United States Patent Application Publication	20250265283
Kind Code	A1
Publication Date	August 21, 2025
Inventor(s)	Nomula; Jagadeshwar et al.

SYSTEM TO ASSIST USERS OF A SOFTWARE APPLICATION

Abstract

A computer-implemented method for agent to agent communication is disclosed. The method comprises receiving a request from a user via a first virtual agent that represents the user; initiating a communication session between the first virtual agent and a second virtual agent representing a service provider to communicate a desired action of the request; and exchanging, by the first and second virtual agents, one or more proposals to fulfill the received request. The method also comprises negotiating, by the first and second virtual agents, terms for performing the desired action based on the exchanged proposals; executing, by the second virtual agent server, the desired action based on the negotiated terms; and communicating, by the first virtual agent server, a response to the user upon the execution of the desired action.

Inventors: **Nomula; Jagadeshwar (Fremont, CA), Yalamarthy; Vineel (Rajahmundry, IN)**

Applicant: **Voicemonk, Inc.** (Sunnyvale, CA)

Family ID: **1000008586932**

Assignee: **Voicemonk, Inc. (Sunnyvale, CA)**

Appl. No.: **19/202161**

Filed: **May 08, 2025**

Related U.S. Application Data

parent US continuation 17346243 20210613 PENDING child US 19202161
parent US continuation 15391837 20161227 ABANDONED child US 17346243
parent US continuation-in-part 15356512 20161118 parent-grant-document US 11068954 child US 15391837
us-provisional-application US 62257722 20151120
us-provisional-application US 62275043 20160105
us-provisional-application US 62318762 20160405

Publication Classification

Int. Cl.: **G06F16/3329** (20250101)

U.S. Cl.:

CPC **G06F16/3329** (20190101);

Background/Summary

CROSS-REFERENCE TO RELATED APPLICATION [0001] This application claims priority under 35 U.S.C. § 120 as a continuation of currently pending U.S. patent application Ser. No. 17/346,243 filed Jun. 13, 2021, which claims priority under 35 U.S.C. § 120 as a continuation of U.S. patent application Ser. No. 15/391,837 filed Dec. 27, 2016, which is a continuation-in-part of currently pending U.S. patent application Ser. No. 15/356,512 filed Nov. 18, 2016, which is a non-provisional application claiming the benefit under 35 U.S.C. § 119(e) of Provisional Patent Application Nos. 62/257,722, 62/275,043 and 62/318,762 filed Nov. 20, 2015, Jan. 5, 2016 and Apr. 5, 2016, respectively. The entire disclosures of all said applications are thereby incorporated by reference.

BACKGROUND
[0002] Unless otherwise indicated herein, the materials described in this section are not prior art to the claims in this application and are not admitted to being prior art by inclusion in this section.
FIELD OF INVENTION
[0003] The subject matter in general relates to the field of virtual assistant systems. More particularly, but not exclusively, the subject matter relates to a virtual assistant system that makes suggestions and executes actions corresponding to events in which preferences of multiple entities matter.
DISCUSSION OF RELATED ART
[0004] In our day-to-day lives we face many situations that require coordination between multiple people. However, multiple opinions arise when there are multiple people, and it becomes tough to come up with recommendations that are suitable to everyone's taste, resulting in a lot of time, effort and coordination needed to come to a decision which is acceptable by everyone. Conventional systems are incapable of making suggestions and executing actions corresponding to events in which preferences of multiple entities matter. Also, there are no provisions to take tastes and opinions of multiple people into consideration. Further, there are no systems which may help the users in fulfilling the actions that were selected based on preferences of multiple people.
[0005] Customers who wish to place an order, complete a transaction or clear doubts regarding a product or a service generally try to contact the customer service of the respective organization. Customers' communication with call centers mostly take place over phone calls or e-mails and frequently leads to confusions and errors in communication between the customer and the call center representative, especially with communication network issues, hearing issues and misinterpretation of instructions related to functioning/use of products. Further, there are instances where the call center representative may not understand the needs or requirements of the customer, which leaves them unable to help the customer. Such situations lead to customer dissatisfaction and often result in bad reviews and complaints made by the customer regarding the organization.
[0006] Conventional systems do not help customers when they face confusions while trying to communicate with a call center representative. Further, there is no provision to support and help the call center representative in case they need further help while communicating with the customer.

[0007] Further, in conventional systems, customers talk to the customer service of the business on the phone. Therefore, the current customer service experiences are restricted to a communication wherein the customer can only hear an agent's speech on a telephone line. Such voice based communication has limitations in terms of how well one can express and address issues, and how well one can interpret and understand information presented through such communication.
[0008] An image and/or video are worth 1000 sentences. However, the current customer service experiences are restricted to a communication wherein the user can only hear an agent's speech on a telephone line.
[0009] Thus, the conventional systems failed to solve the above problems, which result in inconveniences to planners and people who are coordinating between multiple persons. Further, with the current increase in the coordinated work between multiple people at residences, offices and other campuses, it is of prime importance to have improved systems.

SUMMARY

[0010] A computer-implemented method for agent to agent communication is disclosed. The method comprises receiving a request from a user via a first virtual agent that represents the user; initiating a communication session between the first virtual agent and a second virtual agent representing a service provider to communicate a desired action of the request; and exchanging, by the first and second virtual agents, one or more proposals to fulfill the received request. The method also comprises negotiating, by the first and second virtual agents, terms for performing the desired action based on the exchanged proposals; executing, by the second virtual agent server, the desired action based on the negotiated terms; and communicating, by the first virtual agent server, a response to the user upon the execution of the desired action.
[0011] Further, a computer-implemented method for presenting a response is disclosed. The method comprises receiving a natural language request from a first user and analyzing the request to determine a desired action and a context of the desired action. The method also comprises based on the analysis, dynamically selecting a particular external application programming interface (API) to fulfill the request and transmitting the request to the selected API and receiving response data from the selected API. The method further comprises integrating the response data into a natural language output that answers the request from the user; and presenting the integrated output to the user.
[0012] Other objects, features, and advantages of the present invention will become apparent from the following detailed description. It should be understood that however, that the detailed description and specific examples, while indicating preferred embodiments of the present invention, are given by way of illustration only and various modifications may naturally be performed without deviating from the present invention.

Description

BRIEF DESCRIPTION OF THE DIAGRAMS

[0013] Embodiments are illustrated by way of example and not limitation in the Figures of the accompanying drawings, in which like references indicate similar elements and in which:
[0014] FIG. **1** depicts an exemplary architecture of a virtual agent server **100** for assisting a user;
[0015] FIG. **2** depicts a system **200** including the virtual agent server **100** for assisting a user, in accordance with an embodiment;
[0016] FIG. **3** depicts a flowchart of an exemplary method **300** for assisting users using a virtual agent server **100**, in accordance with an embodiment;
[0017] FIG. **4** depicts a system **400** including the virtual agent servers **100***a* and **100***b* for assisting a user, in accordance with an embodiment;
[0018] FIG. **5** depicts a flowchart of an exemplary method **500** for optimizing a user's desired

actions, in accordance with an embodiment;
[0019] FIG. **6** depicts a system **600** including the virtual agent server **100** for assisting users, in accordance with an embodiment;
[0020] FIG. **7** depicts a flowchart of an exemplary method **700** for assisting users using a user's mobile device **202**, in accordance with an embodiment;
[0021] FIGS. **8-11** depict exemplary user interfaces for assisting users using a user's mobile device **202**, in accordance with an embodiment; and
[0022] FIG. **12** depicts a system including the virtual agent server **100** for assisting, users in a conversation with a customer service representative **208**, in accordance with an embodiment.

