US007398209B2

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

(10) **Patent No.: US 7,398,209 B2**
(45) **Date of Patent: Jul. 8, 2008**

(54) **SYSTEMS AND METHODS FOR RESPONDING TO NATURAL LANGUAGE SPEECH UTTERANCE**

(75) Inventors: **Robert A. Kennewick**, Seattle, WA (US); **David Locke**, Redmond, WA (US); **Michael R. Kennewick, Sr.**, Bellevue, WA (US); **Michael R. Kennewick, Jr.**, Bellevue, WA (US); **Richard Kennewick**, Woodinville, WA (US); **Tom Freeman**, Mercer Island, WA (US)

(73) Assignee: **VoiceBox Technologies, Inc.**, Kirkland, WA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 890 days.

(21) Appl. No.: **10/452,147**

(22) Filed: **Jun. 3, 2003**

(65) **Prior Publication Data**

US 2004/0044516 A1 Mar. 4, 2004

Related U.S. Application Data

(60) Provisional application No. 60/384,388, filed on Jun. 3, 2002.

(51) **Int. Cl.**
G10L 15/18 (2006.01)
G10L 15/22 (2006.01)
G06F 17/20 (2006.01)

(52) **U.S. Cl.** **704/255**; 704/257; 704/275; 707/3

(58) **Field of Classification Search** 704/255, 704/257, 270, 270.1, 275; 707/3, 4, 5
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,155,743 A 10/1992 Jacobs 375/28

(Continued)

FOREIGN PATENT DOCUMENTS

WO WO 01/78065 10/2001

OTHER PUBLICATIONS

Reuters, "IBM to Enable Honda Drivers to Talk to Cars", Charles Schwab & Co., Inc., Jul. 28, 2002, 1 page.

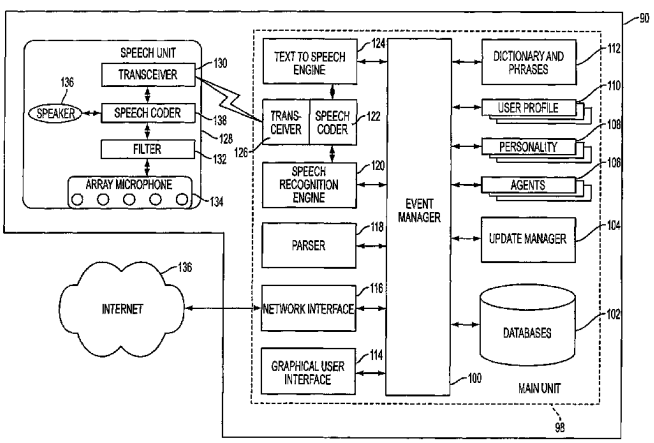
(Continued)

Primary Examiner—Martin Lerner
(74) *Attorney, Agent, or Firm*—Pillsbury Winthrop Shaw Pittman, LLP

(57) **ABSTRACT**

Systems and methods for receiving natural language queries and/or commands and execute the queries and/or commands. The systems and methods overcomes the deficiencies of prior art speech query and response systems through the application of a complete speech-based information query, retrieval, presentation and command environment. This environment makes significant use of context, prior information, domain knowledge, and user specific profile data to achieve a natural environment for one or more users making queries or commands in multiple domains. Through this integrated approach, a complete speech-based natural language query and response environment can be created. The systems and methods creates, stores and uses extensive personal profile information for each user, thereby improving the reliability of determining the context and presenting the expected results for a particular question or command.

14 Claims, 7 Drawing Sheets



MICROSOFT CORP.
EXHIBIT 1001

U.S. PATENT DOCUMENTS

5,386,556	A	1/1995	Hedin et al.	395/600
5,424,947	A	6/1995	Nagao et al.	364/419.08
5,471,318	A	11/1995	Ahuja et al.	358/400
5,475,733	A	12/1995	Eisdorfer et al.	379/52
5,499,289	A	3/1996	Bruno et al.	379/220
5,517,560	A	5/1996	Greenspan	379/114
5,533,108	A	7/1996	Harris et al.	379/201
5,537,436	A	7/1996	Bottoms et al.	375/222
5,539,744	A	7/1996	Chu et al.	370/60
5,557,667	A	9/1996	Bruno et al.	379/201
5,563,937	A	10/1996	Bruno et al.	379/201
5,590,039	A	12/1996	Ikeda et al.	395/759
5,633,922	A	5/1997	August et al.	379/220
5,675,629	A	10/1997	Raffel et al.	379/58
5,708,422	A	1/1998	Blonder et al.	340/825.34
5,721,938	A	2/1998	Stuckey	395/754
5,722,084	A	2/1998	Chakrin et al.	455/551
5,742,763	A	4/1998	Jones	395/200.3
5,752,052	A	5/1998	Richardson et al.	395/759
5,754,784	A	5/1998	Garland et al.	395/200.49
5,761,631	A	6/1998	Nasukawa	704/9
5,794,050	A	8/1998	Dahlgren et al.	395/708
5,799,276	A	8/1998	Komissarchik et al.	
5,802,510	A	9/1998	Jones	707/2
5,832,221	A	11/1998	Jones	375/200.36
5,878,385	A *	3/1999	Bralich et al.	704/9
5,878,386	A	3/1999	Coughlin	704/10
5,895,466	A	4/1999	Goldberg et al.	707/5
5,911,120	A	6/1999	Jarett et al.	455/417
5,918,222	A	6/1999	Fukui et al.	
5,926,784	A	7/1999	Richardson et al.	704/9
5,933,822	A	8/1999	Braden-Harder et al.	707/5
5,953,393	A	9/1999	Culbreth et al.	379/88.25
5,963,894	A	10/1999	Richardson et al.	704/9
5,963,940	A	10/1999	Liddy et al.	707/5
5,987,404	A	11/1999	Della Pietra et al.	704/9
5,991,721	A	11/1999	Asano et al.	704/257
5,995,119	A	11/1999	Cosatto et al.	345/473
6,009,382	A	12/1999	Martino et al.	704/1
6,014,559	A	1/2000	Amin	455/413
6,021,384	A	2/2000	Gorin et al.	704/1
6,044,347	A	3/2000	Abella et al.	704/272
6,049,602	A	4/2000	Foladare et al.	379/265
6,058,187	A	5/2000	Chen	380/21
6,078,886	A	6/2000	Dragosh et al.	704/270
6,081,774	A	6/2000	de Hita et al.	704/9
6,101,241	A	8/2000	Boyce et al.	379/88.01
6,119,087	A	9/2000	Kuhn et al.	704/270
6,134,235	A	10/2000	Goldman et al.	370/352
6,144,667	A	11/2000	Doshi et al.	370/401
6,160,883	A	12/2000	Jackson et al.	379/230
6,173,279	B1	1/2001	Levin et al.	707/5
6,175,858	B1	1/2001	Bulfer et al.	709/206
6,185,535	B1	2/2001	Hedin et al.	
6,192,110	B1	2/2001	Abella et al.	379/88.01
6,192,338	B1	2/2001	Haszto et al.	704/257
6,195,651	B1 *	2/2001	Handel et al.	707/2
6,208,972	B1	3/2001	Grant et al.	
6,219,346	B1	4/2001	Maxemchuk	370/338
6,219,643	B1	4/2001	Cohen et al.	704/257
6,233,556	B1	5/2001	Teunen et al.	704/250
6,233,559	B1	5/2001	Balakrishnan	
6,246,981	B1	6/2001	Papineni et al.	704/235
6,272,455	B1	8/2001	Hoshen et al.	704/1
6,292,767	B1	9/2001	Jackson et al.	704/1
6,314,402	B1	11/2001	Monaco et al.	704/275
6,366,886	B1	4/2002	Dragosh et al.	704/270.1
6,385,646	B1	5/2002	Brown et al.	709/217
6,393,428	B1	5/2002	Miller et al.	707/102
6,404,878	B1	6/2002	Jackson et al.	379/221.01
6,408,272	B1	6/2002	White et al.	
6,411,810	B1	6/2002	Maxemchuk	455/453
6,415,257	B1 *	7/2002	Junqua et al.	704/275
6,418,210	B1	7/2002	Sayko	379/142.15
6,420,975	B1	7/2002	DeLine et al.	
6,430,285	B1	8/2002	Bauer et al.	379/265.01
6,434,523	B1	8/2002	Monaco	704/257
6,434,524	B1	8/2002	Weber	704/257
6,442,522	B1	8/2002	Carberry et al.	704/257
6,446,114	B1	9/2002	Bulfer et al.	709/206
6,453,153	B1	9/2002	Bowker et al.	455/67.4
6,453,292	B2	9/2002	Ramaswamy et al.	704/235
6,456,711	B1	9/2002	Cheung et al.	379/265.09
6,466,654	B1 *	10/2002	Cooper et al.	379/88.01
6,498,797	B1	12/2002	Anerousis et al.	370/522
6,499,013	B1	12/2002	Weber	704/257
6,501,833	B2	12/2002	Phillips et al.	
6,501,834	B1	12/2002	Milewski et al.	379/93.24
6,523,061	B1	2/2003	Halverson et al.	709/202
6,532,444	B1	3/2003	Weber	704/257
6,539,348	B1	3/2003	Bond et al.	704/9
6,553,372	B1	4/2003	Brassell et al.	707/5
6,556,973	B1	4/2003	Lewin	
6,560,576	B1	5/2003	Cohen et al.	704/270
6,567,778	B1	5/2003	Chao Chang et al.	704/257
6,567,797	B1	5/2003	Schuetze et al.	707/2
6,570,555	B1	5/2003	Prevost et al.	345/156
6,570,964	B1	5/2003	Murveit et al.	379/67.1
6,574,624	B1	6/2003	Johnson et al.	707/5
6,594,257	B1	7/2003	Doshi et al.	370/352
6,604,077	B2	8/2003	Dragosh et al.	704/270.1
6,611,692	B2	8/2003	Raffel et al.	455/552
6,614,773	B1	9/2003	Maxemchuk	370/337
6,615,172	B1	9/2003	Bennett et al.	
6,629,066	B1	9/2003	Jackson et al.	704/9
6,650,747	B1	11/2003	Bala et al.	379/265.06
6,691,151	B1 *	2/2004	Cheyer et al.	709/202
6,721,706	B1 *	4/2004	Strubbe et al.	704/275
6,735,592	B1	5/2004	Neumann et al.	707/101
6,741,931	B1	5/2004	Kohut et al.	
6,757,718	B1	6/2004	Halverson et al.	709/218
6,901,366	B1 *	5/2005	Kuhn et al.	704/275
6,937,977	B2	8/2005	Gerson	
6,980,092	B2	12/2005	Turnbull et al.	
6,996,531	B2	2/2006	Korall et al.	
7,020,609	B2	3/2006	Thrift et al.	
7,027,975	B1	4/2006	Pazandak et al.	
7,062,488	B1 *	6/2006	Reisman	707/8
7,092,928	B1	8/2006	Elad et al.	
7,127,400	B2 *	10/2006	Koch	704/270.1
7,137,126	B1	11/2006	Coffman et al.	
7,289,606	B2	10/2007	Sibal et al.	379/52
2002/0143535	A1	10/2002	Kist et al.	
2002/0188602	A1	12/2002	Stubler et al.	707/3
2003/0112267	A1	6/2003	Belrose	345/728
2003/0182132	A1	9/2003	Niemoeller	
2004/0193420	A1	9/2004	Kennewick et al.	
2005/0021826	A1	1/2005	Kumar	

OTHER PUBLICATIONS

Lin, Bor-shen, et al., "A Distributed Architecture for Cooperative Spoken Dialogue Agents with Coherent Dialogue State and History", ASRU'99, 1999, 4 pages.

Kuhn, Thomas, et al., "Hybrid In-Car Speech Recognition for Mobile Multimedia Applications", Vehicular Technology Conference, IEEE, Jul. 1999, pp. 2009-2013.

Belvin, Robert, et al., "Development of the HRL Route Navigation Dialogue System", Proceedings of the First International Conference on Human Language Technology Research, San Diego, 2001, pp. 1-5.

Lind, R., et al., "The Network Vehicle—A Glimpse into the Future of Mobile Multi-Media", *IEEE Aerosp. Electron. Systems Magazine*, vol. 14, No. 9, Sep. 1999, pp. 27-32.

Zhao, Yilin, "Telematics: Safe and Fun Driving", *IEEE Intelligent Systems*, vol. 17, Issue 1, 2002, pp. 10-14.
Chai et al., "Mind: A Semantics-Based Multimodal Interpretation Framework for Conversational System", *Proceedings of the International Class Workshop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems*, Jun. 2002, pp. 37-46.
Cheyer et al., "Multimodal Maps: An Agent-Based Approach", *International Conference on Cooperative Multimodal Communication* (CMC/95), May 24-26, 1995, pp. 111-121.
Elio et al., "On Abstract Task Models and Conversation Policies" in Workshop on Specifying and Implementing Conversation Policies, *Autonomous Agents '99*, Seattle, 1999, 10 pages.
Turunen, "Adaptive Interaction Methods in Speech User Interfaces", Conference on Human Factors in Computing Systems, Seattle, Washington, 2001, pp. 91-92.

