



(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2003/0212558 A1**

**Matula**

(43) **Pub. Date: Nov. 13, 2003**

(54) **METHOD AND APPARATUS FOR DISTRIBUTED INTERACTIVE VOICE PROCESSING**

(57) **ABSTRACT**

(76) Inventor: **Valentine C. Matula**, Granville, OH (US)

A distributed interactive voice processing system is disclosed that distributes IVR capabilities to one or more agent terminals associated with a call center. One or more agent terminals support the traditional functions of a live agent, as well as additional IVR capabilities to support the functions of a virtual or automated agent. A call management system manages the distributed IVR resources by monitoring the availability status of the live and automated IVR agents in the distributed interactive voice processing system and distributing telephone calls and other types of communications to the appropriate agents. The availability status of the automated IVR agents is based on estimated available CPU cycles on the agent terminal. A received call is routed to the extension on the agent terminal associated with the selected IVR channel. A VXML process associated with the selected IVR channel obtains the appropriate application software from a central voiceXML application server and executes the VXML page(s) to play the appropriate prompts from the IVR script to the caller and to collect DTMF or speech recognized inputs from the caller. The distributed design allows centralized application management while using distributed resources to provide improved IVR availability.

Correspondence Address:  
**Ryan, Mason & Lewis, LLP**  
**Suite 205**  
**1300 Post Road**  
**Fairfield, CT 06430 (US)**