DETAILED DESCRIPTION

[0023] The following detailed description includes references to the accompanying drawings, which form part of the detailed description. The drawings show illustrations in accordance with example embodiments. These example embodiments are described in enough details to enable those skilled in the art to practice the present subject matter. However, it will be apparent to one of ordinary skill in the art that the present invention may be practised without these specific details. In other instances, well-known methods, procedures and components have not been described in detail so as not to unnecessarily obscure aspects of the embodiments. The embodiments can be combined, other embodiments can be utilized or structural and logical changes can be made without departing from the scope of the invention. The following detailed description is, therefore, not to be taken as a limiting sense.
[0024] In this document, the terms “a” or “an” are used, as is common in patent documents, to include one or more than one. In this document, the term “or” is used to refer to a non-exclusive “or,” such that “A or B” includes “A but not B,” “B but not A,” and “A and B,” unless otherwise indicated.
[0025] It should be understood that the capabilities of the invention described in the present disclosure and elements shown in the figures may be implemented in various forms of hardware, firmware, software, non-transitory recordable medium or combinations thereof.

OVERVIEW

[0026] The embodiments disclose techniques used to solve problems in communication between multiple entities with the help of a virtual agent server **100**. A virtual agent server **100** can help a user and a group of people to decide on a task. Sometimes, when there are a number of people involved in an activity, deciding on a single course of action becomes difficult due to the differences in opinions of the people involved. In such a case, a user can use the virtual agent server **100** to help in deciding on an acceptable course of action. The virtual agent server **100** can understand natural speech and will determine preferences of people who are parties to an event. It will then come up with overall best-suited suggestions by taking everyone's preferences into consideration. The user and other people involved can choose whichever suggestions they prefer, and the virtual agent server **100** will execute the selected suggestions based on the feedback of the user and others. Further, the virtual agent server **100** can search for multimedia files that are relevant to the suggestions and share them with the user. The virtual agent server **100** can also communicate with external applications to optimize and execute the user's desired actions such that the optimizations are in favour of the user.
[0027] The disclosed system may be used in any chat, voice or phone-based communication. The virtual agent server **100** can identify multimedia files such as images, audio and video clips that are relevant to the topic of communication. Further, the virtual agent server **100** can stream or share the identified multimedia files with one or more users. Hence, the virtual agent server **100** can be used for event planning, task planning, customer support, and other related applications.
[0028] Further, the embodiments disclose techniques used to solve problems in customer service with the help of virtual agent servers **100**. A virtual agent server **100** can help a user who has contacted a customer service representative.

Virtual Agent System

[0029] FIG. **1** represents an exemplary architecture of a virtual agent server **100**, in accordance with an embodiment. The virtual agent server **100** may include a Natural Language Understanding (NLU) module **102** to understand the speech of the users, a preference module **104** to determine preferences of the users, a location module **106**, a suggestion generation module **108**, a controller module **110**, a security module **112**, a multimedia module **114**, an execution module **116** and an optimization processor module **118**.

[0030] The virtual agent server **100** may receive input identifying a primary user and one or more secondary users, wherein the primary user and the secondary users are parties in an event that needs coordination between the primary and secondary users. Further, the virtual agent server **100** may receive one or more inputs to identify one or more preferences of the primary user and the secondary user. The preferences may correspond to one or more categories of preferences. Subsequently, the virtual agent server **100** may assign one or more relative weightage to preferences within a category and generate one or more suggestions for the primary and secondary users based on the relative weightage. The virtual agent server **100** may then communicate with the primary and secondary users to help them execute their selected suggestion.

[0031] In an implementation, a primary user may communicate with the virtual agent server **100** using their user's mobile device **202***a*. The primary user may request the virtual agent server **100** to initiate communication with one or more secondary users using their mobile device **202***b* in order to include them as a party in the event being coordinated by the primary user. Alternatively, the primary user may contact one or more secondary users by themselves as depicted in FIG. **8**.

[0032] In an implementation, the Natural Language Understanding Module **102** (hereafter called NLU module **102**) may be used by the virtual agent server **100** to understand the natural speech of the primary or secondary users.

[0033] In an implementation, the NLU module **102** may receive the primary and/or secondary user's natural speech as an input. This natural speech may be in the form of audio or text. Further, the NLU module **102** may parse information from the primary and/or secondary user's natural language speech in order to collect information about the primary and/or secondary user's desired action and context of the desired action.

[0034] In an implementation, the input to the NLU module **102** may be received through one or more of the following situations: the primary user may communicate with the virtual agent server **100**, or with one or more secondary users. The input may further include inputs received from the primary or secondary users in previous conversations. The NLU module **102** may generate text from the primary or secondary user's speech. In an implementation, the virtual agent server **100** may label parts of the text of the primary or secondary user. These labels may be used by the virtual agent server **100** to understand the speech of the primary or secondary users.

[0035] In an implementation, for example, in case the user comments “I'm looking for Indian restaurants which are close to my house”, the virtual agent server **100** may label “looking for” as ‘desired action’, “restaurants” as ‘desired event’, “Indian” as ‘desired cuisine’, and “close to my home” as ‘desired distance’.

[0036] The labels for the text may be generated using Hidden Markov Models or Conditional Random Field models. Alternatively, the NLU module **102** may use one or more slot-filling algorithms or manually configured rules or natural language analysis or a machine learning classifier to determine the primary or secondary user's desired action and context from the primary or secondary user's natural speech. Subsequently, the NLU module **102** may communicate the primary or secondary user's desired action and context to other modules of the virtual agent server **100**.

[0037] The preference module **104** be used to determine the preferences of the primary or secondary users. These preferences may be used by the virtual agent server **100** to create relevant suggestions for the primary or secondary users.

