TITLE: CONTROLLING NAVIGATION PATHS OF A SPEECH-RECOGNITION PROCESS

ABSTRACT

A method and computer program for controlling navigation paths in a speech-recognition process in which prompts are provided to a user, and actions are taken based on comparison of the user's spoken responses to stored speech elements. The method includes organizing the prompts, actions, and speech elements in a hierarchy of nodes that include group nodes and terminal nodes, and displaying the nodes in a hierarchical list that indicates navigation paths in the speech-recognition process. A user can alter the navigation paths in the speech-recognition process by editing the position of the nodes in the hierarchical list. The speech-recognition process may be a call routing process.
FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

| AL | Albania                       | ES | Spain                      |
| AM | Armenia                      | FI | Finland                    |
| AT | Austria                      | FR | France                     |
| AU | Australia                    | GA | Gabon                      |
| AZ | Azerbaijan                   | GB | United Kingdom             |
| BA | Bosnia and Herzegovina       | GE | Georgia                    |
| BB | Barbados                     | GH | Ghana                      |
| BE | Belgium                      | GN | Guinea                     |
| BF | Burkina Faso                 | GR | Greece                     |
| BG | Bulgaria                     | HU | Hungary                    |
| BJ | Benin                        | IE | Ireland                    |
| BR | Brazil                       | IL | Israel                     |
| BY | Belarus                      | IS | Iceland                    |
| CA | Canada                       | IT | Italy                      |
| CF | Central African Republic     | JP | Japan                      |
| CG | Congo                        | KE | Kenya                      |
| CH | Switzerland                  | KG | Kyrgyzstan                 |
| CI | Côte d’Ivoire                | KP | Democratic People’s Republic of Korea |
| CM | Cameroon                     | KR | Republic of Korea          |
| CN | China                        | KZ | Kazakhstan                 |
| CZ | Czech Republic               | LC | Saint Lucia                |
| DE | Germany                      | LI | Liechtenstein              |
| DK | Denmark                      | LK | Sri Lanka                  |
| EE | Estonia                      | LR | Liberia                    |
| LS | Lesotho                      | LT | Lithuania                  |
| LU | Luxembourg                   | LV | Latvia                     |
| MC | Monaco                       | MD | Republic of Moldova        |
| MG | Madagascar                   | MK | The former Yugoslavia      |
| ML | Mali                         | MN | Mongolia                   |
| MR | Mauritania                   | MW | Malawi                     |
| MX | Mexico                       | NE | Niger                      |
| NL | Netherlands                  | NO | Norway                     |
| NZ | New Zealand                  | PL | Poland                     |
| PT | Portugal                     | RO | Romania                    |
| RU | Russian Federation           | SD | Sudan                      |
| SE | Sweden                       | SG | Singapore                  |
| SI | Slovenia                     | SK | Slovakia                   |
| SN | Senegal                      | SZ | Swaziland                  |
| TD | Chad                         | TG | Togo                       |
| TJ | Tadjikistan                   | TM | Turkmenistan               |
| TR | Turkey                       | TT | Trinidad and Tobago        |
| UA | Ukraine                      | US | United States of America   |
| UZ | Uzbekistan                   | VN | Viet Nam                   |
| YU | Yugoslavia                   | ZW | Zimbabwe                   |
CONTROLLING NAVIGATION PATHS OF A SPEECH-RECOGNITION PROCESS

Background of the Invention

Businesses often offer main phone numbers as an entry point into their phone network. Customers who call a main number frequently need assistance in reaching a particular person or department. Speech-based automated attendant (autoattendant) software uses voice-recognition technology to quickly forward calls from a main number to a particular extension. For example, an autoattendant might ask "What is the name of the employee you are trying to reach?" then analyze a caller's spoken reply to determine a call destination. Automating this task reduces the burden of offering main business phone numbers.

Unfortunately, information management tasks can partially offset the advantages of installing a speech-based autoattendant. For example, scripting the autoattendant's queries and responses to caller speech ("call flow") can become time consuming. Additionally, coordinating the different speech generating, speech recognizing, and phone directory data can require significant database management efforts.

