

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 C.F.R. § 1.53(b)(2).

CERTIFICATE OF MAILING BY "EXPRESS MAIL"

Express Mail Label No.: EE698244522US

Date of Deposit: January 4, 2000

I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 C.F.R. § 1.10 on the date indicated above and is addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231.

Andrea C. Morales

Andrea C. Morales

Docket Number	43794-30001.00	Type a plus sign (+) inside this box	<input type="checkbox"/>
---------------	----------------	--------------------------------------	--------------------------

INVENTOR(S)/APPLICANT(S)			
LAST NAME	FIRST NAME	MID INIT	RESIDENCE (CITY AND STATE/FOREIGN COUNTRY)
Kuwadekar	Sanjeev	R.	Northridge, California
Desai	Adesh	R.	Northridge, California
Sodhi	Deepak		Northridge, California
Kovatch	Alex	L.	Manhattan Beach, California

TITLE OF THE INVENTION (280 characters max)

VOICE RESPONSE SYSTEM

CORRESPONDENCE ADDRESS

Wen Liu
 Morrison & Foerster LLP
 555 West Fifth Street, Suite 3500
 Los Angeles, California 90013-1024
 (213) 892-5321

ENCLOSED APPLICATION PARTS (check all that apply)

- Specification Number of Pages 14 Small Entity Statement
 Drawing(s) Number of Sheets 4

METHOD OF PAYMENT (check one)

- | | | |
|--|-----------------------------------|----------|
| <input checked="" type="checkbox"/> A check or money order is enclosed to cover the Provisional filing fees. | PROVISIONAL FILING FEE AMOUNT(\$) | \$150.00 |
| <input type="checkbox"/> The Assistant Commissioner is hereby authorized to charge filing fees and credit
Deposit Account No.: 03-1952. | | |

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

- No.
 Yes, the name of the U.S. Government agency and the Government contract number are:

Respectfully submitted,

SIGNATURE *Wen Liu*

Date: January 4, 2000

TYPED or PRINTED NAME: Wen Liu

REGISTRATION NO.: (32,822)

- Additional inventors are being named on separately numbered sheets attached hereto.

USE ONLY FOR A PROVISIONAL APPLICATION FOR PATENT

VOICE RESPONSE SYSTEM (HeyAnita)

ABSTRACT

A voice response system (HeyAnita) and methods for navigating any network and using facilities and applications provided by various destination nodes within the network. No change is required in the applications provided by the destination nodes. A user can control and navigate the system with no prior knowledge of the system via self-discovery facilities provided as part of a learning system that adapts itself to the user.

SUMMARY

Users spoken utterance is analyzed and matched with an index of destinations. A list of valid destinations is produced and the user is the guided along the path with pre-recorded voice prompts. The user accessing the system can control the navigation via more speech and/or telephone keypad entry. The intent of the system is to be able to come up with a single choice destination amongst the many offered within the system.

The decision to choose a valid destination is driven by a variety of factors

- User preferences
- User profile derived from usage pattern history
- User responses
- Advertiser rules
- Utterance match weightage
- Active context
- Call origin
- Call date/time
- Call length

The destination that is derived earlier is then accessed via spoken utterance and/or telephone keypad entry. User specific information about the destination is derived from the user profile and the current call context and is used to offer access to the facilities offered by the destination. The facilities offered are specific to the application provided by the destination node.

User responses and queries are appropriately translated to the destination format and vice versa. All of the interaction is via concatenated pre-recorded or synthesized voice segments or fragments.

FEATURES AND FUNCTIONALITY

One of the primary components of the HeyAnita system is the Anita Server, which consists of several components to implement the following functionality and features:

1. Wait for an incoming call
2. When a call is received, listen to user's voice as commands and/or free-form speech or telephone keypad entry
3. Decompose spoken utterance into proprietary commands using proprietary word-mapping techniques and voice recognition grammar
4. Ask relevant questions in order to determine user preferences and context
5. Identify the destination using proprietary search algorithms within the destination tree
6. Navigate to the destination and retrieve requested information
7. Translate retrieved information into voice prompts
8. Generate commercials based on user preferences, usage history patterns and context
9. Intermix commercials and information in a seamless manner to generate unique entertaining experience for the user
10. Return information back to the user in the form of concatenated speech fragments and/or synthesized voice