(21) Appl. No.: **10/140,850**

(22) Filed: **May 7, 2002**

**Publication Classification**

(51) **Int. Cl.<sup>7</sup> ..... G10L 13/00**

(52) **U.S. Cl. .... 704/260**

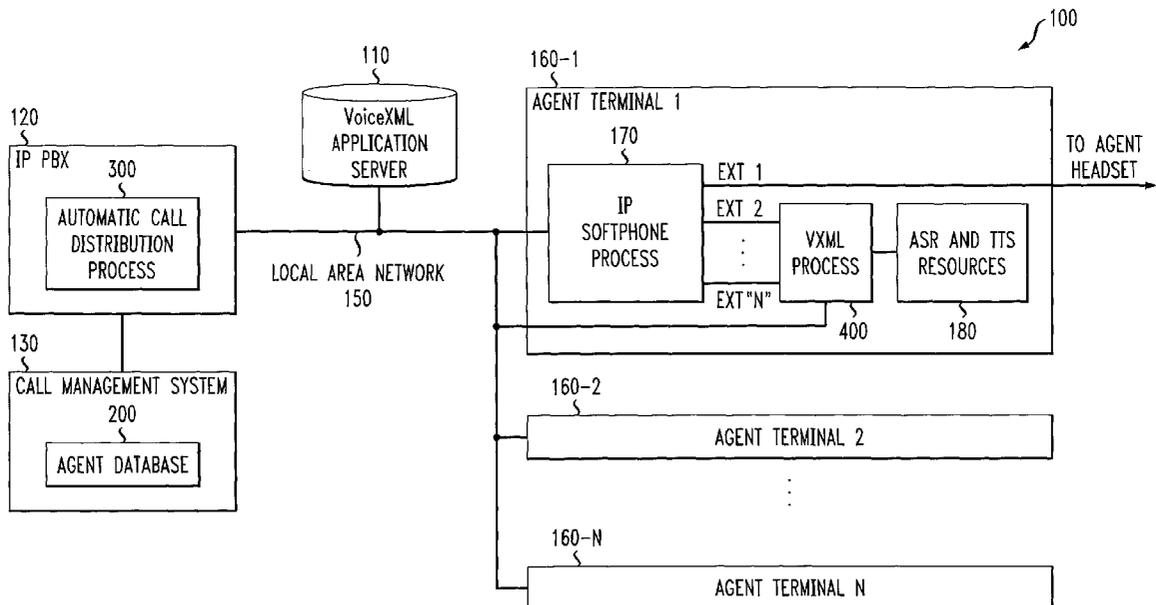


FIG. 1

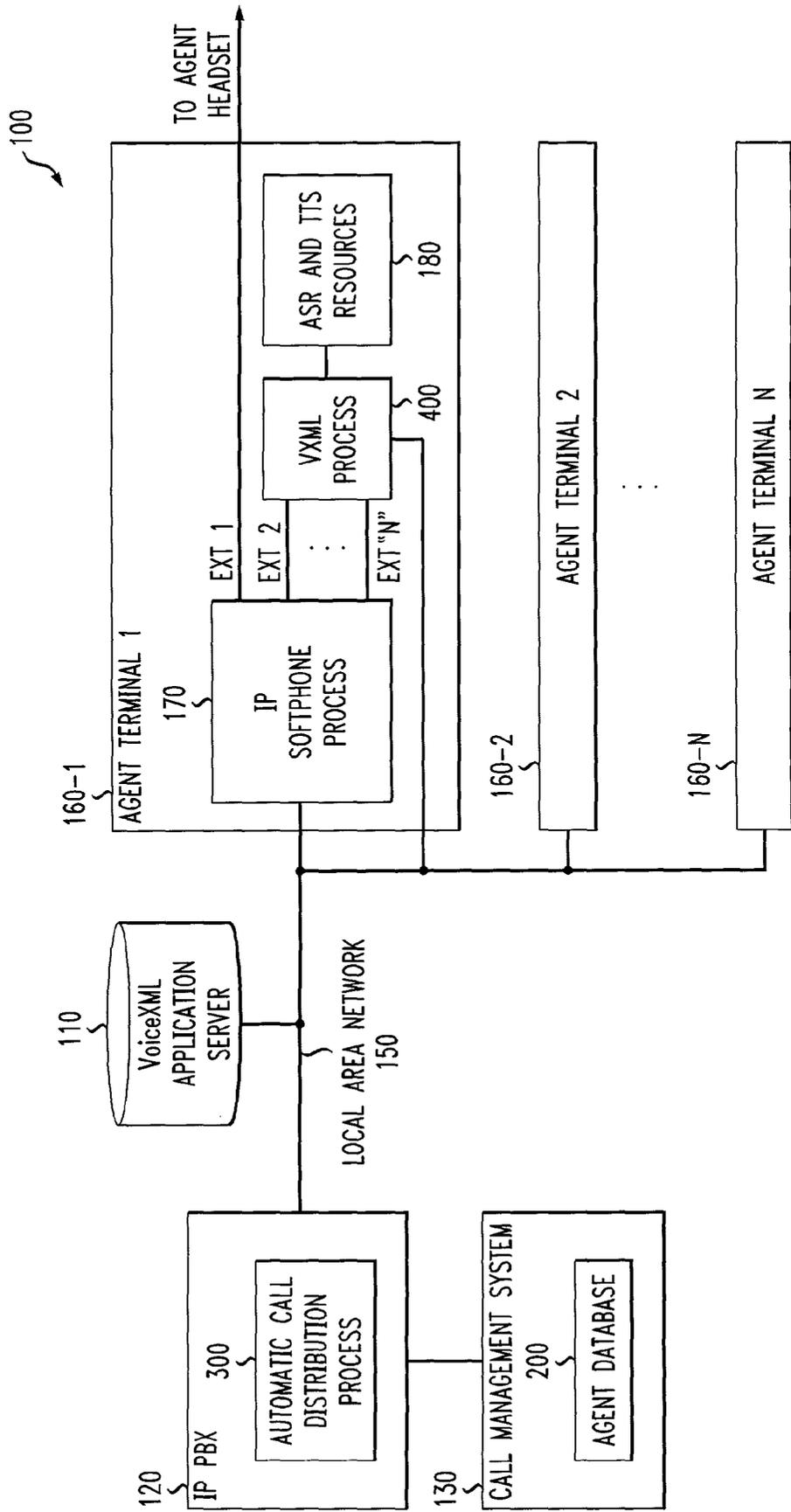


FIG. 2

AGENT DATABASE - 200

AGENT IDENTIFIER	AGENT ADDRESS	AGENT SKILLS	AVAILABILITY
JOHN SMITH	AGENT TERMINAL 1, EXTENSION 1	FLUENT IN ENGLISH AND SPANISH; CUSTOMER DISPUTE RESOLUTION	ON BREAK
IVR AGENT 234	AGENT TERMINAL 1, EXTENSION 2	IVR CAPABILITIES	AVAILABLE
:	:	:	:
IVR AGENT 456	AGENT TERMINAL 2, EXTENSION 5	IVR CAPABILITIES	BUSY

