

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
20 January 2011 (20.01.2011)

PCT

(10) International Publication Number  
**WO 2011/008554 A2**

(51) International Patent Classification: Not classified

(21) International Application Number:

PCT/US2010/040435

(22) International Filing Date:

29 June 2010 (29.06.2010)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

61/221,211 29 June 2009 (29.06.2009) US

12/826,297 29 June 2010 (29.06.2010) US

(71) Applicant (for all designated States except US): **AVAYA INC.** [US/US]; 211 Mount Airy Road, Basking Ridge, NJ 07920-2332 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **BUFORD, John** [US/US]; 39 Wilson Road, Princeton, NJ 08540 (US). **KOLBERG, Mario** [DE/GB]; 215 Cumbernauld Road, Glasgow G69 9ND (GB). **KRISHNASWAMY, Venkatesh** [IN/US]; 29 Crimson King Drive, Holmdel, NJ 07733 (US). **DHARA, Krishna, Kishore** [IN/US]; 201 Crocus Ct., Dayton, NJ 08810 (US). **WU, Xiaotao** [CN/US]; 503 Copperfield Lane, Metuchen, NJ 08840 (US).

(74) Agent: **DIMOV, Kiril**; DEMONT & BREYER, LLC, 100 Commons Way, Ste. 250, Holmdel, NJ 07733 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished upon receipt of that report (Rule 48.2(g))

(54) Title: INTERACTION DETECTION BETWEEN WEB-ENABLED AND CALL-RELATED FEATURES

(57) Abstract: A method is provided for detecting interactions between call-related and web-enabled features. In some embodiments, communication application components are configured to transmit an indication to a feature interaction detector every time a feature is invoked by the user. When the interaction detector receives an indication that a feature is being invoked, it compares the invoked feature against features that have previously been invoked (or executed) and determines whether the invoked feature interacts with any of the features that have been previously executed.



WO 2011/008554 A2

## **Interaction Detection Between Web-Enabled and Call-Related Features**

### **Cross Reference to Related Applications**

[0001] The present application claims priority to provisional application 61/221,211 filed on June 29, 2009, entitled Feature Interactions Between Internet Services and Telecommunication Services, which is herein incorporated by reference.

[0002] The underlying concepts, but not necessarily the language, of the following cases are incorporated by reference:

- (1) U.S. Application 12/392,429 and U.S. Provisional Application 61/145,700, filed on February 25, 2009 and January 19, 2009, respectively.
- (2) U.S. Application 12/392,445 and to U.S. Provisional Application 61/145,703, filed on February 25, 2009 and January 19, 2009, respectively.
- (3) U.S. Application 12/392461 and U.S. Provisional Application 61/145,706, filed on February 25, 2009 and January 19, 2009, respectively.

[0003] If there are any contradictions or inconsistencies in language between this application and one or more of the cases that have been incorporated by reference that might affect the interpretation of the claims in this case, the claims in this case should be interpreted to be consistent with the language in this case.

### **Field of the Invention**

[0004] The present invention relates to telecommunications in general, and, more particularly, to feature interaction detection.

### **Background of the Invention**

[0005] Over the years a wide variety of communications *features* (also referred to as *services*) have been developed, such as call forwarding, three-way calling, music on hold, and so forth. When two or more features are applied to a telephone call, however, it is possible that an interaction between the features might cause unexpected or unwanted behavior, such as permitting policies to be circumvented or causing the call to fail. For example, suppose a call is set up via a meet-me conferencing feature, and then a music-on-hold feature is activated during the call. If one of the call participants goes on hold, then all of the other parties on the call will also hear the music.

### **Summary of the Invention**

**[0006]** Communication features can be classified as call-related and web-enabled. Call-related features are provided by telephone network nodes, such as PBX switches, service provider switches, service provider feature servers, and others. Examples of such features include call forwarding, three-way calling, voice mail delivery, *etc.* Web-enabled features, on the other hand, are provided by servers that are not part of the telephone network and are oftentimes delivered over the Internet. Examples of web-enabled features are delivery of voicemail messages over the email, general email access, and Facebook™ access.

**[0007]** Web services are one type of web-enabled features. Web services are typically application programming interfaces (APIs) which are executed on remote system hosting the requested services. Web services are software components that are described via WSDL and are capable of being accessed via standard web-based protocols such as SOAP and HTTP.