ANITA SERVER - ARCHITECTURE

Fig 1 is a schematic representation of the Anita Server Architecture. The Anita Server is a fault tolerant, scaleable, remotely manageable, multi-threaded NT Service. This comprises the following components:

a. Anita Telephone Interface

Implements call management features such as ring and hangup detection, call switch-over, call transfer, call waiting and tromboning. This also implements functionality to transform computer audio files (.wav files) to audio streams that can be played on a telephone and to detect user utterances on the phone line to pass them on to the Anita Speech Recognition Engine. This may be implemented using Dialogic system software version DNA 3.2 and Nuance Speech recognition system version 6.2.

b. Anita Speech Recognition Engine

Translates spoken utterances to a set of text phrases. This engine supports a number of languages and is speaker independent. This may be implemented using Nuance Speech recognition system version 6.2. This engine serves as input to the Anita Natural Language Engine, described below.

c. Anita Natural Language Engine

Converts natural language sentences to a set of structured commands. These structured commands are then used to drive Anita Query Engine. This implementation is entirely proprietary to HeyAnita.com. The Anita Natural Language Engine in conjunction with Anita Query Engine identify destination nodes and the applications that are available to the user. This engines serves as input to the Anita Query Engine, described below.

d. Anita Query Engine

Maps commands to an application defined using the HeyAnita.com function library (see Appendix A) and state machine definition language. An example of an application would be to obtain weather information using Yahoo! Web site. This would provide a user of the system the capability of listening to weather information for a set of cities or zip codes. The Anita Query Engine does the following:

- 1) Play voice prompts for the user to exactly identify an application
- 2) Generate web URLs to initiate execution of the selected application
- 3) Hand over control to the Anita State Machine and Web Parser, described below

e. Anita State Machine and Web Parser

Anita State Machine and Web Parser executes state machines written using a proprietary function library. This retrieves information web sites and other applications that are enabled for this operation. In addition, its web-parsing function also allows Anita Query Engine to retrieve web pages from any conventional web site on the Internet and convert unstructured HTML data into meaningful structured data. It is not mandatory to make changes to existing web sites to make them work with Anita State Machine and Web Parser. An example of this would be the operations performed to pass in a zip code to the Yahoo web site, execute the form to retrieve the results, select and format the results, play relevant information in the form of concatenated speech fragments. In this scenario the Yahoo! web site was not modified to support the operations nor was it aware that a voice-enabled application was using its HTML based services.

f. Anita Profiler

During each user session, Anita Query Engine transfers relevant information to Anita Profiler. Anita Profiler captures and filters this information to build a repository of user preferences, navigational history and usage patterns. Anita Profiler recognizes the phone number of the incoming caller and can work without any user registration.

g. Anita Ad Mixer

Implements complex algorithms to create an entertaining experience for the user by mixing advertisements and information in a seamless manner. This algorithm is based on a variety of factors such as user preferences and usage patterns, advertisers' rules and currently active context.

h. Anita Prompt Generator

Converts text phrases to audio prompts. Unlike most other text-to-speech engine, Anita Prompt Generator implements algorithms to generate prompts in natural human voice using concatenated speech fragments rather than digitally created voice. However, in cases of completely unstructured text, Anita Prompt Generator uses Text-To-Speech software. This software may be based on Fonix Corporation TTS engine.

i. Anita Repository

All the Anita components are meta-data driven. All the data required to drive these components is stored in Anita Repository. This allows Anita developers to generate new voice applications in a matter of hours by simply adding the necessary meta-data to Anita Repository. This meta-data is stored in the form of relational database tables.

j. Anita Replication Engine

Smart replication engine that allows distribution of Anita Repository information to multiple Anita Servers in a reliable manner. This algorithm uses user preferences and usage patterns to replicate only the necessary information in order to avoid replication storms. In addition to Anita Repository data, Anita Replication Engine also distributes and applies software updates to all Anita Servers including itself.

k. Anita Manager

Implements a set of standard interfaces for remotely monitoring and managing Anita Server components. These interfaces are used by Anita Toolbox to remotely monitor and manage Anita Server components.