* cited by examiner

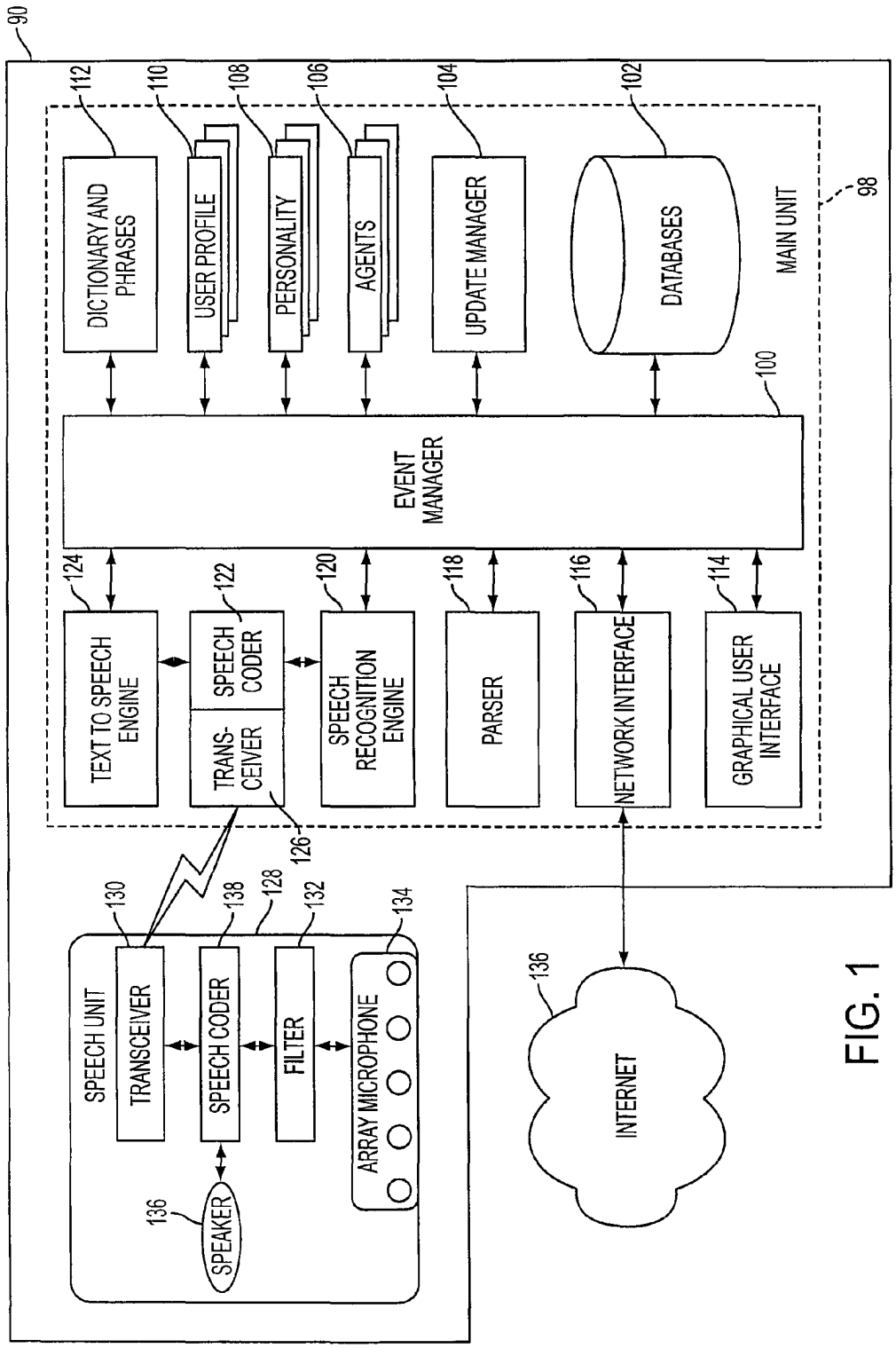

90
112
110
108
106
104
102
DICTIONARY AND PHRASES
USER PROFILE
PERSONALITY
AGENTS
UPDATE MANAGER
DATABASES
MAIN UNIT
98
100
EVENT MANAGER
124
122
120
118
116
114
TEXT TO SPEECH ENGINE
SPEECH CODER
TRANS-CEIVER
SPEECH RECOGNITION ENGINE
PARSER
NETWORK INTERFACE
GRAPHICAL USER INTERFACE
126
130
138
128
132
134
SPEECH UNIT
TRANSCEIVER
SPEECH CODER
FILTER
ARRAY MICROPHONE
136
SPEAKER
136
INTERNET

FIG. 1

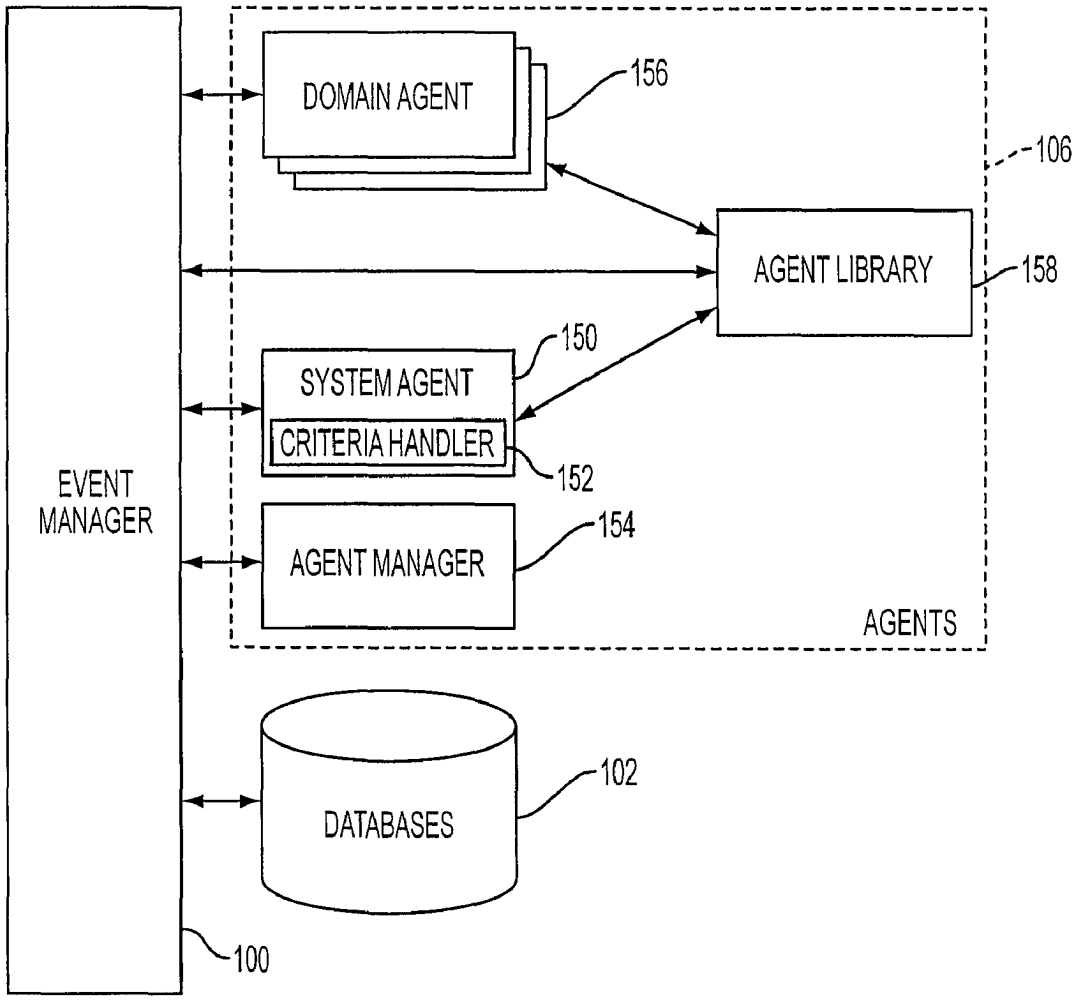

DOMAIN AGENT
156
106
AGENT LIBRARY
158
150
SYSTEM AGENT
CRITERIA HANDLER
152
EVENT MANAGER
154
AGENT MANAGER
AGENTS
102
DATABASES
100

FIG. 2

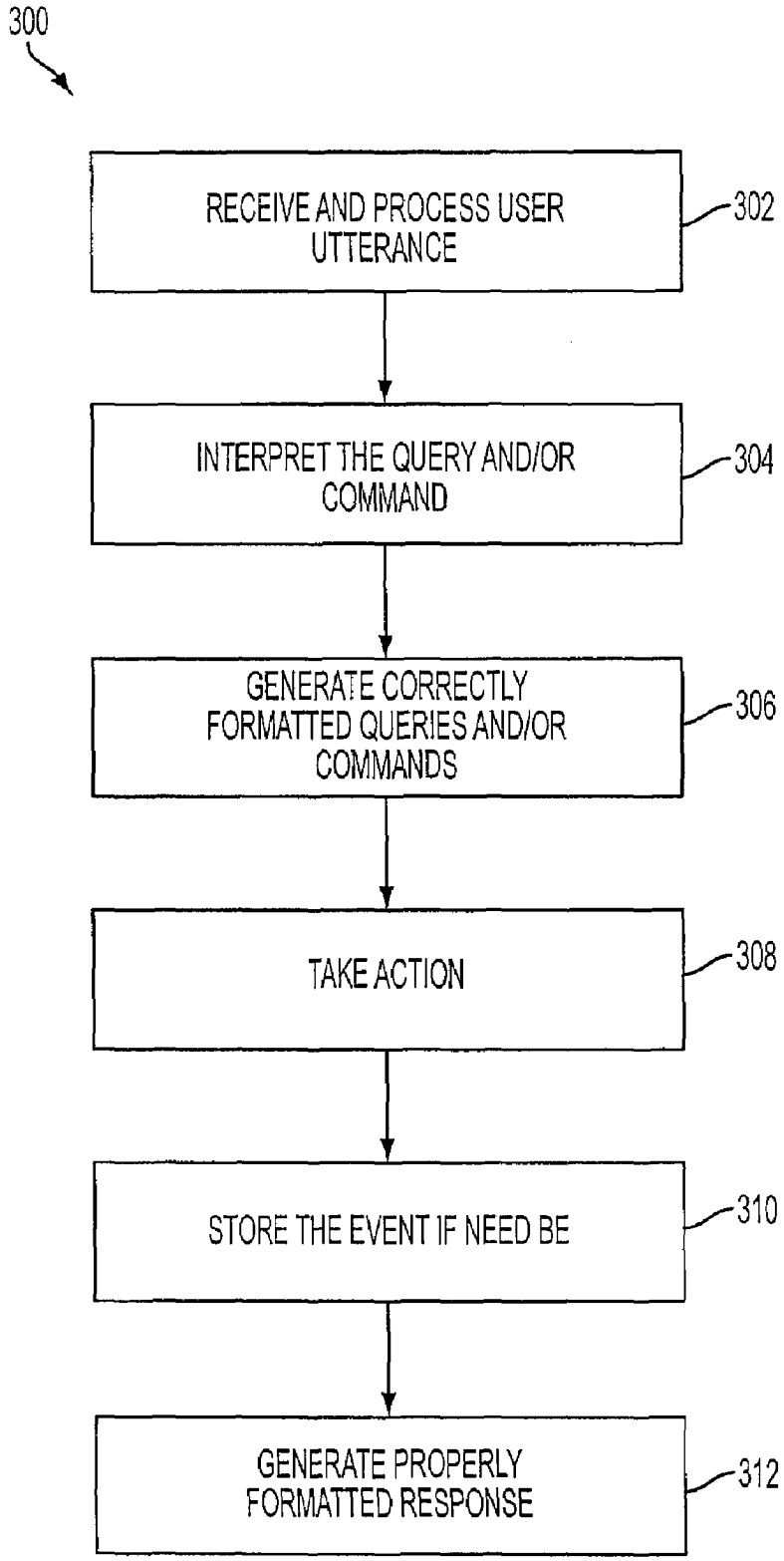

300
RECEIVE AND PROCESS USER UTTERANCE
302
INTERPRET THE QUERY AND/OR COMMAND
304
GENERATE CORRECTLY FORMATTED QUERIES AND/OR COMMANDS
306
TAKE ACTION
308
STORE THE EVENT IF NEED BE
310
GENERATE PROPERLY FORMATTED RESPONSE
312