**[0008]** Previous work in feature interaction detection has focused on call-related features. For instance, the method of feature interaction detection described in provisional application 61/145,700 uses Session Initiation Protocol (SIP) messages to detect feature interactions. The SIP messages are used to carry indications that features are invoked. This approach, however, is not adaptable to web-enabled features because web-enabled features are not guaranteed to be provided via signaling paths, such as those used to carry Session Initiation Protocol (SIP) messages. Moreover, the web-enabled features may be provided by application widgets that themselves lack Session Initiation Protocol (SIP) capability, and instead rely on other types of communications to request and receive web-enabled features. Therefore, the need exists for a method for detecting interactions between call-related and web-enabled communication features.

**[0009]** The present invention addresses this need by providing a method for detecting interactions between call-related and web-enabled features. In some embodiments, a communication application component is configured to transmit an indication to a feature interaction detector when a feature provided by the component is invoked. When the feature interaction detector receives an indication, it determines whether the invoked feature interacts with other features that have been executed or invoked. In some embodiments, the determination is performed in accordance with pre-defined feature definitions and feature interaction rules.

[0010] It is to be understood that the above embodiments and their instances are provided to better illustrate different aspects of the present invention. They are in no way exhaustive of the full scope of the invention. The following disclosure teaches examples of some embodiments of the present invention in detail.

### **Brief Description of the Drawings**

[0011] **Figure 1** depicts a schematic diagram of system 100.

[0012] **Figure 2** depicts a schematic diagram of the salient components of terminal 110.

[0013] **Figure 3** depicts a schematic diagram of the salient components of the illustrative embodiment of the present invention.

[0014] **Figure 4** depicts a schematic diagram of the salient components of the illustrative embodiment of the present invention.

[0015] **Figure 5** depicts a flowchart of the salient tasks associated with the operation of the illustrative embodiment of the present invention.

[0016] **Figure 6** depicts a flowchart of the salient subtasks associated with the execution of task 570.

[0017] **Figure 7** depicts a flowchart of the salient tasks associated with the operation of an alternative embodiment of the present invention.

### **Detailed Description**

[0018] The terms appearing below are defined for use in this specification, including the appended claims, as follows:

- “**browser based**” means an application that is accessed via a web browser. A browser-based application can be retrieved by the browser from the memory of the communication terminal on which the browser is executing or from a remote server.
- “**feature invocation**” means requesting at least one of the execution, enabling, and activation of the feature.
- “**execute**” means at least one of the following:
  - (1) causing a feature to provide a one-time service; for example, when a search feature is executed, it provides a set of search results; when the search feature is executed for a second time, it provides a second

set of search results;

- (2) causing a feature to become active; once a feature has become active, it starts providing its services to the user on a continuous basis; for example, when a call screening feature is active, it operates block all calls to numbers on a do-not-call list over the time period during which the feature is active; and
  - (3) causing a feature to provide its functionality to the user.
- **"feature"** means a service which is provided to the user of a communication terminal by at least one of (1) the communication terminal OR (2) a server that is remote to the communication terminal OR (3) both the communication terminal and the server that is remote to the communication terminal.
  - **"call-related feature"** means a feature that possess the following characteristics:
    - (1) the feature directly changes the state of a real-time communication session between a communication terminal and another node on a communication network (*e.g.*, another communication terminal, voicemail server, *etc.*), and
    - (2) when the feature is not provided by software executing on a communication terminal, the feature is invoked by the communication terminal over a signaling path that is used for the conduct of real-time communication sessions.
  - **"signaling path"** is a communication path between two nodes in a communication network (*e.g.*, communication terminals, proxy servers, registrar servers, *etc.*) that is used by a communication terminal to send and receive signaling that is used for controlling the state of real-time communication sessions, such as telephone calls and video calls, that are enabled by the communication network.
  - **"web-enabled feature"** means a feature that possesses the following characteristics:
    - (1) the feature does not directly change the states of real-time communication sessions; and
    - (2) when the feature is not provided by software executing on a

communication terminal, the feature is invoked by the communication terminal over a communication path that is at least partially disjoint from any possible signaling path between the communication terminal and another communication network node.

**[0019] Figure 1** depicts a schematic diagram of system 100. In accordance with the illustrative embodiment of the present invention, system 100 comprises terminal 110, terminal 150, server 130, server 140, and network 170.