Summary of the Invention

In general, the invention facilitates control of navigation paths in a speech-recognition process in which prompts are provided to a user, and actions are taken based on comparison of the user's spoken responses to stored speech
elements. The invention organizes the prompts, actions, and speech elements in a hierarchy of nodes comprising group nodes and terminal nodes and displays the nodes in a hierarchical list that indicates navigation paths in the speech-recognition process. A user can alter the navigation paths in the speech-recognition process by editing the position of the nodes in the hierarchical list.

Embodiments may include one or more of the following features. Altering the navigation paths can be done by interacting with a graphical user interface. The user can edit the position of a node in the hierarchical list by a drag and drop operation that can also edit the position of hierarchically included nodes. A user can add, delete, or edit nodes. The prompts, actions, and speech elements can be organized as records in a relational database.

The user may also be presented with a display of hierarchically included nodes of a selected group node in a separate list of hierarchically included nodes. The user may be able to collapse or expand group nodes to alter the display of the hierarchical list.

The speech-recognition process can be a call routing process. The call routing process can include forwarding calls to phone extensions or playing speech files.

Advantages may include one or more of the following.

Controlling call flow with an intuitive user interface reduces the burden of call flow management. Further, the ability to process call flow based on
hierarchical data can not only help callers reach an appropriate extension but can also speed autoattendant response by narrowing acceptable responses to any particular prompt.

The invention may be implemented in hardware or software, or a combination of both. Preferably, the technique is implemented in computer programs executing on programmable computers that each include a processor, a storage medium readable by the processor (including volatile and non-volatile memory and/or storage elements), at least one input device, and at least one output device. Program code is applied to data entered using the input device to perform the functions described above and to generate output information. The output information is applied to one or more output devices.

Each program is preferably implemented in a high level procedural or object oriented programming language to communicate with a computer system. However, the programs can be implemented in assembly or machine language, if desired. In any case, the language may be compiled or interpreted language.

Each such computer program is preferably stored on a storage medium or device (e.g., ROM or magnetic diskette) that is readable by a general or special purpose programmable computer for configuring and operating the computer when the storage medium or device is read by the computer to perform the procedures described in this document. The system may also be considered to be implemented as a computer-readable storage medium, configured with a
computer program, where the storage medium so configured causes a computer to
operate in a specific and predefined manner.

Other features and advantages will be apparent from the following
detailed description, including the drawings, and from the claims.

5 Brief Description of the Drawing

FIGS. 1A-1D are diagrams illustrating autoattendant functions.

FIG. 2 is a diagram of a computer platform that includes autoattendant
components.

FIG. 3 is a diagram of autoattendant components.

FIG. 4 is a diagram of table interrelations in an autoattendant relational
database.

FIG. 5 is a diagram of hierarchy records.

FIG. 6 is a flowchart illustrating the relationship between hierarchically
organized nodes and call flow navigation paths.

FIG. 7 is a screen display of a graphical user interface (GUI) that manages
an autoattendant.

FIGS. 8A-8D are screen displays of autoattendant GUI dialogs.

Description of the Preferred Embodiments

Referring to FIGS. 1A-1D, an autoattendant configuration 10 forwards
incoming calls 12 received by a switch 20, such as a PBX (private branch
exchange), to an autoattendant 24. The autoattendant 24 can ask a caller questions
to determine which extension 14, 16, 18, or 22 the caller wants to reach. The
autoattendant 24 instructs the switch 20 to connect the incoming call 12 with the determined extension as shown in FIG. 1B.

Different extensions may belong to different groups. For example, extensions 16 and 18 may be phones of employees in a sales department 26. If provided with a description of this hierarchical department/employee relationship, the autoattendant 24 can ask a caller which department they would like to reach. After determining the caller's response, the autoattendant 24 can either forward the call to an extension in the sales department 26 or ask the caller which extension 16 or 18 within the sales department 26 the caller wishes to reach.

As shown in FIG. 1C, the autoattendant 24 can also process calls originating from extensions connected to the switch 20, for example, when an employee 14 needs to talk with another employee in a particular department 26. After analyzing the caller's responses to questions, the autoattendant 24 can connect the caller as shown in FIG. 1D.

