

DYNAMIC COMMUNICATIONS ROUTING TO DISPARATE ENDPOINTS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present disclosure claims the priority benefit of U.S. Provisional Patent No.
5 62/820,500 filed March 19, 2019 and titled “Dynamic Communications Routing to Disparate
Endpoints,” the disclosure of which is incorporated by reference

FIELD

[0002] The present disclosure relates generally to facilitating routing of communications.
10 More specifically, techniques are provided to dynamically route messages having multiple intents
between bots and terminal devices during communication sessions configured with multi-channel
capabilities.

SUMMARY

[0003] The term embodiment and like terms are intended to refer broadly to all of the subject
15 matter of this disclosure and the claims below. Statements containing these terms should be
understood not to limit the subject matter described herein or to limit the meaning or scope of the
claims below. Embodiments of the present disclosure covered herein are defined by the claims
below, not this summary. This summary is a high-level overview of various aspects of the
disclosure and introduces some of the concepts that are further described in the Detailed
20 Description section below. This summary is not intended to identify key or essential features of
the claimed subject matter, nor is it intended to be used in isolation to determine the scope of the
claimed subject matter. The subject matter should be understood by reference to appropriate
portions of the entire specification of this disclosure, any or all drawings and each claim.

[0004] Certain embodiments of the present disclosure include a computer-implemented
25 method. The method may include receiving one or more variables associated with a client device.
The client device may be operated by a client. The method may further include receiving a
message for the client from a network device. The message may include a first intent and a second
intent. The method may further include parsing the message to identify the first intent and the

second intent. The first intent may be associated with a first actionable item and the second intent may be associated with a second actionable item. The method may further include analyzing the first intent and the second intent to determine a prioritization for executing the first actionable item and the second actionable item. The prioritization may indicate that the first actionable item should
5 be executed first and the second actionable item should be executed second. The method may further include feeding the first intent and the second intent into a machine learning model. The machine learning model may identify a first endpoint for the first intent and a second endpoint for the second intent by optimizing the one or more variables associated with the client device. The method may further include routing the first intent to the first endpoint. The first endpoint may
10 thereafter execute the first actionable item. The method may further include routing the second intent to the second endpoint. The second endpoint may thereafter execute the second actionable item.

[0005] Certain embodiments of the present disclosure include a system. The system may include one or more data processors; and a non-transitory computer-readable storage medium
15 containing instructions which, when executed on the one or more data processors, cause the one or more data processors to perform the methods described above and herein.

[0006] Certain embodiments of the present disclosure include a computer-program product tangibly embodied in a non-transitory machine-readable storage medium, including instructions configured to cause a data processing apparatus to perform the methods described above and
20 herein.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The present disclosure is described in conjunction with the appended figures:

[0008] FIG. 1 shows a block diagram of an embodiment of a network interaction system;

[0009] FIG. 2 shows a block diagram of another embodiment of a network interaction system;

25 [0010] FIGS. 3A-3C show block diagrams of other embodiments of a network interaction system that includes a connection management system;

[0011] FIG. 4 shows a representation of a protocol-stack mapping of connection components' operation;

[0012] FIG. 5 represents a multi-device communication exchange system according to an embodiment;

[0013] FIG. 6 shows a block diagram of an embodiment of a connection management system;

[0014] FIG. 7 shows a block diagram of a network environment for dynamically switching

5 between bots and terminal devices during communication sessions;

[0015] FIG. 8 shows a block diagram representing a network environment for dynamically selecting endpoints across multiple channel environments;

[0016] FIG. 9 shows a block diagram representing a network environment for enhancing endpoint selection using machine-learning techniques; and

10 [0017] FIG. 10 shows an example process for routing messages between bots and terminal devices during a communication session.

[0018] FIG. 11 shows a block diagram representing a network environment in which a system for orchestrating conversations driven by artificial intelligence (AI) may be implemented.

15 [0019] FIG. 12 shows a block diagram representing exemplary information flow within an AI-driven orchestration of a conversation.

[0020] FIGs. 13A-13C shows exemplary conversations that have been orchestrated by AI-driven systems.

[0021] In the appended figures, similar components and/or features can have the same reference label. Further, various components of the same type can be distinguished by following the reference label by a dash and a second label that distinguishes among the similar components. If only the first reference label is used in the specification, the description is applicable to any one of the similar components having the same first reference label irrespective of the second reference label.

DETAILED DESCRIPTION

25 [0022] The ensuing description provides preferred examples of embodiment(s) only and is not intended to limit the scope, applicability or configuration of the disclosure. Rather, the ensuing

description of the preferred examples of embodiment(s) will provide those skilled in the art with an enabling description for implementing a preferred examples of embodiment. It is understood that various changes can be made in the function and arrangement of elements without departing from the spirit and scope as set forth in the appended claims.

5 [0023] FIG. 1 shows a block diagram of an embodiment of a network interaction system 100 which implements and supports certain embodiments and features described herein. Certain embodiments relate to establishing a connection channel between a network device 105 (which can be operated by a user 110) and a terminal device 115 (which can be operated by an agent 120). In certain embodiments, the network interaction system 100 can include a client device 130
10 associated with a client 125.

[0024] In certain embodiments, a user 110 can be an individual browsing a web site or accessing an online service provided by a remote server 140. A client 125 can be an entity that provides, operates, or runs the web site or the online service, or individuals employed by or assigned by such an entity to perform the tasks available to a client 125 as described herein. The
15 agent 120 can be an individual, such as a support agent tasked with providing support or information to the user 110 regarding the website or online service. Out of a large number of agents, a subset of agents may be appropriate for providing support or information for a particular client 125. The agent 120 may be affiliated or not affiliated with the client 125. Each agent can be associated with one or more clients 125. In some non-limiting examples, a user 110 can be an
20 individual shopping an online store from a personal computing device, a client 125 can be a company that sells products online, and an agent 120 can be a representative employed by the company. In various embodiments, the user 110, client 125, and agent 120 can be other individuals or entities.

[0025] While FIG. 1 shows only a single network device 105, terminal device 115 and client
25 device 130, an interaction system 100 can include multiple or many (e.g., tens, hundreds or thousands) of each of one or more of these types of devices. Similarly, while FIG. 1 shows only a single user 110, agent 120 and client 125, an interaction system 100 can include multiple or many of each of one or more of such entities. Thus, it may be necessary to determine which terminal device is to be selected to communicate with a given network device. Further complicating matters,

a remote server 140 may also be configured to receive and respond to select network-device communications.

[0026] A connection management system 150 can facilitate strategic routing of communications. A communication can include a message with content (e.g., defined based on
5 input from an entity, such as typed or spoken input). The communication can also include additional data, such as data about a transmitting device (e.g., an IP address, account identifier, device type and/or operating system); a destination address; an identifier of a client; an identifier of a webpage or webpage element (e.g., a webpage or webpage element being visited when the communication was generated or otherwise associated with the communication) or online history
10 data; a time (e.g., time of day and/or date); and/or destination address. Other information can be included in the communication. In some instances, connection management system 150 routes the entire communication to another device. In some instances, connection management system 150 modifies the communication or generates a new communication (e.g., based on the initial communication). The new or modified communication can include the message (or processed
15 version thereof), at least some (or all) of the additional data (e.g., about the transmitting device, webpage or online history and/or time) and/or other data identified by connection management system 150 (e.g., account data associated with a particular account identifier or device). The new or modified communication can include other information as well.

[0027] Part of strategic-routing facilitation can include establishing, updating and using one or
20 more connection channels between network device 105 and one or more terminal devices 115. For example, upon receiving a communication from network device 105, connection management system 150 can first estimate to which client (if any) the communication corresponds. Upon identifying a client, connection management system 150 can identify a terminal device 115 associated with the client for communication with network device 105. In some instances, the
25 identification can include evaluating a profile of each of a plurality of agents (or experts or delegates), each agent (e.g., agent 120) in the plurality of agents being associated with a terminal device (e.g., terminal device 115). The evaluation can relate to a content in a network-device message. The identification of the terminal device 115 can include a technique described, for example, in U.S. Application Number 12/725,799, filed on March 17, 2010, which is hereby
30 incorporated by reference in its entirety for all purposes.

[0028] In some instances, connection management system 150 can determine whether any connection channels are established between network device 105 and a terminal device associated with the client (or remote server 140) and, if so, whether such channel is to be used to exchange a series of communications including the communication.

5 [0029] Upon selecting a terminal device 115 to communicate with network device 105, connection management system 150 can establish a connection channel between the network device 105 and terminal device 115. In some instances, connection management system 150 can transmit a message to the selected terminal device 115. The message may request an acceptance of a proposed assignment to communicate with a network device 105 or identify that such an
10 assignment has been generated. The message can include information about network device 105 (e.g., IP address, device type, and/or operating system), information about an associated user 110 (e.g., language spoken, duration of having interacted with client, skill level, sentiment, and/or topic preferences), a received communication, code (e.g., a clickable hyperlink) for generating and transmitting a communication to the network device 105, and/or an instruction to generate and
15 transmit a communication to network device 105.

[0030] In one instance, communications between network device 105 and terminal device 115 can be routed through connection management system 150. Such a configuration can allow connection management system 150 to monitor the communication exchange and to detect issues (e.g., as defined based on rules) such as non-responsiveness of either device or extended latency.
20 Further, such a configuration can facilitate selective or complete storage of communications, which may later be used, for example, to assess a quality of a communication exchange and/or to support learning to update or generate routing rules so as to promote particular post-communication targets.

[0031] In some embodiments, connection management system 150 can monitor the
25 communication exchange in real-time and perform automated actions (e.g., rule-based actions) based on the live communications. For example, when connection management system 150 determines that a communication relates to a particular item (e.g., product), connection management system 150 can automatically transmit an additional message to terminal device 115 containing additional information about the item (e.g., quantity of item available, links to support
30 documents related to the item, or other information about the item or similar items).

[0032] In one instance, a designated terminal device 115 can communicate with network device 105 without relaying communications through connection management system 150. One or both devices 105, 115 may (or may not) report particular communication metrics or content to connection management system 150 to facilitate communication monitoring and/or data storage.