ANITA SERVER - PROCESS

1. When a user calls, Anita Telephone Interface(1) receives the call and hands it over to Anita Speech Recognition Engine(2).
2. Anita Speech Recognition engine(2) converts spoken utterance into text and sends it to Anita Natural Language Engine(3) for further processing.
3. Anita Natural Language Engine(3) interprets Natural Language text and sends structured commands to Anita Query Engine(4).

4. Anita Query Engine(4) takes into consideration all of the governing factors such as user preferences, user context, usage patterns and history to determine an end destination node for the user's request.
5. Anita Query Engine(4) generates web queries needed to fulfill user's request and sends them to the Anita State Machine and Web Parser(8).
6. Anita State Machine and Web Parser(8) browses the web to retrieve information requested by the user(11). It parses each received page to convert unstructured text into structured datasets.
7. While Anita State Machine and Web Parser(8) is busy retrieving the requested information, Anita Query Engine(4) asks Anita Prompt Generator(6) to generate context-sensitive voice prompts. It also sends a request to Anita Profiler to add generated queries to the user's profile(7).
8. Anita Prompt Generator(6) asks Anita Ad Generator(9) to create a set of entertaining commercials based on user's preferences and context.
9. Anita Ad Generator(9) asks Anita Profiler(10) for the user preference and usage history data and uses it to select appropriate commercials.
10. Anita Prompt Generator(6) creates an audio stream based on commercials and web information returned by Anita State Machine and Web Parser(8) and sends it to Anita Telephone Interface(12).

ANITA TOOLBOX

To complement the features and functions of the Anita Server, the Anita Toolbox provides a comprehensive set of tools to facilitate business partners and developers to:

- 1) Voice-enable existing web-sites and/or applications
- 2) Build voice-enabled v-applications. This uses the function library to build state machines that can be executed by the Anita State Machine and Web Parser
- 3) Remotely monitor and manage multiple Anita Servers

ANITA SERVER - LOGICAL STRUCTURE

Fig. 2 is a schematic representation of the logical internal structure of Anita Server:

Anita Server consists of three logical servers. These servers could be implemented on one physical box or multiple physical boxes based on the size and load at each Anita site. If they are implemented on multiple boxes, all the boxes are connected on a single high-bandwidth LAN segment.

a. Anita Phone Server

Anita Phone Server implements computer telephony interface using CTI hardware and Anita Telephone Interface, Anita Speech Recognition Engine, Anita Prompt Generator. It connects to one or more digital lines to accept telephone calls.

b. Anita Application Server

Anita Application Server implements Anita applications using Anita Natural Language Engine, Anita Query Engine, Anita State Machine and Web Parser, Anita Profiler and Anita Ad Mixer. This server is connected to Internet using high-bandwidth lines. It also implements smart replication using Anita Replication Engine.

c. Anita Database Server

Anita Database Server implements Anita Repository database.

HeyAnita INFRASTRUCTURE

Fig. 3 is a schematic representation of the overall HeyAnita global infrastructure that comprises Anita Servers in various countries, cities, and other locales. The Anita Servers communicate with each other via a network such as the Internet. The Anita Replication Engine in the Anita Servers distributes Anita Repository information to other Anita Servers. Anita Monitoring Stations are provided to monitor and manage the interaction between the Anita Servers. The Anita Monitoring Stations may be Anita Servers which are configured for monitoring as their primary function. They may be similar to the Anita Managers.

HeyAnita.com Usage Scenarios***User Profile***

User is registered with the system. She wants HeyAnita.com to recognize her based on the caller id of her phone without asking for any password/pin. She wants to travel to New York city and wants the system to help her make arrangements. She prefers using Expedia.com for Travel reservations and has a New York office in Manhattan.

Feature: Free Form Query

Users can use plain language utterances to navigate the system.

Example 1: Weather

Assistant: How can I help you?

User: What is the weather forecast?

Assistant: Weather in your home city of Los Angeles is sunny and seventy degrees with a high expected to be seventy-six and a low expected to be sixty. Would you like the extended forecast for Los Angeles?

User: What's the weather like in New York?

Assistant: It's sunny and seventy degrees with a high expected to be seventy-six and a low expected to be sixty. Would you like the extended forecast for New York?

Example 2: Buying a CD

Assistant: How can I help you?

User: I want to buy the new Guns and Roses CD

Assistant: Please wait while I find the cheapest price for you. CD-now has it for eleven dollars and ten cents. Would you like to buy it now?