FIG. 3

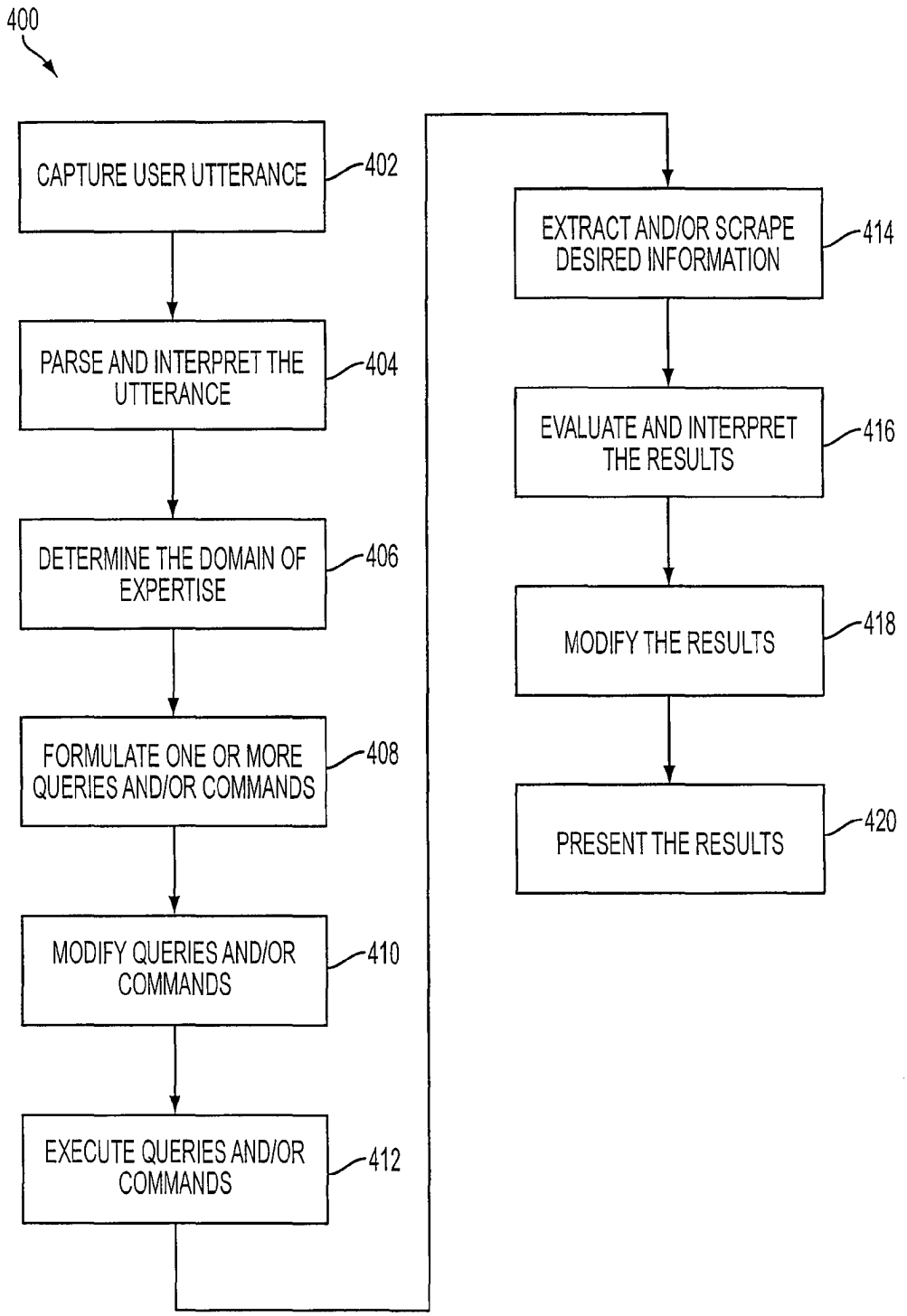

400
CAPTURE USER UTTERANCE
402
PARSE AND INTERPRET THE UTTERANCE
404
DETERMINE THE DOMAIN OF EXPERTISE
406
FORMULATE ONE OR MORE QUERIES AND/OR COMMANDS
408
MODIFY QUERIES AND/OR COMMANDS
410
EXECUTE QUERIES AND/OR COMMANDS
412
EXTRACT AND/OR SCRAPE DESIRED INFORMATION
414
EVALUATE AND INTERPRET THE RESULTS
416
MODIFY THE RESULTS
418
PRESENT THE RESULTS
420

FIG. 4A

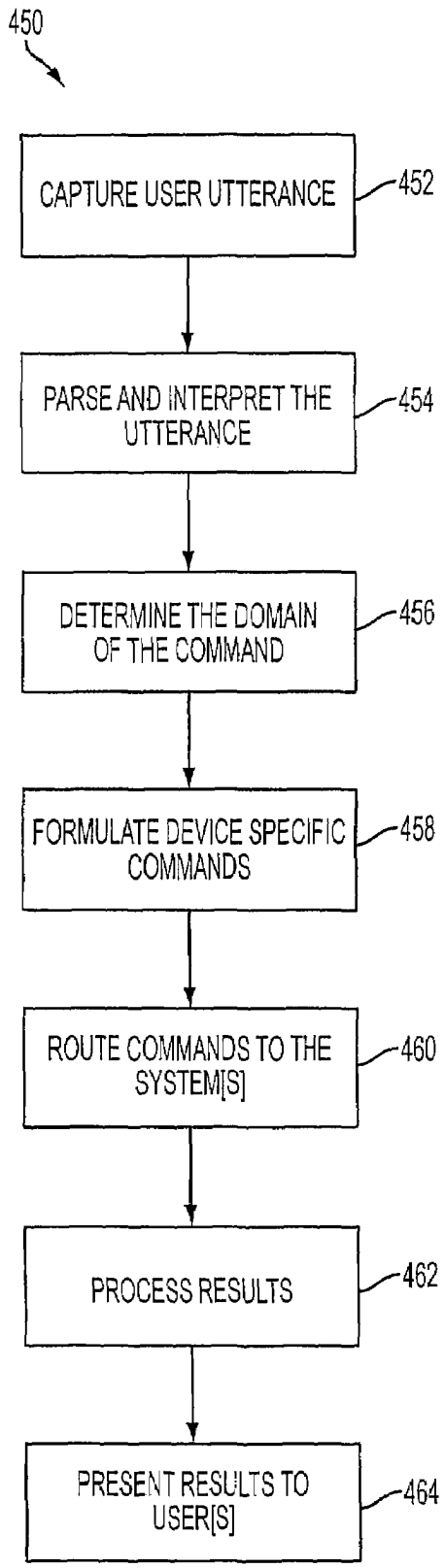

450
CAPTURE USER UTTERANCE
452
PARSE AND INTERPRET THE UTTERANCE
454
DETERMINE THE DOMAIN OF THE COMMAND
456
FORMULATE DEVICE SPECIFIC COMMANDS
458
ROUTE COMMANDS TO THE SYSTEM[S]
460
PROCESS RESULTS
462
PRESENT RESULTS TO USER[S]
464

FIG. 4B

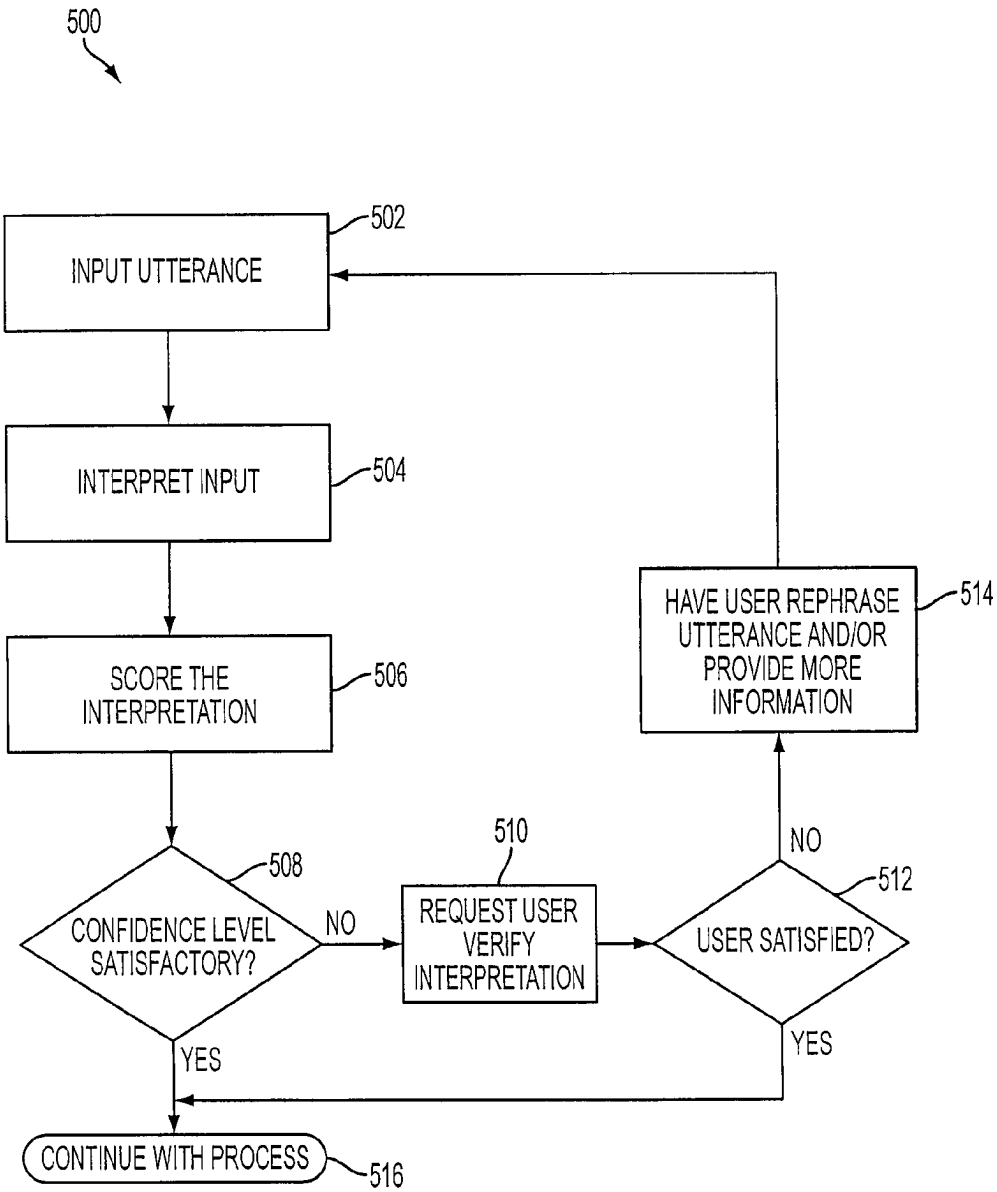

500
INPUT UTTERANCE
502
INTERPRET INPUT
504
SCORE THE INTERPRETATION
506
CONFIDENCE LEVEL SATISFACTORY?
508
NO
YES
REQUEST USER VERIFY INTERPRETATION
510
USER SATISFIED?
512
NO
YES
HAVE USER REPHRASE UTTERANCE AND/OR PROVIDE MORE INFORMATION
514
CONTINUE WITH PROCESS
516

FIG. 5

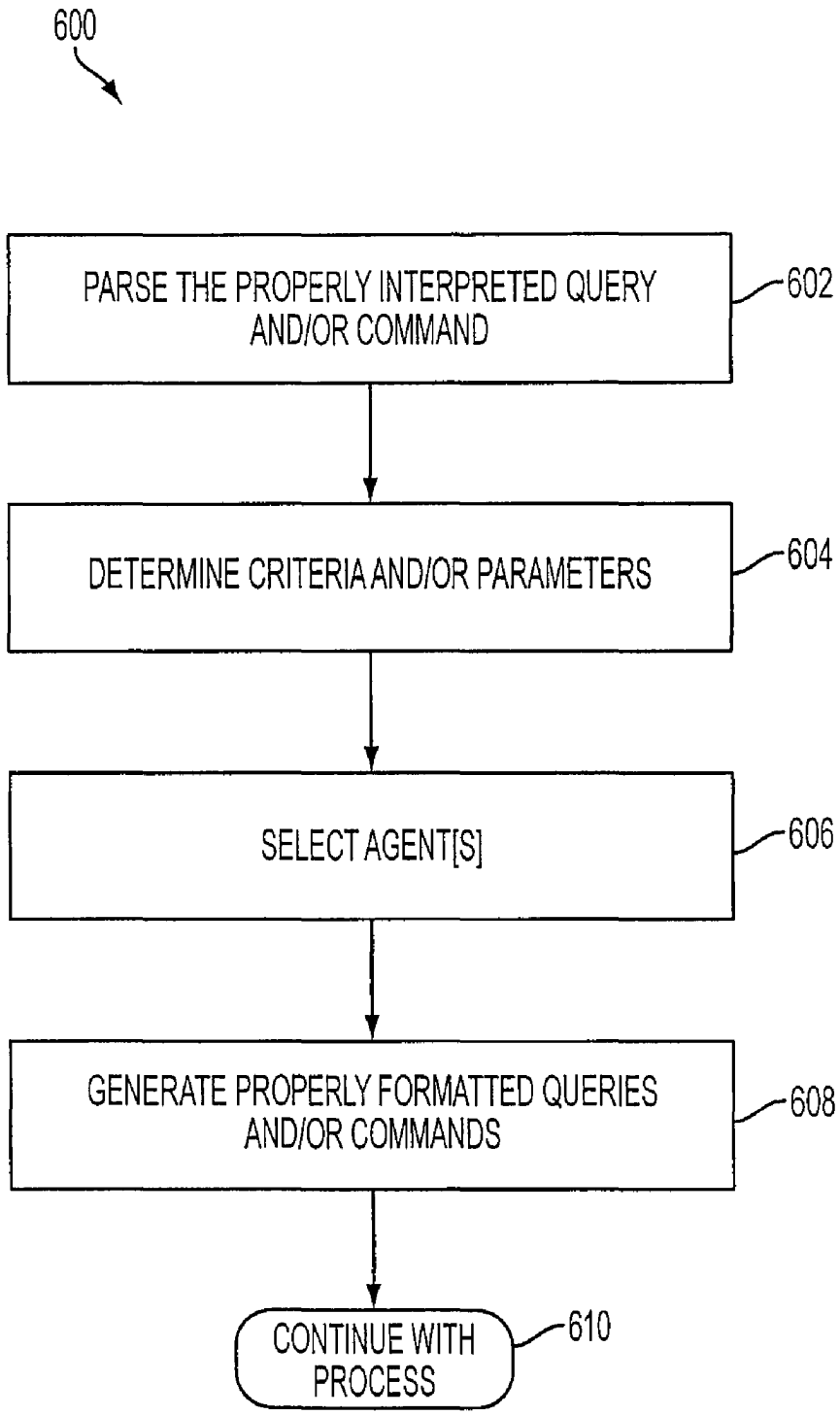

600
PARSE THE PROPERLY INTERPRETED QUERY AND/OR COMMAND
602
DETERMINE CRITERIA AND/OR PARAMETERS
604
SELECT AGENT[S]
606
GENERATE PROPERLY FORMATTED QUERIES AND/OR COMMANDS
608
CONTINUE WITH PROCESS
610

FIG. 6

SYSTEMS AND METHODS FOR RESPONDING TO NATURAL LANGUAGE SPEECH UTTERANCE

This application claims priority from U.S. Provisional Patent Application Ser. No. 60/384,388, filed Jun. 3, 2002, the disclosure of which is hereby incorporated by reference by its entirety.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to the retrieval of online information or processing of commands through a speech interface. More specifically, the invention is a fully integrated environment allowing users to submit natural language speech questions and commands. Information that is sought may be obtained from a wide range of disciplines, making local and network queries to obtain the information, and presenting results in a natural manner even in cases where the question asked or the responses received are incomplete, ambiguous or subjective. The invention may further allow users to control devices and systems either locally or remotely using natural language speech commands.

