accordance with a preferred embodiment;

FIG. **17** is a flowchart of the logic for accessing the centrally stored profile in accordance with a preferred embodiment;

FIG. **18** is a flowchart of the interaction logic between a user and the integrator for a particular supplier in accordance with a preferred embodiment;

FIG. **19** is a flowchart of the agent processing for generating a verbal summary in accordance with a preferred embodiment;

FIG. **20** illustrates a display login in accordance with a preferred embodiment;

FIG. **21** illustrates a managing daily logistics display in accordance with a preferred embodiment;

FIG. **22** illustrates a user main display in accordance with a preferred embodiment;

FIG. **23** illustrates an agent interaction display in accordance with a preferred embodiment;

FIG. **24** is a block diagram of an active knowledge management system in accordance with a preferred embodiment;

FIG. **25** is a block diagram of a back end server in accordance with a preferred embodiment; and

FIG. **26** is a block diagram of a magic wall in accordance with a preferred embodiment.

DETAILED DESCRIPTION

A preferred embodiment of a system in accordance with the present invention is preferably practiced in the context of

a personal computer such as an IBM compatible personal computer, Apple Macintosh computer or UNIX based workstation. A representative hardware environment is depicted in FIG. **1**, which illustrates a typical hardware configuration of a workstation in accordance with a preferred embodiment having a central processing unit **110**, such as a microprocessor, and a number of other units interconnected via a system bus **112**. The workstation shown in FIG. **1** includes a Random Access Memory (RAM) **114**, Read Only Memory (ROM) **116**, an I/O adapter **118** for connecting peripheral devices such as disk storage units **120** to the bus **112**, a user interface adapter **122** for connecting a keyboard **124**, a mouse **126**, a speaker **128**, a microphone **132**, and/or other user interface devices such as a touch screen (not shown) to the bus **112**, communication adapter **134** for connecting the workstation to a communication network (e.g., a data processing network) and a display adapter **136** for connecting the bus **112** to a display device **138**. The workstation typically has resident thereon an operating system such as the Microsoft Windows NT or Windows/95 Operating System (OS), the IBM OS/2 operating system, the MAC OS, or UNIX operating system. Those skilled in the art will appreciate that the present invention may also be implemented on platforms and operating systems other than those mentioned.

A preferred embodiment is written using JAVA, C, and the C++ language and utilizes object oriented programming methodology. Object oriented programming (OOP) has become increasingly used to develop complex applications. As OOP moves toward the mainstream of software design and development, various software solutions require adaptation to make use of the benefits of OOP. A need exists for these principles of OOP to be applied to a messaging interface of an electronic messaging system such that a set of OOP classes and objects for the messaging interface can be provided.

OOP is a process of developing computer software using objects, including the steps of analyzing the problem, designing the system, and constructing the program. An object is a software package that contains both data and a collection of related structures and procedures. Since it contains both data and a collection of structures and procedures, it can be visualized as a self-sufficient component that does not require other additional structures, procedures or data to perform its specific task. OOP, therefore, views a computer program as a collection of largely autonomous components, called objects, each of which is responsible for a specific task. This concept of packaging data, structures, and procedures together in one component or module is called encapsulation.

In general, OOP components are reusable software modules which present an interface that conforms to an object model and which are accessed at run-time through a component integration architecture. A component integration architecture is a set of architecture mechanisms which allow software modules in different process spaces to utilize each others capabilities or functions. This is generally done by assuming a common component object model on which to build the architecture.

It is worthwhile to differentiate between an object and a class of objects at this point. An object is a single instance of the class of objects, which is often just called a class. A class of objects can be viewed as a blueprint, from which many objects can be formed.

OOP allows the programmer to create an object that is a part of another object. For example, the object representing a piston engine is said to have a composition-relationship with the object representing a piston. In reality, a piston engine comprises a piston, valves and many other components; the fact that a piston is an element of a piston engine can be logically and semantically represented in OOP by two objects.

OOP also allows creation of an object that "depends from" another object. If there are two objects, one representing a piston engine and the other representing a piston engine wherein the piston is made of ceramic, then the relationship between the two objects is not that of composition. A ceramic piston engine does not make up a piston engine. Rather it is merely one kind of piston engine that has one more limitation than the piston engine; its piston is made of ceramic. In this case, the object representing the ceramic piston engine is called a derived object, and it inherits all of the aspects of the object representing the piston engine and adds further limitation or detail to it. The object representing the ceramic piston engine "depends from" the object representing the piston engine. The relationship between these objects is called inheritance.

When the object or class representing the ceramic piston engine inherits all of the aspects of the objects representing the piston engine, it inherits the thermal characteristics of a standard piston defined in the piston engine class. However, the ceramic piston engine object overrides these ceramic specific thermal characteristics, which are typically different from those associated with a metal piston. It skips over the original and uses new functions related to ceramic pistons. Different kinds of piston engines have different characteristics, but may have the same underlying functions associated with it (e.g., how many pistons in the engine, ignition sequences, lubrication, etc.). To access each of these functions in any piston engine object, a programmer would call the same functions with the same names, but each type of piston engine may have different/overriding implementations of functions behind the same name. This ability to hide different implementations of a function behind the same name is called polymorphism and it greatly simplifies communication among objects.

With the concepts of composition-relationship, encapsulation, inheritance and polymorphism, an object can represent just about anything in the real world. In fact, our logical perception of the reality is the only limit on determining the kinds of things that can become objects in object-oriented software. Some typical categories are as follows:

- Objects can represent physical objects, such as automobiles in a traffic-flow simulation, electrical components in a circuit-design program, countries in an economics model, or aircraft in an air-traffic-control system.
- Objects can represent elements of the computer-user environment such as windows, menus or graphics objects.
- An object can represent an inventory, such as a personnel file or a table of the latitudes and longitudes of cities.
- An object can represent user-defined data types such as time, angles, and complex numbers, or points on the plane.

With this enormous capability of an object to represent just about any logically separable matters, OOP allows the software developer to design and implement a computer program that is a model of some aspects of reality, whether that reality is a physical entity, a process, a system, or a composition of matter. Since the object can represent anything, the software developer can create an object which can be used as a component in a larger software project in the future.