5 [0033] As mentioned, connection management system 150 may route select communications to a remote server 140. Remote server 140 can be configured to provide information in a predetermined manner. For example, remote server 140 may access defined one or more text passages, voice recording and/or files to transmit in response to a communication. Remote server 140 may select a particular text passage, recording or file based on, for example, an analysis of a
10 received communication (e.g., a semantic or mapping analysis).

[0034] Routing and/or other determinations or processing performed at connection management system 150 can be performed based on rules and/or data at least partly defined by or provided by one or more client devices 130. For example, client device 130 may transmit a communication that identifies a prioritization of agents, terminal-device types, and/or topic/skill
15 matching. As another example, client device 130 may identify one or more weights to apply to various variables potentially impacting routing determinations (e.g., language compatibility, predicted response time, device type and capabilities, and/or terminal-device load balancing). It will be appreciated that which terminal devices and/or agents are to be associated with a client may be dynamic. Communications from client device 130 and/or terminal devices 115 may
20 provide information indicating that a given terminal device and/or agent is to be added or removed as one associated with a client. For example, client device 130 can transmit a communication with IP address and an indication as to whether a terminal device with the address is to be added or removed from a list identifying client-associated terminal devices.

[0035] Each communication (e.g., between devices, between a device and connection
25 management system 150, between remote server 140 and connection management system 150 or between remote server 140 and a device) can occur over one or more networks 170. Any combination of open or closed networks can be included in the one or more networks 170. Examples of suitable networks include the Internet, a personal area network, a local area network (LAN), a wide area network (WAN), or a wireless local area network (WLAN). Other networks
30 may be suitable as well. The one or more networks 170 can be incorporated entirely within or can

include an intranet, an extranet, or a combination thereof. In some instances, a network in the one or more networks 170 includes a short-range communication channel, such as a Bluetooth or a Bluetooth Low Energy channel. In one embodiment, communications between two or more systems and/or devices can be achieved by a secure communications protocol, such as secure sockets layer (SSL) or transport layer security (TLS). In addition, data and/or transactional details may be encrypted based on any convenient, known, or to be developed manner, such as, but not limited to, Data Encryption Standard (DES), Triple DES, Rivest-Shamir-Adleman encryption (RSA), Blowfish encryption, Advanced Encryption Standard (AES), CAST-128, CAST-256, Decorrelated Fast Cipher (DFC), Tiny Encryption Algorithm (TEA), eXtended TEA (XTEA), Corrected Block TEA (XXTEA), and/or RC5, etc.

[0036] A network device 105, terminal device 115 and/or client device 130 can include, for example, a portable electronic device (e.g., a smart phone, tablet, laptop computer, or smart wearable device) or a non-portable electronic device (e.g., one or more desktop computers, smart appliances, servers, and/or processors). Connection management system 150 can be separately housed from network, terminal and client devices or may be part of one or more such devices (e.g., via installation of an application on a device). Remote server 140 may be separately housed from each device and connection management system 150 and/or may be part of another device or system. While each device, server and system in FIG. 1 is shown as a single device, it will be appreciated that multiple devices may instead be used. For example, a set of network devices can be used to transmit various communications from a single user, or remote server 140 may include a server stack.

[0037] A software agent or application may be installed on and/or executable on a depicted device, system or server. In one instance, the software agent or application is configured such that various depicted elements can act in complementary manners. For example, a software agent on a device can be configured to collect and transmit data about device usage to a separate connection management system, and a software application on the separate connection management system can be configured to receive and process the data.

[0038] FIG. 2 shows a block diagram of another embodiment of a network interaction system 200. Generally, FIG. 2 illustrates a variety of components configured and arranged to enable a

network device 205 to communicate with one or more terminal devices 215. The depicted instance includes nine terminal devices 215 included in three local-area networks 235.

[0039] In some instances, a communication from network device 205 includes destination data (e.g., a destination IP address) that at least partly or entirely indicates which terminal device is to receive the communication. Network interaction system 200 can include one or more inter-network connection components 240 and/or one or more intra-network connection components 255 that can process the destination data and facilitate appropriate routing.

[0040] Each inter-network connection components 245 can be connected to a plurality of networks 235 and can have multiple network cards installed (e.g., each card connected to a different network). For example, an inter-network connection component 245 can be connected to a wide-area network 270 (e.g., the Internet) and one or more local-area networks 235. In the depicted instance, in order for a communication to be transmitted from network device 205 to any of the terminal devices, in the depicted system, the communication must be handled by multiple inter-network connection components 245.

[0041] When an inter-network connection component 245 receives a communication (or a set of packets corresponding to the communication), inter-network connection component 245 can determine at least part of a route to pass the communication to a network associated with a destination. The route can be determined using, for example, a routing table (e.g., stored at the router), which can include one or more routes that are pre-defined, generated based on an incoming message (e.g., from another router or from another device) or learned.

[0042] Examples of inter-network connection components 245 include a router 260 and a gateway 265. An inter-network connection component 245 (e.g., gateway 265) may be configured to convert between network systems or protocols. For example, gateway 265 may facilitate communication between Transmission Control Protocol/Internet Protocol (TCP/IP) and Internetwork Packet Exchange/Sequenced Packet Exchange (IPX/SPX) devices.

[0043] Upon receiving a communication at a local-area network 235, further routing may still need to be performed. Such intra-network routing can be performed via an intra-network connection component 255, such as a switch 280 or hub 285. Each intra-network connection component 255 can be connected to (e.g., wirelessly or wired, such as via an Ethernet cable)

multiple terminal devices 215. Hub 285 can be configured to repeat all received communications to each device to which it is connected. Each terminal device can then evaluate each communication to determine whether the terminal device is the destination device or whether the communication is to be ignored. Switch 280 can be configured to selectively direct
5 communications to only the destination terminal device.

[0044] In some instances, a local-area network 235 can be divided into multiple segments, each of which can be associated with independent firewalls, security rules and network protocols. An intra-network connection component 255 can be provided in each of one, more or all segments to facilitate intra-segment routing. A bridge 280 can be configured to route communications across
10 segments 275.

[0045] To appropriately route communications across or within networks, various components analyze destination data in the communications. For example, such data can indicate which network a communication is to be routed to, which device within a network a communication is to be routed to or which communications a terminal device is to process (versus ignore). However,
15 in some instances, it is not immediately apparent which terminal device (or even which network) is to participate in a communication from a network device.

[0046] To illustrate, a set of terminal devices may be configured so as to provide similar types of responsive communications. Thus, it may be expected that a query in a communication from a network device may be responded to in similar manners regardless to which network device the
20 communication is routed. While this assumption may be true at a high level, various details pertaining to terminal devices can give rise to particular routings being advantageous as compared to others. For example, terminal devices in the set may differ from each other with respect to (for example) which communication channels are supported, geographic and/or network proximity to a network device and/or characteristics of associated agents (e.g., knowledge bases, experience,
25 languages spoken, availability, general personality or sentiment, etc.). Accordingly, select routings may facilitate faster responses that more accurately and/or completely respond to a network-device communication. A complication is that static routings mapping network devices to terminal devices may fail to account for variations in communication topics, channel types, agent availability, and so on.

[0047] FIGS. 3A-3C show block diagrams of other embodiments of a network interaction system 300a-c that includes a connection management system. Each of the depicted systems 300a-c show only 2 local-area networks 235 for simplicity, though it can be appreciated that embodiments can be extended to expand the number of local-area networks. Each of systems 300a-c include a connection management system 350, which can identify which terminal device is to communicate with network device 205, can establish and manage (e.g., maintain or close) connection channels, can determine whether and when to re-route communications in an exchange, and so on. Thus, connection management system 350 can be configured to dynamically, and in real-time, evaluate communications, agent availability, capabilities of terminal devices or agents, and so on, to influence routing determinations.

[0048] In FIG. 3A, connection management system 350 is associated with each of network device 205 and a remote server 340 (e.g., connection management system 350a is associated with network device 205 and connection management system 350b is associated with remote server 340). For example, connection management system 350a and/or connection management system 350b can be installed or stored as an application on each of network device 205 and remote server 340, respectively. Execution of the application(s) can facilitate, for example, a communication between network device 205 and remote server 340 to identify a terminal device 215 selected to participate in a communication exchange with network device 205. The identification can be made based on one or more factors disclosed herein (e.g., availability, matching between a communication's topic/level of detail with agents' or terminal devices' knowledge bases, predicted latency, channel-type availability, and so on).

[0049] A client device 330 can provide client data indicating how routing determinations are to be made. For example, such data can include: indications as to how particular characteristics are to be weighted or matched or constraints or biases (e.g., pertaining to load balancing or predicted response latency). Client data can also include specifications related to when communication channels are to be established (or closed) or when communications are to be re-routed to a different network device. Client data can be used to define various client-specific rules, such as rules for communication routing and so on.

[0050] Connection management system 350b executing on remote server 340 can monitor various metrics pertaining to terminal devices (e.g., pertaining to a given client), such as which

communication channels are supported, geographic and/or network proximity to a network device, communication latency and/or stability with the terminal device, a type of the terminal device, a capability of the terminal device, whether the terminal device (or agent) has communicated with a given network device (or user) before and/or characteristics of associated agents (e.g., knowledge bases, experience, languages spoken, availability, general personality or sentiment, etc.).
Accordingly, connection management system 350b may be enabled to select routings to facilitate faster responses that more accurately and/or completely respond to a network-device communication based on the metrics.

[0051] In the example depicted in FIG. 3A, a communication exchange between network device 205 and remote server 340 can facilitate early identification of a destination address. Network device 205 may then use the destination address to direct subsequent communications. For example, network device 205 may send an initial communication to remote server 340 (e.g., via one or more inter-network connections and a wide-area network), and remote server 340 may identify one or more corresponding clients. Remote server 340 may then identify a set of terminal devices associated with the one or more corresponding clients and collect metrics for those terminal devices. The metrics can be evaluated (e.g., by remote server 340) so as to select a terminal device to involve in a communication exchange, and information pertaining to the terminal device (e.g., an IP address) can be sent to network device 205. In some embodiments, remote server 340 may continuously or periodically collect and evaluate metrics for various terminal devices and store evaluation results in a data store. In such embodiments, upon identifying a set of terminal devices associated with the one or more corresponding clients, remote server 340 can access the stored evaluation results from the data store and select a terminal device to involve in the communication exchange based on the stored evaluation results.