2. Background of the Related Art

A machine's ability to communicate with humans in a natural manner remains a difficult problem. Cognitive research on human interaction shows that a person asking a question or giving a command typically relies heavily on context and the domain knowledge of the person answering. On the other hand, machine-based queries (e.g., questions, commands, requests, and/or other types of communications) may be highly structured and are not inherently natural to the human user. Thus, human questions and machine processing of queries may be fundamentally incompatible. Yet the ability to allow a person to make natural language speech-based queries remains a desirable goal.

Much work covering multiple methods has been done in the fields of natural language processing and speech recognition. Speech recognition has steadily improved in accuracy and today is successfully used in a wide range of applications. Natural language processing has previously been applied to the parsing of speech queries. Yet, current systems are generally unable to provide a complete environment for users to make natural language speech queries and receive natural sounding responses. There remains a number of significant barriers to creation of a complete natural language speech-based query and response environment.

SUMMARY OF THE INVENTION

According to an aspect of the invention, one object of the invention is to overcome these and other drawbacks of prior systems.

According to one aspect of the invention, in order for machines (e.g., computer devices) to properly respond to a question and/or a command that is in a natural language form, machine processable queries and commands may be formulated after the natural language question and/or command has been parsed and interpreted.

In another aspect of the invention, systems are provided which may overcome the deficiencies of prior systems through the application of a complete speech-based information query, retrieval, presentation and command environment. This environment makes maximum use of context, prior information, domain knowledge, and user specific profile data to achieve a natural environment for one or more users

making queries or commands in multiple domains. Through this integrated approach, a complete speech-based natural language query and response environment may be created. Further, at each step in the process, accommodation may be made for full or partial failure and graceful recovery. The robustness to partial failure is achieved through the use of probabilistic and fuzzy reasoning at several stages of the process. This robustness to partial failure promotes the feeling of a natural response to questions and commands.

According to another aspect of the invention, the system comprises a speech unit interface device that receives spoken natural language queries, commands and/or other utterances from a user, and a computer device or system that receives input from the speech unit and processes the input (e.g., retrieves information responsive to the query, takes action consistent with the command and performs other functions as detailed herein). The system may further generate a natural language speech response in response to the user and/or generate a natural language speech message without any prompting from a user such as an alert message.

According to another aspect of the invention, the speech unit may be incorporated into the computer device or system, or may be a separate structure. If a separate structure, the speech unit may be connected to the computer device via a wired or wireless connection. If a wireless connection, a base unit may be connected to the computer, internally or externally, to communicate with the speech unit.

According to another aspect of the invention, the computer device or system may comprise a stand alone or networked PC, a PDA or other portable computer device, or other computer device or system. For convenience, these and other computer alternatives shall be referred to simply as a computer. One aspect of the invention comprises software that is installed onto the computer, where the software comprises one or more of the following modules: a speech recognition module for capturing the user input; a parser for parsing the input, a text to speech engine module for converting text to speech; a network interface for enabling the computer to interface with one or more networks; a graphical user interface module; and an event manager for managing events. Preferably; the event manager is in communication with a dictionary and phrases module, a user profile module that enables user profiles to be created, modified, and accessed, a personality module that enables various personalities to be created and used, an agent module, an update manager, and one or more databases.

According to another aspect of the invention domain specific behavior and information is organized into agents. Agents are autonomous executables that receive, process and respond to user questions, queries and commands. The agents provide complete, convenient and re-distributable packages or modules of functionality, typically for a specific domain or application. Agents can be complete packages of executable code, scripts, links to information, and other forms of communication data, required to provide a specific package of functionality, usually in a specific domain. In other words, an agent includes everything that is needed to extend the functionality of the invention to a new domain. Further, agents and their associated data can be updated remotely over a network as new behavior is added or new information becomes available. Agents can use system resources and the services of other, typically more specialized, agents. Agents can be distributed and redistributed in a number of ways including on removable storage media, transfer over networks or attached to emails and other messages. An update manager is used to add new agents to the system or update existing agents.

According to another aspect of the invention, license management capabilities allowing the sale of agents by third parties to one or more users on a one time or subscription basis may be provided. In addition, users with particular expertise may create agents, update existing agents by adding new behaviors and information, and making these agents for other users.

In order to enhance the natural query and response environment, the system may format results in a manner enhancing the understandability to the user. The optimal formatting and presentation depends on the context of the questions, the contents of the response being presented, the history of the interaction with the user, the user's preferences and interests and the nature of the domain. Information presented in a rigid, highly formatted, or structured manner seems unnatural to most people. Thus a system simulates some aspects of human "personality". In some cases, the presentation of the response and the terms used are randomized so they do not appear rigidly formatted or mechanical. The use of other simulated personality characteristics is also desirable. For example, a response that may be upsetting to the user is best presented in a sympathetic manner. Finally, the results of many queries are long text strings, lists, tables or other lengthy sets of data. Natural presentation of this type of information presents particular challenges. Simply reading the long response is generally not preferred. Therefore the system parses the most important sections from the response and, at least initially, only reports these. Determining what parts of a long response are presented depends on the context of the questions, the contents of the response being presented, the history of the interaction with the user, the user's preferences and interests and the nature of the domain. At the same time the system gives the user interactive control over what information and how much information is being presented, to stop the response all together, or to take other actions.

According to another aspect of the invention, the system may process and respond to both questions or queries and commands. Keywords or context are used to determine if the user's utterance is a command or query. Some utterances can include both aspects of a command and a query or question. For example, a user may say, "record my favorite TV program". A query is required to determine the name, the channel, and time for the users favorite TV program. A command must be executed to set a video recorder to capture this program.

For each user utterance including a question or query or set of questions or queries, the system may perform multiple steps possibly including:

capture of the user's question or query through accurate speech recognition operating in a variety of real-world environments;

parse and interpret the question or query;

determine the domain of expertise required and context, invoking the proper resources, including agents;

formulate one or more queries to one or more local and/or network data sources or sending appropriate commands to local or remote devices or the system itself;

perform required formatting, variable substitutions and transformations to modify the queries to a form most likely to yield desired results from the available sources;

execute the multiple queries or commands in an asynchronous manner and dealing gracefully with failures;

extract or scrape the desired information from the one or more results, which may be returned in any one of a number of different formats;

evaluate and interpret the results, including processing of errors, gathered and combine them into a single best result judged to be "best" even if the results are ambiguous, incomplete, or conflicting;

perform required formatting, variable substitutions and transformations to modify the results to a form most easily understood by the user; and

present the compound result, through a text to speech engine, to the user in a useful and expected manner.

The above steps preferably are performed while accounting for the domain of expertise required, the context for the question or command, domain specific information, the history of the user's interaction, user preferences, information sources or commands available, and responses obtained from the sources. At each stage probabilistic or fuzzy set decision and matching methods can be applied to deal with inconsistent, ambiguous, conflicting and incomplete information or responses. In addition, the use of asynchronous queries that may result in rapid and graceful failure of some queries or commands may allow the system to robustly return results quickly and in a manner that seems natural to the user.

Many everyday questions are inherently subjective and result in answers that are a matter of option or consensus as much as fact. Such questions are often ad hoc in their nature, as well. The system, according to another aspect of the invention, may use probabilistic and fuzzy set decision and matching methods to first identify the subjective nature of the question and to evaluate a range of possible answers, selecting the one answer or few answers that best represent the type of result desired by the user. The context and expected results from a particular question may be highly dependent on the individual asking the question. Therefore, the system may create, store and use extensive personal profile information for each user. Information in the profile may be added and updated automatically as the user uses the system or may be manually added or updated by the user or by others. Domain specific agents may collect, store and use specific profile information, as may be required for optimal operations. Users can create commands for regularly used reports, automatically generated alerts, and other queries and for the formatting and presentation of results. The system may use profile data in interpreting questions, formulating queries, interpreting results of queries and presenting answers to the user. Examples of information in a user profile includes, history of questions asked, session histories, formatting and presentation preferences, special word spelling, terms of interest, special data sources of interest, age, sex, education, location or address, place of business, type of business, investments, hobbies, sports interests, news interests and other profile data.

To create a natural question and response environment, the system according to one aspect of the invention, may attempt to provide rapid responses without requiring any additional information. The system may determine the mostly likely context or domain for a user's question or command, for example, by using a real-time scoring system or other technique. Based on this determination, the system may invoke the correct agent. The agent makes one or more queries and rapidly returns a formatted response. Thus, a user can receive a direct response to a set of questions, each with a different response or context. In some cases, the available information, including the query results, may not adequately answer the question. In such a situation, the user may be asked one or more questions to resolve the ambiguity. Additional queries may then be made before an adequate response is made. In these cases, the system may use context information, user profile information and domain specific information to minimize the interaction with the user required to deliver a response.

If the confidence level of the domain or context score is not high enough to ensure a reliable response, the system can request that the user verify the question or command is correctly understood. In general, the question may be phrased to indicate the context of the question including all criteria or parameters. If the user confirms that the question is correct the system may proceed to produce a response. Otherwise, either the user can rephrase the original question, perhaps adding additional information to remove ambiguity, or the system may ask one or more questions to attempt to resolve the ambiguity or other actions may taken.

Although the system, according to another aspect of the invention, is intended to be able to accept most any natural language question or command, ambiguity can still be a problem. To assist users formulate concise questions and commands the system may support a voice query language. The language may help users clearly specify the keywords or contexts of the question or command along with the parameters or criteria. The system may provide built in training capabilities to help the user learn the best methods to formulate their questions and commands.

In order to make the responses to user's questions and commands seem more natural, the system may employ one or more dynamically invokeable personalities. Personalities have specific characteristics, which simulate the behavioral characteristics of real humans. Examples of these characteristics include, sympathy, irritation, and helpfulness. The personality also randomizes aspects of responses, just as a real human would do. This behavior includes randomization of terms used and the order of presentation of information. Characteristics of the personality are invoked using probabilistic or fuzzy set decision and matching methods, and using criteria including the context for the question, the history of the user's interaction, user preferences, information sources available, responses obtained from the sources.

According to another aspect of the invention, special procedures to present information that is in the form of long text strings, tables, lists or other long response sets may be employed. Simply presenting a long set of information in an ordered manner is not natural nor what most users have in mind. The system, using for example, probabilistic or fuzzy set matching methods, may extract the most relevant information to the user and presents these subsets first. Further the system may provide commands allowing the user to skip through the list, find keywords or key information in the list or stop processing the list altogether.

In addition to the ability to process user questions and commands, the invention may provide local or remote control functions for the system or for other devices. Users may initiate commands locally or remotely. Several techniques for executing remote operations are possible, for example, the use of a telephone or other audio connection. Other remote command techniques may be used. The processing of these commands is performed in a nearly identical manner to a question. One difference being that the result of the command is generally an action rather than a response. Examples of control operations supported by the invention include, finding and playing music, film, games or other entertainment media, control of the appliances or other electronic or electro-mechanical devices, security systems control, audio, text and video conferencing (including Voice over IP conferencing), operating integrated messaging environments and other control operations. By using an optional analog or digital telephone interface or a data network interface, the invention can provide remote control capability of the computer or and auxiliary devices connected to the computer. For example, if the computer is connected via an interface to one or more auxiliary devices, a remote user can use this capability to remotely control or program the operation of the auxiliary device. For example, this enables recording of a television program, checking the status of a home security system, checking voice mail messages, hearing and responding to email messages, and much more. For each user command utterance the system may execute a number of steps possibly including:

- capture the user's command through accurate speech recognition operating in a variety of real-world environments;
- parse and interpret the command;
- determine the domain for the command and context, invoking the proper resources, including agents;
- formulate device specific commands for the system or external devices;
- route command to system, or external devices, including external devices connected to data networks;
- receive and process results of command, including errors; and
- optionally, provide response to user indicating the success or failure of the command, and possibly including state information.

The system may be used in conjunction with a wide range of platform environments. These environments include, as a peripheral to a PC, a portable or wearable computer or other computer, embedded in a PC or other computer, on a personal digital assistant (e.g., a PDA with wireless networking capability, in a specialized device such as a bar code scanner or payment terminal, on a wireless telephone, or other platforms). If the invention is used with a mobile or portable device that has position location capability, the location data can be used by the invention as part of the context for user questions. A user may use the system on one or more devices. In this case history and profile information for the user may be synchronized between the multiple devices on a periodic basis or other basis.

According to another aspect of the invention, the system may be deployed in a network of devices using common base of agents, data, information, user profiles and histories. Each user can then interact with, and receive the same services and applications at any location equipped with the required device on the network. For example, multiple devices on which the invention is deployed, and connected to a network, can be placed at different locations throughout a home, place of business or other location. In such a case, the system may use the location of the particular device addressed by the user as part of the context for the questions asked.