**[0020]** Terminal 110 is a cellular telephone. Terminal 110 is connected to the communication network of a telephone service provider. The communication network of the telephone service provider enables terminal 110 to participate in real-time communication sessions, such as telephone calls, video calls, *etc.* In the illustrative embodiment, the communication network of the telephone service provider is a SIP network. The transport-layer network for the SIP network is a packet-switched network (*e.g.*, the Internet, LAN, WAN, *etc.*). It will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention in which the communication network uses another type of signaling (*e.g.*, GSM signaling, CDMA signaling, *etc.*) and operates in conjunction with another type of transport-layer network (*e.g.*, a cellular network, the public switched telephone network, *etc.*).

**[0021]** In the illustrative embodiment, terminal 110 is engaged in a communication session with terminal 150. The communication session is a Session Initiation Protocol (SIP) telephone call. The telephone call is carried over a signaling path and media path. The signaling path is used for the exchange of communication protocol signaling. And the media path is used for the exchange of voice. It will be clear to those skilled in the art, after reading this disclosure, in which another communication protocol is used (*e.g.*, Skype<sup>™</sup>, H.323, SS7, *etc.*).

**[0022]** Server 130 is a node in the communication network of the telephone service provider of terminal 110. In the illustrative embodiment of the present invention server 130 operates as a SIP proxy and is used in establishing the communication session between terminal 110 and terminal 150. In addition, server 130 is capable of providing one or more call-related services, such as call-blocking. Server 130 and terminal 110 communicate via a signaling path.

**[0023]** It will be clear to those skilled in the art, after reading this disclosure, that alternative embodiments of the present invention can be devised in which server 130 is

a node in another type of node (e.g., a post office switch, a PBX switch). Furthermore, it will be clear to those skilled in the art, after reading this disclosure, that alternative embodiments of the present invention in which server 130 is a node in another type of network e.g., SIP network, VoIP network, GSM network, CDMA network, Public Switched Telephone Network, etc.). And still furthermore, it will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention in which server 130 is a node dedicated to executing feature interaction detector (FID) 300. (See, e.g., Figure 3, etc.)

**[0024]** Server 140 provides access to one or more web-enabled features. In particular, server 140 is a server that executes an information retrieval application. The information retrieval application is run as a web-service and it is described in the discussion with respect to Figure 2.

**[0025]** Terminal 110 communicates with server 140 via a web service path. The web service path is a communication path that is at least partially disjoint from the signaling path. That is, at least one leg of the web service path is not used for the sending and transmission of communication protocol signaling.

**[0026]** In some embodiments, the web service path is used for the transmission of web-service messages. The web-service messages are encapsulated inside HTTP messages and use SOAP to perform remote procedure calls that are necessary for the invocation of web-enabled features.

**[0027]** In alternative embodiments, the communication path and the web service path are established by terminal 110 via separate interfaces. The web service path is established via a Wi-Fi interface, whereas the signaling part is established via telephony interface (e.g., GSM interface, etc.). Terminal 110 is connected to the Internet via an access point, such as a wireless LAN router, etc. At the same time, terminal 110 uses the telephony interface to connect to a telephony network (e.g. GSM network, etc.) via a cellular network base station.

**[0028]** Network 170 is a communications network that transports signals between server 130, server 140, and terminal 150. In accordance with the illustrative embodiment of the present invention, network 170 is the Internet, but it will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention in which network 150 is another type of communication network (e.g. a local area network, the Public Switched Telephone Network, a cellular network, wide area network, etc.).

[0029] Network 170 comprises a telecommunication domain and web service domain. In the illustrative embodiment, the telecommunication domain is an administrative domain comprising server(s) that are managed by a telephone service provider. The web service domain is an administrative domain that is not managed by the telephone service provider. Server 130 is part of the telecommunication domain while server 140 is part of the web service domain.

[0030] In some embodiments, the telecommunication domain is a private network (or network with restricted access such as the telephone service provider's network), and the web-service domain another network (e.g., the Internet, etc.). In these embodiments, the telecommunications domain and the web service domain are connected via one or more gateways and have distinct address spaces.