[0052] In **FIG. 3B**, connection management system 350 can be configured to serve as a relay and/or destination address. Thus, for example, a set of network devices 205 may transmit communications, each identifying connection management system 350 as a destination. Connection management system 350 can receive each communication and can concurrently monitor a set of terminal devices (e.g., so as to generate metrics for each terminal device). Based on the monitoring and a rule, connection management system 350 can identify a terminal device 215 to which it may relay each communication. Depending on the embodiment, terminal device

communications may similarly be directed to a consistent destination (e.g., of connection management system 350) for further relaying, or terminal devices may begin communicating directly with corresponding network devices. These embodiments can facilitate efficient routing and thorough communication monitoring.

5 [0053] The embodiment depicted in **FIG. 3C** is similar to that in FIG. 3B. However, in some embodiments, connection management system 350 is directly connected to intra-network components (e.g., terminal devices, intra-network connections, or other).

[0054] It will be appreciated that many variations of FIGS. 3A-3C are contemplated. For example, connection management system 350 may be associated with a connection component
10 (e.g., inter-network connection component 245 or intra-network connection component 255) such that an application corresponding to connection management system 350 (or part thereof) is installed on the component. The application may, for example, perform independently or by communicating with one or more similar or complementary applications (e.g., executing on one or more other components, network devices or remotes servers).

15 [0055] **FIG. 4** shows a representation of a protocol-stack mapping 400 of connection components' operation. More specifically, FIG. 4 identifies a layer of operation in an Open Systems Interaction (OSI) model that corresponds to various connection components.

[0056] The OSI model can include multiple logical layers 402-414. The layers are arranged in an ordered stack, such that layers 402-412 each serve a higher level and layers 404-414 is each
20 served by a lower layer. The OSI model includes a physical layer 402. Physical layer 402 can define parameters physical communication (e.g., electrical, optical, or electromagnetic). Physical layer 402 also defines connection management protocols, such as protocols to establish and close connections. Physical layer 402 can further define a flow-control protocol and a transmission mode.

25 [0057] A link layer 404 can manage node-to-node communications. Link layer 404 can detect and correct errors (e.g., transmission errors in the physical layer 402) and manage access permissions. Link layer 404 can include a media access control (MAC) layer and logical link control (LLC) layer.

[0058] A network layer 406 can coordinate transferring data (e.g., of variable length) across nodes in a same network (e.g., as datagrams). Network layer 406 can convert a logical network address to a physical machine address.

[0059] A transport layer 408 can manage transmission and receipt quality. Transport layer 408 can provide a protocol for transferring data, such as a Transmission Control Protocol (TCP). Transport layer 408 can perform segmentation/desegmentation of data packets for transmission and can detect and account for transmission errors occurring in layers 402-406. A session layer 410 can initiate, maintain and terminate connections between local and remote applications. Sessions may be used as part of remote-procedure interactions. A presentation layer 412 can encrypt, decrypt and format data based on data types known to be accepted by an application or network layer.

[0060] An application layer 414 can interact with software applications that control or manage communications. Via such applications, application layer 414 can (for example) identify destinations, local resource states or availability and/or communication content or formatting. Various layers 402-414 can perform other functions as available and applicable.

[0061] Intra-network connection components 422, 424 are shown to operate in physical layer 402 and link layer 404. More specifically, a hub can operate in the physical layer, such that operations can be controlled with respect to receipts and transmissions of communications. Because hubs lack the ability to address communications or filter data, they possess little to no capability to operate in higher levels. Switches, meanwhile, can operate in link layer 404, as they are capable of filtering communication frames based on addresses (e.g., MAC addresses).

[0062] Meanwhile, inter-network connection components 426, 428 are shown to operate on higher levels (e.g., layers 406-414). For example, routers can filter communication data packets based on addresses (e.g., IP addresses). Routers can forward packets to particular ports based on the address, so as to direct the packets to an appropriate network. Gateways can operate at the network layer and above, perform similar filtering and directing and further translation of data (e.g., across protocols or architectures).

[0063] A connection management system 450 can interact with and/or operate on, in various embodiments, one, more, all or any of the various layers. For example, connection management

system 450 can interact with a hub so as to dynamically adjust which terminal devices the hub communicates. As another example, connection management system 450 can communicate with a bridge, switch, router or gateway so as to influence which terminal device the component selects as a destination (e.g., MAC, logical or physical) address. By way of further examples, a connection management system 450 can monitor, control, or direct segmentation of data packets on transport layer 408, session duration on session layer 410, and/or encryption and/or compression on presentation layer 412. In some embodiments, connection management system 450 can interact with various layers by exchanging communications with (e.g., sending commands to) equipment operating on a particular layer (e.g., a switch operating on link layer 404), by routing or modifying existing communications (e.g., between a network device and a terminal device) in a particular manner, and/or by generating new communications containing particular information (e.g., new destination addresses) based on the existing communication. Thus, connection management system 450 can influence communication routing and channel establishment (or maintenance or termination) via interaction with a variety of devices and/or via influencing operating at a variety of protocol-stack layers.

[0064] FIG. 5 represents a multi-device communication exchange system 500 according to an embodiment. System 500 includes a network device 505 configured to communicate with a variety of types of terminal devices over a variety of types of communication channels.

[0065] In the depicted instance, network device 505 can transmit a communication over a cellular network (e.g., via a base station 510). The communication can be routed to an operative network 515. Operative network 515 can include a connection management system 520 that receives the communication and identifies which terminal device is to respond to the communication. Such determination can depend on identifying a client to which that communication pertains (e.g., based on a content analysis or user input indicative of the client) and determining one or more metrics for each of one or more terminal devices associated with the client. For example, in FIG. 5, each cluster of terminal devices 530a-c can correspond to a different client. The terminal devices may be geographically co-located or disperse. The metrics may be determined based on stored or learned data and/or real-time monitoring (e.g., based on availability).

[0066] Connection management system 520 can communicate with various terminal devices via one or more routers 525 or other inter-network or intra-network connection components. Connection management system 520 may collect, analyze and/or store data from or pertaining to communications, terminal-device operations, client rules, and/or user-associated actions (e.g.,
5 online activity) at one or more data stores. Such data may influence communication routing.

[0067] Notably, various other devices can further be used to influence communication routing and/or processing. For example, in the depicted instance, connection management system 520 also is connected to a web server 540. Thus, connection management system 520 can retrieve data of interest, such as technical item details, and so on.

10 [0068] Network device 505 may also be connected to a web server (e.g., including a web server 545). In some instances, communication with such a server provided an initial option to initiate a communication exchange with connection management system 520. For example, network device 505 may detect that, while visiting a particular webpage, a communication opportunity is available and such an option can be presented.

15 [0069] One or more elements of communication system 500 can also be connected to a social-networking server 550. Social networking server 550 can aggregate data received from a variety of user devices. Thus, for example, connection management system 520 may be able to estimate a general (or user-specific) behavior of a given user or class of users.

[0070] FIG. 6 shows a block diagram of an embodiment of a connection management system
20 600. A message receiver interface 605 can receive a message. In some instances, the message can be received, for example, as part of a communication transmitted by a source device (e.g., housed separately from connection management system 600 or within a same housing), such as a network device or terminal device. In some instances, the communication can be part of a series of communications or a communicate exchange, which can include a series of messages or message
25 exchange being routed between two devices (e.g., a network device and terminal device). This message or communication exchange may be part of and/or may define an interaction between the devices. A communication channel or operative channel can include one or more protocols (e.g., routing protocols, task-assigning protocols and/or addressing protocols) used to facilitate routing and a communication exchange between the devices.

[0071] In some instances, the message can include a message generated based on inputs received at a local or remote user interface. For example, the message can include a message that was generated based on button or key presses or recorded speech signals. In one instance, the message includes an automatically generated message, such as one generated upon detecting that a network device is presenting a particular app page or webpage or has provided a particular input command (e.g., key sequence). The message can include an instruction or request, such as one to initiate a communication exchange.

[0072] In some instances, the message can include or be associated with an identifier of a client. For example, the message can explicitly identify the client (or a device associated with the client); the message can include or be associated with a webpage or app page associated with the client; the message can include or be associated with a destination address associated with a client; or the message can include or be associated with an identification of an item (e.g., product) or service associated with the client. To illustrate, a network device may be presenting an app page of a particular client, which may offer an option to transmit a communication to an agent. Upon receiving user input corresponding to a message, a communication may be generated to include the message and an identifier of the particular client.

[0073] A processing engine 610 may process a received communication and/or message. Processing can include, for example, extracting one or more particular data elements (e.g., a message, a client identifier, a network-device identifier, an account identifier, and so on). Processing can include transforming a formatting or communication type (e.g., to be compatible with a particular device type, operating system, communication-channel type, protocol and/or network).

[0074] A message assessment engine 615 may assess the (e.g., extracted or received) message. The assessment can include identifying, for example, one or more categories or tags for the message. Examples of category or tag types can include (for example) topic, sentiment, complexity, and urgency. A difference between categorizing and tagging a message can be that categories can be limited (e.g., according to a predefined set of category options), while tags can be open. A topic can include, for example, a technical issue, a use question, or a request. A category or tag can be determined, for example, based on a semantic analysis of a message (e.g., by identifying keywords, sentence structures, repeated words, punctuation characters and/or non-

article words); user input (e.g., having selected one or more categories); and/or message-associated statistics (e.g., typing speed and/or response latency).

[0075] In some instances, message assessment engine 615 can determine a metric for a message. A metric can include, for example, a number of characters, words, capital letters, all-capital words or instances of particular characters or punctuation marks (e.g., exclamation points, question marks and/or periods). A metric can include a ratio, such as a fraction of sentences that end with an exclamation point (or question mark), a fraction of words that are all capitalized, and so on.

[0076] Message assessment engine 615 can store a message, message metric and/or message statistic in a message data store 620. Each message can also be stored in association with other data (e.g., metadata), such as data identifying a corresponding source device, destination device, network device, terminal device, client, one or more categories, one or more stages and/or message-associated statistics). Various components of connection management system 600 (e.g., message assessment engine 615 and/or an interaction management engine 625) can query message data store 620 to retrieve query-responsive messages, message metrics and/or message statistics.