Multiple users may use the system at different times or during interleaved or overlapping sessions. The system recognizes the user either by name or voice. Once the user is recognized, the system invokes the correct profile. If multiple users are addressing the system in overlapping or interleaved sessions, the system determines which user is stating each question or command and applies the correct profile and context. For applications requiring security, the user is verified, typically by using voiceprint matching or requesting a password or pass-phrase from the user. When multiple users are engaged in interleaved sessions, the system gracefully resolves conflicts using a probabilistic or fuzzy set decision method. This process simulates the manner in which a human would address multiple questions. For example, the system may answer short questions first at times, while answering questions in the order received at other times.

Given the desire for domain specific behavior, user specific behavior and domain specific information, the system, according to another aspect of the invention, may allow both

users and content providers to extend the system capabilities, add data to local data sources, and add references to network data sources. To allow coverage of the widest possible range of topics, the system allows third party content developers to develop, distribute and sell specialized or domain specific system programs and information. Content is created though creation of new agents, scripting existing agents, adding new data to agents or databases and adding or modifying links to information sources. Distribution of this information is sensitive to the users interests and use history and to their willingness to pay or not.

According to another aspect of the invention, the system may allow users to disseminate their knowledge without programming. The system may include mechanisms that allow users to post and distribute agents and information in their particular areas of expertise, to improve system capability. Further, users can extend the system and configure it to their own preferences, add information to their profile to define new questions or queries, extend and modify existing questions and queries, add new data sources, update data sources, set preferences and specify presentation parameters for results.

Since the invention may operate in many environments, including environments with background noise, point noise sources and people holding conversations, filtering of speech input may be performed. The invention may use, for example, either one-dimensional or two-dimensional array microphones to receive human speech. The array microphones can be use fixed or employ dynamic beam forming techniques. The array pattern may be adjusted to maximize gain in the direction of the user and to null point noise sources. Speech received at the microphones may then be processed with analog or digital filters to optimize the bandwidth, cancel echoes, and notch-out narrow band noise sources. Following filtering, the system may use variable rate sampling to maximize the fidelity of the encoded speech, while minimizing required bandwidth. This procedure may be particularly useful in cases where the encoded speech is transmitted over a wireless network or link.

Some example applications for the invention include but are not limited to, the following:

- white pages and yellow pages lookups to find email addresses, telephone numbers, street addresses and other information for businesses and individuals;
- personal address book, calendars and reminders for each user;
- automatic telephone dialing, reading and sending emails and pages by voice and other communications control functions;
- map, location and direction applications;
- movie or other entertainment locator, review information and ticket purchasing;
- television, radio or other home entertainment schedule, review information and device control from a local or remote user;
- weather information for the local area or other locations;
- stock and other investment information including, prices, company reports, profiles, company information, business news stories, company reports, analysis, price alerts, news alerts, portfolio reports, portfolio plans, etc.;
- flight or other scheduled transportation information and ticketing;
- reservations for hotels, rental cars and other travel services;
- local, national and international news information including headlines of interest by subject or location, story summaries, full stories, audio and video retrieval and play for stories;
- sports scores, news stories, schedules, alerts, statistics, back ground and history information, etc.;
- ability to subscribe interactively to multimedia information channels, including sports, news, business, different types of music and entertainment, applying user specific preferences for extracting and presenting information;
- rights management for information or content used or published;
- horoscopes, daily jokes and comics, crossword puzzle retrieval and display and related entertainment or diversions;
- recipes, meal planning, nutrition information and planning, shopping lists and other home organization related activities;
- as an interface to auctions and online shopping, and where the system can manage payment or an electronic wallet;
- management of network communications and conferencing, including telecommunications, email, instant messaging, Voice over IP communications and conferencing, local and wide area video and audio conferencing, pages and alerts;
- location, selection, management of play lists and play control of interactive entertainment from local or network sources including, video on demand, digital audio, such as MP3 format material, interactive games, web radio and video broadcasts;
- organization and calendar management for families, businesses and other groups of users including the management of, meetings, appointments, and events; and
- interactive educational programs using local and network material, with lesson material level set based on user's profile, and including, interactive multimedia lessons, religious instruction, calculator, dictionary and spelling, language training, foreign language translation and encyclopedias and other reference material.

It will be appreciated that the foregoing statements of the features of the invention are not intended as exhaustive or limiting, the proper scope thereof being appreciated by reference to this entire disclosure and reasonably apparent variations and extensions thereof.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will be described by reference to the preferred and alternative embodiments thereof in conjunction with the drawings in which:

FIG. **1** is an overall diagrammatic view according to one embodiment of the invention;

FIG. **2** is a schematic block diagram showing the agent architecture according to one embodiment of the invention;

FIG. **3** is a high level process for receiving natural language speech-based queries and/or commands and generating a response according to one embodiment of the invention;

FIG. **4**A is a process for receiving natural language speech-based queries and/or commands and generating a response according to one embodiment of the invention;

FIG. **4**B is a process for receiving natural language speech-based commands in order to locally or remotely control functions of devices according to one embodiment of the invention;

FIG. **5** is a process for correctly interpreting a user's utterance according to one embodiment of the invention; and

FIG. **6** is a process for determining the proper domain agents to invoke and the properly formatted queries and/or commands that is to be submitted to the agents according to one embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The fact that most natural language queries are incomplete in their definition is a significant barrier to natural human query-response interaction between humans and machines. Further, some questions can only be interpreted in the context of previous questions, knowledge of the domain, or the user's history of interests and preferences. Thus, some natural language questions may not be easily transformed to machine processable form. Compounding this problem, many natural language questions are ambiguous or subjective. In these cases, the formation of a machine processable query and returning of a natural language response is difficult at best.

In order for machines (e.g., computer devices) to properly respond to a question or a command that is in a natural language form, machine processable queries and commands may be formulated after the natural form question or command has been parsed and interpreted. Depending on the nature of the question or command, there may not be a simple set of queries and commands that will return an adequate response. Several queries and commands may need to be initiated and even these queries and commands may need to be chained or concatenated to achieve a complete result. Further, no single available source may contain the entire set of results required. Thus multiple queries and/or commands, perhaps with several parts, may be needed for multiple data sources, which can be both local or on a network. Not all of these sources, queries and/or commands will return useful results or any results at all. Useful results that are returned are often embedded in other information, and from which they may need to be extracted. For example, a few key words or numbers often need to be "scraped" from a larger amount of other information in a text string, table, list, page or other information. At the same time, other extraneous information such as graphics or pictures needs to be removed to process the response. In any case, the multiple results must be evaluated and combined to form the best possible answer, even in the case where some queries do not return useful results or fail entirely. In cases where the question is ambiguous or the result inherently subjective, determining the best result to present is a complex process. Finally, to maintain a natural interaction, responses need to be returned rapidly to the user. Managing and evaluating complex and uncertain queries while maintaining real-time performance is a significant challenge.

The following detailed description refers to the accompanying drawings, and describes exemplary embodiments of the present invention. Other embodiments are possible and modifications may be made to the exemplary embodiments without departing from the spirit, functionality and scope of the invention. Therefore, the following detailed descriptions are not meant to limit the invention.

The present invention provides a complete speech-based information query, retrieval, processing and presentation environment. In addition, the invention may be useful for control of the system itself and/or external devices. This integrated environment makes maximum use of context, prior information and domain and user specific profile data to achieve a natural environment for one or more users making queries or stating commands in multiple domains. Through this integrated approach, a complete speech-based natural language query, command and response environment may be created. FIG. **1** shows a schematic diagram view of a system **90** according to an embodiment of the invention.

The system **90** may include a main unit **98** and a speech unit **128**. Alternatively, the system **98** may only comprise of the main unit **98**, the speech unit **128** being a completely separate system. The event manager **100** may mediate interactions between other components of the main unit **98**. The event manager **100** provides a multi-threaded environment allowing the system **98** to operate on multiple commands or questions from multiple user sessions without conflict and in an efficient manner, maintaining real-time response capabilities.

Agents **106** contain packages of both generic and domain specific behavior for the system **98**. Agents **106** may use nonvolatile storage for data, parameters, history information, and locally stored content provided in the system databases **102** or other local sources. User specific data, parameters, and session and history information that may determine the behavior of agents **106** are stored in one or more user profiles **110**. Data determining system personality characteristics for agents are stored in the one or more personality module **108**. The update manager **104** manages the automatic and manual loading and updating of agents **106** and their associated data from the Internet **136** or other network through the network interface **116**.

The main user interface for the system **90**, according to an embodiment of the invention, is through one or more speech units **128**. The speech unit **128** includes one or more microphones, for example array microphone **134**, to receive the utterances of the user. The speech received at the microphone **134** may be processed by filter **132** and passed to the speech coder **138** for encoding and compression. In one embodiment, a transceiver module **130** transmits the coded speech to the main unit **98**. Coded speech received from the main unit **98** is detected by the transceiver **130**, then decoded and decompressed by the speech coder **138** and annunciated by the speaker **136**.

The one or more speech units **128** and the main unit **98** may communicate over a communication link. The communication link can include a wired or wireless link. According to one embodiment, the communication link comprises an RF link. The transceiver **130** on the speech unit communicates coded speech data bi-directionally over the communication link with the transceiver **126** on the main unit **98**. According to another embodiment, RF link may use any standard local area wireless data protocols including the IEEE 802.11, Bluetooth or other standards. Alternatively, an infrared data link conforming to any suitable standard such as IrDA or other infrared standards can be used. In an alternative embodiment, wires connect the speech unit **128** and the main unit **98**, eliminating the need for one speech coder **138**. Other wired or wireless analog or digital transmission techniques can be used.

Coded speech received at the transceiver **126** on the main unit **98** is passed to the speech coder **122** for decoding and decompression. The decoded speech is processed by the speech recognition engine **120** using data in the dictionary and phrases module **112** and received from the agents **106**. The recognized words and phrases are processed by the parser **118**, which transforms them into complete commands and questions using data supplied by an agent **106**. The agent **106** then process the commands or questions. The agent **106** create queries to local databases **102** or though the network interface **116** to data sources on the Internet **136** or other networks. Commands typically result in actions taken by the system **90** itself (i.e., pause or stop), or to a remote device or data source (i.e., download data or program, or control a remote device), through the network interface to the Internet or other data interface.

The agents **106** return results of questions as responses to users. The response may be created using the results of information queries, the system personality **108** and the user pref-

erences or other data in the user profile **110**. The agents **106** generally present these results using the speech unit **128**. The agents **106** create a response string, which is sent to the text to speech engine **124**. The text to speech engine **124** creates the required utterances, which may be encoded and compressed by the speech coder **122**. Once coded, the utterances are transmitted from the main unit **98** by the transceiver **126** to the transceiver **130** on the speech unit **128**. The utterance is then decoded and decompressed by the speech coder **138** and output by the speaker **136**.

The graphical user interface **114** can be used as a substitute or complement to the speech interface. For example, the graphical user interface **114** can be used to view and interact with graphical or tabular information in a manner more easily digested by the user. The graphical user interface can show system state and history in a more concise manner than the speech interface. Users can use the graphical user interface to create or extend agents **106**. These operations can include scripting of agents, adding data to the agent or databases **102** used by the agent, adding links to information sources.

The system **90** may comprise of different types of agents. In one embodiment of the invention, generic and domain specific behavior and information may be organized into domain agents. A system agent, on the other hand, provides default functionality and basic services. The domain specific agents provide complete, convenient and redistributable packages or modules for each application area. In other words, a domain agent includes everything needed to extend or modify the functionality of the system **90** in a current or new domain. Further, domain agents and their associated data can be updated remotely over a network as new behavior is added or new information becomes available. Domain agents may use the services of other, typically more specialized, agents and the system agent. Agents are distributed and redistributed in a number of ways including on removable storage media, transfer over networks or attached to emails and other messages. The invention may provide license management capability allowing the sale of agents by third parties to one or more users on a one time or subscription basis. In addition, users with particular expertise may create agents, update existing agents by adding new behaviors and information and making these agents to other users. A block diagram of an agent architecture according to an embodiment of the invention is shown in FIG. **2**.

Agents **106** receive and return events to the event manager **100**. Both system agents **150** and domain agents **156** receive questions and commands from the parser **118**. Based on keywords in the questions and commands and the structures of the questions and commands, the parser invokes the required agent[s]. Agents use the nonvolatile storage for data, parameters, history information and local content provided in the system databases **102**. When the system starts-up or boots-up the agent manager **154** may load and initialize the system agent **150** and the one or more domain agents **156**. At shutdown the agent manager unloads the agents. The agent manager **154** also performs license management functions for the domain agents **156** and content in the databases **102**.

The system agent **150** manages the criteria handlers **152**, which handle specific parameters or values (criteria) used to determine context for questions and commands. Both the system agent **150** and the domain agents,**156** use the criteria handlers **152**. The various domain agents **156** can use the services of the system agent **150** and of other, typically more specialized, domain agents **156**. The system agent **150** and the domain agents **156** use the services of the agent library **158**, which contains utilities for commonly used functions. The library may include utilities for text and string handling, network communications, database lookup and management, fuzzy and probabilistic evaluation, text to speech formats, and other utilities.

Domain agents **156** can be data-driven, scripted or created with compiled code. A base of generic agent is used as the starting point for data-driven or scripted agents. Agents created with compiled code are typically built into dynamically linkable or loadable libraries. Developers of agents can add new functionality to the agent library **158** as required. Details of agent distribution and update, and agent creation or modification are discussed in sections below.