Example 3: Stock Quotes

Assistant: How can I help you?

User: I want to know the stock price for Microsoft

Assistant: Microsoft is currently trading at eighty-eight dollars.

Feature: Clarification Questions

If the user's query does not narrow the result set down to one destination then clarification questions are asked.

Example 1: Weather

Assistant: How can I help you?
 User: Please tell the traffic conditions
 Assistant: Please say the freeway and direction that you are interested in
 User: 4 oh 5 South
 Assistant: 405 South is clear from Northridge to Lax, then slows down to Long Beach, then is clear to Irvine

Example 2: Buying a CD

Assistant: How can I help you?
 User: I want to buy CDs
 Assistant: Would you like to buy CDs from Amazon, CD Now or find the cheapest price.
 User: I like Amazon.
 Assistant: Please tell me the name of the CD or the artist...

Example 3: Post Office Directions

Assistant: How can I help you?
 User: Post Office
 Assistant: Say stamps to buy stamps, say directions to get the directions to the post office or say shipping to get shipping status for parcels at the post office
 User: I want directions.
 Assistant: The nearest post office is ...

Example 4: Travel to New York

Assistant: How can I help you?
 User: I want to fly to New York tonight
 Assistant: Please wait while I find the best fare using Expedia..... The fare to New York is \$356. Would you like to buy it
 User: Please do and book me a hotel near my office.
 Assistant: Your preferred hotel Sheraton in Manhattan is \$227. Would you like to book it
 User: Please do and also get me a rental car.
 Assistant: You are all set. Thanks for using HeyAnita.com

Upon Arrival in New York

User: I need directions to the hotel.
HeyAnita recognizes that the call originates from a JFK airport phone number
 Assistant: Directions to your hotel in Manhattan.....

Feature: Organized Catalog

The way in which data is added and stored is also important creating a navigable application. Information is organized in a "tree" structure as shown in Fig. 4. Fig. 4

demonstrates the organized tree of information which helps to show how the clarification questions would be asked while narrowing down the search.

Unlike with the internet, the creator of a VRU can plan and control the creation and growth of this tree so that it remains usable.

Feature: Self-Discovering Features

While traveling down through the tree the user can discover the functions and features of the nodes below.

Each parent node describes the set of features in the child node.

Examples: Shopping = Buy Books, Buy Electronics
 Buy Electronics = Buy CD Players, Buy VCRs

 News = Headlines, Weather, Financial Sports
 Sports = Football, Basketball, Soccer
 Football = Football Headlines, Football Scores, Football Odds
 Football Headlines = ESPN Football Headlines, CBS Football Headlines

Feature: Context Sensitive Results

It is important to point out how this tree concept also gives **context** to the search as well. For example, if the user just said "Amazon" from the context of the main menu then the user would be asked if they wanted to "buy books from Amazon" or to "buy CDs from Amazon" but if the user said the same thing from the context of the books sub-tree then they would be taken directly to the section where they can buy books from Amazon.

Feature: User Preferences

HeyAnita is a learning system. It keeps on accumulating information about how users interact with it and modifies its search mechanism based on users' navigational history and preferences.

Example: If it finds that a particular user always buys books from Amazon, it will take him directly to "Buy Books from Amazon" when he says, "Buy Books"

APPENDIX A**[comCreateObject]**

Creates a COM object and returns a reference pointer

Parameters

strProgId

Return Value

IDispatch

[comDeleteObject]

Deletes a previously created COM object

Parameters

var(IDispatch)

[comGetProperty]

Gets the value of the named property

Parameters

var(IDispatch)	strPropertyName	nArgCount	param1
param2	param3	param4	param5
param7	param8	param9	param10

Return Value

SMVar

[comInvokeMethod]

Invokes a method on a previously created COM object

Parameters

var(IDispatch)	strMethodName	nArgCount	param1
param2	param3	param4	param5
param7	param8	param9	param10

Return Value

SMVar

[comSetProperty]

Sets the value of the named property

Parameters

var(IDispatch)	strPropertyName	NewValue
----------------	-----------------	----------

[dbExec]

Executes a SQL statement for a relational database

Parameters

strSQLQuery	ReturnField	param1	param2	param3
param4	param5	param6	param7	param8
param9	param10	param11	param12	param13
param14	param15	param16		