Instead of routing an incoming call 12, the autoattendant 24 can perform further call processing. For example, the autoattendant 24 can play speech files of technical support information or road directions.

Referring to FIG. 2, an autoattendant 24 can include a computer system 32 that includes a processor 36, memory 34, and other components such as bus interface circuits (not shown). The computer platform 24 includes a standard PC type keyboard 28, a pointing device such as a mouse 30, and a monitor 27. The computer system 32 includes a mass storage element 38 such as a CD, floppy
disk, hard disk, etc. The computer system 32 receiving incoming calls through a line card 37. Portions of mass storage element 38 are transferred to memory 34 and processor 36 in the course of operation.

Mass storage element 32 includes autoattendant management software 40, data 44, and voice user interface (VUI) software 42. The management software 40 provides a graphical user interface (GUI) that displays autoattendant information on the monitor 27 and enables a manager to quickly edit and configure data 44 to provide a desired call flow. The VUI 42 processes incoming calls based on the data 44 as arranged by the management software 40. Locking techniques permit the management software 40 to alter data 44 without interrupting VUI 42 service.

Referring to FIG. 3, data 44 includes different relational databases 50 and 52, however, other implementations use a single relational database. Each database 50 and 52 corresponds to a different call flow and produces different prompt 54 and 58 and grammar 56 and 60 files.

Prompt files 54 and 58 include indexed signal information used by the VUI 42 to produce autoattendant speech. For example, after accessing a relational database 50 or 52 to determine which prompt to play, the VUI 42 may retrieve prompt file 54 or 58 information needed to produce a particular prompt (e.g., "Thank you for calling our business"). Prompt files 54 and 58 can include both prerecorded and site specific prompts.
Grammar files 56 and 60 include indexed signal information that represents the different speech elements the autoattendant can recognize in response to a particular prompt (e.g., "sales, please"). After analyzing caller speech, the VUI 42 can access a relational database 50 or 52 to determine how the autoattendant should respond (e.g., forwarding the call to an extension or playing another prompt).

Compiling software 64 produces grammar files 56 and 60 from relational database 50 and 52 records. Compiling can occur either incrementally, en masse for better run-time retrieval, or both when the manager initiates an incremental compilation and allows automatic scheduling of an off hours compilation.

The management software 40 service manager 66 enables a manager to assign different call flows, as embodied in different databases 50 and 52, to different incoming channels 62. For example, a business may have a set of phone lines for customer inquiries and another set for technical support. The VUI 42 can examine a call directing file 48 to determine which relational database 50 or 52 provides assigned call flow. Many channels 62 can simultaneously use the same relational database 50 or 52.

Referring to FIG. 4, each relational database, such as database 50, includes a call flow configuration (configuration) record 64. The configuration record 64 stores data describing general parameters such as the type of switch connected to the autoattendant and the language being used (e.g., English or Spanish). The configuration record 64 can also store information that indicates
normal business hours, holidays, and an extension (e.g., voice mail or an operator) for handling messages received after hours.

The configuration record 64 also stores a configuration type identifier that controls the prompts the VUI 42 uses to query a caller. For example, a "Basic" configuration produces prompts for a flat configuration that does not nest extensions or other groups within groups. For example, a basic configuration type might prompt "Please say the name of the person or department you would like to reach" and forward the call based on the callers response.

A "Department-Name" configuration produces prompts for a multi-level configuration that nests extensions, groups, etc. within other groups. A department-name configuration might prompt a caller: "To reach a party please say the name of their department or wait for a list of departments." If the caller responds with a department name, the VUI 42 can list the names of people in that department for selection.

Each relational database 50 includes a table of hierarchy records 66. Referring also to FIG. 5, each hierarchy record 66 describes a node in a hierarchy 68. A node can be a group node 26, 130, 132, 134, or a terminal node. A terminal node can represent an extension 14, 16, 18, 24, 140, a speech file 136, 138, or a file that includes further call processing instructions (not shown). A group 26, 130, 132, 134 can hierarchically include (i.e., parent) any of the other node types. The connections between nodes form different navigation paths a caller can
navigate. Referring again to FIG. 4, a hierarchy table 66 record includes a unique identification number used as an index into other database 50 tables.