The invention, according to another embodiment of the invention, may provide capabilities to distribute and update system agents **150**, domain agents **156**, agent library components **158**, databases **102**, and dictionary and phrase entries **112** over wireless or wired networks **136**, including dial-up networks using the update manager **104**. The network interface **116** may provide connections to one or more networks. The update manager **104** may also manages the downloading and installation of core system updates. The agent manager **154** may perform license management functions for the domain agents and the databases. The update manager **104** and agent manager **154** may perform these functions for all agents and database content including, agents and content available to all users or agents and content only available to certain users. Examples of agent and database components added or updated on a periodic basis include:

- agents for new domains;
- additional domain knowledge for agents;
- new keywords for a domain, which can include names of politicians, athletes, entertainers, names of new movies or songs, etc. who have achieved recent prominence;
- links to a preferred set of information sources for the domains covered including links for, entertainment, news, sports, weather, etc.;
- updates to domain information based on, for example, changes to tax laws, company mergers, changing political boundaries; and
- updates to content, including dictionaries, encyclopedias and almanacs; and
- other content and database components.

When a user requires or selects a new domain agent **156** or database element **102**, the update manager **104** may connect to their source on the network **136** though the network interface **116**, download and install the agent or data. To save system resources and to comply with any license conditions, the update manager **104** may uninstall agents that are no longer in use. In one embodiment of the invention, the update manager **104** periodically queries one or more sources of the licensed agents and database components to locate and download updates to agent executables, scripts or data as they become available. Alternatively, the agent sources may initiate the downloading of agent updates of the registered or licensed agents to the update manager as they become available.

The agent manager **154** may provide a license management client capable of executing most any license terms and conditions. When a particular agent **106** and/or database element **102** is required by a command, the agent manager **154** verifies that the use of the agent or data element is within the allowed terms and conditions, and if so, invokes the agent or allows access to the data element. License management schemes that can be implemented through the agent manager **154** include outright purchase, subscription for updates, one time or limited time use. Use of shared agents and data elements (such as

those downloaded from web sites maintained by groups of domain experts) may also be managed by the agent manager **154**.

If a question or command requires an agent, currently not loaded on the system, the agent manager **154** may search the network **136** through the network interface **116** to find a source for a suitable agent. Once located, the agent can be loaded under the control of the update manager **104**, within the terms and conditions of the license agreement as enforced by the agent manger.

New commands, keywords, information, or information sources can be added to any domain agent **156** by changing agent data or scripting. These configuration capabilities may allow users and content developers to extend and modify the behavior of existing domain agents **156** or to create new domain agents **156** from a generic agent without the need to create new compiled code. Thus, the modification of the domain agents **156** may range from minor data-driven updates by even the most casual users, such as specifying the spelling of words, to development of complex behavior using the scripting language as would typically be done by a domain expert. The user can create and manage modifications to domain agents **156** through speech interface commands or using a graphical user interface **114**. User-specific modifications of domain agents **156** are stored in conjunction with the user's profile **110** and accessed by the domain agent **156** at run-time.

The data used to configure data driven agents **156** are structured in a manner to facilitate efficient evaluation and to help developers with organization. These data are used not only by the agents **156**, but also by the speech recognition engine **120**, the text to speech engine **124**, and the parser **118**. Examples of some major categories of data include:

1. Content packages include questions or commands. Each command or question or group of commands or questions includes contexts used for creation of one or more queries. The agent **156** passes a regular grammar expression to the parser **118** for evaluation of a context or question. An initial or default context is typically supplied for each command or question. The command or question includes a grammar for the management and evaluation of the context stack.
2. Page lists or pointers to other local or network content sources. For each page or content source there is a pointer (e.g. URL, URI, or other pointer) to the page or source. Each page has specific scraping information used to extract the data of interest. The scraping information includes matching patterns, HTML or other format parsing information.
3. A response list, determining the response of the agent **156** to a particular command or question given the context, the user profile and the information retrieved. Responses can include diagnostic error messages or requests for more information if the question or command cannot yet be resolved from the known information. Responses can be based on or dependent on thresholds or probabilistic or fuzzy weights for the variables.
4. Substitution lists containing variable substitutions and transformations, often applied by the agents **150**, **156** in the formatting of queries and results. For example, a stock domain specific agent **156** would use a substitution list of company trading symbols, company names and commonly used abbreviations. Substitutions and transformations can be performed on commands and questions to create precise queries, which can be applied against one or more information sources or to results for creating more meaningful output to the user. Substitution lists also include information for optimally dealing with structured information, such as HTTP formatted page parsing and evaluation.
5. Personalities used for responses. Personalities are constructed by combining multiple traits in a weighted manner. Weights can be specified for each agent's domain area to create one or more specific personalities. Examples of personality traits include, sarcasm, humor, irritation, and sympathy, and other traits.
6. Public and user specific parameters for sources, substitutions, transformations, variables or criteria. The public parameter lists are part of the agent package **156**. The user specific parameters are contained in the user profile **110**.

Commands and questions are interpreted, queries formulated, responses created and results presented based on the users personal or user profile **110** values. Personal profiles may include information specific to the individual, their interests, their special use of terminology, the history of their interactions with the system, and domains of interest. The personal profile data may be used by the agents **106**, the speech recognition engine **120**, the text to speech engine **124**, and the parser **118**. Preferences can include, special (modified) commands, past behavior or history, questions, information sources, formats, reports, and alerts. User profile data can be manually entered by the user and/or can be learned by the system **90** based on user behavior. User profile values may include:

spelling preferences;
date of birth for user, family and friends;
income level;
gender;
occupation;
location information such as, home address, neighborhood, and business address;
car type;
telecommunications and other service providers and services;
financial and investment information;
synonyms (i.e., a nick name for someone);
special spelling;
keywords;
transformation or substitution variables;
domains of interest; and
other values.

End users may use the data driven agent **156** extension and modification facilities and values stored in user profiles **110** to create special reports, packages of queries, alerts and output formats. A single alert or report can be configured to use multiple data sources, values, and other variables (i.e., time, location, etc .) to condition or otherwise determine when an alert should be sent. For example, an alert can be generated by sampling a stock price every fifteen minutes and sending an alert if the price drops below some value. To create a report, the user first specifies a set of commands or questions. Next, the user creates or selects a format for the report. Finally the user may name the report. A report can have variable parameters. For example, a user may create a company stock report, and execute the report by stating its name and the company name, which gives the user selected information and in a specified format for that company. In another example, a user can create a "morning" report, which presents selected multimedia information from different sources (news, sports, traffic, weather) in the order and formats desired. Alerts and reports can be created using only voice commands and responses, commands and responses through the graphical

user interface **114**, or a combination of the two. To create a report, alert, or other specialized behavior, the user performs a number of steps including:

- specify the command to run a report or alert;
- specify the question or questions, including keywords, used for a query;
- set the criteria for running the report such as on command or when a particular condition is met;
- define preferred information sources;
- define preferences for order of result evaluation by source, value, etc.;
- specify the presentation medium for a report or alert, such as an email, the text to speech engine, a message to a pager, or a text and graphics display; and
- specify the preferred format for the report, such as information to be presented, order of information to be presented, preferred abbreviations or other variable substitutions.

Filtering and noise elimination may be important in facilitating the various functionalities of the system **90**. The accurate recognition and parsing of the user's speech requires the best possible signal to noise ratio at the input to the speech recognition engine **120**. To accomplish the required improvements an array microphone **134** and a filter **132** are employed. According to an embodiment of the invention, the microphone array, filters and speech coder **138** are physically separated from the main unit **98** into a speech unit **128**, and connected using a wireless link. Since bandwidth on a wireless connection is at a premium, the speech coder dynamically adapts the digitization rate and compression of the captured speech.

The speech unit **128** may use an array of microphones **134** to provide better directional signal capture and noise elimination than can be achieved with a single microphone. The microphone array can be one-dimensional (a linear array) or two-dimensional (a circle, square, triangle or other suitable shape). The beam pattern of the array can be fixed or made adaptive though use of analog or digital phase shifting circuitry. The pattern of the active array is steered to point in the direction of the one or more users speaking. At the same time nulls can be added to the pattern to notch out point or limited area noise sources. The use of the array microphone also helps reduce the cross talk between output from the text to speech engine **124** through the speaker **136** and detection of the user's speech.

The microphone can be complemented with an analog or digital (i.e., Voice over IP) speech interface. This interface allows a remote user to connect to the system and interact with it in the same manner possible if they were physically present.

The speech unit **128**, according to an embodiment of the invention, may use an analog or digital filter **132** between the array microphone **134** and the speech coder **138**. The pass band of the filter is set to optimize the signal to noise ratio at the input to the speech recognition engine **120**. In some embodiments, the filter is adaptive, using band shaping combined with notch filtering to reject narrow-band noise. In one embodiment of the invention, the system **90** may employ adaptive echo cancellation in the filter. The echo cancellation helps prevent cross talk between output from the text to speech engine and detection of the user's speech as well as suppression of environmentally caused echoes. Algorithms comparing the background noise to the signal received from the user's speech are used to optimize the band-shaping parameters of the adaptive filter. Those skilled in the art will be familiar with multiple techniques used to construct suitable digital and analog filters.

The speech received by the array microphone **134** and passed through the filter **132** may be sent to the speech digitizer or coder **138**. The speech coder **138** may use adaptive lossy audio compression to optimize bandwidth requirements for the transmission of the coded speech to the speech recognition engine **120** over a wireless link. The lossy coding may be optimized to preserve only the components of the speech signal required for optimal recognition. Further, the lossy compression algorithms that may be used may be designed to prevent even momentary gaps in the signal stream, which can cause severe errors in the speech recognition engine. The digitized speech may be buffered in the coder and the coder may adapt the output data rate to optimize the use of the available bandwidth. The use of the adaptive speech coder is particularly advantageous when a band-limited wireless link is used between the coder and the speech recognition engine.

In an alternative embodiment, the array microphone can be replaced by a set of physically distributed microphones or a headset worn by the user. The distributed microphones can be placed in different parts of a room or in different rooms. The distributed microphones can create a three-dimensional array to improve signal to noise ratio. The headset may use a wireless or wired connection.

While the invention is intended to be able to accept most any natural language question or command, ambiguity can still be a problem. To assist users formulate concise questions and commands the system **90** may support a voice query language. The language is structured to allow a variety of queries with minimal ambiguity. Thus, the voice query language helps users clearly specify the keywords or contexts of the question or command along with the parameters or criteria. The language provides a grammar to clearly specify the keyword used to determine the context and a set of one or criteria or parameters. A user asking a question or stating a command in the voice query language is nearly always guaranteed to receive a response.

The voice query language may be sensitive to the contents of the context stack. Thus, a following-on question can be asked using an abbreviated grammar, since keywords and criteria can be inherited from the stack. For example, the user can simply ask about another keyword if the criteria of the question remain constant.

The system **90**, may provide built in training capabilities to help the user learn the best methods to formulate their questions and commands. The interactive training allows the user to audibly or visibly see the machine interpretation of their queries and provides suggestions on how to better structure a query. Using the interactive training a user can quickly become comfortable with the voice query language and at the same time learn how to optimize the amount of information required with each step of a dialog.

The output of the speech coder **122** may be fed to the speech recognition engine **120**. The speech recognition engine **120** recognizes words and phrases, using information in the dictionary and phrase tables **112**, and passes these to the parser **118** for interpretation. The speech recognition engine **120** may determine the user's identity by voice and name for each utterance. Recognized words and phrases may be tagged with this identity in all further processing. Thus, as multiple users engage in overlapping sessions, the tags added by the speech recognition engine **120** to each utterance allows other components of the system **90** to tie that utterance to the correct user and dialog. The user recognition capability may further be used as a security measure for applications, such as auctions or online shopping, where this is required. Voice characteristics of each user may be contained in the user profile **110**.

A user may start a dialog with the system **90** when they first address it. This can be done by speaking a generic word ("computer") or addressing a specific name ("Fred"), which may be generally tied to a system personality **108**. Once the user starts the dialog, it may be recognized by the speech recognition engine **120**, using unique characteristics of the user's speech. At the end of a dialog or to interrupt a dialog, the user may utter a dismissal word ("good bye").

According to another embodiment of the invention, the system **90** may employ a speech recognition engine **120 1.24** seeding for improved word recognition accuracy, using data from the dictionary and phrase tables **112**, user profiles **110**, and the agents **106**. At the same time, the fuzzy set possibilities or prior probabilities for the words in the dictionary and phrase tables may be dynamically updated to maximize the probability of correct recognition at each stage of the dialog. The probabilities or possibilities may be dynamically updated based on a number of criteria including the application domain, the questions or commands, contexts, the user profile and preferences, user dialog history, the recognizer dictionary and phrase tables, and word spellings.

For uncommon words or new vocabulary words, a user may be given the option to spell the words. The spelling may be done by saying the names or the letters or using a phonetic alphabet. The phonetic alphabet can be a default one or one of the user's choosing.

Alternatively, when a user uses a word that is not recognized at all or is not correctly recognized by the speech recognition engine **120** then the user may be asked to spell the word. The speech recognition engine **120** determines this condition based on confidence level for the scoring process. The word is looked up in the dictionary **112** and the pronunciation for the word is added to either the dictionary, the agent **106**, or the user's profile **110**. The word pronunciation can then be associated with the domain, the question, the context and the user. Though this process the speech recognition engine learns with time and improves accuracy. To assist users in spelling words an individualized phonetic alphabet can be used. Each user can modify the standard phonetic alphabets with words, which they can remember more easily.

Once the words and phrases have been recognized by the speech recognition engine **120**, the tokens and user identification is passed to the parser **118**. The parser **118** examines the tokens for the questions or commands, context and criteria. The parser **118** determines a context for an utterance by applying prior probabilities or fuzzy possibilities to keyword matching, user profile **110**, and dialog history. The context of a question or command determines the domain and thereby, the domain agent **156**, if any, to be evoked. For example, a question with the keywords "temperature" implies a context value of weather for the question. The parser dynamically receives keyword and associated prior probability or fuzzy possibility updates from the system agent **150** or an already active domain agent **156**. Based on these probabilities or possibilities the possible contexts are scored and the top one or few are used for further processing.