205

210

215

220

230

235

240

250

FIG. 3

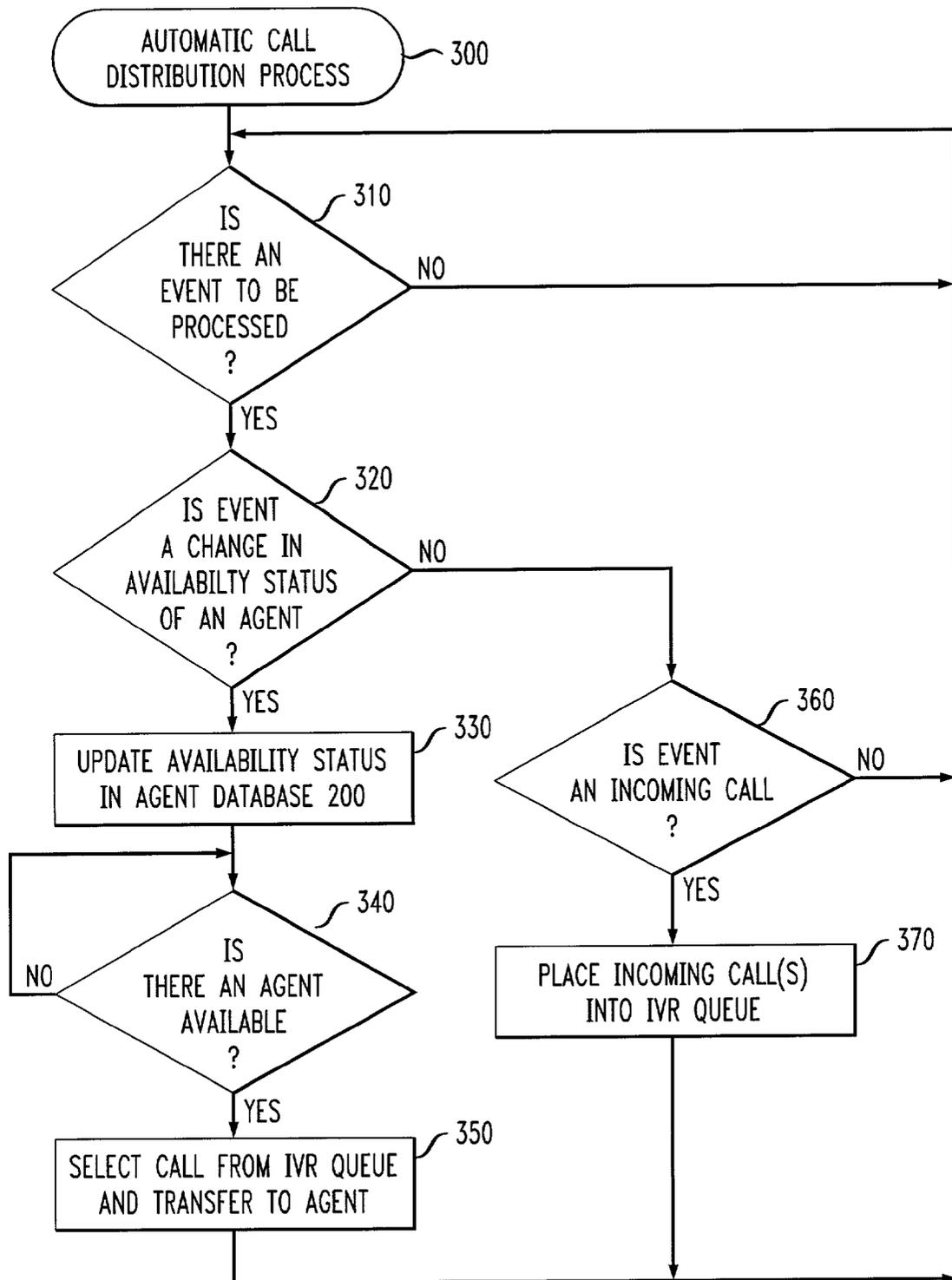
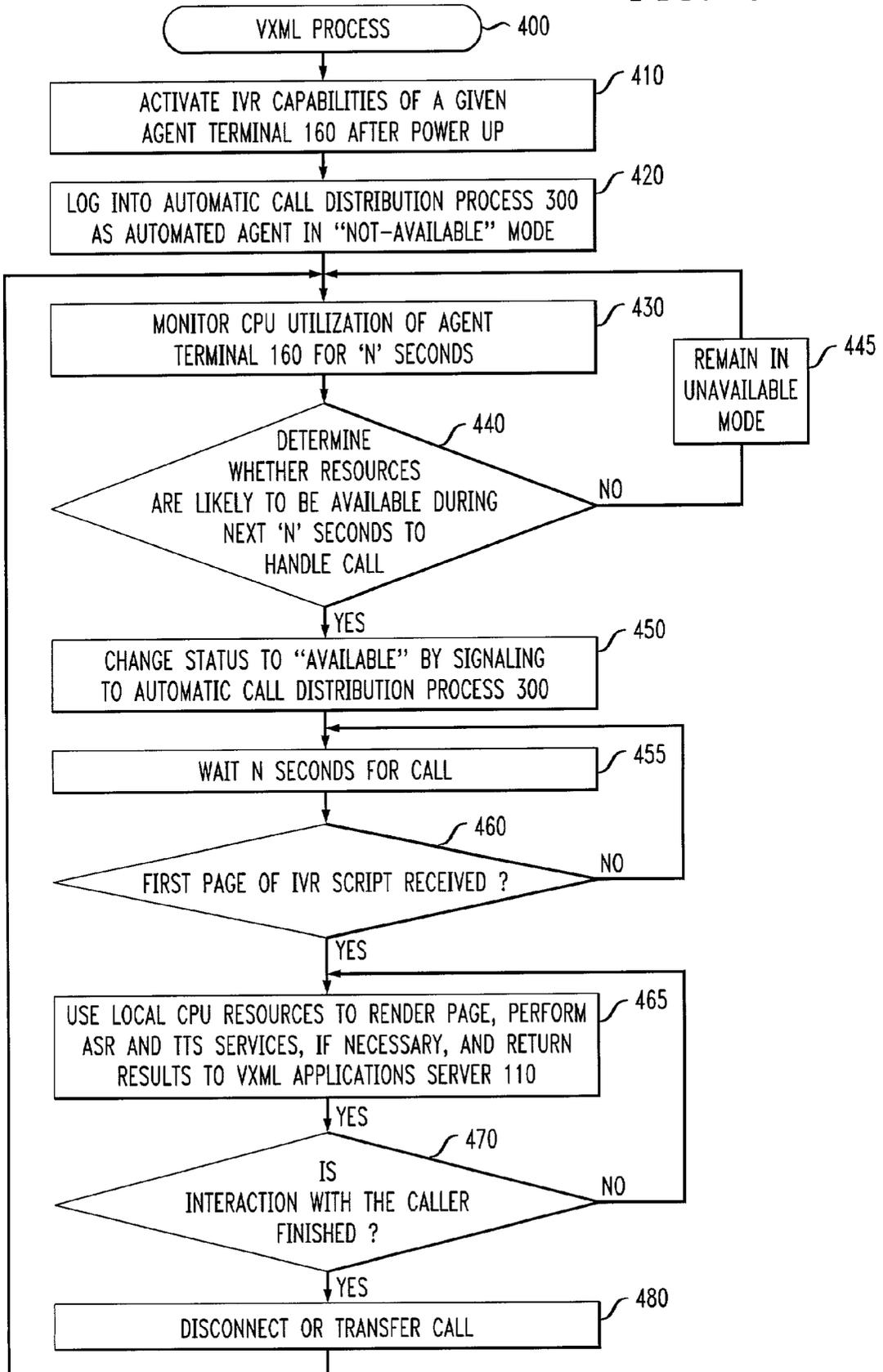