[0038] In an implementation, the preference module **104** may receive inputs from other modules of the virtual agent server **100** and process the inputs to determine the preferences of the primary and secondary users.
[0039] In an implementation, the preference module **104** may also determine one or more preferences of the primary user and/or the secondary user based on a conversation between two or more of, the primary user, the secondary user and the virtual agent server **100**. The conversation may be a present conversation or one or more past conversations which may be received from an external database or server. The virtual agent server **100** may identify one or more preferences based on the frequency of certain words used in the conversation. Further, the virtual agent server **100** may assign weightage to the preferences based on the frequency of the words used in the conversation.
[0040] Further, the preference module **104** may communicate with one or more databases to receive a history of the primary or secondary users. The preference module **104** may represent the preferences of the primary or secondary users by creating a matrix of their preferences along one or more parameters.
[0041] In an implementation, a representative location of the entire group of primary and/or secondary users may be derived from a conversation history or from the primary and/or secondary user. In another implementation, a centroid may be computed by laying out the longitude and latitude of the primary and/or secondary users.
[0042] In an implementation, the preference module **104** may convert the strength of the user's preferences into a number. The strength of the user's preferences may be measured by features such as frequency of usage of the preference, tone of voice, pitch of voice, frequency of repeated words, history or social profile of the primary or secondary users, among others. Weightage may be assigned to one or more preferences based on the above mentioned features. A weighted table may be constructed for the primary and secondary users by considering their user preference scores and frequency of the preferences of the primary and secondary users.
[0043] In an implementation, the location module **106** may be used to receive or determine one or more location information related to the primary or secondary users. The virtual agent server **100** may use this location information as a preference to generate suggestions for the primary and secondary users.
[0044] Further, the location module **106** may communicate the determined locations of the primary and secondary users with other modules of the virtual agent server **100**.
[0045] In an implementation, the location of the primary and secondary users may be derived from a previous or present conversation between at least two of the primary user, secondary user, and the virtual agent server **100**. Alternatively, the location of the primary or secondary users may be determined based on the GPS location information of the primary and/or secondary users' mobile devices **202***a*, **202***b*.
[0046] In an implementation, the location information of the primary or secondary users may include one or more current locations where the primary or secondary users are currently present at. Further, the location information may include an address of a preferred location, wherein the virtual agent server **100** has determined, using the preference module **104**, that the primary or secondary user shows preference for the preferred location over other locations.
[0047] In an implementation, the location information shared by the location module **106** with other modules may include an address of one or more locations of consensus. The virtual agent server **100** may be configured to determine one or more locations of consensus as explained below. These locations of consensus may be determined based on the location information corresponding to the primary user or the secondary user. Further, the suggestions may also be based on the locations of consensus.
[0048] In an implementation, the suggestion generation module **108** may take into consideration the different preferences of the primary and secondary users to generate suggestions that may be

used by the primary and secondary users.
[0049] In an implementation, the suggestion generation module **108** may receive information from the other modules of the virtual agent server **100**. The suggestion generation module **108** may receive location information from the location module **106** and preference information from the preference module **104**. Further, the suggestion generation module **108** may receive information regarding the desired action and context of the desired action of the primary user from the NLU module **102**.
[0050] In an implementation, the suggestion generation module **108** may take the preferences of the primary and secondary users; and receive a user profile and/or a user history from one or more databases. Further, the suggestion generation module **108** may process the received information to generate suggestions applicable to the primary and secondary users; and rank the generated suggestions so that the virtual agent server **100** may suggest the higher ranked suggestions before the lower ranked suggestions. To generate suggestions, the preferences of the primary or secondary users may be converted into a single user vector for search and recommendations use cases. The user vector may comprise elements that hold information representing one or more of the location of the various users, set of preferences of the primary or secondary users, and history of the users, amongst others. A group user vector may then be used to rank the elements, which may be further used during various actions such as search and suggestion, among others.
[0051] As an example, the suggestion generation module **108** may create a user vector for a user 'Dan' as shown below. The desired action and context of user Dan may be to locate a restaurant where he may have lunch. One or more information about user Dan received from databases and present and previous conversations comprising location information and food preferences may be included in the elements of the user vector as follows:
[0052] User Dan: [0053] location: {37.4292 N, 122.1381} [0054] preferences: {Italian 4.1}, {Thai 3.5}, {Indian:2.0} [0055] favourite dishes: {Lasagne 10.1}, {Drunken Noodles 8.5}
[0056] In an implementation, in case the group size comprising the primary and secondary users increases, there may not be many preferences that match the interest of all users. In this case, the suggestion generation module **108** may construct a query with less number of restrictions to get a higher number of suggestion results. This query may result in one or more suggestions that may be preferred by more primary or secondary users. Further, the suggestion generation module **108** may rank the results of this query. Subsequently, the ranked results may be suggested to the primary and secondary users. The suggestion generation module **108** may use one or more of the following methods to rank the suggestions.
[0057] In an implementation, the first iteration of ranking may be done by an algorithm such as a simple Euclidean distance between a combination of user and query vector and result vector. Further, the second iteration of ranking may use an expensive Gradient Boosted. Decision Tree learn to rank algorithm with features from the user vector.
[0058] In another implementation, the ranking may be generated by communicating with one or more contacts of the primary or secondary users and requesting them for their advice or opinion to identify one or more suggestions. The contacts of the primary user may be retrieved from one or more social networks or from a contact list on the primary user's mobile device **202***a*. In case the primary user has not given any explicit input regarding which contact's opinion is to be requested, the virtual agent server **100** may rank the contacts of the primary and/or secondary users in social network/contacts to optimize the requesting process for a timely feedback from the contacts. Further, one or more contacts may be selected based on the ranking to receive their opinion or advice regarding the preferences of the primary or secondary users. In an implementation, ranking of social contacts may use a Gradient Boosted Decision Tree learn to rank algorithm.
[0059] In an implementation, the virtual agent server **100** may decide to ask the primary user for feedback in case the primary user has explicitly asked the virtual agent server **100** to do so. Alternatively, the virtual agent server **100** may run a machine learning classifier to deter mine if it