The parser **118** uses a scoring system to determine the mostly likely context or domain for a user's question and/or command. The score is determined from weighting a number of factors including, the user profile **110**, the domain agent's data content and previous context. Based on this scoring, the system **90** invokes the correct agent. If the confidence level of the score is not high enough to ensure a reliable response, the system **90** may ask the user to verify whether the question and/or command is correctly understood. In general, the question that is asked by the system **90** may be phrased to indicate the context of the question including all criteria or parameters. For example, the question can be in the form of: "Did I understand that you want such-and-such" If the user confirms that the question is correct the system proceeds to produce a response. Otherwise, the user can rephrase the original question, perhaps adding additional information to remove ambiguity, or the system can ask one or more questions to attempt to resolve the ambiguity.

Once the context for the question or command has been determined, the parser **118** can invoke the correct agent **156**, **150**. To formulate a question or command in the regular grammar used by agents, the parser **118** may determine required and optional values for the criteria or parameters. These criteria may have been explicitly supplied by the user or may need to be inferred. The parser **118** makes use of the criteria handlers **152** supplied by the system agent **150**. The criteria handlers **152** provide context sensitive procedures for extracting the criteria or parameters from the user's question or command. Some criteria are determined by executing algorithms in the agent, while others may be determined by applying probabilistic of fuzzy reasoning to tables of possible values. Prior probabilities or fuzzy possibilities and associated values are received from a number of sources including the history of the dialog, the user profile **110**, and the agent. Based on user responses, the prior probabilities or fuzzy possibilities are updated as the system learns the desired behavior. For a weather context, examples of criteria include, location, date and time. Other criteria can include command criteria (i.e., yes/no, on/off, pause, stop), and spelling. Special criteria handlers are available from the system agent for processing lists, tables, barge-in commands, long strings of text and system commands.

The criteria handlers **152** operate iteratively or recursively on the criteria extracted to eliminate ambiguity. This processing helps reduce the ambiguity in the user's question or command. For example, if the user has a place name (or other proper noun) in their utterance the parser **118** can use services of the domain agent **156** to look up tables in the databases **102** for place names or can attempt to determine which word is the proper noun from the syntax of the utterance. In another example, the user asks "what about flight one hundred and twenty too?". The parser and domain agent use flight information in the database and network information along with context to determine the most plausible interpretation among; flight **100** and flight **20** also, flight **100** and flight **22**, flight **122**, and the like.

Once the context and the criteria are determined, the parser **118** may form the question or command in a standard format or hierarchical data structure used for processing by the agents **150**, **156**. The parser **118** may fill in all required and some optional tokens for the grammar of the context. Often the tokens must be transformed to values and forms acceptable to the agents. The parser obtains the required transformations from the agents, dialog history or user profile **110**. Examples of transformations or substitutions performed by the parser on tokens include:

- substituting a stock symbol for a company name or abbreviation;
- substituting a numerical value for a word or words;
- adding a zip code to an address; and
- changing a place or other name to a commonly used standard abbreviation.

The agents **150**, **156** may receive a command or question once the parser **118** has placed it in the required standard format. Based on the context, the parser **118** evokes the correct agent to process the question or command.

Commands can be directed to the system **90** or to an external entity. System commands are generally directed to the

system agent **150**. Commands for external entities are generally processed by a domain agent **156**, which includes the command context and behavior for the external entity.

Specific questions are generally directed to one of the domain agents **156**. Based on the question or context and the parameters or criteria, the domain agent creates one or more queries to one or more local or external information sources. Questions can be objective or subjective in nature. Results for objective questions can often be obtained by structured queries to one or more local or network information sources. Even for objective questions, the system **90** may need to apply probabilistic or fuzzy set analysis to deal with cases of conflicting information or incomplete information. Information to answer subjective questions is generally obtained by one or more ad-hoc queries to local or network data sources, followed by probabilistic or fuzzy set evaluation of the one results to determine a best answer.

Once the domain agent **156** has formulated the one or more queries, they may be sent to local and/or network information sources. The queries are performed in an asynchronous manner to account for the fact that sources respond at different speeds or may fail to respond at all. Duplicate queries are sent to different information sources to ensure that at least one source responds with a useful result in a timely manner. Further, if multiple results are received in a timely manner, they can be scored by the system to determine which data is most reliable or appropriate. Examples of data sources accommodated include, HTTP data sources, sources with meta-data in various formats including XML, entertainment audio, video and game files including MP3, databases using query languages and structured responses such as SQL, and other data sources.

The local information sources can be stored in one or more system databases **102** or can be on any local data storage such as a set of CDs or DVDs in a player or other local data storage. Network information sources can be connected to the Internet **136** or other network and accessed through a series of plug-ins or adaptors, known as pluggable sources, in the network interface **116**. The pluggable sources in the network interface **116** may be capable of executing the protocols and interpreting the data formats for the data sources of interest. The pluggable sources may provide information scraping data and procedures for each source to the domain agents **156**. If a new type of data source is to be used, a new plug-in or adaptor can be added to the network interface **116**.

The domain agent **156** evaluates the results of the one or more queries as they arrive. The domain agent **156** scores the relevance of the results based on results already received, the context, the criteria, the history of the dialog, the user profile **110** and domain specific information using probabilistic or fuzzy scoring techniques. Part of the dialog history is maintained in a context stack. The weight of each context for the scoring is based on the relevance of one context to another and the age of the contexts. Other scoring variables can be associated through the context stack. Contexts can also be exclusive, so that previous contexts have no weight in the scoring.

Based on the continuous scoring processes, the domain agent **156** may determine if a single best answer can be extracted. For most questions the desired result has a set of tokens that must be found to formulate an answer. Once a value has been found for each of these tokens the results are ready for presentation to the user. For example, for a question on weather, the tokens can include the date, day of week, predicted high temperature, predicted low temperature, chance of precipitation, expected cloud cover, expected type of precipitation and other tokens. Results processed in this manner include error messages. For subjective questions this determination is made by determining a most likely answer or answers, extracted by matching of the results received. If no satisfactory answer can be inferred from the results of the query the agent can do one of the following:

1. Ask the user for more information, typically through the speech interface, and based on the results obtained formulate new queries. This approach is applied when an irresolvable ambiguity arises in the formulation of a response.
2. Formulate new queries based on the results received from the first set of queries. This approach is typically applied in cases where the responses received do not contain all the required information. Information sources to query can be inferred from the results already obtained (i.e., links in an HTML document) or from other sources. Using this approach one or more sets of queries and responses can be chained without the need for action by the user.
3. Wait for additional queries to return results.

In any case, the domain agent **156** may continue to make queries and evaluate results until a satisfactory response is constructed. In doing so, the agent can start several overlapping query paths or threads of inquiry, typically mediated by the event manager **100**. This technique, combined with the use of asynchronous queries from multiple data sources provides the real-time response performance required for a natural interaction with the user.

The domain agent **156** may apply conditional scraping operations to each query response as it is received. The conditional scraping actions depend on the context, the criteria, user profile **110**, and domain agent coding and data. For each token to be extracted, a scraping criteria **152** may be created using the services of the system agent **150**. The scraping criteria use format specific scraping methods including, tables, lists, text, and other scrapping methods. One or more scraping criteria can be applied to a page or results set. Once additional results are received, the domain agent **156** can create new scraping criteria to apply to results already acquired. The conditional scarping process removes extraneous information, such as graphics, which need not be further processed or stored, improving system performance.

The domain agent **156** may strongly influence how the results are presented. For instance, once the domain agent **156** has created a satisfactory response to a question, the agent **156** may format that response for presentation. Typically, the domain agent **156** formats the response into the markup format used by the text to speech engine **124**. The domain agent **156** may also format the result presentation using available format templates and based on the context, the criteria, and the user profile **110**. Variable substitutions and transformations may be performed by the agent **156** to produce a response best understood and most natural to the user. Further, the order of presentation of tokens and the exact terminology used to create a more natural response to the user may be varied by the agent **156**. The domain agent **156** may also select the presentation personality **108** to be used.

The domain agent **156**, may select the presentation template, determine order of presentation for tokens and determine variable substitutions and transformations using probabilistic or fuzzy set decision methods. The template used to form the presentation can be from the domain agent itself or from the user profile **110**. The user profile **110** can completely specify the presentation format or can be used to select and then modify an existing presentation format. Selection and formatting of presentation template can also depend on the presentation personality modules **108**. At the same time, the characteristics of the personality used for the response are

dynamically determined using probabilities or fuzzy possibilities derived from the context, the criteria, the domain agent itself and the user profile **110**.

The domain agent **156** may apply a number of transformations to the tokens before presentation to the user. These variable substitutions and transformations are derived from a number of sources including, domain information carried by the agent, the context, the token values, the criteria, the personality module **108** to be used, and the user profile **110**. Examples of variable substitutions and transformations include:

- substitution of words for numbers;
- substitution of names for acronyms or symbols (i.e., trading symbols);
- use of formatting information derived from the information sources (i.e., HTML tags);
- nature of the response including, text, long text, list, table;
- possible missing information or errors;
- units for measurement (i.e., English or metric); and
- preferred terminology from the user profile **110** or presentation personality **108**.

The system **90**, according to another embodiment of the invention, may provide special purpose presentation capabilities for long text strings, tables, lists and other large results sets. Domain agents **156** may use special formatting templates for such results. The system agent **150** may provide special criteria handlers **152** for presentation and user commands for large results sets. The presentation templates used by the domain agents **156** for large results sets typically include methods for summarizing the results and then allowing the user to query the result in more detail. For example, initially only short summaries, such as headlines or key numbers, are presented. The user can then query the results set further. The criteria handlers **152** provide users with the capability to browse large results sets. Commands provided by the criteria handlers **152** for large results sets include, stop, pause, skip, rewind, start, and forward.

Some information, in formats such as video, pictures and graphics, may be best presented in a displayed format. The domain agents **156** may apply suitable presentation templates in these cases and present the information through the graphical user interface **114**. The system agent **150** provides special criteria handlers **152** for presentation and user commands for display presentation and control.

FIG. **3** is a high level process **300** for receiving natural language speech-based queries and/or commands and generating a response according to an embodiment of the invention. The process **300** comprises of individual steps, several of which may be repetitive in order to accommodate partial failures. The process **300** may begin when a user's utterance is received and processed at **302**. The user utterance may be a query and/or a command and may be of a natural language speech-based form. After receiving the natural language query and/or command, the query and/or command may be processed so that the data is in a more machine understandable format such as in a digital format. Once the utterance has been converted into an understandable format, the utterance may be processed so that at least an approximate meaning of the utterance can be determined at **304**. This may be accomplished by, for example, the use of stored dictionaries, phrases, user profiles, domain agent data, and other data. Based on the approximate meaning of the utterance, one or more correctly formatted queries and/or commands may be generated. A user created query and/or command may require that multiple queries and/or commands may be needed in order to generate the desired response or action. Further, depending upon the context (e.g., who is the user, what is the subject of the query and/or command directed to, at what time or location is the query and/or command being directed to, and other parameters used in defining the query and/or command) in which the query and/or command has been submitted, a specific format for the query and/or command may be required. For instance, each of the domain agents **156** may require that queries and/or commands be formatted in a specific manner. Thus at **306**, one or more properly formatted queries and/or commands may be generated. Steps **304** and **306** may be operations which may have to be repeated in order to obtain the correct interpretation of the user utterance and result in the desired response and/or action. Once the properly formatted queries and/or commands are generated, the appropriate action may be taken at **308**. This may require that the properly formatted queries and/or commands be sent to specific domain agent[s], information source[s], device[s], or other appropriate destination[s] that can fulfill the requirements of the query and/or command. Once the appropriate action[s] have been executed, this event may be recorded to, for example, the user's profile, database and/or one or more agents at **310**. Such data may be useful for future user inquires and commands. After the action[s] have been performed, a response, if need be, may be generated and forwarded to the user and/or third parties at **312**. In the case of a query for retrieving data, the response would contain the requested information. In the case of a command, the response may be a confirmation that a specific action[s] has been executed. The response may be in the form of a natural language format. The response may also be formatted to reflect a particular personality or tone to the response in order to make the response more "human." The response may be relayed to the user and/or third parties as an audio message and/or a visual message displayed on a user interface.