Records in name detail 70 and group detail 72 tables further describe each hierarchy table 66 record. For example, a name detail table 70 record includes an employee's name and an extension the VUI 42 can use to forward an incoming call. A group detail table 72 record includes a group name, but does not include extension data since a group in a department-name typed configuration does not result in call forwarding until further caller querying (e.g., "who in the department would you like to speak with?").

A pronunciation table 74 describes words in both the name detail 70 (e.g., a person's name) and group detail 72 (e.g., the name of the group) tables. For example, a name of "John Doe" contains two words and is represented by two corresponding pronunciation table 74 records. A pronunciation table 74 record includes both the word and the phonemes that construct the word. For example, the phonemes "ji", "ah" and "n" describe the word "John." The compilation process (64 in FIG. 3) stores the collected phonemes as an entry in a grammar file along with the unique identification number of the hierarchy table 66 record that corresponds to the pronunciation record 74. When a caller speaks, the VUI 42 detects phonemes in the caller's speech, checks the phonemes against phonemes in the grammar file, and retrieves the hierarchy table 66 record that corresponds with the grammar file phonemes that match the caller's speech. For example, when a caller says "John", the VUI 42 finds j-ah-n in a grammar file and searches
the relational database 50 hierarchy table 66 for the corresponding hierarchy 66 record. If the hierarchy 66 record stores a name, the VUI 42 can forward the caller to the extension stored in a corresponding name detail 70 record. If the hierarchy 66 record stores a speech file reference, the VUI 42 can play the speech file to the caller. If the hierarchy 66 record stores a group, the VUI 42 can play a group-level prompt to further query a caller.

The management system 40 can import data into a database 50 from a variety of sources including CSV (Comma Separated Value) files and any ODBC (Open Database Connectivity) compliant data source (e.g., Microsoft Excel™ or Accesses) provided a manager supplies an appropriate ODBC driver. After a manager links fields in the imported data source with autoattendant database fields, the autoattendant can load each data source record into hierarchy 66, name detail 70, and group detail 72 records. The autoattendant automatically produces corresponding pronunciation table 74 records. The manager can also specify whether an import record (e.g., a particular person's information) overwrites an existing record or is ignored. By importing pre-existing human resources data files, managers can quickly begin using an autoattendant without laborious data entry.

The database 50 also includes data that controls the prompts the VUI 42 plays to a caller. The autoattendant data 44 includes pre-recorded prompts for responses to predefined events (e.g., the caller fails to respond or the caller says something not in the grammar file). Referring again to FIG. 5, the prompts
correspond to caller navigation to different hierarchy nodes (records). For example, navigating to a configuration node 64 can trigger a message telling a caller that a call occurred after hours. Navigating to a group node 26, 130, 132, 134 can trigger a group prompt telling a caller to choose a particular employee or other node within a group. Each node can have several associated prompts. The VUI 42 can choose prompts based on caller behavior. For example, the VUI 42 can keep track how many times a caller fails to respond and play a series of different prompts before terminating the call.

Referring back to FIG. 4, the template prompt table 73 stores references to pre-recorded prompts in a prompt file. A manager can record over a pre-recorded prompt, perhaps including business specific information, producing an overriding prompt table 75 record that references a different prompt in the prompt file. During a call, the VUI 42 first checks the prompt table 75 for a prompt record before checking the template prompt table 73. The VUI 42 can then retrieve corresponding prompt file information to produce autoattendant speech.

Referring to FIG. 6, call flow follows the hierarchy defined in the database. After receiving an incoming call, the VUI 42 positions the caller at the root of the hierarchy, the configuration node 64 (110). Thereafter, the VUI 42 engages the caller in "conversation" that controls navigation through the hierarchy (114-124).