Return Value

SMVar

[mopGetProperty]**Gets the value of the named property**

Parameters

strPropName

Return Value

String

[mopNavigate]**Navigates to a given URL**

Parameters

Page Params=NULL Headers=NULL OpType=[HTTP_POST]

[mopOpenConnection]**Opens a connection to a given web site**

Parameters

Site=NULL Port=80

[mopRunMacro]**Executes a previously defined macro**

Parameters

lMacroID strStartLabel lFlags

[mopSetProperty]**Sets the value of the named property**

Parameters

strPropName strPropVal

[mopWaitForMacro]**Waits for a given time**

Parameters

LTimeout

[opAssign]**Assigns a value**

Parameters

type arg1 operator arg2

Return Value

SMVar

[opGosub]**Transfer control to the subroutine**

Parameters

strLabel	Param1	Param2	Param3	Param4
Param5	Param6	Param7	Param8	Param9
Param10				

[opFunctionHeader]**Function Header statement for a code fragment**

Parameters

Param1	Param2	Param3	Param4	Param5
Param6	Param7	Param8	Param9	Param10

[opIfGoto]**Transfer of control statement**

Parameters

arg1 strCondition arg2 strLabel

[opLoadAndRunStateMachine]

Transfer control to and executes another state machine

Parameters

SMAAppID nClearVariables strStartLabel

[opNull]

Does nothing, NULL statement

[opMakeDate]

Creates date from input values

Parameters

Year Month Day Hour Minute HourFormat

Return Value

Date

[opMakeString]

Makes a string from input parameters

Parameters

Param1 Param2 Param3 Param4 Param5
Param6 Param7 Param8 Param9 Param10

Return Value

String

[opReturn]

Returns control back to the calling statement

[popGetPos]

Returns the position for a given Tag

Return Value

%Position

[popGetTagEnd]

Finds the end position for the given tag

Parameters

strTag dwFlags=0 strErrorLabel

Return Value

%Position

[popSearch]

Searches for a string in the retrieved data stream

Parameters

SearchString lMaxPos=MAX_INT dwFlags=0 strErrorLabel

Return Value

%Position

[popSeek]

Positions the pointer to the given offset

Parameters

Position strErrorLabel

[popTag]**Creates a property with a given name**

Parameters

Name EndString lMaxPos=MAX_INT dwFlags=0 strErrorLabel

Return Value

%Position

[popTag2]**Creates a property with a given name using a number of characters as offset**

Parameters

Name NumChars=MAX_INT

Return Value

%Position

[nuFindDestinations]**Finds destinations within the tree for a given user utterance**

Parameters

LFlags

[nuGetSlotVal]**Returns the value of a given slot for voice recognition system**

Parameters

strSlotName Type

Return Value

SMVar

[nuGetStr]**Retrieves a string/utterance by the user**

Parameters

nTimeout	nFlags	Grammar1	Grammar2	Grammar3
Grammar4	Grammar5	Grammar6	Grammar7	Grammar8
Grammar9	Grammar10	Grammar11	Grammar12	Grammar13
Grammar14	Grammar15	Grammar16	Grammar17	Grammar18
Grammar19	Grammar20			

Return Value

lStat

[nuLoadDestinations]**Loads destinations for a given node in the tree**

Parameters

LTreeID

[nuMakeCall]**Sets up all necessary initialization information and dials a phone number**

Parameters

strPhoneNumber nTimeout lFlags

[nuPlayDestinations]**Plays voice files associated with a destination**

Parameters

strPreSingle	strPostSingle	strPreMultiple
strPostMultiple	lPreSilence	lPostSilence
		lFlags

[nuPlayFile]

Plays a given voice file

Parameters

strFilename lPreSilence lPostSilence lFlags

[nuRecord]

Records a voice file from user utterance

Parameters

strRecordFile nMaxSecs nMaxSilence strTerminators
lFlags**[nuRegisterMagicWord]**

Registers a word to be treated as a jump/magic word for immediate recognition

Parameters

strPhraseGrammar strGotoLabel lFlags

[nuSayDate]

Plays back the given string as a date using the given format

Parameters

strDate strFormat lPreSilence lPostSilence lFlags

[nuSayFile]

Plays a file (over the voice line).