needs to contact other users to get feedback on the restaurant in the conversation.
[0060] In an implementation, the suggestion generation module **108** may be configured to determine one or more locations of consensus. These locations of consensus may be locations which may be acceptable by a majority of the primary and secondary users. These locations of consensus may be determined based on location information corresponding to the primary user or the secondary user. Further, the suggestions generated by the suggestion generation module **108** may also be based on the locations of consensus.
[0061] In an implementation, the suggestion generation module **108** may further communicate with other external systems **404**, such as social network platforms **210**, related to the primary or secondary users to collect information that may help in generating suggestions. In an embodiment, the suggestion generation module **108** may communicate with a calendar application on the primary and/or secondary user's mobile device **202***a*, **202***b* to determine the time and date of the user's free time or appointments. Further, the suggestion generation module **108** may communicate with external applications such as Google maps to determine estimated time, locations and routes. Such information collected from external systems **404** may be used to generate suggestions for the primary or secondary users.
[0062] As an example, the primary user may like an Indian restaurant which is 5 miles away. The virtual agent server **100** may have contacted one or more external systems **404** and determined that there is an accident on route to the restaurant. Further, the virtual agent server **100** may have communicated with the calendar application of the primary and/or secondary user's mobile device **202***a*, **202***b* and determined that the primary or secondary user has a meeting scheduled in 10 minutes. In this case, the virtual agent server **100** may alert the user that it may be tough for the primary or secondary user to attend the meeting if he chooses that Indian restaurant. The virtual agent server **100** may generate a suggestion to the primary or secondary user such as “I can show you the driving directions to the restaurant you chose but it looks like you have a meeting in another 10 minutes and there is an accident on the way. You might be late to the meeting if we proceed”.
[0063] In an implementation, the generated suggestion may be communicated to the primary or secondary users as one or more of a spoken dialog, a text message or an e-mail, among others. The generation of the suggestion may be alerted to the primary or secondary user by using a spoken dialog, a ring tone or a phone vibration.
[0064] In an implementation, the controller module **110** may be used to coordinate actions between the other modules of the virtual agent server **100**. The controller module **110** may comprise the main instructions to be followed by the virtual agent server **100** while taking into consideration the preferences of the multiple users and suggesting actions to be chosen by the users which may be further implemented by the virtual agent server **100**.
[0065] In an implementation, the controller module **110** may determine the best alerting mechanism to be used for alerting the primary or secondary user in case of a generated suggestion. The controller module **110** may use a combination of rules configured against events and a Machine Learning/Predictive classifier to determine the type of alert that may be used in a given scenario. Some variables that may be used as input into the algorithm/rules are: [0066] a) Is the user moving as measured by the phone location? [0067] b) What is the type of the message? Is it a date meet or lunch meet with co-workers? [0068] c) Is the user in an important meeting? [0069] d) How many people are currently engaged with the user? [0070] e) How far is the phone to the user? Is the phone in a pocket or on a table or in hand?
[0071] In an implementation, the answers to one or more of these questions may be measured by tracking the time lag between changes in the position of a user's mobile device **202**. For instance, in an example it may take an approximately constant time for the primary or secondary user to move user's mobile device **202** from his pocket to his hand and complete an unlock action or click action on the screen of the user's mobile device **202**. The virtual agent server **100** may keep a track of time

periods when the user may have put the user's mobile device **202** down. Alternatively, the virtual agent server **100** may instruct a user's mobile device **202** sensors to log metadata information about the surroundings of the user's mobile device **202**.
[0072] In an implementation, the controller module **110** may determine that the virtual agent server **100** may have stopped communication with the user's mobile device **100**. This may have occurred either due to network failure or server malfunctioning. In such scenarios, in an implementation there may be a virtual agent client **212** on the primary and/or secondary user's mobile device **202***a*, **202***b* wherein the virtual agent client **212** may have a preconfigured time period to receive a response from the virtual agent server **100**. In case the virtual agent server **100** does not respond within the preconfigured time period, the virtual agent client **212** may make decisions with the available information using one or more software codes and data lookup tables stored locally on the user's mobile device.
[0073] In an implementation, the virtual agent client **212** on the primary and/or secondary user's mobile device **202***a*, **202***b* may push data and logical steps needed for the user to make decisions about the generated suggestions in case the virtual agent server **100** has stopped communication with the user's mobile device **100** due to low or no network. The virtual agent client **212** on the primary and/or secondary user's mobile device **202***a*, **202***b* may push the data and the logical steps of the virtual agent process either as a periodic process or after running a machine learning classifier using features such as user location and network availability in the location.
[0074] In an implementation, the code may be a JavaScript code for the virtual agent client **212** to execute, wherein data may be stored in a nested hashmap structure. The hashmap data structure may assign unique keys to values to be stored. The values may be stored in a map by forming a key-value pair. The values may be retrieved at any point by passing the assigned key through an appropriate command.
[0075] As an example, consider a user trying to find a restaurant when the virtual agent server **100** may stop communicating with the primary and/or secondary user's mobile device **202***a*, **202***b:*
[0076] User: “I am looking for an Indian Restaurant” [0077] Virtual agent server **100**: “It looks like you are around Fremont. Based on your food taste, you might like food from the restaurant Shalimar” (Virtual agent server **100** may show pictures of dishes at Shalimar in the conversation using implementations discussed above) [0078] User: “Call Shalimar”
[0079] Further, the virtual agent server **100** may stop communication with the user. The virtual agent client **212** may determine that there is no connection to the virtual agent server **100**. The virtual agent client **212**, based on the context of user location and network type might run a Machine Learning classifier algorithm and execute the action instructed by the user. In this case, when the user says “Call Shalimar”, the virtual agent client **212** may forward the phone number of the selected restaurant and the code to call the phone number to the user's mobile device **100**. The user's mobile device **100** may then initiate a call to the restaurant “Shalimar” when the user says “Call Shalimar”, even though it can't communicate with the virtual agent server **100**.
[0080] The discussed implementations may be applicable to all customer-to-business interactions including one or more of buying products on retail websites and buying airline tickets, among others. The disclosure discusses techniques in the context of natural speech interaction between users and virtual agent servers **100**. These techniques may be easily extended to natural text interactions as seen in conventional chat applications.
[0081] In an implementation, the security module **112** may be used to maintain the security of the user. The virtual agent server **100** may ensure that the security of the primary or secondary users is not breached. This may be accomplished by not sharing any unnecessary or extra information while completing the primary or secondary users' request while matching the questions against slots. Further, the security module **112** may restrict revealing any extra information unless explicitly approved by the primary or secondary user.
[0082] In an implementation, the multimedia module **114** may be used by the virtual agent server

100 to search and share multimedia files and suggestions with the primary or secondary users to give them a better understanding of the generated suggestions. The virtual agent server **100** may stream one or more multimedia such as images, files, audio or video, among other multimedia related to context of the conversation. In an implementation, this may be accomplished by training a Convolutional Neural Network with tagged multimedia files.

[0083] In an implementation, the multimedia module **114** may be used by the virtual agent server **100** to identify one or more multimedia files that the user may be interested in based on the user's intent from one or more text-based chat and/or phone communication. Further, the multimedia module **114** may share the identified multimedia files with one user and receive that user's feedback regarding the identified multimedia files. Subsequently, the multimedia module **114** may share one or more of the identified multimedia files with another user based on the feedback received. The multimedia files and textual tags may be stored in a database as explained below in FIG. **6**.

[0084] In an implementation, the execution module **116** may be used by the virtual agent server **100** to receive and process feedback from the primary and/or secondary users corresponding to the generated suggestions. Further, the execution module **116** may execute one or more of the suggestions that have been selected by the primary or secondary users using the feedback and the context derived from the natural speech of the primary or secondary users. The virtual agent server **100** may receive feedback from the primary or secondary users regarding the generated suggestion that they may have selected. In case the selected suggestion may be executed without contacting any external system **404**, the execution module **116** may directly initiate the execution of the selected suggestion.

[0085] In an implementation, there may be situations wherein the virtual agent server **100** may need to communicate with one or more external systems **404** in order to execute the selected suggestion. In this case, the execution module **116** may first determine the type of external system **404** it needs to communicate with.

[0086] In an implementation, the external system **404** may be a software application comprising a web application **206** or it may be a natural person such as a customer service representative **208**. In case the execution module **116** has determined that it needs to contact an external web application **206**, the virtual agent server **100** may call an external Application Programming Interface and execute the selected suggestion with the parsed parameters from the natural language speech of the primary or secondary users. In case the external system **404** doesn't comprise of an Application Programming Interface, the execution module **116** may need to search for a phone number or customer service application related to the external system **404** in order to execute the suggestion selected by the primary or secondary users.