[0077] An interaction management engine 625 can determine to which device a communication is to be routed and how the receiving and transmitting devices are to communicate. Each of these determinations can depend, for example, on whether a particular network device (or any network device associated with a particular user) has previously communicated with a terminal device in a set of terminal devices (e.g., any terminal device associated with connection management system 600 or any terminal device associated with one or more particular clients).

[0078] In some instances, when a network device (or other network device associated with a same user or profile) has previously communicated with a given terminal device, communication routing can be generally biased towards the same terminal device. Other factors that may influence routing can include, for example, whether the terminal device (or corresponding agent) is available and/or a predicted response latency of the terminal device. Such factors may be considered absolutely or relative to similar metrics corresponding to other terminal devices. A re-routing rule (e.g., a client-specific or general rule) can indicate how such factors are to be assessed and weighted to determine whether to forego agent consistency.

[0079] When a network device (or other network device associated with a same user or account) has not previously communicated with a given terminal device, a terminal-device selection can be performed based on factors such as, for example, an extent to which various agents' knowledge base corresponds to a communication topic, availability of various agents at a given time and/or over a channel type, types and/or capabilities of terminal devices (e.g., associated with the client). In one instance, a rule can identify how to determine a sub-parameter to one or more factors such as these and a weight to assign to each parameter. By combining (e.g., summing) weighted sub-parameters, a parameter for each agent can be determined. A terminal device selection can then be made by comparing terminal devices' parameters.

[0080] With regard to determining how devices are to communicate, interaction management engine 625 can (for example) determine whether a terminal device is to respond to a communication via (for example) SMS message, voice call, video communication, etc. A communication type can be selected based on, for example, a communication-type priority list (e.g., at least partly defined by a client or user); a type of a communication previously received from the network device (e.g., so as to promote consistency), a complexity of a received message, capabilities of the network device, and/or an availability of one or more terminal devices. Appreciably, some communication types will result in real-time communication (e.g., where fast message response is expected), while others can result in asynchronous communication (e.g., where delays (e.g., of several minutes or hours) between messages are acceptable).

[0081] Further, interaction management engine 625 can determine whether a continuous channel between two devices should be established, used or terminated. A continuous channel can be structured so as to facilitate routing of future communications from a network device to a specified terminal device. This bias can persist even across message series. In some instances, a representation of a continuous channel (e.g., identifying an agent) can be included in a presentation to be presented on a network device. In this manner, a user can understand that communications are to be consistently routed so as to promote efficiency.

[0082] In one instance, a parameter can be generated using one or more factors described herein and a rule (e.g., that includes a weight for each of the one or more factors) to determine a connection parameter corresponding to a given network device and terminal device. The parameter may pertain to an overall match or one specific to a given communication or communication series.

Thus, for example, the parameter may reflect a degree to which a given terminal device is predicted to be suited to respond to a network-device communication. In some instances, a parameter analysis can be used to identify each of a terminal device to route a given communication to and whether to establish, use or terminate a connection channel. When a parameter analysis is used to both address a routing decision and a channel decision, a parameter relevant to each decision may be determined in a same, similar or different manner.

[0083] Thus, for example, it will be appreciated that different factors may be considered depending on whether the parameter is to predict a strength of a long-term match versus one to respond to a particular message query. For example, in the former instance, considerations of overall schedules and time zones may be important, while in the latter instance, immediate availability may be more highly weighted. A parameter can be determined for a single network-device/terminal-device combination, or multiple parameters can be determined, each characterizing a match between a given network device and a different terminal device.

[0084] To illustrate, a set of three terminal devices associated with a client may be evaluated for potential communication routing. A parameter may be generated for each that relates to a match for the particular communication. Each of the first two terminal devices may have previously communicated with a network device having transmitted the communication. An input from the network device may have indicated positive feedback associated with an interaction with the communication(s) with the first device. Thus, a past-interact sub-parameter (as calculated according to a rule) for the first, second and third devices may be 10, 5, and 0, respectively. (Negative feedback inputs may result in negative sub-parameters.) It may be determined that only the third terminal device is available. It may be predicted that the second terminal device will be available for responding within 15 minutes, but that the first terminal device will not be available for responding until the next day. Thus, a fast-response sub-parameter for the first, second and third devices may be 1, 3 and 10. Finally, it may be estimated a degree to which an agent (associated with the terminal device) is knowledgeable about a topic in the communication. It may be determined that an agent associated with the third terminal device is more knowledgeable than those associated with the other two devices, resulting in sub-parameters of 3, 4 and 9. In this example, the rule does not include weighting or normalization parameters (though, in other instances, a rule may), resulting in parameters of 14, 11 and 19. Thus, the rule may indicate that

the message is to be routed to a device with the highest parameter, that being the third terminal device. If routing to a particular terminal device is unsuccessful, the message can be routed to a device with the next-highest parameter, and so on.

[0085] A parameter may be compared to one or more absolute or relative thresholds. For example, parameters for a set of terminal devices can be compared to each other to identify a high parameter to select a terminal device to which a communication can be routed. As another example, a parameter (e.g., a high parameter) can be compared to one or more absolute thresholds to determine whether to establish a continuous channel with a terminal device. An overall threshold for establishing a continuous channel may (but need not) be higher than a threshold for consistently routing communications in a given series of messages. This difference between the overall threshold and threshold for determining whether to consistently route communication may be because a strong match is important in the continuous-channel context given the extended utility of the channel. In some embodiments, an overall threshold for using a continuous channel may (but need not) be lower than a threshold for establishing a continuous channel and/or for consistently routing communications in a given series of messages.

[0086] Interaction management engine 625 can interact with an account engine 630 in various contexts. For example, account engine 630 may look up an identifier of a network device or terminal device in an account data store 635 to identify an account corresponding to the device. Further, account engine 630 can maintain data about previous communication exchanges (e.g., times, involved other device(s), channel type, resolution stage, topic(s) and/or associated client identifier), connection channels (e.g., indicating – for each of one or more clients – whether any channels exist, a terminal device associated with each channel, an establishment time, a usage frequency, a date of last use, any channel constraints and/or supported types of communication), user or agent preferences or constraints (e.g., related to terminal-device selection, response latency, terminal-device consistency, agent expertise, and/or communication-type preference or constraint), and/or user or agent characteristics (e.g., age, language(s) spoken or preferred, geographical location, interests, and so on).

[0087] Further, interaction management engine 625 can alert account engine 630 of various connection-channel actions, such that account data store 635 can be updated to reflect the current channel data. For example, upon establishing a channel, interaction management engine 625 can

notify account engine 630 of the establishment and identify one or more of: a network device, a terminal device, an account and a client. Account engine 635 can (in some instances) subsequently notify a user of the channel's existence such that the user can be aware of the agent consistency being availed.

5 **[0088]** Interaction management engine 625 can further interact with a client mapping engine 640, which can map a communication to one or more clients (and/or associated brands). In some instances, a communication received from a network device itself includes an identifier corresponding to a client (e.g., an identifier of a client, webpage, or app page). The identifier can be included as part of a message (e.g., which client mapping engine 640 may detect) or included
10 as other data in a message-inclusive communication. Client mapping engine 640 may then look up the identifier in a client data store 645 to retrieve additional data about the client and/or an identifier of the client.

[0089] In some instances, a message may not particularly correspond to any client. For example, a message may include a general query. Client mapping engine 640 may, for example,
15 perform a semantic analysis on the message, identify one or more keywords and identify one or more clients associated with the keyword(s). In some instances, a single client is identified. In some instances, multiple clients are identified. An identification of each client may then be presented via a network device such that a user can select a client to communicate with (e.g., via an associated terminal device).

20 **[0090]** Client data store 645 can include identifications of one or more terminal devices (and/or agents) associated with the client. A terminal routing engine 650 can retrieve or collect data pertaining to each of one, more or all such terminal devices (and/or agents) so as to influence routing determinations. For example, terminal routing engine 650 may maintain a terminal data store 655, which can store information such as terminal devices' device types, operating system,
25 communication-type capabilities, installed applications accessories, geographic location and/or identifiers (e.g., IP addresses). Some information can be dynamically updated. For example, information indicating whether a terminal device is available may be dynamically updated based on (for example) a communication from a terminal device (e.g., identifying whether the device is asleep, being turned off/on, non-active/active, or identifying whether input has been received
30 within a time period); a communication routing (e.g., indicative of whether a terminal device is

involved in or being assigned to be part of a communication exchange); or a communication from a network device or terminal device indicating that a communication exchange has ended or begun.

[0091] It will be appreciated that, in various contexts, being engaged in one or more communication exchanges does not necessarily indicate that a terminal device is not available to
5 engage in another communication exchange. Various factors, such as communication types (e.g., message), client-identified or user-identified target response times, and/or system loads (e.g., generally or with respect to a user) may influence how many exchanges a terminal device may be involved in.

[0092] When interaction management engine 625 has identified a terminal device to involve
10 in a communication exchange or connection channel, it can notify terminal routing engine 650, which may retrieve any pertinent data about the terminal device from terminal data store 655, such as a destination (e.g., IP) address, device type, protocol, etc. Processing engine 610 can then (in some instances) modify the message-inclusive communication or generate a new communication (including the message) so as to have a particular format, comply with a particular protocol, and
15 so on. In some instances, a new or modified message may include additional data, such as account data corresponding to a network device, a message chronicle, and/or client data.

[0093] A message transmitter interface 660 can then transmit the communication to the terminal device. The transmission may include, for example, a wired or wireless transmission to a device housed in a separate housing. The terminal device can include a terminal device in a same
20 or different network (e.g., local-area network) as connection management system 600. Accordingly, transmitting the communication to the terminal device can include transmitting the communication to an inter- or intra-network connection component.