FIG. 4



## METHOD AND APPARATUS FOR DISTRIBUTED INTERACTIVE VOICE PROCESSING

### FIELD OF THE INVENTION

[0001] The present invention relates generally to call centers or other call processing systems in which various types of communications, including voice messages, electronic mail messages, facsimiles, text messages, and Internet service requests, are distributed among a number of service agents for handling.

### BACKGROUND OF THE INVENTION

[0002] Call centers distribute telephone calls and other types of communications to available customer service representatives, referred to herein as "agents," using well-known automatic call distribution techniques. In a call surplus condition, where there are more incoming calls than agents available to process the incoming calls, the call center maintains one or more call queues to maintain the received incoming calls in an active state until an agent with the necessary skill becomes available to process the incoming call.

[0003] Call centers often employ interactive voice response (IVR) systems to efficiently provide callers with information in the form of recorded messages and to obtain information from callers using keypad or voice responses to recorded queries. The IVR systems are often referred to as "automated agents" or "virtual agents." Existing call centers, such as the CONVERSANT® System for Interactive Voice Response, commercially available from Avaya Inc., are typically implemented as centralized server-based interactive customer service solutions. A number of IVR systems employ the Voice eXtensible Markup Language (VoiceXML) to create IVR scripts. Typically, a media server receives a call, obtains the appropriate VoiceXML page from a dedicated local or remote server and then executes the VoiceXML page for the caller.