[0031] Terminal 150 is a cellular telephone. Those skilled in the art will recognize, after reading this disclosure, how to make and use alternative embodiments of the present invention in which one of terminals 110 and 150 is another communication device, such as, for example, and without limitation, a smart phone, desk set telephone, portable digital assistant (PDA), media streaming server, desktop computer, etc. In the illustrative embodiment, terminals 110 and 150 use the Session Initiation Protocol (SIP). But it will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention in which any signaling protocol is used (e.g., ss7, H.323, etc.).

[0032] **Figure 2** depicts a schematic diagram of the salient components of terminal 110. In the illustrative embodiment of the present invention, terminal 110 comprises communication application 200.

[0033] Communication application 200 is browser-based software for the conduct of communication sessions (e.g., telephone calls, video calls, teleconferences, etc.). In accordance with the illustrative embodiment, communication application 200 uses the Session Initiation Protocol (SIP). It will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention in which any other communication protocol is used (e.g., H.323, Skype, etc.).

[0034] Although, in the illustrative embodiment, communication application 200 is a browser-based application, it will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention, in which communication application 200 is any type of communication application known in the art (e.g., non-browser based, etc.). In the illustrative embodiment, communication application 200 comprises a plurality of widgets. It will be clear to those skilled in the

art, after reading this disclosure, how to make and use alternative embodiment of the present in which communication application 200 comprises other type(s) of components. Furthermore, it will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiment of the present in which communication application 200 is monolithic. That is, all of its services (or features) of the application are programmed together as a monolith body of code.

**[0035]** As noted, communication application 200 comprises a plurality of widgets. A widget is a component that can be added to a web page (e.g., HTML based document, etc.) by an end user without requiring additional programming of the component. Widgets can provide both web-enabled and call-related features to the user. Widgets have the benefit of being reusable. In accordance with the illustrative embodiment, the widgets are capable of generating one or more event notifications (or indications) when their services are invoked. The event notifications are directed to feature interaction detector (FID) 300 (See, e.g. Figure 3, etc.) and used to detect feature interactions. Although, in the illustrative embodiment, widgets are used to provide the features of communication application 200, it will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention in which other types of components are used (e.g., non-web-page based components, etc.).

**[0036]** In some embodiments, the widgets are independently addressable by feature interaction detector (FID) 300. Each widget has a unique identifier which permits feature interaction detector (FID) 300 to identify it and transmit information to it. Furthermore, in some embodiments, the independent addressing capability permits feature interaction detector (FID) 300 to poll widgets directly for the occurrence of feature events, such as feature invocation, feature activation, etc.

**[0037]** The widgets are used to provide access to both call-related and web-enabled features. Specifically, some of the widgets provide call-related features while others provide web-enabled features. The features provided by different widgets interact. The interactions are detected by Feature Interaction Detector (FID) 330. (See, e.g., Figure 3, etc.) In accordance with the illustrative embodiment, communication application comprises six widgets: telephony widget, contact list widget, call screening widget, information retrieval widget, call forwarding widget, and personal assistant widget. The user interfaces for these widgets are depicted in Figure 2 as telephony widget 210, contact list 220, screening list interface 230, call forwarding information retrieval tool 250, call forwarding tool 240, personal assistant 260, and message widget 270.

**[0038]** Telephony widget 210 provides access to a telephony feature. The telephony feature provides basic telephone call functionality. The telephony feature is a call-related feature.

**[0039]** Contact list interface 220 is the interface to a contact-list feature. The contact list feature allows users of communication application 200 to browse a list of contacts. It will be clear to those skilled in the art how to implement the contact-list feature.

**[0040]** Screening list interface 230 is a user interface that provides access to an outgoing call screening feature. This feature enables users to put one or more users (*i.e.*, telephone numbers, URIs, *etc.*) on a do-not-call list. Once a telephone number is put on this list, the outgoing call screening feature will prevent terminal 110 from initiating and/or establishing a communication session with that telephone number. The outgoing call screening feature is implemented by server 130. Server 130 stores the do-not-call list and blocks communication attempts to numbers that are present in the list. The outgoing call screening feature is a call-related feature.

**[0041]** Information retrieval tool 250 provides access to an information retrieval feature. The information retrieval feature is a web-enabled feature. Furthermore, the information retrieval feature is a composite feature. It comprises a search feature and a call feature.

**[0042]** The call feature is used to start communications sessions with telephone numbers that are retrieved by the search feature. The call feature is implemented by software executing on terminal 110. The call feature is a call-related feature.