[0094] Systems and methods for dynamically switching between bots and terminal devices (e.g., operated by live agents) during communication sessions with network devices (e.g., operated
25 by users) is provided. In some implementations, bots can be configured to autonomously communicate with network devices. Further, bots can be configured for a specific capability. Examples of capabilities can include updating database records, providing updates to users, providing additional data about the user to agents, determining a user's intent and routing the user to a destination system based on the intent, predicting or suggesting responses to agents

communicating with users, escalating communication sessions to include one or more additional bots or agents, and other suitable capabilities. In some implementations, while a bot is communicating with a network device (e.g., operated by the user) during a communication session (e.g., using a chat-enabled interface), a communication server can automatically and dynamically
5 determine to switch the bot with a terminal device. For example, bots can communicate with users about certain tasks (e.g., updating a database record associated with a user), whereas, terminal devices can communicate with users about more difficult tasks (e.g., communicating using a communication channel to solve a technical issue).

[0095] In some implementations, determining whether to switch between a bot and a terminal
10 device during a communication session can be based on an analysis of one or more characteristics of the messages in a communication session. Further, a dynamic sentiment parameter can be generated to represent a sentiment of messages, conversations, entities, agents, and so on. For example, in cases where the dynamic sentiment parameter indicates that the user is frustrated with the bot, the system can automatically switch the bot with a terminal device so that a live agent can
15 communicate with the user. *See* U.S. Serial No. 15/171,525, filed June 2, 2016, the disclosure of which is incorporated by reference herein in its entirety for all purposes. In some examples, determining whether to switch between the bots and terminal devices can be performed without a prompt from a user. The determination can be performed automatically at the communication server based any number of factors, including characteristics of the current messages in the
20 communication session (e.g., chat), characteristics of previous messages transmitted by the user in previous communication sessions, a trajectory of a characteristic (e.g., a sentiment) over multiple messages in a conversation, or additional information associated with the user (e.g., profile information, preference information, and other suitable information associated with the user).

[0096] **FIG. 7** shows a block diagram of a network environment for dynamically switching
25 between bots and terminal devices during communication sessions. In some implementations, network environment 700 can include network device 705, communication server 710, terminal device 715, and bot 720. Communication server 710 can be a server with one or more processors with at least one storage device, and can be configured to perform methods and techniques described herein. For example, communication server 710 can manage communication sessions
30 between network devices (e.g., operated by users) and terminal devices (e.g., operated by agents).

Communication server 710 can establish a communication channel between network device 705 and terminal device 715 so that network device 705 and terminal device 715 can communicate with each other during a communication session. A communication session can facilitate the exchange of one or more messages between network device 705 and terminal device 715. The present disclosure is not limited to the exchange of messages during a communication session. Other forms of communication can be facilitated by the communication session, for example, video communication (e.g., a video feed) and audio communication (e.g., a Voice-Over-IP connection).

[0097] In some implementations, communication server 710 can establish a communication channel between network device 705 and bot 720. Bot 720 can be code that, when executed, is configured to autonomously communicate with network device 705. For example, bot 720 can be a bot that automatically generates messages to initiate conversations with the user associated with network device 705 and/or to automatically respond to messages from network device 705. In addition, communication server 710 can be associated with a platform. Clients (e.g., an external system to the platform) can deploy bots in their internal communication systems using the platform. In some examples, clients can use their own bots in the platform, which enables clients to implement the methods and techniques described herein into their internal communication systems.

[0098] In some implementations, bots can be defined by one or more sources. For example, data store 730 can store code representing bots that are defined (e.g., created or coded) by clients of the communication server. For example, a client that has defined its own bots can load the bots to the communication server 710. The bots defined by clients can be stored in client bots data store 730. Data store 740 can store code representing bots that are defined by third-party systems. For example, a third-party system can include an independent software vendor. Data store 750 can store code representing bots that are defined by an entity associated with communication server 710. For example, bots that are coded by the entity can be loaded to or accessible by communication server 710, so that the bots can be executed and autonomously communicate with users. In some implementations, communication server 710 can access bots stored in data store 730, data store 740, and/or data store 750 using cloud network 760. Cloud network 760 may be any network, and can include an open network, such as the Internet, personal area network, local

area network (LAN), campus area network (CAN), metropolitan area network (MAN), wide area network (WAN), wireless local area network (WLAN), a private network, such as an intranet, extranet, or other backbone.

[0099] In addition, terminal device 715 can be operated by an agent. Terminal device 715 can be any portable (e.g., mobile phone, tablet, laptop) or non-portable device (e.g., electronic kiosk, desktop computer, etc.). In some instances, the agent can access a website using a browser that is running on terminal device 715. For example, the website can include a console or platform that is running on the browser of terminal device 715. The agent can be logged into the platform using the browser. One or more login credentials (e.g., username, password, and the like) can be used to authenticate the agent's identity before allowing the agent to gain access to the console or web applications included in the console. Examples of a console can include a platform that includes one or more APIs (application programming interfaces), a dashboard including one or more functions, a web-hosted application running on a web browser (without the need for downloading plug-ins) that is capable of establishing or joining a communication session, and other suitable interfaces. Further, the console can include one or more web applications or functions that can be executed. The web applications or functions can be executed at the browser, at communication server 710, a local server, a remote server, or other suitable computing device. For example, the web applications, native applications, or functions can enable an agent to communicate with a user, and to view communications between the user and one or more bots.

[00100] In some implementations, communication server 710 can be configured to dynamically switch between bot 720 and terminal device 715 during a particular communication session. For example, communication server 710 can facilitate a communication session between network device 705 and bot 720. Bot 720 can be configured to autonomously communicate with network device 705 by exchanging one or more messages with the network device 705 during the communication session. Communication server 710 can dynamically determine whether to switch bot 720 with terminal device 715 (or in some cases, vice versa) so that a live agent can communicate with network device 705, instead of bot 720. In some implementations, the switching can be performed without a prompt from the network device 705 or terminal device 715. For example, the switching can be based on message parameters (e.g., scores representing sentiment

of a message or series of messages) of the messages exchanged between the network device 705 and the bot 720, without prompting the network device 705 to request a terminal device.

[00101] In some implementations, communication server 710 can determine to switch between bot 720 and terminal device 715 automatically based on characteristics of the messages exchanged between the bot 720 and the network device 705. In some instances, analyzing the text of a message to determine the characteristic (e.g., the message parameter) can include an analysis of textual or non-textual attributes associated with the message. For example, communication server 710 can extract one or more lines of text included in the message from network device 705. Communication server 710 can identify whether the one or more lines of text include an anchor. Examples of an anchor include a string of text associated with a polarity (e.g., sentiment or intent, the word “frustrated” corresponding to a negative polarity or frustrated polarity, the word “happy” corresponding to a positive polarity, and so on). For example, a term “dispute” for one client can be negative, but can be neutral or positive for a second client. In some instances, anchors can be dynamically determined using supervised machine learning techniques. For example, one or more clustering algorithms can be executed on stored messages to find patterns within the stored messages. The clustered messages can be further filtered and evaluated to determine the anchor. Further, one or more words near the identified anchor can be parsed for amplifiers. An example of an amplifier is a term that increases or decreases an intensity associated with the polarity of the anchor, such as “really,” “not really,” “kind of,” and so on. The characteristic can include, for example, the speed of typing, the number of special characters used in the message (e.g., exclamation points, question marks, and so on), a semantic analysis of a message (e.g., by identifying keywords, sentence structures, repeated words, punctuation characters and/or non-article words); user input (e.g., having selected one or more categories); and/or message-associated statistics (e.g., response latency).

[00102] As a non-limiting example, the message parameter can be a numerical value that indicates the high intensity of the negative polarity (e.g., a message parameter of 20 on a scale of 0-100, with lower numbers indicating a negative polarity and higher numbers indicating a positive polarity). An algorithm can be used to calculate the message parameter. For example, the algorithm may be based on supervised machine learning techniques. In a further example, if the term “kind of” is near the anchor “don’t like” (e.g., as in the sentence “I kind of don’t like”), the term “kind

of” may be identified as an amplifier term that indicates a medium intensity of the negative polarity. In this case, a message parameter can be generated based on the identification of the medium intensity of the negative polarity. As a non-limiting example, the message parameter can be a numerical value that indicates the medium intensity of the negative polarity (e.g., a message
5 parameter of 40, as opposed to the message parameter of 20). In some instances, the message parameter can be used to determine which secondary queue is to store the communication.

[00103] In some implementations, the characteristic of a message can be the sentiment associated with the message. The message parameter can represent the sentiment of the message. For example, if the sentiment of the message is happy, the message parameter can be a certain
10 value or range of values, whereas, if the sentiment of the message is angry, the message parameter can be another value or range of values. Determining whether to switch between the bots and the terminal device can be based on the message parameter, which is continuously and automatically updated with each new message received at communication server 710.

[00104] In some implementations, communication server 710 may recommend or predict
15 responses to messages received from network device 705. For example, communication server 710 can include a message recommendation system, which can evaluate messages received from network device 705 and use a machine-learning model to recommend responses to those received messages. The message recommendation system can display a set of recommended messages on terminal device 715 to assist the agent in communicating with network device 705.

[00105] **FIG. 8** shows a block diagram representing network environment 800 for dynamically
20 selecting endpoints across multiple communication channels. In some implementations, network environment 800 may include network device 805, terminal device 810, and communication server 820. Network device 805 may be similar to network device 705, and thus, a description is omitted here for the sake of brevity. Terminal device 810 may be similar to terminal device 715, and thus,
25 a description is omitted here for the sake of brevity. Communication server 820 may be similar to Communication server 710, and thus, a description is omitted here for the sake of brevity.

[00106] Communication server 820 may establish or facilitate the establishment of a communication channel between network device 805 and terminal device 810. As illustrated in FIG. 8, communication server 820 may establish communication channel C 840, which enables

network device 805 and terminal device 810 to exchange one or more messages. As a non-limiting example, communication channel C 840 may be a web chat feature of a website, communication channel B 835 may be a chat application running on a mobile device (e.g., a smart phone), and communication channel A 830 may be a voice over Internet Protocol (VOIP) audio channel that
5 enables the agent to communicate with the user.