Parameters

strFilename lPreSilence lPostSilence lFlags

[nuSayNum]

Says a number using the supplied format.

Parameters

Value strFormat lPreSilence lPostSilence lFlags

[nuSayPhrase]

First tries to play a prerecorded file (if the phrase has been previously prerecorded) and then uses the TTS (Text To Speech) engine if no file exists.

Parameters

strPhrase lPreSilence lPostSilence lFlags

[nuSetProperty]

Passes a value along to the recognition engine and stores it in the given property name.

Parameters

strPropName strPropVal

[nuSpellPhrase]

Says the value of each character in the string. Ex. S-P-E-L-L

Parameters

strString lPreSilence lPostSilence lFlags

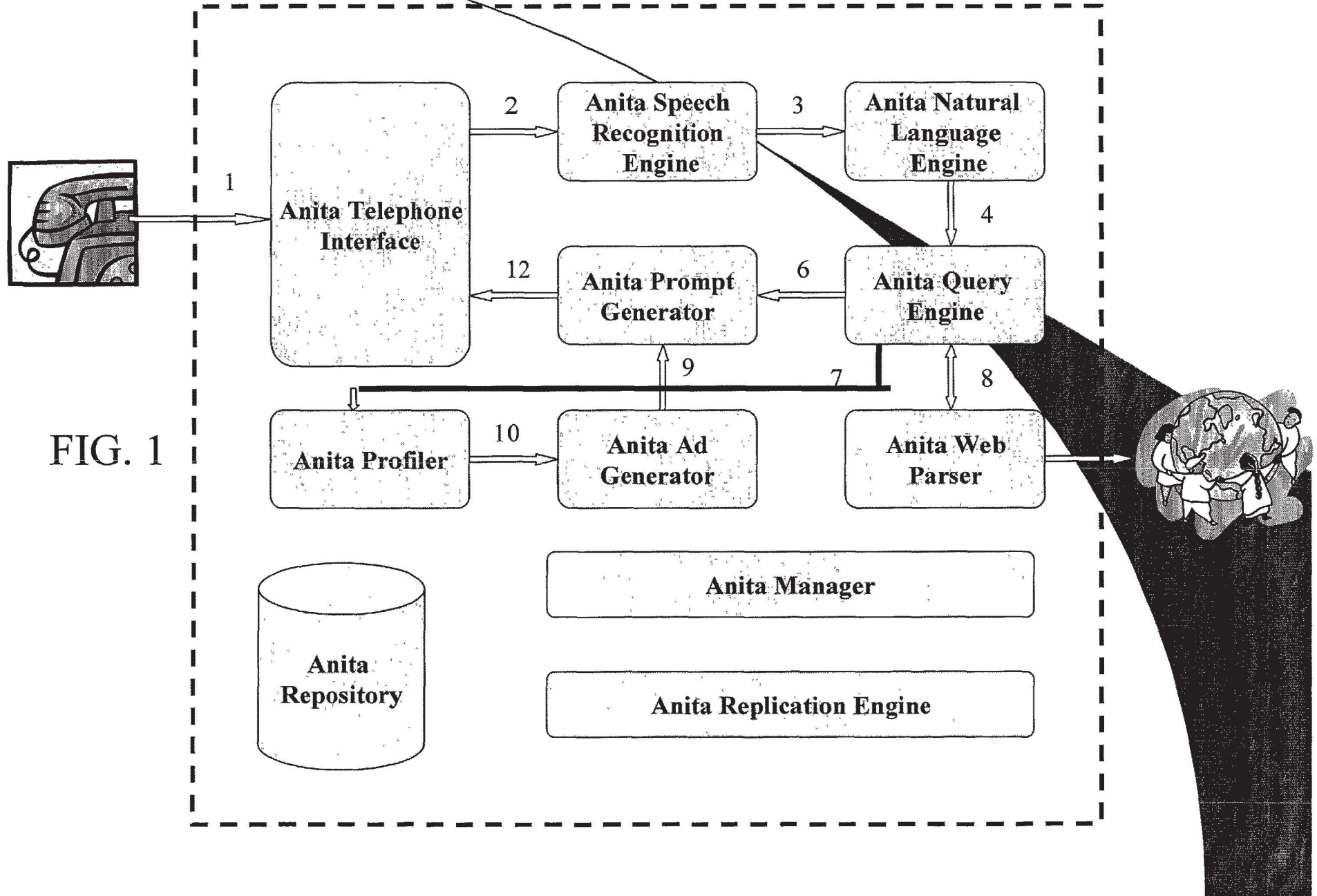
[nuTransferCall]

Makes an outgoing call to the number supplied and transfers the caller to that line.