**[0043]** The search feature is a web service. It is implemented as a software application that executes on server 140 and communicates with information retrieval tool via a web service interface. The search feature is a web-enabled feature.

**[0044]** It should be noted that the information retrieval feature is considered to be a web-enabled feature because it is a composite feature that comprises at least one web-enabled feature.

[0045] The information retrieval feature enables users to obtain the contact information of other users. In using the information retrieval feature, the user types in the name of a person (*e.g.*, John Peterson, *etc.*) in text field 256 and then presses search button 258. When button 258 is pressed, the search feature of information retrieval tool 250 transmits a search query to the search feature executing on server 140. The search feature consults a database and returns the contact information of the person specified in the search query.

[0046] Furthermore, information retrieval feature provides a user with a call button which can be used to establish a communication session with the user whose contact information is retrieved. When button 252 is pressed, the call feature of the information retrieval feature establishes a communication session with the telephone number retrieved by the search feature.

[0047] In some embodiments, information retrieval tool 250 uses a Wi-Fi connection (and a Wi-Fi interface of terminal 110) to contact server 140, and uses a telephony connection (and a GSM interface of terminal 110) to establish the call. In these embodiments, the communication network of the telephone service provider of terminal 110 is uninvolved in the provision of the search feature component of the information retrieval feature.

[0048] Call forwarding tool 240 enables the automatic forwarding of telephone calls that are incoming to terminal 110 when the call recipient is busy or not answering. When call forwarding tool 240 is enabled, all incoming calls are automatically forwarded to a specified number (*e.g.*, 123-456-7890). The call forwarding always feature is a call-related feature.

[0049] Personal assistant 260 is a front-end to a virtual assistant feature. The virtual assistant is a composite feature. It is composed of both call-related and web-enabled features. It comprises a call-related feature to answer incoming calls and communicate information about scheduled appointments. It also comprises a web-enabled feature that is charged with the tasks of consulting calendars and determining a preferred time to schedule the appointments.

[0050] In operation, the virtual assistant answers incoming telephone calls when a call recipient is busy or not answering. Moreover, the virtual assistant consults calendars and appointment books, and schedules appointments for the calling party to call back. The virtual assistant communicates the scheduled appointment by using text-to-speech translation. Generated speech is transmitted to the calling party via telephone connection that is established through the SIP network of server 130.

[0051] Message widget 270 is a widget for displaying message(s) (e.g., text messages, graphic messages, audio messages, *etc.*) from feature interaction detector (FID) 300. (See, e.g., Figure 3, *etc.*).

### **Interaction Between Call-Related Features and Web-Enabled Features**

[0052] The features of communication application 200 are capable of interacting with one another. For instance, the outgoing call screening and the information retrieval feature can interact sometimes. As depicted in Figure 2, the telephone number of a person (John Peterson) is searched for and retrieved by information retrieval tool 250. Call Button 252, can be used to start a telephone call with the person (*i.e.*, John Peterson).

[0053] At the same time, however, the telephone number of the person (*i.e.*, John Peterson) is listed in outgoing screening list 230. This causes the call screening feature of communication application 200 to block all outgoing calls that are made to the number.

[0054] Stated succinctly, a conflict exists in the operation of the call screening and information retrieval features. If the user clicks on button 252, one of two things will happen. Either a telephone call with number "1-123-456-7890" will be established, or the telephone call setup attempt will fail because it is blocked by the outgoing call screening feature. In either scenario, one of the features is not going to operate as expected by the user.

[0055] Notably, if the interaction, or conflict, is caught ahead of time, it can be mitigated. For instance, a message can be transmitted to communication application 200 warning the user that call button 252 may not work as expected because of conflicts with the call-blocking feature. In the illustrative embodiment, such a message is displayed by message widget 270.

[0056] Another example of feature interaction is the interaction between call forwarding tool 240 and personal assistant 260. The former forwards calls automatically when a recipient is not available, while the latter answers these calls on behalf of the recipient to schedule a call-back appointment. Only one of the two features can execute, however. An incoming call can either be answered by the virtual assistant or forwarded to a third-party number, but not both. If this conflict is caught ahead of time, it can be mitigated by allowing only the call forwarding tool to execute while blocking the execution of the personal assistant.