FIG. **4**A is a process **400** for receiving natural language speech-based queries and/or commands and generating a response using the system **90** according to another embodiment of the invention. The process **400** may begin when a user's utterance (i.e., user query and/or command) is captured through accurate speech recognition operating in a variety of real-world environments at **402**. Once the utterance is captured, it is parsed and interpreted to determine the query and/or command that is contained in the utterance at **404**. Next, review of the query and/or command and determine the domain of expertise required and the context of the query, invoking the proper resources including, for example, agents at **406**. The original query and/or command submitted by the user will often require that multiple queries and/or commands be generated. For instance, suppose a user is interested in retrieving the value of her stock portfolio. The user may utter "please get the value of my stock portfolio." The system **90** may review this request together with stored data such as the user's profile and determine keywords such as "get the value" and "my stock portfolio." The system **90** may then generate queries to determine the stocks in the user's portfolio, the number of shares and the source for current pricing information. These queries may then be sent to one or more agent domains, such as a domain which may access a database containing the user's profile and a domain which accesses stock pricing sources to determine the answers to these questions. Queries may be sent to these domain agents or sources in order to obtain the desired data. Thus, at **408** formulate one or more queries that are sent to one or more local and/or network data sources and/or send appropriate commands to local or remote devices or the system itself. The queries are then sent to the designated agent[s]. The agents may then, in turn, generate their own queries and/or commands to be sent to, for example, local or remote information sources to

retrieve needed data. The agent generated queries and/or commands may be formatted according to the requirements of the target sources and variable substitutions and transformations are performed to modify the queries to a form most likely to yield desired results from the available sources at **410**. Once the queries are formatted correctly, they may be executed in an asynchronous manner and dealing gracefully with failures at **412**. As a result of the execution of the queries, results may be returned by the domain agents and/or sources. The system **90** may then extract or scrape the desired information from the one or more results, which may be returned in any one of a number of different formats at **414**. That is, the results sought by a user may be the summary or the results of further processing of information obtained from several sources for example. Next, the results may be evaluated and interpreted including processing of errors, and gathering and combining them into a single best result judged to be "best" even if the results are ambiguous, incomplete, or conflicting at **416**. Once the best results are determined, perform any required formatting, variable substitutions and transformations to modify the results to a form most easily understood by the user at **418**. Finally, the compound results may be presented through the text to speech engine **124**, to the user in a useful and expected manner at **420**. The process **400** may be performed while accounting for the domain of expertise required, the context in which the question or command is presented, the domain specific information available, the history of the user's interaction, the user preferences, the information sources or commands that are available, and responses obtained from the sources. At each stage of the process **400**, probabilistic or fuzzy set decision and matching methods may be applied to deal with inconsistent, ambiguous, conflicting and incomplete information or responses. In addition, the use of asynchronous queries that may result in rapid and graceful failure of some queries or commands may allow the system **90** to robustly return results quickly and in a manner that seems natural to the user.

FIG. **4**B is a process **450** for receiving natural language speech-based commands in order to locally or remotely control functions of the system **90** or for other devices according to another embodiment of the invention. The process **450** may begin when a user's utterance (i.e., user query and/or command) is captured through accurate speech recognition operating in a variety of real-world environments at **452**. Once the utterance is captured, it is parsed and interpreted to determine the command that is contained in the utterance at **454**. Next, the command is reviewed and the domain determined for the command and context, invoking the proper resources including selecting of agents at **456**. The original command submitted by the user will often require that multiple queries and/or commands be generated. For instance, suppose a user is interested in recorded his favorite television program. The user may utter "please record my favorite TV program." The system **90** may review this request together with stored data such as the user's profile and determine keywords such as "record" and "my favorite TV program." The system **90** may then generate queries to determine the name, the channel, and time for the user's favorite TV program. These queries may then be sent to a domain agent, which may access a database containing the user's profile and determine the answers to these questions. A command may then be sent to a video recorder ordering the video recorder to record the selected TV program. Thus, at **458** formulate one or more queries that are to be sent to one or more local and/or network data sources and/or send appropriate commands to local or remote devices or the system itself. Next, route the generated commands to the appropriate system[s] and/or external devices at **460**. Once the commands have been executed, results including errors may be received and processed at **462**. The results of the executed commands may be optionally presented to the user at **464**.

Some of the steps depicting in FIG. **4**A may actually require multiple steps that may also be repetitive. For instance, in step **404**, the user's utterance is parsed and interpreted in order to determine the meaning of the utterance. The system **90** may make an initial interpretation of the utterance based on the data stored in, for example, the dictionary and phrases module **112**, user profiles **110**, agents **106** and the databases **102**. A real-time scoring system or other techniques may be applied to the interpretation results in order to generate a domain or a context score. If the confidence level of the domain or context score is not high enough to ensure a reliable response, the system **90** can request that the user verify the question or command is correctly understood. In general, the question may be phrased to indicate the context of the question including all criteria or parameters. If the user confirms that the question is correct, the system **90** may proceed to produce a response. Otherwise, either the user can rephrase the original question, perhaps adding additional information to remove ambiguity, or the system may ask one or more questions to attempt to resolve the ambiguity or other actions may taken. FIG. **5** is a process **500** for correctly interpreting a user's utterance according to one embodiment of the invention. The process **500** generally represents steps **402** and **404** of process **400**. Initially the user's utterance is inputted into the system **90** at **502**. The input is then interpreted at **504** using, for example, various sources of stored data such as user profiles, agent data, dictionary and phrases, and other relevant data. The interpretation is then scored using, for instance, a weighted scoring system as described previously at **506**. Once a score is obtained, a determination is made as to whether the confidence level of the interpretation is satisfactory at **508**. In other words the determination relates to whether the score or confidence level given to the interpretation exceeds a certain value. If the confidence level is determined to be unsatisfactory then a request may be submitted to the user requesting that the user verify the interpretation at **510** and **512**. If the user is unsatisfied with the interpretation, he/she may be asked to rephrase the utterance and/or provide additional information at **514**. Once the user provides the rephrased utterance and/or additional information, the process **500** returns to the beginning at **502**. If, on the other hand, the user is satisfied with the interpretation, then the process **400** for receiving a natural language speech-based queries and/or commands and generating a response may continue at **516**.

Once the meaning of the utterance has been determined, the appropriate domain agent[s] and the query and/or commands properly formatted for the agent[s] may be determined. Referring to FIG. **6**, a process **600** is illustrated for determining the proper domain agent[s] to invoke and the proper formatting of queries and/or commands that are to be submitted to the agents **106**, as generally depicted in steps **406-408** of FIG. **4**A according to one embodiment of the invention. In order to formulate a question or command in the regular grammar used by agents **106**, a determination may be made as to the required and optional values for the criteria or parameters of the queries and/or commands. These criteria may have been explicitly supplied by the user or may need to be inferred. Thus, the properly interpreted query and/or command that is generated by, for example, steps **404** and **454** of FIGS. **4**A and **4**B, is parsed at **602**. The content of the properly interpreted query and/or command is then analyzed to determine the criteria and/or parameters of the properly interpreted

query and/or command at **604**. A context sensitive procedure for extracting the criteria or parameters from the properly interpreted query and/or command may be used. Some criteria are determined by executing algorithms in the agent, while others may be determined by applying probabilistic of fuzzy reasoning to tables of possible values. Prior probabilities or fuzzy possibilities and associated values are received from a number of sources including the history of the dialog, the user profile **110**, and the agent. Based on user responses, the prior probabilities or fuzzy possibilities are updated as the system **90** learns the desired behavior. For a weather context, examples of criteria include, location, date and time. Other criteria can include command criteria (i.e., yes/no, on/off, pause, stop), and spelling. The process for determining criteria may be iterative or recursive in order to eliminate ambiguity in the user's question or command. For example, if the user has a place name (or other proper noun) in their utterance, tables in the databases **102** may be reviewed for place names or an attempt may be made to determine which word is the proper noun from the syntax of the utterance. In another example, the user asks "what about flight one hundred and twenty too?" In such a situation, flight information in the database and network information along with context may be used to determine the most plausible interpretation among: flight one-hundred and flight twenty also, flight one-hundred and flight twenty-two flight one hundred twenty-two. and the like. Once the parameters and criteria for the query and/or command have been established, the proper agents **106** may be selected at **606**. After selecting the agents **106**, a properly formatted query and/or command that will be submitted to the agents **106** may be determined at **608**. The query and/or command may be in a standard format or may be a hierarchical data structure used for processing by the agent **106**. In order to submit the properly formatted query and/or command to the agent **106**, all of the required and some optional tokens for the grammar of the context may be filled in. Often the tokens must be transformed to values and forms acceptable to the agents. The required transformations may be obtained from the agents, dialog history or user profile **110**. Examples of transformations or substitutions that may be performed were provided above. Once the properly formatted queries and/or commands are generated, the process (e.g., process **400**) may continue at **610**.

We claim:

1. A method responsive to a user generated natural language speech utterance, comprising:

receiving the user generated natural language speech utterance, the received user utterance containing at least one request;

maintaining a dynamic set of prior probabilities or fuzzy possibilities usable at each stage of processing the received user utterance;

recognizing words and phrases contained in the received utterance using information in one or more dictionary and phrase tables;

parsing the recognized words and phrases to determine a meaning of the utterance, wherein determining the meaning includes determining a context for the at least one request contained in the utterance based on one or more keywords contained in the recognized words and phrases;

selecting at least one domain agent based on the determined meaning, the selected domain agent being an autonomous executable that receives, processes, and responds to requests associated with the determined context;

formulating the at least one request contained in the utterance in accordance with a grammar used by the selected domain agent to process requests associated with the determined context;

invoking the selected domain agent to process the formulated request; and

presenting results of the processed request to the user, the presented results generated as a result of the invoked domain agent processing the formulated request.

2. The method according to claim **1**, wherein recognizing the words and phrases contained in the received utterance includes:

dynamically updating the information in the dictionary and phrase tables based on the dynamic set of prior probabilities or fuzzy possibilities;

determining an identity of the user based on voice characteristics of the utterance; and

associating the recognized words and phrases and a pronunciation of the recognized words and phrases with the identity of the user, the determined context, and the at least one request when the recognized words and phrases satisfy a predetermined confidence level.

3. The method according to claim **2**, wherein recognizing the words and phrases contained in the received utterance further includes:

requesting the user to spell one or more of the words and phrases when the recognized words and phrases fail to satisfy the predetermined confidence level; and

updating the information in the dictionary and phrase tables based on a phonetic alphabet spelling of the requested words and phrases, the phonetic alphabet spelling provided by the user in one or more subsequent natural language speech utterances.

4. The method according to claim **1**, wherein determining the context for the at least one request contained in the utterance includes:

matching the keywords contained in the recognized words and phrases to a dynamically updated set of keywords, the dynamically updated set of keywords associated with the dynamic set of prior probabilities or fuzzy possibilities;

scoring one or more possible contexts for the matched keywords using one or more of a profile of the user, data content associated with the domain agents, or a context stack of one or more recent contexts; and

determining the context by selecting one or more of the possible contexts having a score satisfying a predetermined confidence level.

5. The method according to claim **4**, wherein determining the context for the at least one request contained in the utterance further includes:

requesting the user to verify the recognized words and phrases when none of the possible contexts have a score satisfying the predetermined confidence level; and

determining the context based on information provided by the user in one or more subsequent natural language speech utterances.

6. The method according to claim **1**, wherein formulating the at least one request in accordance with the grammar used by the selected domain agent includes:

determining one or more required values and one or more optional values for a request in the grammar used by the selected domain agent;

extracting one or more criteria and one or more parameters from the keywords contained in the recognized words and phrases using procedures sensitive to the determined context;

inferring one or more criteria and one or more parameters using the dynamic set of prior probabilities or fuzzy possibilities; and

transforming the extracted and the inferred criteria and parameters into a format compatible with the grammar used by the selected domain agent, the transformed criteria and parameters including all of the required values and one or more of the optional values.

7. The method according to claim **1**, wherein the at least one request contained in the utterance includes at least one question and at least one command, wherein the invoked domain agent processes the formulated request by:

directing the at least one command to local or remote processing devices; and

querying local or network information sources to resolve information requested in the at least one question.

8. The method according to claim **7**, wherein querying the local or network information sources to resolve the information requested in the at least one question includes:

submitting a plurality of asynchronous queries to the information sources, the plurality of queries including a plurality of duplicate queries submitted to distinct information sources;

a synchronously receiving a plurality of results of the queries from the information sources; and

scoring a relevance of each of the asynchronously received results using the dynamic set of prior probabilities or fuzzy possibilities to determine one or more best responses to the at least one question.

9. The method according to claim **8**, wherein determining the one or more best responses to the at least one question includes:

determining one or more tokens required to formulate a response to the question;

extracting one or more values for the determined tokens from the asynchronously received results, the one or more values extracted by applying one or more scraping criteria to the asynchronously received results;

evaluating the extracted values using the dynamic set of prior probabilities or fuzzy possibilities, wherein evaluating the extracted values includes resolving ambiguous, incomplete, or conflicting information; and

selecting one or more best values for the determined tokens from the evaluated values, wherein the one or more best responses include the one or more selected best values.

10. The method according to claim **9**, wherein determining the one or more best responses to the at least one question further includes:

determining that the dynamic set of prior probabilities or fuzzy possibilities has failed to resolve the ambiguous, incomplete, or conflicting information; and

resolving the ambiguous, incomplete, or conflicting information using subsequently received results of one or more the plurality of asynchronous queries that remain pending.

11. The method according to claim **9**, wherein determining the one or more best responses to the at least one question further includes:

determining that the dynamic set of prior probabilities or fuzzy possibilities has failed to resolve the ambiguous, incomplete, or conflicting information; and

resolving the ambiguous, incomplete, or conflicting information by submitting one or more additional queries to the information sources, wherein submitting the additional queries includes inferring the information sources to be queried based on results of one or more of the plurality of asynchronous queries that have already been received.

12. The method according to claim **9**, wherein determining the one or more best responses to the at least one question further includes:

determining that the dynamic set of prior probabilities or fuzzy possibilities has failed to resolve the ambiguous, incomplete, or conflicting information; and

resolving the ambiguous, incomplete, or conflicting information by requesting the user to provide additional information for resolving the ambiguous, incomplete, or conflicting information.

13. The method according to claim **9**, wherein presenting results of the processed request to the user includes:

selecting a format template for presenting the one or more best responses;

selecting a personality for presenting the one or more best responses;

determining an order for presenting the tokens contained in the response; and

performing variable substitutions and transformations on the best values selected for the tokens, wherein performing the variable substitutions and transformations includes using the dynamic set of prior probabilities or fuzzy possibilities to vary a terminology of the best values.

14. The method according to claim **13**, wherein presenting results of the processed request to the user further includes:

formatting the best responses in accordance with the selected format template, wherein a text to speech engine reads the formatted best responses to create a system generated speech utterance that presents the best responses to the user.

* * * * *