[0004] The centralization of such IVR resources makes the administration and maintenance of these resources more efficient and also potentially more reliable due to centralized fault monitoring. Furthermore, the redundancy provided by most IVR systems helps to ensure the continued availability of the customer service capabilities that are critical to the efficient operation of a call center and improves the ability to recover from a failure by allowing IVR tasks to continue on an alternate device upon a detected failure. A centralized implementation also allows efficient maintenance of the IVR application processes, including the proprietary call flows and scripting languages, which may require modification or upgrade. IVR systems must typically interface with various subsystems required by the call center, including digital and analog voice circuits, dual tone multiple frequency (DTMF) processing systems, speech recognition systems and text-to-speech processing systems. Thus, the specialized circuitry required by the IVR systems to interface with other systems provides additional motivation for such centralized designs.

[0005] The agents in a call center typically employ computer terminals that are interconnected using a local area network (LAN). When a call is assigned to an available agent, information about the caller or the call (or both), is often routed with the call to the agent's terminal, for presentation to the agent. In this manner, the agent is better

prepared to process the call in an efficient manner with up-to-date call-specific information. It has been found that the processing resources distributed among the various agents in a call center are generally under utilized as agents perform call processing functions.

[0006] A need therefore exists for a method and apparatus for distributing IVR functions among the agent terminals in a call center. A further need exists for a method and apparatus for creating an IVR system using spare processor cycles from agent terminals to supplement or replace centralized or dedicated IVR servers.

### SUMMARY OF THE INVENTION

[0007] Generally, a distributed interactive voice processing system is disclosed that distributes IVR capabilities to one or more agent terminals associated with a call center. An agent terminal in accordance with the present invention is a computer system, such as a personal computer, work station or personal digital assistant, that supports the traditional functions of a live agent, as well as additional IVR capabilities to support the functions of a virtual or automated agent. The distributed interactive voice processing system includes a call management system that manages the distributed IVR resources in accordance with the present invention. Generally, the call management system monitors the availability status of the live and automated IVR agents in the distributed interactive voice processing system and distributes telephone calls and other types of communications to the appropriate agents. The availability status of the automated IVR agents is based on estimated available CPU cycles on the agent terminal.

[0008] When a call is received by the distributed interactive voice processing system, the call is routed using well-known techniques to the extension on the agent terminal of the selected IVR channel and the remote IVR process handles the call. The agent terminals include a VXML process (or another process that provides HTML-based support for IVR) and, optionally, a set of automatic speech recognition (ASR) and text-to-speech (TTS) resources that support the IVR capabilities of the automated IVR agent. When a call is assigned to a distributed IVR extension of a given agent terminal, the associated VXML process obtains the appropriate application page and IVR script from a central voiceXML application server. The VXML process executes the VXML page to play the appropriate prompts from the IVR script to the caller and to collect DTMF or speech recognized inputs from the caller. At the end of a given call, the VXML process will either disconnect the call or request the IP PBX switch to transfer the call to the appropriate call center agent or application.

[0009] In this manner, the present invention efficiently utilizes otherwise "idle" resources of an agent's terminal to provide IVR channels for everyday use or for peak use when centralized server systems are fully loaded or overloaded. The distributed design allows centralized application management while using distributed resources that provide improved IVR availability.

[0010] A more complete understanding of the present invention, as well as further features and advantages of the present invention, will be obtained by reference to the following detailed description and drawings.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0011] FIG. 1 illustrates distributed interactive voice processing system accordance with the present invention;

[0012] FIG. 2 is a sample record from an agent database incorporating features of the present invention;

[0013] FIG. 3 is a flow chart describing an exemplary implementation of a distributed IVR management process incorporating features of the present invention; and

[0014] FIG. 4 is a flow chart describing an exemplary implementation of a VXML process incorporating features of the present invention.

## DETAILED DESCRIPTION

[0015] FIG. 1 illustrates a distributed interactive voice processing system 100 in accordance with the present invention. As shown in FIG. 1, the distributed interactive voice processing system 100 includes a voiceXML application server 110, an Internet Protocol (IP) private branch exchange (PBX) 120, a call management system 130 and one or more agent terminals 160-1 through 160-N (hereinafter, collectively referred to as agent terminals 160), interconnected by a local area network (LAN) 150. According to one aspect of the present invention, each agent terminal 160 includes capabilities to support the traditional functions of a "live agent," such as an IP Softphone process 170, as well as additional IVR capabilities to support the functions of an "automated agent." An IP Softphone emulates a traditional telephone in a known manner. It is noted that the IVR functions that are deployed on the agent terminal 160 of a given live agent may not be related to the work of the given live agent. It is further noted that while the exemplary embodiment employs voiceXML (VXML), other techniques that provide HTML-based support for IVR, such as Speech Application Language Tags (SALT), described, for example, in Speech Application Language Tags (SALT), Technical White Paper, downloadable from [www.saltforum.org](http://www.saltforum.org), could be employed, as would be apparent to a person of ordinary skill in the art, based on the disclosure herein. SALT is a speech interface markup language that extends existing markup languages to enable multi-modal and telephony access to the Internet.