[0087] In both cases, the execution module **116** may need to collect one or more parameters related to the context of the execution. These parameters may include one or more of conversation summary, user's first name and last name, time, date, location and operating system of the primary and/or secondary user's mobile device **202***a*, **202***b*, among others. The possible ways in which this information may be collected are follows:

[0088] a) Question and Answering problem: In this implementation, the context, conversation history and inference from the conversation are used for answering a query regarding the value of the parameter for the Application Programming Interface. Attributes such as meta-data of the Application Programming Interface and the collected parameters are used to construct the question. In an implementation, this may be done by using manually developed pattern matching rules. Alternatively, a combination of Memory Networks, Recurrent Neural Network models with pattern matching rules may be used to answer questions for the Application Programming Interface. Crowdsourcing platforms may be used to create training data for training neural networks.

[0089] b) Slot filling Algorithms: In this implementation, slot filling algorithms may be used to figure out the parameters for the Application Programming Interface after passing the conversation context through one or more entity recognition algorithms.

[0090] c) Manual rules: Manual rules may be configured against the textual patterns using regular expressions and grammars to parse out the parameters.
[0091] In another implementation, the virtual agent **100** may determine the parameters for the Application Programming Interface call from the natural language of the primary and/or secondary user, by storing, searching, and inferring information from the user dialog history and runtime context. In an implementation, the past “n” dialogs are retrieved and one or more words represented by tokens may be annotated with labels using entity name recognition techniques. Further, inference rules may be run on the annotated dialogs to derive any extra information, following which, parameters for the Application Programming Interface call are gathered.
[0092] In an implementation, the optimization processor module **118** may be used by a virtual agent server **100***a* in case it needs to communicate with another virtual agent server **100***b* representing an external system **404** to execute a user's actions, as depicted in FIG. **4**. In this case, the optimization processor module **118** may be used to determine optimization options corresponding to the user's desired actions, communicating the optimization options with the other virtual agent server **100***b*, if allowed by the user. Further, the optimization processor module **118** may be used to determine which optimization option is to be executed.
[0093] FIG. **2** depicts a system **200** comprising a virtual agent server **100** which may communicate with primary and secondary users' mobile devices **202***a* and **202***b*; a web application **206**, a customer service representative **208**, a social network webpage **210**, and a virtual agent client **212** as depicted.
[0094] In an implementation, the primary and secondary user's mobile device **202***a*, **202***b* may include mobile phones, palmtops, PDAs, tablet PCs, notebook PCs, laptops and computers, among other computing devices. In an embodiment, the primary and secondary user's mobile device **202***a*, **202***b* may include any electronic device equipped with a browser to communicate with the virtual agent server **100**. The user's mobile device **202** may belong to a primary or a secondary user who may use it to communicate with the virtual agent server **100**.
[0095] In an implementation, the primary and/or secondary user's mobile device **202***a*, **202***b* may include a virtual agent client **212** which may communicate with the virtual agent server **100** and share inputs related to the primary or secondary user with the virtual agent server **100**. Further, the virtual agent client **212** may execute one or more instructions in case communication with the virtual agent server **100** has failed, as described in embodiments above.
[0096] In an implementation, the virtual agent server **100** may be implemented in the form of one or more processors with a memory coupled to the one or more processors with one or more communication interfaces. The virtual agent server **100** may communicate with one or more external systems **404** and one or more primary and/or secondary user's mobile device **202***a*, **202***b* through a communication network. It may be noted that some of the functionality of the virtual agent server **100** may be implemented in the primary and/or secondary users' mobile devices **202***a* and **202***b*.
[0097] In an implementation, the communication network may include a wired network, a wireless network, or a combination of wired network and wireless network. For example, the communication network may include local area network, wide area network, and metropolitan area network, among others.
[0098] In an implementation, the social network webpage **210** may comprise of any webpage related to a social networking application used by the primary and/or secondary users through their primary and secondary users' devices **202***a* and **202***b*.
[0099] In an implementation, the virtual agent server **100** may communicate with a primary user through their primary user's mobile device **202***a* and secondary users through their user's mobile devices **202***b*. The virtual agent server **100** may receive inputs from the primary and/or secondary users, and determine their desired action and a context of their desired action. Further, the virtual agent server **100** may determine preferences of the primary and/or secondary users, and may

generate suggestions for the primary and/or secondary users. In case the primary and/or secondary users select a suggestion, the virtual agent server **100** may communicate with one or more external systems **404** such as the web application **206** or a natural person such as the customer service representative **208** in order to execute the selected suggestion.

[0100] In an implementation, the virtual agent server **100** may be required to contact the customer service representative **208** in case options are not available. The customer service for the business may be handled by a human i.e. a customer service representative **208** or by a virtual agent server **100**. In both cases, the virtual agent server **100** may need to understand a natural language interaction wherein the virtual agent server **100** operating on behalf of the user may be required to answer questions about the user's desired action. In an implementation, this may be accomplished by parsing the natural language evaluating probabilities against configured slots and providing one or more answers to the customer service representative **208** or the virtual agent server **100**. The virtual agent server **100** may ensure that the transaction is acknowledged. The acknowledgement may then be communicated to the user in the form of a message or notification. As an example, in case the virtual agent server **100** was required to contact the customer service representative **208** of a hotel, the virtual agent server **100** may converse with the customer service representative **208** using natural language to reserve bookings on behalf of the user as instructed, and receive an acknowledgement corresponding to the booking. Further, virtual agent server **100** may communicate the following message to the user through the primary user's mobile device **202***a* "The hotel bookings at Hilton for 2 days have been confirmed".

[0101] In an implementation, the virtual agent server **100** may receive inputs from a current conversation between a primary user and a customer service representative **208**. The virtual agent server **100** may convert the conversation to text to generate one or more metadata such as information about the speaker and emotional response of the speaker, among others. The annotated text may then be analysed for one or more actions such as placing an order/transaction, complaints, bad customer service etc. This analysis may be done using one or more of slot filling algorithms, manually configured rules, natural language analysis for pre-configured actions or a Machine Learning classifier. The actions may then be completed by a virtual agent server **100** by calling external API's and/or making phone calls.

[0102] FIG. **3** depicts a method **300** for a virtual agent server **100** to determine one or more preferences of primary or secondary users to generate suggestions and execute the selected suggestions, in accordance with an implementation. The method **300** may be implemented by a server or an application on a primary and/or secondary user's mobile device **202***a*, **202***b*.

[0103] In an implementation, the virtual agent server **100** may be instructed by the primary user to include one or more secondary users in a conversation between the primary user and the virtual agent server **100**. The virtual agent server **100** may be configured to obtain input identifying the secondary user from one or more of a phone number of the secondary user, voice characteristics of the secondary user or metadata corresponding to the secondary user, among others. The virtual agent server **100** may communicate with one or more external servers to obtain metadata corresponding to the secondary user.