[00107] Communication server 820 may configure bot 825 to autonomously communicate with network device 805. In some implementations, bot 825 may access and execute one or more protocols that enable bot 825 to communicate with network device 805 using communication channel C 840. Continuing with the non-limiting example above, bot 825 may access and execute
10 a protocol for communicating over the web chat feature of the website. In this example, the protocol may include a coding language specific to the web chat feature for exchanging messages using the web chat feature. The protocol may include code that, when executed, converts a message (e.g., a string of text or other content) inputted by an agent at terminal device 810 into structured content (e.g., content separated into independent data fields), and maps the structured content to
15 elements of the web chat feature of the website. As input is received at terminal device 810 (e.g., by the agent), bot 825 can translate the structured content to the elements of the web chat feature to enable the message to be communicated using the web chat feature.

[00108] In some implementations, bot 825 can also be configured to communicate with network device 805 using communication channel B 835. Communication channel B 835 can be a different
20 communication channel from communication channel C 840. Further, communication channel B 835 may require different elements to facilitate communication than the elements required for communication channel C 840. Bot 825 can be configured to translate the structured content to the elements of communication channel B 835. Continuing with the non-limiting example described above, communication channel B 835 may be an in-app chat feature of a native application running
25 on a smart phone. One or more elements may be required in order to facilitate communication using communication channel B 835. For example, FACEBOOK MESSENGER may be the native application running on the smart phone. In this example, the one or more elements of FACEBOOK MESSENGER may be templates specific to FACEBOOK MESSENGER that are required to facilitate communication using FACEBOOK MESSENGER. The protocol that enables bot 825 to
30 communicate using communication channel B 835 may map the structured content to the templates

of the FACEBOOK MESSENGER native application in order to transmit the structured content as a message within the FACEBOOK MESSENGER application.

[00109] In some examples, a mobile application (e.g., a mobile native application) may include executable code (stored in the mobile device or at one or more external servers) that can be
5 executed using the operating system of the network device (e.g., a smartphone). In some examples, the mobile application may include a hybrid mobile application that is comprised of native user interface (UI) components (generated and stored at the mobile device), but is written in an interpreted language (e.g., using Web-based coding languages). The present disclosure is not limited to mobile native applications or hybrid applications, and thus, any type of mobile
10 application may be used in the methods described herein.

[00110] In some implementations, bot 825 can also be configured to communicate with network device 805 using communication channel A 830. Communication channel A 835 can be a different communication channel from communication channel C 840 and communication channel B 835. Further, communication channel A 830 may require different elements to facilitate communication
15 than the elements required for communication channel C 840 and for communication channel B 835. Bot 825 can be configured to translate the structured content to the elements of communication channel A 830. Continuing with the non-limiting example described above, communication channel A 830 may be a VOIP audio communication link between network device 805 and terminal device 810. One or more elements may be required in order to facilitate
20 communication using communication channel A 830. The protocol may include a mapping of the structured content to the elements associated with communication channel A 830.

[00111] In some implementations, communication server 820 may be configured to dynamically, autonomously, and/or automatically transfer a communication session between different communication channels, so that bot 825 can continuously communicate with network
25 device 805, regardless of the communication channel. For example, network device 805 may be communicating with terminal device 810 using a first communication channel 845 (i.e., communication channel C 840). Network device 805 may transmit a message indicating that the user operating network device 805 intends to change the communication channel currently being used for the communication session. For example, network device 805 may indicate that second
30 communication channel 850 is the target communication channel for continuing the

communication session with terminal device 810. Bot 825 can automatically detect the indication that the communication channel should be changed from first communication channel 845 to second communication channel 850. For example, bot 825 may continuously evaluate messages exchanged during the communication session to detect that the communication channel should be changed. Upon detecting the indication that the communication channel should be changed, communication server may identify the user identifier associated with network device 805. For example, user data database 815 may store user identifiers for various users. A user identifier may be a string of text and/or numbers that uniquely identifies a network device. If, at any given time, communication server 820 determines that the same user identifier is associated with two active communication channels, communication server 820 can recognize that the network device is requesting to continue a communication session but to change the communication channels.

[00112] Communication server 820 may be configured to support continuity between different communication channels. For example, the target communication channel (e.g., second communication channel 850) can be automatically used by bot 825 to continue the communication session with network device 805, but using second communication channel 850, instead of first communication channel 845. In some implementations, bot 825 may automatically transmit a message to network device 805 using second communication channel 850. Transmitting the message to network device 805 may indicate to network device 805 that the transfer of communication channels is complete. In some implementations, communication server 820 may automatically detect that the communication channel has been changed from first communication channel 845 to second communication channel 850. For example, communication server 820 may recognize the user identifier associated with network device 805 when network device 805 is communicating with bot 825 using first communication channel 845. If network device 805 begins using second communication channel 850 (e.g., without indicating the intention to change communication channels during the communication session), communication server 820 can automatically detect that the user identifier for network device 805 is currently associated with two active communication channels (e.g., first communication channel 845 and second communication channel 850). Communication server 820 can detect that first communication channel 845 is associated with a recent history of messages (e.g., messages transmitted or exchanged within the last five minutes) and that second communication channel 850 is not associated with a recent history of messages (e.g., within the last few minutes). As a result, communication server 820 can

determine that network device 805 is requesting to transfer the communication session from first communication channel 845 to second communication channel 850. Communication server 820 can implement the transfer by accessing the protocol associated with second communication channel 850, and executing bot 825 using the accessed protocol to enable bot 825 or terminal device 810 to communicate with network device 805 using second communication channel 850, instead of using first communication channel 845.

[00113] In some implementations, one or more machine-learning techniques can be used to identify patterns in the communication channel usage of network device 805. For example, the usage of communication channels by network device 805 can be tracked and recorded (and stored as historical data). Machine-learning techniques can be applied to the historical data to identify which communication channel network device 805 is most likely to use when communicating with a particular entity (e.g., bot, company, terminal device, agent, and so on). When initiating communications from terminal device 810 (or bot 825 or any other terminal device) to network device 805, communication server 820 can establish a communication channel of the type that network device 805 is most likely to use (based on the results of the machine learning techniques). As network device 805 begins to use a different communication channel more frequently, communication server 820 can identify this changing trend and initiate communication sessions using the most used or most frequently used communication channel.

[00114] FIG. 9 shows a block diagram representing network environment 900 for enhancing endpoint selection using machine-learning techniques. Network environment 900 may include network device 905 (operated by a user) communication server 910, bot 915 and terminal device 920. Communication server 910 can facilitate the establishment of a communication channel that enables network device 905 and at least one bot 915 and terminal device 920 to communication.

[00115] Communication server 910 may include intelligent routing system 925, message recommendation system 930, and message data store 935. Each of intelligent routing system 925 and message recommendation system 930 may include one or more computing devices with a processor and a memory that execute instructions to implement certain operations. In some implementations, intelligent routing system 925 may be a bot configured to intelligently route communications received from network devices to the appropriate destination. Intelligent routing system 925 may include one or more processors configured to execute code that causes one or

more machine-learning techniques or artificial intelligence techniques to intelligently route messages. In some implementations, intelligent routing system 925 can execute one or more machine-learning techniques to train a model that predicts a destination associated with a message received from network device 905.

5 **[00116]** As a non-limiting example, intelligent routing system 925 may receive a message from network device 905 through a communication channel established or facilitated by communication server 910 (e.g., a native application configured to enable users to communicate with each other across various devices). Intelligent routing system 925 may evaluate the incoming message according to certain embodiments described above. For example, intelligent routing system 925
10 may evaluate the content (e.g., text, audio clips, images, emoticons, or other suitable content) included in the received message using a trained machine-learning model. The content of the message can be inputted into the machine-learning model to generate a predicted destination (e.g., a particular terminal device or bot). The machine-learning model may be continuously trained based on feedback signal 940 received from network device 905. In some implementations,
15 intelligent routing system 925 may request an acknowledgement from network device 905 of the predicted destination. As a non-limiting example, intelligent routing system 925 may evaluate the message using a machine-learning technique, and a result of the evaluation may include a predication that bot 915 is the destination for the message. To confirm, intelligent routing system 925 may automatically request feedback signal 940. For example, feedback signal 940 may include
20 a request for network device 905 to acknowledge whether bot 915 is the correct destination for the message (e.g., “Is Technical Support the correct destination?”). If network device 905 transmits the acknowledgement that bot 915 is the correct destination (e.g., the destination intended by the user operating network device 905), then intelligent routing system 925 may train the machine-learning model to predict that future messages including the exact or similar content (e.g., a
25 threshold of similarity, such as 10 percent difference in content) as the received message are to be routed to bot 915. However, if intelligent routing system 925 receives feedback signal 940 indicating that bot 915 is not the correct or intended destination for the received message, but rather terminal device 920 was the correct or intended destination, intelligent routing system 925 can train the machine-learning model that future messages including the exact or similar content
30 as the received message are to be routed to terminal device 920 (instead of bot 915). In some implementations, intelligent routing system 925 may not immediately update or train the machine-

learning model to route future messages to terminal device 920, but rather, intelligent routing system 925 may wait for a threshold number of incorrect routings to bot 915 before routing all future messages with the exact same or similar content as the received message to terminal device 920. As a non-limiting example, intelligent routing system 925 may begin routing future messages (that were predicted to be routed to bot 915) to terminal device 920 instead of bot 915 after five instances of network devices transmitting feedback signals indicating that bot 915 is not the correct or intended destination.

[00117] In some embodiments, intelligent routing system 925 may select where to route a given message based on bids received to handle a particular request in the message. Intelligent routing system 925 may broadcast an intent to disparate services and determine who wants to bid on handling the request. Bidding parties may respond with their level of confidence in successfully handling the request and a plan to execute handling of the request. Intelligent routing system 925 may evaluate all of the responses from the bidding parties and, based on machine learning policies, determine which bidding party to use for a given message.

[00118] Message data store 935 may store some (e.g., but not all) or all messages received in the past from one or more network devices. Further, message data store 935 may also store some or all messages transmitted by terminal devices or bots during previous communication sessions with network devices. Message data store 935 may also store some or all messages transmitted by network devices to bots during communication sessions. Further, message data store 935 may store some or all messages transmitted by bots to network devices during communication sessions. In some implementations, message data store 935 may be a database of all messages processed (e.g., transmitted by or received at) communication server 910.