[0016] The IVR application (including the IVR scripts) may be written, for example, in VoiceXML and stored on the central VoiceXML application server 110. The centralized VoiceXML application server 110 simplifies version control and distribution of the application software to the distributed agent terminals 160 without significant administrative overhead. As discussed further below in conjunction with FIG. 4, the VoiceXML application server 110 is presented with requests VXML pages in the exemplary embodiment from a VXML process 400 that supports the IVR capabilities of the automated IVR agent. The VoiceXML application server 110 creates a session with the VXML process 400, and returns the first page of the IVR application encoded in VXML script to the VXML process 400 on the agent terminal 160 for interpretation. For each subsequent request, the VoiceXML application server 110 will obtain or dynamically create (or both) subsequent documents to be sent to the requesting VXML process 400 for rendering. When the interaction with the caller is finished, the final VXML page sent from the VoiceXML application server 110 to the

VXML process 400 will instruct the VXML process 400 to terminate or transfer the call to the appropriate call center agent or application. The VoiceXML application server 110 may be embodied, for example, as a Microsoft Web server with Internet Information Services (IIS) package installed and enabled, commercially available from Microsoft Corp. of Redmond, Wash., as modified herein to carry out the features and functions of the present invention.

[0017] The IP PBX 120 includes an automatic call distribution process 300, discussed below in conjunction with FIG. 3, that manages the distributed IVR resources in accordance with the present invention. Generally, the automatic call distribution process 300 monitors the availability status of all live and automated IVR agents in the distributed interactive voice processing system 100 using an agent database 200, discussed below in conjunction with FIG. 2, and distributes telephone calls and other types of communications to the appropriate agents. The IP PBX 120 can provide information to the call management system 130 regarding the availability of individual agents and other call center statistics, for the generation of appropriate reports and forecasts.

[0018] The call management system 130 may be embodied, for example, as the Call Management System®, commercially available from Avaya Inc. of Basking Ridge, N.J., as modified herein to carry out the features and functions of the present invention. The IP PBX 120 may be embodied, for example, as the IP 600 Internet Protocol Communications Server, commercially available from Avaya Inc. of Basking Ridge, N.J., as modified herein to carry out the features and functions of the present invention.

[0019] The IVR capabilities of a given agent terminal 160 appear to the automatic call distribution process 300 as an agent logged onto an instance of a desktop telephone at the corresponding agent terminal 160. When a call is received at the distributed interactive voice processing system 100, the call is routed using well-known techniques to the IVR extension on the agent terminal 160 and the remote IVR process handles the call. The agent terminals 160 can be embodied as any computing system having a processor and memory configured to perform the disclosed features and functions of the present invention, such as a personal computer, workstation or personal digital assistant.

[0020] As shown in FIG. 1, and discussed further below, the agent terminal 160 includes a VXML process 400, discussed below in conjunction with FIG. 4, and a set of automatic speech recognition (ASR) and text-to-speech (TTS) resources 180 that support the IVR capabilities of the automated IVR agent. Generally, when a call is assigned to a distributed IVR extension of a given agent terminal 160, the associated VXML process 400 obtains the appropriate application page from the voiceXML application server 110 and begins to execute the VXML page to play the appropriate prompts from the IVR script to the caller and to collect DTMF or speech recognized inputs from the caller. At the end of a given call, the VXML process 400 will either disconnect the call or request the IP PBX switch 120 to transfer the call to the appropriate call center agent or application.

[0021] Call Management System 130

[0022] As previously indicated, the automatic call distribution process 300 employs an agent database 200, shown

in FIG. 2, to record the availability status of all live and automated IVR agents in the distributed interactive voice processing system 100. The exemplary agent database 200 maintains a plurality of records, such as records 205 through 220, each associated with a different live or automated agent. For each agent identified by an agent identifier in field 230, the agent database 200 includes the extension address of the agent in field 235, an indication of any special skills of the agent in field 240 and an indication of the availability of the agent in field 250. It is noted that the information recorded in fields 230, 235 and 240 would typically be acquired and maintained by the call management system 130, and the information recorded in field 250 would typically be acquired and maintained by the automatic call distribution process 300 (which may in turn provide the availability status to the call management system 130, as indicated above).

