additional value(s) may have been determined during interaction with the 3P agent, and those value(s) also provided in the invocation request to the additional 3P agent. Which of multiple additional 3P agent(s) are selected for transmitting the additional invocation request may be based on the additional input itself (e.g., if it references one of the additional 3P agents by name or characteristic) and/or based on other factors (e.g., the next most popular 3P agent if the additional input is "next agent").

If, at block **294**, the system determines the received additional input does not indicate a desire to switch to a different 3P agent, the system proceeds to block **252**.

FIG. **2**B is a flowchart illustrating an example **200**B of some implementations of certain blocks of the method **200** of FIG. **2**A. For convenience, the operations of the flow chart are described with reference to a system that performs the operations. This system may include various components of various computer systems, such as 3P agent engine **120** and/or one or more other components of automated assistant **110**. Moreover, while operations of the example **200**B are shown in a particular order, this is not meant to be limiting. One or more operations may be reordered, omitted or added.

Example **200**B provides some particular examples of blocks **252**, **260**, **270**, and **280** of FIG. **2**A.

Block **252**A of FIG. **2**B is a particular example of block **252** of FIG. **2**A. At block **252**A, the system selects a 3P intended action based on user input.

Blocks **260**A, **260**B, and **2660**C of FIG. **2**B are a particular example of block **260** of FIG. **2**A. At block **260**A, the system identifies mandatory and optional parameters for the intended action selected at block **252**A.

At block **260**B, the system determines and stores the value(s) for parameter(s), if any, that are indicated in the user input received at block **252**A.

At block **260**C, the system determines whether all mandatory parameters for the intended action have been determined at the most recent iteration of block **260**B. If they have not, the system proceeds to blocks **262** and **264** of FIG. **2**A, then proceeds back to block **260**B to determine and store parameters (if any) in further user input provided in block **264**.

If, at an iteration of block **260**C, all mandatory parameters for the intended action have been determined, the system proceeds to block **270**A. At block **270**A, the system determines whether a single 3P agent for the intended action is resolvable based on the value(s) determined and stored at one or more iterations of block **260**B, past user interactions, and/or other criteria. If not, the system proceeds to blocks **272** and **274** of FIG. **2**A, then proceeds back to block **270**A to determine if further user input provided in block **274** enables resolution of a single 3P agent.

If, at an iteration of block **270**A, a single 3P agent is resolvable, the system proceeds to block **280**A. At block **280**A, the system transmits, to the single 3P agent, an invocation request that includes the parameters determined and stored at one or more iterations of block **260**B. The system then proceeds to block **282** of FIG. **2**A.

FIG. **3** is a flowchart illustrating an example method **300** that may be performed by a third-party agent according to implementations disclosed herein. This system may include various components of various computer systems, such as one or more components of one of the 3P agents **140**A-N. Moreover, while operations of method **300** are shown in a particular order, this is not meant to be limiting. One or more operations may be reordered, omitted or added.

At block **352**, the system receives an invocation request from an automated assistant. The invocation request may include an intended action (e.g., when the system can perform multiple intended actions), value(s) for parameter(s) of the intended action, and/or other data. In some implementations, the invocation request may indicate that the system should effectively or actually take over the dialog. If the invocation request indicates that the system should actually take over the dialog, the system may establish a direct network communications session with a corresponding client device. If the invocation request indicates that the system should effectively take over the dialog, the system may take over the dialog while still communicating with the component that provided the invocation request and/or a related component.

At block **354**, the system generates responsive content based on value(s) for parameter(s) that are included in the invocation request, such as values for mandatory and/or optional parameters.

At block **356**, the system provides the responsive content. For example, where the invocation request indicates the system should effectively take over the dialog and/or should only perform an intended action without engaging in a dialog, the system may transmit the responsive content to the component (or related component) that sent the invocation request. Also, for example, where the invocation request indicates the system should actually take over the dialog, the system may transmit the responsive content to a corresponding client device.

At block **358**, the system receives further content that is in reply to the responsive content provided at block **356**. The further content may include, for example, user input provided in response to the responsive content, a conversion of that user input, and/or annotations of that user input. For example, where an automated assistant or other component serves as an intermediary, the automated assistant may provide a conversion of that user input and/or annotations of that user input.

At block **360**, the system generates further responsive content based on the received further content of block **358**.

At block **362**, the system provides the further responsive content generated at block **360**. The system may optionally then return to block **358** and perform an additional iteration of of blocks **358**, **360**, and **362**. It is understood that in some implementations the system may perform multiple iterations of blocks **358**, **360**, and **362** while steering and engaging in a dialog. The system may exit the method **300** in response to satisfaction of one or more conditions (e.g., performance of the intended action) and may optionally transmit a communication to the component that provided the invocation request of block **352** to indicate that the system is exiting the method **300**.

Blocks **358**, **360**, and **362** are illustrated in broken lines in FIG. **3** to indicate that they may not be performed in some situations. For example, as described herein in some implementations the system may perform an intended action in response to the invocation request of block **352**, generate responsive content at block **354** that indicates an intended action has been performed, provide that responsive content at block **356**, then not receive further content in association with that invocation request.

FIGS. **4**, **5**, and **6** each illustrates a user **101**, a voice-enabled client device **406**, and an example of dialog that may occur between the user **101**, an automated assistant associated with the client device **406**, and a third-party agent. The client device **406** includes one or more microphones and one or more speakers. One or more aspects of the