As discussed, each node has an associated set of prompts. The VUI 42 plays a prompt for the caller's current node position (112) based on caller
behavior (e.g., how many times the caller a visited the same node). The VUI 42 analyzes a caller's response to a prompt by checking the node's associated grammar file (114). The VUI 42 identifies a hierarchy table 66 record that corresponds to the caller's speech. If the caller has specified a name record (i.e., a record with an extension) (118) the autoattendant can forward the call (120). If the caller has instead specified a group record, the VUI 42 advances the caller to the group node corresponding to the caller's speech (124) and begins the prompt/response exchange again (112).

Referring to FIG. 7, the description of the autoattendant data architecture and call flow may have seemed complicated. Fortunately, management software 40 significantly reduces the complexity of managing such a system.

The management software 40 includes a graphical user interface (GUI) 84 constructed from different Microsoft Foundation Class (MFC) controls (e.g., buttons, list controls, and dialogs). The GUI 84 provides a manager with the ability to quickly define and alter the database node hierarchy (FIG. 6) in addition to providing an intuitive relational database management system. The GUI 78 includes a menu bar 86, toolbar buttons 88, and a side-by-side display that includes a hierarchical list display 90, and a display of hierarchically included nodes 92 of a selected group in the hierarchical list display 90.

The hierarchical list display 90 shows an outline of call flow as embodied in the node hierarchy. The hierarchical list display 90 lists the names of the configuration and hierarchy nodes. Alongside each listed name appears a folder
icon 94 and a sign 93 (e.g., "+" or ")"). A sign 93 indicates whether the hierarchical list display 90 shows nodes included in a particular node.

Expanding (e.g., clicking a listing "+" sign 93) a hierarchy node in the hierarchical list display 90 expands the hierarchical list display 90 to show nodes hierarchically included within the expanded node. For example, expanding group node 96 produces a hierarchical list display 90 that includes listings of included group nodes 95 indented relative to the expanded group node 96. Closing (e.g., clicking a listing "-" sign) a hierarchy node conceals nodes within the closed hierarchy node from the hierarchical list display 90. For example, closing a group node 96 would conceal group nodes 95 from display on the hierarchical list display 90.

A manager can manipulate groups from the hierarchical list display 90. For example, a manager can add and delete groups nodes from a configuration. The hierarchical list display 90 also offers a "drag-and-drop" capability. For example, a manager can drag a selected group into another group. Doing so, alters a hierarchy, nesting the selected group within the other group. While presenting a caller with more levels of navigation may seem undesirable, this technique can help a caller quickly winnow through a tremendous amount of information whether extensions, technical support information, etc. Nesting also narrows the possible responses the VUI 42 needs to consider to determine a grammar file match for caller speech speeding VUI 42 response.
The hierarchically included node display 92 shows the contents of a selected hierarchical list display 90 element. For example, selecting a group node 97 in the hierarchical list display 90 changes the selected group elements icon to an open folder and lists hierarchically included groups and extensions in the hierarchically included node display 92. The display 92 can include node information (e.g., name, extension, or remarks). The display 92 can further include management information about each node. For example, if an employee has not recorded a pronunciation of her name, the display 92 can indicate this by marking a node with an exclamation point (not shown). A manager can sort the display 92 by a variety of criteria such as alphabetical order, when the node was added to the hierarchy, etc.

A manager can add, delete, and edit display 92 elements. A manager can also move elements (i.e., groups or names) into a different positions in the call hierarchy by dragging-and-dropping the element into a different group in either the hierarchically included node display 90 or the hierarchical list display 92. The management system 40 alters database contents based on these actions. This allows a manager to quickly reorganize data and alter call flow.

Referring to FIGS. 8A and 8B, GUI dialogs provide easy management of database information. For example, selecting a configuration node (from FIG. 7) for editing produces the tabbed dialogs shown in 8A and 8B. In FIG. 8A, a manager can edit information in dialog fields that describe a configuration record. In FIG. 8B, a manager can alter the configuration level prompt messages issued
by the VUI 42 in response to events caused by navigation to a configuration node including initial call processing.