[00119] Message recommendation system 930 may analyze the database of messages stored at message data store 935. In some implementations, message recommendation system 930 may evaluate the messages stored at message data store 935 using one or more machine-learning algorithms or artificial intelligence algorithms. For example, message recommendation system 930 may execute one or more clustering algorithms, such as K-means clustering, means-shift clustering, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering, Expectation-Maximization (EM) Clustering using Gaussian Mixture Models (GMM), and other suitable machine-learning algorithms, on the database of messages stored in message data store

935. In some implementations, a recurrent neural network (RNN) or a convolutional neural network (CNN) may be used to predict response messages to assist the agent. In some implementations, message recommendation system 930 may use support vector machines (SVM), supervised, semi-supervised, ensemble techniques, or unsupervised machine-learning techniques to evaluate all previous messages to predict responses to incoming messages received from network devices during communication sessions. For example, message recommendation system 930 may evaluate the content of messages received from network devices (or messages received at communication server 910 from bots or terminal devices) and compare the results of the evaluation to the one or more clusters of previous messages stored in message data store 935. Once the cluster is identified, message recommendation system 930 can identify the most relevant response messages based on a confidence threshold. For example, an incoming message (e.g., received at communication server 910 from network device 905) may correspond to a technical issue based on the content of the incoming message. Message recommendation system 930 can identify that the incoming message corresponds to a technical issue based on an evaluation of the content of the incoming message (e.g., text evaluation). Message recommendation system 930 can access message data store 935 to identify the cluster of messages associated with technical issues. Message recommendation system 930 can select one or more responses messages within the cluster of messages based on a confidence threshold. As a non-limiting example, a confidence algorithm can be executed to generate a confidence score. A confidence score may be a percentage value where the lower the percentage, the less likely the response is a good prediction for the incoming message, and the higher the percentage, the more likely the response is a good prediction for the incoming message. A minimum confidence threshold may be defined as a measure of certainty or trustworthiness associated with each discovered pattern. Further, an example of a confidence algorithm may be the Apriori Algorithm, similarity algorithms indicating similarity between two data sets, and other suitable confidence algorithms.

[00120] FIG. 10 shows an example process for routing a conversation between bots and terminal devices during a communication session with a network device. At step 1005, one or more variables associated with a client device may be received and tracked. The client device may be operated by a client, as described further herein. The one or more variables may include, for example, costs, customer profiles (e.g., general versus VIP status, new versus returning customer), experience, historical data (e.g., past conversation transcripts, data regarding past conversations

by the user or those sharing similarities with the user), or any other input. Such variables provide context that may be considered in determining what actionable item is warranted, how such actionable item is to be performed, and what entity is designated to perform the actionable item. For example, a client may decide that they are interested in reducing costs, so as few agents as possible should be used, and instead bot processing should be performed so as to automate processes, even if that means lower customer satisfaction. Another client may be okay with costs, but instead focus on customer experience. These variables may affect whether a message is routed to an agent (via a terminal device), bot, automation, or other recipient device in relation to different actionable items, as well as the conditions of the routing (e.g., communication channel). Such variables may be continually tracked throughout the course of a conversation. As the user reveals new information, for example, such new information may represent new variables that may then be aggregated with other (e.g., historical) variables to inform subsequent routing decisions.

[00121] At step 1010, a conversation with the client may be monitored. The conversation may begin with a message sent to or received from a network device of the client by an agent or bot. The message may include information that is indicative of one or more intents. In some embodiments, the message may be in natural language, i.e., conversational. For example, the message may state, “I want to change my address and pay my bill.” Regardless of whether the user is communicating with a human agent, a bot, automation, etc., or channel of such communication, each user message may be continually monitored, tracked, analyzed. For example, a conversation may begin on a social media page, but may then be routed to various messaging applications, email applications, telephones, or other communication channels. Regardless of changes in channel or medium, therefore, all messages in the conversation may be monitored and evaluated in context.

[00122] At step 1015, each message in the conversation may be parsed to identify what the indicated intents are. For example, the message, “I want to change my address and pay my bill” may be parsed to identify an intent to “change_address” and an intent to “pay_bill”). Each of the intents may be associated with an actionable item defined by one or more policies. The routing policies or rules may be defined by the entity or business. Each routing rule may specify different variables or conditions, as well as one or more actionable items to take when the variables or conditions are met. For example, the intent to “change_address” may be associated with an actionable item in which the user is routed to a bot programmed to query the user about the updated

address and to update an existing user profile. Likewise, the intent to “pay_bill” may be associated with an actionable item in which the user is routed to an automation programmed to initiate payment toward his or her bill.

[00123] In some embodiments, routing policies may specify different actionable items based on the variables that are tracked in step 1005. For example, VIP customers may not be routed to either bot or automation, but rather routed to human agents on a prioritized basis. Different conditions of the conversation (e.g., calendar, geographic location, historical routes, preferences, weather, etc.) may also result in different routing decisions.

[00124] At step 1020, the first intent and the second intent are analyzed to determine a prioritization for executing the actionable item(s). The prioritization may indicate the order in which each actionable item are performed. Continuing the above example, it may be determined that the address change should be completed first, because that information may be needed in order to effectively pay the bill. The prioritization may also be determined based on other data, such as the most efficient order to execute the actionable items. For example, if the first actionable item is quicker to execute, it may be executed first.

[00125] At step 1025, the intent(s) may be fed into a machine learning model to obtain a concertation plan. The machine learning model may evaluate the variables and the intents in conjunction with the applicable routing rules to identify one or more endpoints to which to route the conversation. The conversation plan may therefore include the identified endpoints, which may further be ordered based on prioritization. For example, the endpoints may include a first endpoint for the first intent and a second endpoint for the second intent. The identification of endpoints may differ based on the different variables associated with the conversation. For example, if the one or more variables indicate that a user seeking to change an airline flight may have a preference for maximizing cost savings (rather than time savings), the user may be presented with flight options that have been filtered to maximize cost savings. If the one or more policies indicate that customer satisfaction should be maximized, variables related to current sentiment (e.g., level of customer satisfaction or lack thereof) may be weighted towards routing to a human agent as much as possible. Other variables may include processing speed for given tasks, accuracy of tasks performed by bots and agents, and the like.

[00126] At steps 1030, the conversation routed to the endpoints in the order specified by the prioritization. For example, the first intent may be routed to the first endpoint. The first endpoint may thereafter execute the first actionable item, e.g., a change of address executed by a bot. The network device may then be transferred from the first endpoint to the second endpoint, e.g., from a bot to a terminal device, and the second endpoint may thereafter execute the second actionable item, e.g., a bill pay.

[00127] The conversation may be routed sequentially based on the prioritization of the respective associated intents. In some embodiments, however, new intents may emerge that may be assigned high priority, resulting in a different route decision for the conversation than originally identified in the conversation plan. The conversation may therefore experience switching from an original endpoint to a new endpoint based on such dynamic, real-time evaluation and decisions. In some cases, the user may be further queried regarding the indicated new intent to confirm the presence of the new intent, as well as to obtain context or variables to consider in making switching decisions.

[00128] In step 1035, it may be determined whether a new intent is indicated by the conversation. Such determination may be based on querying the user whether they have any further needs or requests. In some embodiments, such determination may be indicated by other information associated with the conversation. A new intent may also be identified where an originally-identified intent is determined to have changed. If a new intent appears to have been indicated, the method may return to step 1005. As noted above, variables and conversations may be continually monitored. As such, different messages from the user or changed conditions may indicate new intents. For example, a conversation regarding purchase of a single item (e.g., television) may elicit interest in certain features that are absent from the item (e.g., surround sound system). Other examples may include flight change requests where changing weather patterns may indicate that certain connection points are likely to experience delay. Likewise, during bill payment, the user may express dismay at certain costs, which may prompt the system to present promotions and offers to reduce costs. In returning to step 1005, however, all the contextual information related to previous messages in the conversation (e.g., variables, intents) may continue to be considered.

[00129] If no new intent is indicated, the method may proceed to step 1040, in which the results and outcomes of the conversation may be add to the machine learning model. Such data regarding conversational results and outcomes may be analyzed and used to refine current policies that may be applied to future routing decisions.

5 [00130] FIG. 11 shows a block diagram representing a network environment in which a system for orchestrating conversations driven by artificial intelligence (AI) may be implemented. The network environment may include a conversational operating system (OS) 1110, context warehouse 1120, concierge bot 1130, conversation orchestrator 1140, and external service(s) 1150.

[00131] Conversational OS 1110 may be inclusive a variety of conversational modules for
10 analyzing conversations and performing various conversational functions. Conversational OS 1110 may include, for example, modules configured to perform natural language understanding (NLU), intent services, contextual services, sentiment analysis, feedback collection, dialog management, service provider discovery and assignment, action determination, dispatch, and conversational intelligence. Conversational OS 1110 may therefore support automated
15 conversation, as well as track conversations with human agents. The analysis may be applied to chat transcripts from past conversations and may also provide real-time visibility and aggregated analysis of current conversations. Different entities and brands may specify rules for how conversations with customers or clients are to be managed. In addition, various learning models may be built and refined over time based on conversations conducted by agents (human or bot) of
20 the entity or brand.

[00132] Context warehouse 1120 may include any contextual information that may be relevant to a conversation, including historical data, current data, and user-specific data. Such contextual data may supply one or more of the variables considered in step 1005. Contextual warehouse 1120 may further communicate with various external systems 1150 to obtain contextual data regarding
25 the conversation. The data regarding the conversation may be aggregated and considered throughout the course of a conversation, as well as used to update context warehouse 1120 on a continual basis. Such data may thereafter be analyzed and used to identify trends and patterns, as well as refine various rules and policies.

[00133] Concierge bot 1130 may serve as an initial endpoint in a conversation. Concierge both 1130 may be programmed to elicit data from the user, for example, that may be indicative of one or more intents. The conversation may thereafter be routed to different bots and other types of endpoints (e.g., human agents, automations) based on current intent(s).