Parameters

strNumber nTimeout lFlags

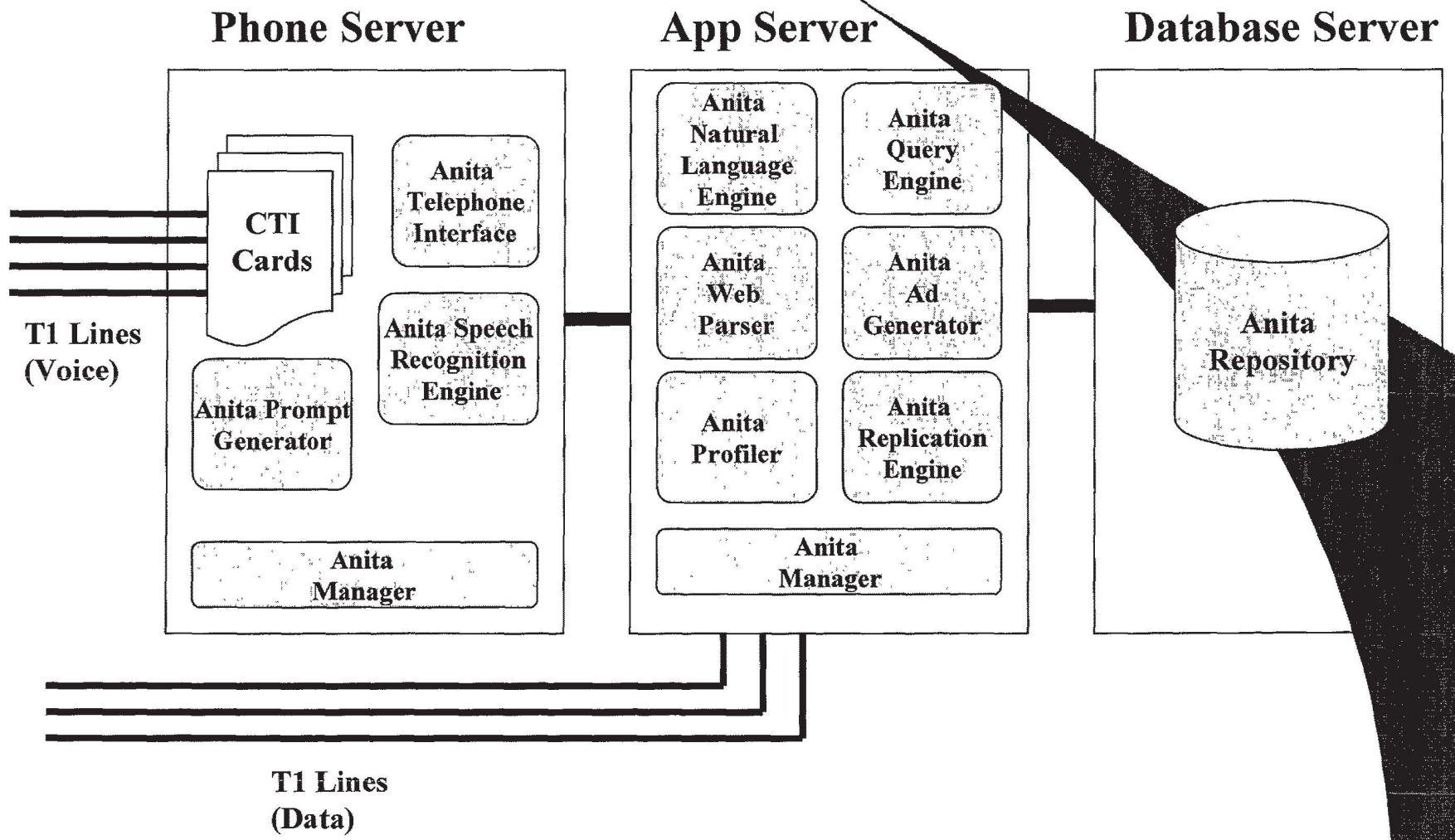
Anita Server Architecture



CONFIDENTIAL