Selecting a dialog's "OK" button 100 saves the edited information in the relational database, in this case, potentially updating the configuration record and adding new prompt records. The management software further records - when database changes occur to coordinate record locking. The "Close" button 102 discards edits. The GUI presents a manager with a "Keep changes made" dialog when the manager ends a management session. Those familiar with database concepts will recognize that the "OK" button resembles a SQL INSERT command while the "Keep changes made" dialog causes a database commit or rollback. Familiarity with such database concepts, however, is unnecessary for system management since the GUI presents these database concepts in dialog buttons familiar even to casual word-processing users.

Referring to FIG. 8C, another dialog permits editing of group node information and optionally recording a pronunciation of the group name. Altering group node information can alter the node's description in the hierarchy table and potentially add, delete, or modify prompt and pronunciation records. Again, the management system 40 conceals this cascade of database changes from a manager to ease autoattendant management.

Referring to FIG. 8D, selecting a name node produces a name properties dialog. In this dialog, a manager can alter an employee's extension or alter the phonemes that construct the employee's name. Similar to the group node dialog, a
manager can record a pronunciation of the employee's name or let the management software generate a pronunciation based on spelling. The management software 40 also allows individual employees to remotely (i.e., from any phone) record pronunciation of their own name. Any alterations can produce or alter pronunciation records and produce entries in a prompt file.

Other embodiments are within the scope of the following claims. The techniques described should not be considered limited to autoattendant functions, but instead can be incorporated into a variety of applications.
What is claimed is:

1. A method of controlling navigation paths in a speech-recognition process in which prompts are provided to a user, and actions are taken based on comparison of the user's spoken responses to stored speech elements, the method comprising:

   organizing the prompts, actions, and speech elements in a hierarchy of nodes comprising group nodes and terminal nodes;

   displaying the nodes in a hierarchical list that indicates navigation paths in the speech-recognition process; and

   altering the navigation paths in the speech recognition process in response to the user's editing of the position of the nodes in the hierarchical list.

2. The method of claim 1, wherein altering the navigation paths comprises interacting with a graphical user interface.

3. The method of claim 1, wherein editing the position of a node in the hierarchical list comprises a drag and drop operation which comprises editing the position of hierarchically included nodes.

4. The method of claim 1, wherein organizing the prompts, actions, and speech elements in a hierarchy of nodes comprises organizing records in a relational database.

5. The method of claim 1, wherein organizing the prompts, actions, and speech elements in a hierarchy of nodes comprises adding, deleting, or editing nodes.

6. The method of claim 1, further comprising displaying hierarchically included nodes of a selected group node in a separate list of hierarchically included nodes.
7. The method of claim 1, wherein group nodes in the hierarchical list can be expanded to display hierarchically included nodes or collapsed to hide hierarchically include nodes.

8. The method of claim 1, wherein the speech-recognition process comprises a call routing process.

9. The method of claim 8, wherein terminal nodes comprise phone extensions.

10. The method of claim 9, wherein actions comprise forwarding a call to a phone extension when the user navigates to a phone extension terminal node.

11. The method of claim 8, wherein terminal nodes comprise speech files.

12. A computer program, residing on a computer readable medium, for controlling navigation paths in a speech-recognition process in which prompts are provided to a user, and actions are taken based on comparison of the user's spoken responses to stored speech elements, the program comprising instructions for:

   organizing the prompts, actions, and speech elements in a hierarchy of nodes comprising group nodes and terminal nodes;

   displaying the nodes in a hierarchical list that indicates navigation paths in the speech-recognition process; and

   altering the navigation paths in the speech-recognition process in response to the user's editing of the position of the nodes in the hierarchical list.

13. The computer program of claim 12, wherein altering the navigation paths comprises interacting with a graphical user interface.
14. The computer program of claim 12, wherein editing the position of a node in
the hierarchical list comprises a drag and drop operation which comprises editing the position
of hierarchically included nodes.

15. The computer program of claim 12, wherein organizing the prompts, actions,
and speech elements in a hierarchy of nodes comprises organizing records in a relational
database.

16. The computer program of claim 12, wherein organizing the prompts, actions,
and speech elements in a hierarchy of nodes comprises adding, deleting, or editing nodes.

17. The computer program of claim 12, further comprising displaying
hierarchically included nodes of a selected group node in a separate list of hierarchically
included nodes.