[0023] The automatic call distribution process 300, shown in FIG. 3, manages the distributed IVR resources by monitoring the availability status of all live and automated IVR agents in the distributed interactive voice processing system 100 and distributing telephone calls and other types of communications to the appropriate agents. The automatic call distribution process 300 places calls into an IVR queue as with conventional centralized server systems. As discussed hereinafter, whenever an automated IVR agent (or live agent) that is logged into the queue becomes available, a call is selected from the waiting queue and presented to the agent. A record of each call started and each call completed is made to the call management system 130 for inclusion in status and historical reports.

[0024] As shown in FIG. 3, the automatic call distribution process 300 initially performs a test during step 310 to determine if an event is detected. If it is determined during step 310 that an event is not detected, then program control returns to step 310 until an event to be processed is detected. Once it is determined during step 310 that an event is detected, then program control proceeds to step 320 where a further test is performed during step 320 to determine if event is a change in the availability status of an agent. If it is determined during step 320 that there is no change in the availability status of an agent, then program control proceeds to step 360, discussed below. If, however, it is determined during step 320 that there is a change in the availability status of an agent, then the availability status is updated in the agent database 200 during step 330.

[0025] A further test is performed during step 340 to determine if an agent is available. If it is determined during step 340 that an agent is not available, then program control returns to step 340 until an available agent is detected. If, however, it is determined during step 340 that an agent is available, then a call is selected from the IVR queue and transferred to the available agent during step 350.

[0026] If it is determined during step 320 that there is no change in the availability status of an agent, then a further test is performed during step 360 to determine if the event is an incoming call. If it is determined during step 360 that the event is not an incoming call, then program control returns to step 310 and continues in the manner described above. If it is determined during step 360 that the event is an incoming call, then all incoming calls are placed into an IVR queue (not shown) during step 370, in a known manner.

Program control then terminates or returns to step 310 for continued monitoring of changes in the availability status of agents and distributing telephone calls and other types of communications to the appropriate agents.

#### [0027] Agent Terminal 160

[0028] As previously indicated, the agent terminals 160 employ a VXML process 400, shown in FIG. 4, that supports the IVR capabilities of the automated IVR agent(s). Generally, when a call is assigned to a distributed IVR extension of a given agent terminal 160, the associated VXML process 400 obtains the appropriate application page from the voiceXML application server 110 and begins to execute the VXML page to play the appropriate prompts from the IVR script to the caller and to collect DTMF or speech recognized inputs from the caller. At the end of a given call, the VXML process 400 will either disconnect the call or request the IP PBX switch 120 to transfer the call to the appropriate call center agent or application.

[0029] As shown in FIG. 4, the IVR capabilities of a given agent terminal 160 can be activated during step 410, for example, when the live agent turns on the agent terminal 160. Thus, the VXML process 400 is initiated when the agent turns on the agent terminal 160, together with the EP Softphone process 170, and optional ASR and TTS processes. The VXML process 400 initially logs into the automatic call distribution process 300 during step 420 as an automated IVR agent in a "not-available" mode. The VXML process 400 monitors the CPU utilization of the agent terminal during step 430 for 'N' seconds. At the end of the 'N' second cycle, the VXML process 400 determines during step 440 whether resources are likely to be available during the next 'N' seconds to handle a call. For example, a minimum number of CPU cycles may be specified as a threshold to determine whether sufficient resources are available. In further variations, the availability of sufficient resources can be based, for example, on the rate of change of CPU cycles, or observations of the user mouse or another user input device (which assumes that additional CPU cycles will be consumed by the live agent following such a user input).

[0030] If it is determined during step 440 that resources are not likely to be available, the VXML process 440 stays in a "not-available" mode during step 445, and returns to step 430. If it is determined during step 440 that resources are likely to be available, the VXML process 400 changes its status during step 450 to an "available" mode by asking the IP Softphone process 170 to signal to the automatic call distribution process 300 that the virtual Agent VXML process is available. The VXML process 400 begins a timer during step 455 for 'N' seconds, and waits for a call. If the automatic call distribution process 300 determines that the virtual agent should receive a call from the IVR queue, the call will be routed to IP Softphone software 170, with the destination marked for the extension of the VXML process 400 (such as extension 2 through N in FIG. 1).