[0104] At step **302**, inputs from the primary and secondary users and their location information may be received or determined by the various modules included in the virtual agent server **100**. Further, using previous and present inputs, the virtual agent server **100** may determine one or more preferences of the primary and secondary users as depicted at step **304**. The user inputs and their preferences may be used by the virtual agent server **100** to determine one or more desired actions and the context of the actions desired by the primary and/or secondary users.

[0105] The method at step **306** comprises of receiving inputs from contacts of the primary user, regarding their advice or opinions related to the preferences of the primary user. At step **308**, the virtual agent server **100** may process the preferences of the primary and/or secondary users to generate a broad query. The result of this broad query may result in one or more suggestions that

may correspond to the preferences of the primary and/or secondary users. Further, as shown in step **310**, the virtual agent server **100** may share one or more multimedia files with the primary and/or secondary users, wherein the shared multimedia files may be related to the generated suggestions and/or preferences of the primary and/or secondary users.

[0106] After sharing the generated suggestions, the virtual agent server **100** may receive feedback related to the suggestions from the primary and/or secondary users as shown at step **312**. The virtual agent server **100** may process the received feedback to determine whether any of the suggested generations were selected by the primary and/or secondary users. In case the primary and/or secondary users did not agree on finalizing a suggestion, the virtual agent server **100** may generate further suggestions based on the feedback of the primary/secondary users as shown in step **308**. In case one or more suggestions were selected by the primary and/or secondary users, the virtual agent server **100** may execute one or more actions based on the received feedback as shown at step **314**.

[0107] FIG. **4** depicts a system **400** including a virtual agent server **100***a* assisting a user and a virtual agent server **100***b* assisting an organization, in accordance with an embodiment. In an embodiment, a user's actions may be executed by a virtual agent server **100***a* and an external system **404**'s actions may be executed by a virtual agent server **100***b*. The virtual agent server **100***a* may communicate with the virtual agent server **100***b* through an external application programming interface. In an embodiment, this communication may be in the form of one or more of natural speech interaction or machine understandable language, among others.

[0108] In an implementation, the virtual agent server **100***a* may receive the user's speech or chat **402** as an input from the user or other modules of the virtual agent server **100**. Subsequently, the virtual agent server **100***a* may use the NLU module **102** to process the input to determine an action desired by the user and context of the action. The context and user's desired action may be communicated to the virtual agent server **100***b* and may be incorporated during execution of actions.

[0109] In an implementation, the virtual agent server **100***a* may determine one or more plans of action that may optimize the desired actions of the user. These optimization options may be communicated with the virtual agent server **100***b* representing an external system **404**. The user may determine which information may be shared by using the security module **112**.

[0110] In an implementation, the virtual agent server **100***b* may determine one or more optimization options related to the external system **404** and communicate them to virtual agent **100***a*. The virtual agent server **100***a* of the user may contact the virtual agent server **1001**) of the external system **404** through an external application programming interface call. The virtual agent server **100***a* may communicate the user's desired actions to the virtual agent server **100***b*. The two virtual agent servers **100***a* and **100***b* may communicate with each other to execute the desired action of the user while implementing one or more optimization options.

[0111] In an implementation, the two virtual agent servers **100***a* and **100***b* may communicate using natural language speech which may be understood by the two virtual agent servers **100***a* and **100***b*. The user's context and desired actions may be included as one or more parameters in the speech. Alternatively, the virtual agent servers **100***a* and **100***b* may communicate via one or more application programming interfaces, wherein the user's context and user's desired action are shared as parameters in the application programming interface call.

[0112] In an implementation, the virtual agent servers **100***a* and **100***b* may communicate with one or more databases to receive further information corresponding to the user and the external system **404**.

[0113] In an implementation, the user's speech may be in the form of natural language and may comprise textual words from the user received from the current conversation. In an implementation, the user's input may be taken from previous “m” conversations, where “in” is manually configured or tuned using Machine Learning for an application. The NLU module **102**

may assign one or more weights to the tokens (individual words) in the speech context using Term Frequency Inverted Document Frequency (tfidf) and the recency of the communication session. The speech context may also include explicit inputs or inferences from previous speech interaction sessions decayed using recency of occurrence.

[0114] In an implementation, the virtual agent server **100***b* may be configured to receive and process the action desired by the customer and the context of the action to obtain input for executable action.

[0115] In an implementation, the virtual agent servers **100***a* and **100***b* may be configured to exchange optimization options of the user and the external system **404**, respectively, with each other, where the optimization options correspond to the execution of the action. Further, the virtual agent servers **100***a* and **100***b* may identify if a common optimization option exists between the optimization options shared by the virtual agent servers **100***a* and **100***b*. Subsequently, the virtual agent servers **100***a* and **100***b* may execute one or more actions as per the common optimization option, in case it exists.

[0116] In an implementation, the system **400** may be configured to execute the action as per the preference of the customer in case a common preference does not exist.

[0117] In an implementation, as an example, consider a Virtual agent server **100***b* (hereafter called VB) handling the customer service of a business. A customer may order food using their own virtual agent server **100***a* (hereafter called VU). VU may determine to place an order for one or more dishes on behalf of the user. VU may contact VB and initiate a communication session through an application programming interface call or by conversing through dialogs. The virtual agent servers VB and VU may use natural language understanding techniques to understand each other and to determine further actions from the conversation.

[0118] The payment for the order may be transferred through an online bank transaction using one of the customer's bank accounts. VB may request payment information from VU to complete the order. VU may identify one or more manually configured rules to optimize the payment on behalf of the user. As an example, VU may identify a rule to optimize for one or more parameters such as credit score, cash rewards and offers, among others, which may be applicable to the user's transaction. The manual rule may be configured either by taking the user's feedback or through VU, which may configure default rules for the users. VB may comprise a rule which instructs it to prefer a credit card that charges the least transaction fees. VB may configure the default rule for the merchant. In case the user allows freedom to and VU agent to complete the transaction, VU may optimize the transaction for the user. Similarly, VB may optimize for the business.

[0119] In an implementation, VB may process one or more information corresponding to the user to determine whether they are flexible on the type of card to use to complete the transaction. In case VB and VU belong to the same entity, they might coordinate a transaction which is optimized for both the user and the business, after optimizing on their respective customers. For example, VU may decide to use any one of two credit cards C1, C2 belonging to the user in case they both comprise the same cash rewards and don't show any adverse impact on the credit score of the user. VB may prefer to complete the transaction with credit card C1 in case it charges less on the transaction fees. Hence, VB may negotiate with VU to determine whether credit card C1 may be used instead of C1. VB may accomplish this by using a manual rule configured against payment transaction and requesting VU for information regarding all the credit card types owned by the customer. VU may share the user's information about the credit cards that may be use to complete the transaction after getting an approval from the user to share the information. VB may ask VU to use C1 over C2 for completing the transaction, by passing a list of cards it would prefer to use.