Anita Server Internals

FIG. 2



HeyAnita.com Infrastructure

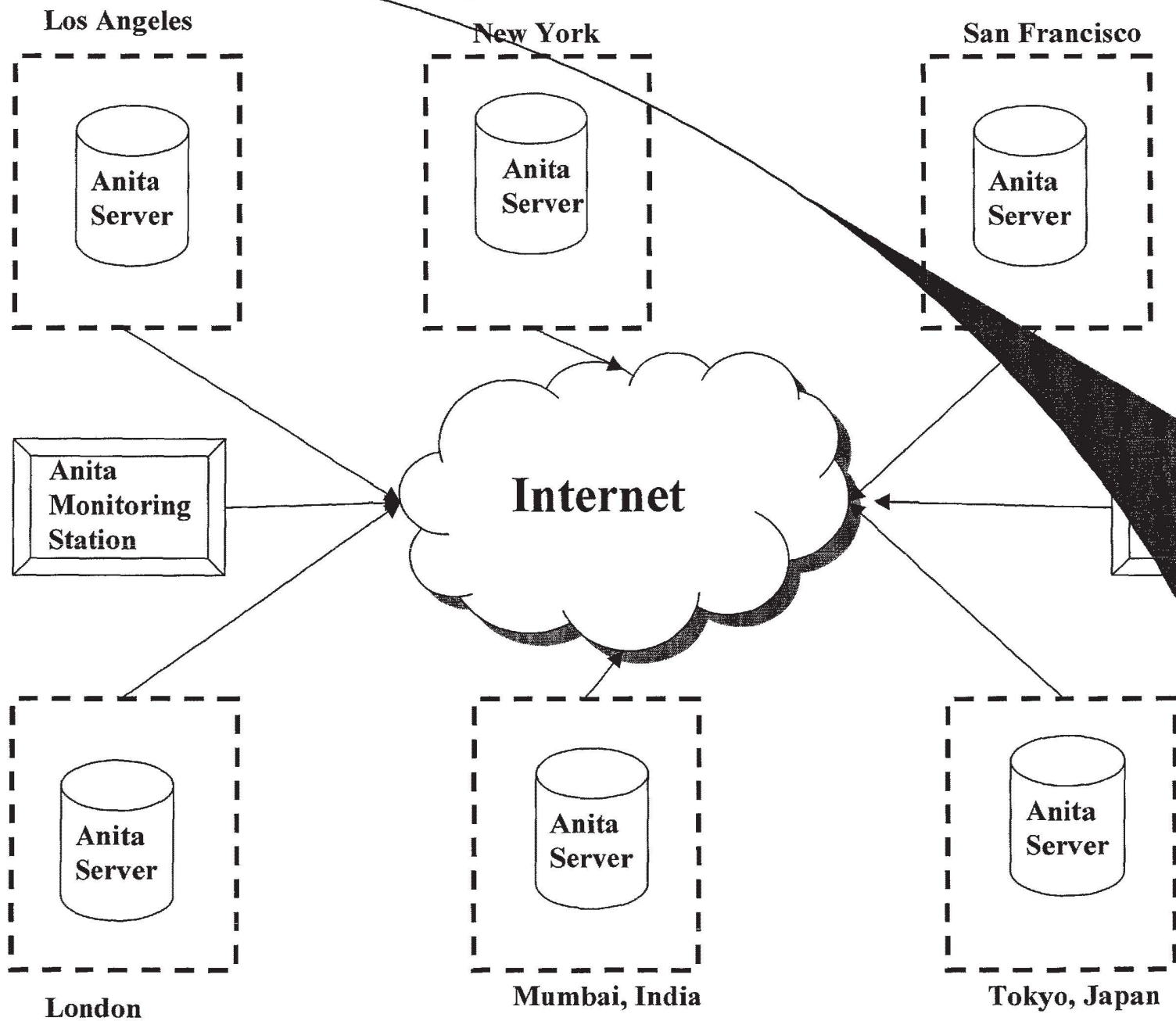


FIG. 3

FIG. 4