[0031] The VXML process 400 receives the telephone number of the calling party and the called number for the call with the incoming call, and makes a request to the VXML applications server 110, for example, via an HTTP request. The VXML applications server 110 returns the first page of the IVR application encoded in VXML script to the VXML process 400 on the agent terminal 160 which is

received during step 460 for interpretation. The VXML process 400 uses local CPU resources during step 465 to render the page, perform ASR and TTS services, if necessary, and return results to the VXML applications server 110.

[0032] A test is performed during step 470 to determine if the interaction with the caller is finished. For example, the final VXML page sent from the VXML applications server 110 to the VXML process 400 can instruct the VXML process 400 to terminate or transfer the call. If it is determined during step 470 that the interaction with the caller is not finished, the VXML process 400 will return to step 465 to process the next IVR script page. If it is determined during step 470 that the interaction with the caller is finished, the VXML process 400 will request the IP Softphone 170 to disconnect or to transfer the call during step 480. As part of standard processing between the automatic call distribution process 300 and the virtual agent, the virtual agent will be put into a "not-available" mode automatically. The VXML process will then return to step 430 to monitor system resources.

[0033] It is noted that if the predicted availability of an agent terminal 160 during step 440 turns out to be incorrect, the terminal 160 must merely time share the available CPU cycles between the functions of the live agent and the IVR functions of the automated agent, in a known manner.

[0034] It is to be understood that the embodiments and variations shown and described herein are merely illustrative of the principles of this invention and that various modifications may be implemented by those skilled in the art without departing from the scope and spirit of the invention.

I claim:

1. A distributed interactive voice processing system, comprising:

a plurality of agent terminals each employed by a corresponding human agent associated with a call center, wherein at least one of said agent terminals supports a corresponding human agent and provides at least one automated interactive voice response (IVR) channel; and

a call management system to distribute communications to an appropriate agent.

2. The distributed interactive voice processing system of claim 1, wherein said call a management system monitors an availability status of said at least one of said agent terminals.

3. The distributed interactive voice processing system of claim 2, wherein said availability status is based on a predicted number of available CPU cycles.

4. The distributed interactive voice processing system of claim 1, wherein said at least one of said agent terminals uses local processing resources to perform at least one interactive voice response function.

5. The distributed interactive voice processing system of claim 4, wherein said at least one interactive voice response function includes rendering an IVR script.

6. The distributed interactive voice processing system of claim 4, wherein said at least one interactive voice response function includes an automatic speech recognition function.

7. The distributed interactive voice processing system of claim 4, wherein said at least one interactive voice response function includes a text-to-speech translation function.

8. The distributed interactive voice processing system of claim 1, wherein said at least one of said agent terminals uses local processing resources to provide IP telephony resources to said human agent.

9. A computing system for use by a human agent in a call center, comprising:

a memory that stores computer-readable code; and

a processor operatively coupled to said memory, said processor configured to implement said computer-readable code, said computer-readable code configured to:

support functions of said human agent; and

provide interactive voice response capabilities to support functions of an automated agent.

10. The computing system of claim 9, wherein said processor is further configured to predict a number of available processing cycles in said processor.

11. The computing system of claim 9, wherein said interactive voice response capabilities includes rendering an IVR script.

12. The computing system of claim 9, wherein said interactive voice response capabilities includes an automatic speech recognition function.

13. The computing system of claim 9, wherein said interactive voice response capabilities includes a text-to-speech translation function.

14. A method for interacting with a caller in a call center, said method comprising the steps of:

receiving a telephone call from a caller; and

processing said telephone call using an interactive voice response channel executing on a computing system dedicated to a human agent.

15. The method of claim 14, further comprising the step of monitoring an availability status of at least one computing system dedicated to a human agent.

16. The method of claim 15, wherein said availability status is based on a predicted number of available CPU cycles.

17. The method of claim 14, wherein said computing system dedicated to a human agent uses local processing resources to perform at least one interactive voice response function.

18. A method for assigning a telephone call to an automated agent in a call center, said method comprising the steps of:

receiving a telephone call from a caller;

monitoring an availability status of at least one computer system upon which said automated agents execute; and

assigning said telephone call to an available automated agent.

19. The method of claim 18, wherein said availability status based on a number of predicted available CPU cycles.

20. The method of claim 18, wherein said at least one computer system upon which said automated agents execute use local processing resources to perform at least one interactive voice response function.

\* \* \* \* \*