5 [00134] Conversation orchestrator 1140 may operate in conjunction with conversational OA 1110, context warehouse 1120, concierge bot 1130 (and/or other endpoints), and external services 1150 to perform the method of FIG. 10. In particular, conversation orchestrator 1140 may continually monitor and analyze ongoing conversations, contexts (e.g, variables, conditions) against one or more policies (e.g, associated with a business, brand, or other entity) to identify
10 actionable item(s) in a manner that is personalized or tailored to the user, brand, and other conditions of the conversation. Such actionable items may include routing decisions that may entail switching among different endpoints, taking over a conversation, etc., so as to best address the intent of the user in relation to the brand.

[00135] FIG. 12 shows a block diagram representing exemplary information flow within an AI-
15 driven orchestration of a conversation. As illustrated, a user (“Tom Dean”) conversation may be analyzed to extract certain context data, including customer attributes and indications of intent. Such context data may be stored in context warehouse 1120. In addition, such context data may be provided to conversation orchestrator 1140, which matches the applicable context data to one or more policies that specify certain suggested actionable items. The actionable items in the
20 example illustrated in FIG. 12 include triggering a billing credit and a personalized response.

[00136] FIGs. 13A-13C shows exemplary conversations that have been orchestrated by AI-driven systems. In particular, FIG. 13A illustrates a graphic user interface presented to the user during a conversation, as well as a diagram of background data and routing actions. The conversation of FIG. 13A begins when the user texts “Hi.” Data regarding the consumer (e.g,
25 “Consumer Info”) may retrieved to identify the name of the user, which may thereafter be used by a routing bot to respond to the user in a personalized way and to further query the user for intent data. As illustrated, the “Consumer Info” includes (previous conversations,” as well as a variety of different user-specific contextual data, which may be used to formulate responses and make decisions within the conversation. As illustrated, the user may indicate a desire to be helped in

relation to a “storm delay.” The background “previous conversation” data may indicate past messages that included “delayed” and associated responses.

[00137] FIG. 13B illustrates a graphic user interface presented to the user during a different conversation, as well as a diagram of policies identified as being relevant in the background. In the illustrated conversation, the user has indicated a “Need to reschedule a flight.” The background analysis may identify which policies are applicable to such intent. The policies may further be filtered and prioritized based on contextual data, including weather data (e.g., policies specific to “Weather_Delay), VIP status (e.g, VIPRule), and other conditions of the conversation.

[00138] FIG. 13C illustrates a graphic user interface presented to the user during a conversation, as well as a diagram of feedback analysis as to whether each routing decision is relevant to the intent of the user. For example, the message may concern “directions,” which may be identified as indicative of an intent for “navigation.directions.” The conversation may be routed to a bot programmed to assist with navigation (e.g, “navigationBot”) based on such intent. As illustrated, the identified bot may have been relevant in 88% of inquiries determined to have the identified intent. In addition, the user feedback may thereafter be used to evaluate user sentiment and satisfaction as to whether the routing decision was indeed relevant to the user’s needs.

[00139] Specific details are given in the above description to provide a thorough understanding of the embodiments. However, it is understood that the embodiments can be practiced without these specific details. For example, circuits can be shown as block diagrams in order not to obscure the embodiments in unnecessary detail. In other instances, well-known circuits, processes, algorithms, structures, and techniques can be shown without unnecessary detail in order to avoid obscuring the embodiments.

[00140] Implementation of the techniques, blocks, steps and means described above can be done in various ways. For example, these techniques, blocks, steps and means can be implemented in hardware, software, or a combination thereof. For a hardware implementation, the processing units can be implemented within one or more application specific integrated circuits (ASICs), digital signal processors (DSPs), digital signal processing devices (DSPDs), programmable logic devices (PLDs), field programmable gate arrays (FPGAs), processors, controllers, micro-

controllers, microprocessors, other electronic units designed to perform the functions described above, and/or a combination thereof.

[00141] Also, it is noted that portions of the embodiments can be described as a process which is depicted as a flowchart, a flow diagram, a data flow diagram, a structure diagram, or a block diagram. Although a flowchart can describe the operations as a sequential process, many of the operations can be performed in parallel or concurrently. In addition, the order of the operations can be re-arranged. A process is terminated when its operations are completed, but could have additional steps not included in the figure. A process can correspond to a method, a function, a procedure, a subroutine, a subprogram, etc. When a process corresponds to a function, its termination corresponds to a return of the function to the calling function or the main function.

[00142] Furthermore, embodiments can be implemented by hardware, software, scripting languages, firmware, middleware, microcode, hardware description languages, and/or any combination thereof. When implemented in software, firmware, middleware, scripting language, and/or microcode, the program code or code segments to perform the necessary tasks can be stored in a machine readable medium such as a storage medium. A code segment or machine-executable instruction can represent a procedure, a function, a subprogram, a program, a routine, a subroutine, a module, a software package, a script, a class, or any combination of instructions, data structures, and/or program statements. A code segment can be coupled to another code segment or a hardware circuit by passing and/or receiving information, data, arguments, parameters, and/or memory contents. Information, arguments, parameters, data, etc. can be passed, forwarded, or transmitted via any suitable means including memory sharing, message passing, ticket passing, network transmission, etc.

[00143] For a firmware and/or software implementation, the methodologies can be implemented with modules (e.g., procedures, functions, and so on) that perform the functions described herein. Any machine-readable medium tangibly embodying instructions can be used in implementing the methodologies described herein. For example, software codes can be stored in a memory. Memory can be implemented within the processor or external to the processor. As used herein the term “memory” refers to any type of long term, short term, volatile, nonvolatile, or other storage medium and is not to be limited to any particular type of memory or number of memories, or type of media upon which memory is stored.

[00144] Moreover, as disclosed herein, the term "storage medium", "storage" or "memory" can represent one or more memories for storing data, including read only memory (ROM), random access memory (RAM), magnetic RAM, core memory, magnetic disk storage mediums, optical storage mediums, flash memory devices and/or other machine readable mediums for storing
5 information. The term "machine-readable medium" includes, but is not limited to portable or fixed storage devices, optical storage devices, wireless channels, and/or various other storage mediums capable of storing that contain or carry instruction(s) and/or data.

[00145] While the principles of the disclosure have been described above in connection with specific apparatuses and methods, it is to be clearly understood that this description is made only
10 by way of example and not as limitation on the scope of the disclosure.

CLAIMS

WHAT IS CLAIMED IS:

1. A computer-implemented method comprising:
 - receiving one or more variables associated with a client device, wherein the client device is operated by a client;
 - receiving a message for the client from a network device, wherein the message includes a first intent and a second intent;
 - parsing the message to identify the first intent and the second intent, wherein the first intent is associated with a first actionable item and the second intent is associated with a second actionable item;
 - analyzing the first intent and the second intent to determine a prioritization for executing the first actionable item and the second actionable item, wherein the prioritization indicates that the first actionable item should be executed first and the second actionable item should be executed second;
 - feeding the first intent and the second intent into a machine learning model, wherein the machine learning model identifies a first endpoint for the first intent and a second endpoint for the second intent by optimizing the one or more variables associated with the client device;
 - routing the first intent to the first endpoint, wherein the first endpoint thereafter executes the first actionable item; and
 - routing the second intent to the second endpoint, wherein the second endpoint thereafter executes the second actionable item.
2. The computer-implemented method of claim 1, wherein the first endpoint is a bot.
3. The computer-implemented method of claim 1, wherein the first endpoint is a terminal device, and wherein the terminal device is operated by an agent.
4. The computer-implemented method of claim 1, wherein the first endpoint is a bot and the second endpoint is a terminal device, and wherein the terminal device is operated by an agent.

5. The computer-implemented method of claim 1, wherein the message is in natural language.

6. A computer-program product tangibly embodied in a non-transitory machine-readable storage medium, including instructions that, when executed by one or more processors, cause the one or more processors to perform operations including:

receiving one or more variables associated with a client device, wherein the client device is operated by a client;

receiving a message for the client from a network device, wherein the message includes a first intent and a second intent;

parsing the message to identify the first intent and the second intent, wherein the first intent is associated with a first actionable item and the second intent is associated with a second actionable item;

analyzing the first intent and the second intent to determine a prioritization for executing the first actionable item and the second actionable item, wherein the prioritization indicates that the first actionable item should be executed first and the second actionable item should be executed second;

feeding the first intent and the second intent into a machine learning model, wherein the machine learning model identifies a first endpoint for the first intent and a second endpoint for the second intent by optimizing the one or more variables associated with the client device;

routing the first intent to the first endpoint, wherein the first endpoint thereafter executes the first actionable item; and

routing the second intent to the second endpoint, wherein the second endpoint thereafter executes the second actionable item.

7. The computer-program product of claim 6, wherein the first endpoint is a bot.

8. The computer-program product of claim 6, wherein the first endpoint is a terminal device, and wherein the terminal device is operated by an agent.

9. The computer-program product of claim 6, wherein the first endpoint is a bot and the second endpoint is a terminal device, and wherein the terminal device is operated by an agent.

10. The computer-program product of claim 6, wherein the message is in natural language.

11. A system comprising:

one or more processors; and

one or more non-transitory machine-readable storage media containing instructions that, when executed on the one or more processors, cause the one or more processors to perform operations including:

receiving one or more variables associated with a client device, wherein the client device is operated by a client;

receiving a message for the client from a network device, wherein the message includes a first intent and a second intent;

parsing the message to identify the first intent and the second intent, wherein the first intent is associated with a first actionable item and the second intent is associated with a second actionable item;

analyzing the first intent and the second intent to determine a prioritization for executing the first actionable item and the second actionable item, wherein the prioritization indicates that the first actionable item should be executed first and the second actionable item should be executed second;

feeding the first intent and the second intent into a machine learning model, wherein the machine learning model identifies a first endpoint for the first intent and a second endpoint for the second intent by optimizing the one or more variables associated with the client device;

routing the first intent to the first endpoint, wherein the first endpoint thereafter executes the first actionable item; and

routing the second intent to the second endpoint, wherein the second endpoint thereafter executes the second actionable item.

12. The system of claim 11, wherein the first endpoint is a bot.

13. The system of claim 11, wherein the first endpoint is a terminal device, and wherein the terminal device is operated by an agent.

14. The system of claim 11, wherein the first endpoint is a bot and the second endpoint is a terminal device, and wherein the terminal device is operated by an agent.

15. The system of claim 11, wherein the message is in natural language.