[0120] In an implementation, this may be done by using an application programming interface wherein the preferred list is passed as a list ordered by preference of credit cards. VU may then determine whether to accept VB's request regarding the preferred credit card. In an embodiment, a virtual agent server **100** may represent VU and VB and may do global optimization for both VB

and VU after optimizing on their respective customers.
[0121] The discussed optimization principles may be applied to one or more types of actions including transactions such as merchandising on e-commerce platform, negotiating charges for a spa service, and negotiating hotel stay rates, among others.
[0122] FIG. **5** may depict a flowchart of an exemplary method **500** for optimizing a user's desired actions, in accordance with an embodiment. This method may be used in case a first virtual agent server **100** and a second virtual agent server **100** that represent a user and a business respectively. At step **502**, the user's desired actions may be determined by the first virtual agent server **100** from one or more inputs corresponding to the user. Further, the first virtual agent server **100** may contact the second virtual agent server **100** to communicate the user's desired actions with the second virtual agent server **100**.
[0123] At step **504**, the first virtual agent server **100** may determine one or more optimization options preferred by the user, and the second virtual agent server **100** may determine one or more optimization options preferred by the business. Subsequently, one or more of the optimization options may be communicated between the first agent and the second virtual agent server **100** which may determine whether any optimization options are present that are common between the optimization options determined by the first and the second virtual agent server **100** as shown at step **506**.
[0124] In case no common optimization options are present, the first virtual agent server **100** and the second virtual agent server **100** may proceed to execute the user's desired actions according to the instructions or context of the user, as shown at step **508**. In case there are one or more common optimization options present, the first virtual agent server **100** and the second virtual agent server **100** may determine which common optimization option to execute. Further, the first virtual agent server **100** and the second virtual agent server **100** may execute the user's desired action according to the selected common optimization as shown at step **510**.
[0125] FIG. **6** may depict a system **600** including the virtual agent server **100** which may share one or multimedia files with users, in accordance with an embodiment. The system may include a primary user's mobile device **202***a*, a secondary user's mobile device **202***b*, and a multimedia database **602**.
[0126] The primary user may initiate communication with a secondary user. In an implementation, there may be more than one secondary user using a secondary user's mobile device **202***b* to communicate with the primary user. The multimedia database **602** may comprise one or more multimedia files, with one or more tags associated with one or more of the multimedia files. Further, the virtual agent server **100** may receive input from the primary and the secondary user, and may process the received input to understand a context of the input. Further, the virtual agent server may identify one or more multimedia files based on the context and the tags. Subsequently, the virtual agent server **100** may share one or more of the identified multimedia files with one or more of the primary and secondary user to enable better understanding between the primary and secondary user.
[0127] In an implementation, the tags may be natural language tags. Further, a convolutional neural network may be used to tag multimedia files. Alternatively multimedia files may be tagged using one or more crowdsourcing platforms. The textual tags and meta-data corresponding to the multimedia files may be stored in a database or an inverted index to optimize the retrieval performance of the virtual agent server **100**.
[0128] In an implementation, the multimedia module **114** may receive one or more of conversation context, user intent, dialog summary and speech of the user converted into text, among other information, from the NLU module **102** through the primary and/or secondary user's mobile devices **202***a* and **202***b*. The conversation context and the dialog summary may then be used to identify tagged multimedia files related to the user's intent by using information retrieval techniques. Further, the virtual agent server **100** may share the identified multimedia with one or

more of the primary and/or secondary users by sharing the multimedia files through their user's mobile device **202***a*, **202***b* as explained below.

[0129] FIG. **7** depicts a flowchart of an exemplary method **700** for sharing multimedia files with primary and/or secondary users through their user's mobile device **202***a*, 2026, in accordance with an embodiment.

[0130] The virtual agent server **100** may receive one or more inputs from the first user and the second user as depicted in step **702**. Further, the virtual agent server **100** may process the received input to determine a context of the input at step **704**.

[0131] At step **706**, the virtual agent server **100** may identify one or more multimedia files from a multimedia database **602** by matching the context of the user to tags assigned to one or more multimedia files. Subsequently, the virtual agent server **100** may share at least one of the identified multimedia files with one of the two users at step **708**.

[0132] At step **710**, the virtual agent server **100** may receive feedback corresponding to at least one of the identified multimedia files from the user with whom it is shared. The feedback may contain a selection or ranking of the identified multimedia files. Further, as shown at step **712**, the virtual agent server **100** may share one or more of the selected or ranked multimedia files with the other user among the two users based on the feedback received.

[0133] In an embodiment, the first user may be a customer and the second user may be a customer service representative **208**. Further, the identified multimedia files may be shared with the customer who may share their feedback regarding the identified multimedia files with the virtual agent server **100**. The identified multimedia files may have been ranked for natural language context by the customer and may be shared with the customer service representative **208** who did not receive any multimedia files yet.

[0134] In an embodiment, the identified multimedia files may be shared with the customer service representative **208** who may share their feedback regarding the identified multimedia files with the virtual agent server **100**. The identified multimedia files may have been ranked for natural language context by the customer service representative **208** and may be shared with the customer who did not receive any multimedia files yet. The sharing of the relevant multimedia files may aide in better understanding between the user and the customer service representative **208**.

[0135] In an implementation, the virtual agent server **100** may be further configured to share at least one of the identified multimedia files in a current conversation between the first user and the second user.

[0136] In another implementation, a single image may be selected from the search results to stream to the users. The virtual agent server **100** may use one or more of a predictive algorithm or machine learning to share one or more multimedia files.

[0137] In another implementation, the multimedia file may include at least one of an image, an audio file, a video file or a document.

[0138] FIG. **8** depicts an exemplary user interface which may show the virtual agent client **212** in the primary user's mobile device **202***a*. The virtual agent client **212** may include one or more options to be selected by the user. The options may include ‘Add a friend’ wherein a primary user may choose to add one or more secondary users to a conversation. An ‘Execute action’ function may be used by the user to instruct the virtual agent client **212** about one or more actions desired by the user. In FIG. **8**, a primary user has selected the ‘Add a friend’ option. The primary user may view a pop-up window with two options as shown. The primary user may choose the ‘Select friend’ option in case the primary user chooses to add a friend to the conversation by using the virtual agent client **212**. Alternatively, the primary user may choose the ‘Select friend’ option in case the primary user chooses to add a friend to the conversation manually.

[0139] FIG. **9** depicts an exemplary user interface which may show the virtual agent client **212** present in the primary user's mobile device **202***a* after a friend (secondary user) of the primary user has been added to a conversation.

[0140] FIG. **10** depicts an exemplary user interface which may show a conversation between the primary user and secondary user on the primary user's mobile device **202***a*. In this conversation, the two users are discussing where to have lunch. The primary user and secondary user may decide to eat burgers at a restaurant near the primary user's house as shown.

[0141] The virtual agent client **212** may determine the context, user's desired actions and preferences from the conversation between the primary and secondary user's conversation as follows. The user's desired action may be to locate a restaurant, the context may be to have lunch, and the preferences may be a restaurant close to the primary user's house where burgers are available.