18. The computer program of claim 12, wherein group nodes in the hierarchical
list can be expanded to display hierarchically included nodes or collapsed to hide
hierarchically included nodes.

19. The computer program of claim 12, wherein the speech-recognition process
comprises a call routing process.

20. The computer program of claim 19, wherein terminal nodes comprise phone
extensions.

21. The computer program of claim 20, wherein actions comprise forwarding a
call to a phone extension when the user navigates to a phone extension terminal node.

22. The computer program of claim 19, wherein 2 terminal nodes comprise speech
files.
FIG. 2

[Diagram of a computer system with labeled components: 27 (monitor), 28 (keyboard), 30 (mouse), 12 (incoming calls), 37 (line cards), 34 (memory), 36 (processor), 32 (management), 40 (write), 42 (read), 44 (data)].

SUBSTITUTE SHEET (RULE 26)
**FIG. 5**

5/10

**CONFIGURATION**

EXTENSION FOR VOICE MAIL SYSTEM

SALES

TECHNICAL SUPPORT

PRODUCT

HELPLINE

**FIG. 6**

INCOMING CALL

PLACE CALLER AT CONFIGURATION NODE (ROOT)

PLAY A PROMPT ASSOCIATED WITH CURRENT NODE

COMPARE CALLER SPEECH WITH SPEECH STORED IN GRAMMAR FILE

DID CALLER RESPOND ACCEPTABLY?

DID RESPONSE CORRESPOND TO A NAME NODE?

DID RESPONSE CORRESPOND TO GROUP NODE?

MOVE TO GROUP NODE

FORWARD CALL TO CORRESPONDING EXTENSION
FIG. 8A

GEM Configuration

Configuration Settings: Prompts

Configuration Name: Example Configuration
Configuration Type: Basic
Number of Digits in extensions: 3
Operator Extension: 0
PBX Configuration: Panasonic_KX-T1232

Business Hours
- Open All Day
- Select Open Hours: 08:00AM to 05:00PM

After Hours Support
- Extension 0
- Attended (Operator)
- Unattended (Mailbox)

Holidays (e.g. 971225, 980101)
980101, 980216, 980525, 980907, 981012, 981126, 981127, 981225

Days Closed: Sun, Mon, Tue, Wed, Thu, Fri, Sat

Remote Voice Mail Access
- Enable
- Extension: 101

Remote Name Recording
- Enable
- Access code: 999

OK, Cancel, Help
FIG. 8B

GEM Configuration

**Prompt List**

- [ ] Recommended
- [ ] Advanced

- after hours main prompt
- business hours main prompt
- closed main prompt
- holiday main prompt

**Tips**

This is the main greeting that outside callers hear when they call after business hours. We recommend that you record your own version of this prompt using your company name. Be sure to let the caller know that your business is closed.

**Text**

Thank you for calling. Our office is currently closed.
Prompt List

- Recommended
- Advanced

- attended non-bargethru list unconfirmed2 prompt
- confirmed main prompt
- unattended bargethru list rejection2 prompt
- unattended bargethru list timeout1 prompt
- unattended bargethru list unconfirmed1 prompt
- unattended bargethru list unconfirmed2 prompt
- unattended non-bargethru list rejection2 prompt
- unattended non-bargethru list timeout1 prompt

Tips

This is the main prompt after the caller has said a department name and has confirmed it. The prompt should ask the caller to say the full name of the person he or she would like to reach. Our prompt also gives the name of the department that was recognized so the caller can verify that he or she was correctly recognized.

Text

Please say the full name of the person you would like to reach.
FIG. 8D

Name Properties

Full Name: Tim Collins
Alternate Names: Timothy Collins
Extension: 103
Remark: Product Management

Name Recording
Using Locally Recorded Recording
Play Record

Pronunciation

<table>
<thead>
<tr>
<th>Word</th>
<th>Pronunciation</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLLINS</td>
<td>k oo l ih n z</td>
</tr>
<tr>
<td>TIM</td>
<td>t ih m</td>
</tr>
<tr>
<td>TIMOTHY</td>
<td>t ih m ah th iy</td>
</tr>
</tbody>
</table>

OK Cancel Help