### **Interactions Between Features that Are Invoked by Different Communication Terminals**

[0057] The illustrative embodiment of the present invention describes interactions between features that are invoked by the same communication terminal. It will be clear to those skilled in the art, however that features that are invoked by different communication terminals can also interact with one another. An example of one such interaction is provided in the "Background" section of this disclosure. The "Background" section describes a conflict between a meet-me conferencing feature and music-on-hold feature. The conference feature is invoked by all terminals that participate in the conference, whereas the music-on-hold feature is invoked by one terminal only. Yet, the invocation of the music-on-hold feature by one terminal affects all other terminals since they also hear the music.

[0058] **Figure 3** depicts a schematic diagram of the salient components of the illustrative embodiment of the present invention. Figure 3 depicts a schematic diagram of feature interaction detector (FID) 300 and its salient components. Feature interaction detector (FID) comprises coordinator module 310, feature definitions 320, interaction definition 330, and list of invoked features 340.

[0059] Coordinator module 310 is logic for the detection and resolution of feature interactions. In the illustrative embodiment, coordinator module 310 is implemented in software. In alternative embodiments, however, coordinator module 310 is implemented in hardware. The operation of coordinator module 310 is further described in the discussion with respect to Figures 5, 6, and 7.

[0060] Record 320 is a record of one or more feature definitions. Each definition comprises at least one of a precondition, post condition, and a triggering event. A precondition is a predicate that must always be true in order for a feature to be executed. A post condition is a condition or predicate that must always be true after the execution of a feature. A triggering event is an occurrence that causes the execution of a feature.

[0061] In accordance with the illustrative embodiment, feature definitions 320 comprises the following feature definitions:

| <b>Feature name</b> | <b>Trigger</b>  | <b>Pre-condition</b>                                                       | <b>Action</b>                    | <b>Post-condition</b>                                                                                  |
|---------------------|-----------------|----------------------------------------------------------------------------|----------------------------------|--------------------------------------------------------------------------------------------------------|
| virtual assistant   | $E_c(incoming)$ | $S_c(C_1) = early,$<br>$C_1 = call(local, remote)$<br>$S_d(local) = busy,$ | bridged extension:<br>$answer()$ | $S_c(C_1) = held,$<br>$S_c(C_2) = confirmed,$<br>$C_2 = call(bridge, remote),$<br>$S_d(local) = busy,$ |

|                            |                                         |                                                                                            |                                                                                            |                                                                                                                                                                                                                       |
|----------------------------|-----------------------------------------|--------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                            |                                         | $S_u(local) = busy,$<br>$S_d(bridge) = avail,$                                             |                                                                                            | $S_u(local) = busy,$<br>$S_d(remote) = busy,$<br>$S_u(remote) = busy,$<br>$S_d(bridge) = busy$                                                                                                                        |
| information retrieval      | $E_d(incoming) \parallel E_d(outgoing)$ | $S_c(C_1) = \neg terminated,$<br>$C_1 = call(local, remote)$<br>$S_i(caller-id) = unavail$ | retrieve-info(caller-id)                                                                   | $S_i(caller-id) = ready$<br>$E_i(ir-done(caller-id))$<br>$S_c(C_1) = \neg terminated$                                                                                                                                 |
| call forwarding            | $E_d(incoming)$                         | $S_c(C_1) = early,$<br>$C_1 = call(local, remote)$                                         | referred-to = select(forwarded-to),<br>connect(remote, forwarded-to),<br>disconnect(local) | $S_c(C_1) = terminated,$<br>$S_c(C_2) = trying proceeding early confirmed,$<br>$C_2 = call(forwarded-to, remote),$<br>$S_d(local) = avail,$<br>$S_u(local) = avail,$<br>$S_d(remote) = busy,$<br>$S_u(remote) = busy$ |
| originating call screening | $E_d(outgoing)$                         | $S_c(C_1) = early,$<br>$C_1 = call(local, remote)$                                         | disconnect(local)                                                                          | $S_c(C_1) = terminated,$<br>$S_d(local) = avail,$<br>$S_u(local) = avail,$<br>$S_d(remote) = avail,$<br>$S_u(remote) = avail$                                                                                         |