[0142] FIG. **11** depicts an exemplary user interface showing a virtual agent client **212** that has determined a suggestion for the primary and secondary user. The suggestion was created using the preferences of the primary and secondary user. This suggestion may be a restaurant “Burger Prince”. A multimedia file comprising an image depicting the restaurant may be shared with the primary user's mobile device **202***a*. Additionally, the virtual agent client **212** may communicate with an external map application to share a map depicting a route from the primary user's house to the suggested restaurant, as shown.

[0143] FIG. **12** depicts a system including the virtual agent server **100** for assisting users in a conversation with a customer service representative **208**, in accordance with an embodiment. The system depicts a user using their user's mobile device **202** in a conversation with a customer service representative **208**. The virtual agent server **100** may receive the conversation as an input. Further, the virtual agent may process and comprehend the input in order to send one or more multimedia files related to the conversation. In this case, the user may have a doubt regarding what to order. Consequently, the user may ask the customer service representative **208** about the best burger available at “Burger Prince” restaurant. The virtual agent server **100** may comprehend this query and conduct an appropriate search. Further, the virtual agent server **100** may share one or more multimedia files with the customer service representative **208**, who may select one image representing the most popular burger at Burger Prince. Further, the customer service representative **208** may share the image with the user on their user's mobile device **202** as depicted in FIG. **12**.

CONCLUSION

[0144] The present invention overcomes the drawbacks of virtual assistant systems, by taking into consideration preferences of multiple users and generating suggestions which may be suitable to most the users. Further, the system helps to enable better understanding between a customer and customer service representative. Additionally, the system helps in selection and sharing of multimedia files between two users or between a customer service representative and a user. Thus, the present invention as discussed in this document with respect to different embodiments will be advantageous at least in optimizing the process of event coordination and execution of actions selected by multiple users. Further, it is advantageous in providing better user experience and decreasing time and effort required by users in coordinating an event. It is also advantageous in helping the user in the process of conversing with a customer representative. Additionally, the system may optimize actions for the user while trying to execute actions on behalf of the user. Additional advantages not listed may be understood by a person skilled in the art considering the embodiments disclosed above.

[0145] It shall be noted that the processes described above are described as sequence of steps; this was done solely for the sake of illustration. Accordingly, it is contemplated that some steps may be added, some steps may be omitted, the order of the steps may be re-arranged, or some steps may be performed simultaneously.

[0146] Although embodiments have been described with reference to specific example embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the system and method described herein. Accordingly, the specification and drawings are to be regarded in an illustrative

rather than a restrictive sense.

[0147] Many alterations and modifications of the present invention will no doubt become apparent to a person of ordinary skill in the art after having read the foregoing description. It is to be understood that the phraseology or terminology employed herein is for the purpose of description and not of limitation. It is to be understood that the description above contains many specifications; these should not be construed as limiting the scope of the invention but as merely providing illustrations of some of the personally preferred embodiments of this invention. Thus, the scope of the invention should be determined by the appended claims and their legal equivalents rather than by the examples given.

Claims

1. A computer-implemented method comprising: receiving a request from a user via a first virtual agent that represents the user; initiating a communication session between the first virtual agent and a second virtual agent representing a service provider to communicate a desired action of the request; exchanging, by the first and second virtual agents, one or more proposals to fulfill the received request; negotiating, by the first and second virtual agents, terms for performing the desired action based on the exchanged proposals; executing, by the second virtual agent server, the desired action based on the negotiated terms; and communicating, by the first virtual agent server, a response to the user upon the execution of the desired action.

2. The method of claim 1, further comprising receiving a context associated with the received request, wherein the context comprises at least one of: location of the user, preferences of the user, schedule, or device state.

3. The method of claim 1, wherein negotiating comprises: sending a proposal from the first virtual agent; receiving a counter-proposal from the second virtual agent; and selecting an acceptance from the first virtual agent for the received counter-proposal.

4. The method of claim 1, wherein negotiating and executing the action are performed without additional user input once the request is received.

5. The method of claim 1, further comprising the second virtual agent generating a prioritized set of proposals based on optimization preferences of the service provider, and the first virtual agent evaluating the prioritized set in view of the preferences of the user.

6. The method of claim 2, wherein the first virtual agent communicates with the second virtual agent through an Application Programming Interface (API) call, and wherein the desired action and the context are provided as parameters of the API call.

7. The method of claim 1, wherein negotiating comprises: sending a proposal from the first virtual agent; receiving a counter-proposal from the second virtual agent; and selecting the proposal from the first virtual agent when no common optimization option exists between the proposal and the counter-proposal.

8. A computer-implemented method comprising: receiving a natural language request from a first user; analyzing the request to determine a desired action and a context of the desired action; based on the analysis, dynamically selecting a particular external application programming interface (API) to fulfill the request; transmitting the request to the selected API and receiving response data from the selected API; integrating the response data into a natural language output that answers the request from the user; and presenting the integrated output to the user.

9. The method of claim 8, wherein dynamically selecting the API comprises using a slot filling algorithm or a question answering model to determine the context.

10. The method of claim 8, wherein the context includes at least one of the user's location, device state, or prior conversation utterances, and wherein the API selection is based on the context.

11. The method of claim 8, wherein selecting the application programming interface comprises generating parameters for the API by retrieving past “n” dialogs, annotating words using entity

name recognition techniques, and applying inference rules on the annotated dialogs.

12. The method of claim 8, further comprising: determining a type of external system to communicate with based on the context of the request, parsing parameters from the request, and transmitting the parameters to the selected API associated with the external system.

13. The method of claim 8, wherein the context and the desired actions are included as one or more parameters in an application programming interface (API) call.

14. The method of claim 8, wherein analyzing the request comprises: processing the natural language request to determine the desired action and context by labeling the natural language request using one of: a machine learning model and one or more slot-filling algorithms to determine the desired action and the context.

15. A system comprising: a first virtual agent configured to: receive a request from a user; and communicate a desired action of the request with a second virtual agent, wherein the first virtual agent and the second virtual agent are configured to: exchange one or more proposals to fulfill the received request; and negotiate terms for performing the desired action based on the exchanged proposals, wherein the second virtual agent is configured to execute the desired action based on the negotiated terms, and wherein the first virtual agent is configured to communicate a response to the user upon the execution of the desired action.

16. The system of claim 15, wherein the first virtual agent is further configured to: receive a context associated with the received request, wherein the context comprises at least one of: location of the user, preferences of the user, schedule, or device state.

17. The system of claim 15, wherein the second virtual agent is configured to generate a prioritized set of proposals based on optimization preferences of the service provider, and wherein the first virtual agent is configured to evaluate the prioritized set in view of the preferences of the user.

18. The system of claim 16, wherein the first virtual agent communicates with the second virtual agent through an Application Programming Interface (API) call.

19. The system of claim 18, wherein the desired action and the context are provided as parameters of the API call.

20. The system of claim 15, wherein the first virtual agent and the second virtual agent are configured to perform the negotiation and the execution of the desired action without additional user input once the request is received.