**Table 1:** Feature definitions

[0062] Below is a list of definitions of the abbreviations used in Table 1:

|                      |   |                                                          |
|----------------------|---|----------------------------------------------------------|
| <b>S</b>             | — | stands for state,                                        |
| <b>C<sub>i</sub></b> | — | stands for call,                                         |
| <b>E</b>             | — | stands for triggering event,                             |
| <b>local</b>         | — | stands for the party that initiates a communication,     |
| <b>remote</b>        | — | stands for a party that is contacted by the local party, |
| <b>u</b>             | — | stands for user,                                         |
| <b>d</b>             | — | stands for device,                                       |
| <b>c</b>             | — | stands for call,                                         |
| <b>v</b>             | — | stands for voicemail,                                    |
| <b>e</b>             | — | stands for email.                                        |

[0063] Thus, the notation for the post condition " $S_c(C_1) = \neg terminated$ ," that is required for the information retrieval feature, means that a user must not be prevented from calling the party whose information is retrieved. That is, the called user must not be on a do-not-call list.

[0064] The notation for the post condition " $S_c(C_1) = terminated$ ," that is required for the originating call screening feature, means that the state of a call to a party must be terminated. In other words, a user is precluded from starting

communication sessions with parties on a "do-not-call list."

[0065] The notation of the triggering event  $E(\text{incoming call})$ , which is a triggering event for both the call forwarding and personal assistant features, means that the features are triggered by the receipt of an incoming call.

[0066] Record 330 is a record of one or more interaction definitions. The interaction definitions specify conditions that can be used by coordinator module 310 to detect feature interactions. For the purposes of this disclosure, the term "**interaction definition**" is defined as an association between (i) at least one of a precondition, postcondition, and triggering event for a first feature and (ii) at least one of a precondition, postcondition, and triggering event for a second feature.

[0067] In the illustrative embodiment, three types of interactions are defined: concurrency conflicts, disabling conflicts, and result conflicts.

#### **Concurrency Conflicts**

[0068] Concurrency conflicts exist when two features have inconsistent preconditions, and thus cannot be executed at the same time. In the illustrative embodiment, concurrency conflicts are defined as follows:

[0069]  $(\exists c_1 \in \text{pre}(F_1)) \wedge (\exists c_2 \in (\text{pre}(F_2)) \wedge \text{trigger}(F_1) = \text{trigger}(F_2) \rightarrow (c_1 \rightarrow \neg c_2)$

[0070] According to this definition, a concurrency conflict exists when: (1) the precondition of one feature is the negation of a precondition of a second feature; and (2) both the first feature and the second feature have the same triggering event.

#### **Disabling Conflicts**

[0071] Disabling conflicts occur when the execution of a first feature leaves system 100 in a state in which a second feature cannot be executed. An example of a disabling conflict is an interaction between the call screening feature and the information retrieval feature. As discussed with respect to Figure 2, when a user is put on the "do-not-call-list" of the call screening feature, that user cannot be called by using call button 252 of the information retrieval feature. In other words, once the call screening feature is executed (or enabled) the system is put in a state in which the user cannot be called any more.

[0072] In the illustrative embodiment, disabling conflicts are defined as follows:

[0073]  $(\exists c_1 \in \text{post}(F_1)) \wedge (\exists c_2 \in \text{pre}(F_2)) \wedge \text{trigger}(F_1) \neq \text{trigger}(F_2) \rightarrow (c_1 \rightarrow \neg c_2)$

[0074] According to this definition, a disabling conflict exists when: (1) a post

condition for a first feature is the negation of a precondition for a second feature, and (2) the first feature and the second feature have non-equivalent triggering events.

### **Results Conflicts**

[0075] Results conflicts occur when two actions would leave the system in an inconsistent (impossible) state and thus cannot be executed simultaneously. An example of a results conflict is the conflict between the call forwarding and personal assistant features. As discussed, the call forwarding feature automatically forwards a call when the call recipient is busy. The personal assistant, on the other hand, answers incoming calls when the call recipient is busy. Since, a call cannot be both automatically forwarded and answered at the same time, the execution of the two features would leave the system (*i.e.*, system 100) in an impossible state.

[0076] In the illustrative embodiment, results conflicts are defined as follows:

[0077]  $(\exists c1 \in post(F1)) \wedge (\exists c2 \in post(F2)) \wedge trigger(F1) = trigger(F2) \rightarrow (c1 \rightarrow \neg c2)$

[0078] According to this definition, a results conflict exists when: (1) a post condition for a first feature is the negation of a post condition for a second feature; and (2) the first feature and the second feature have the same triggering event.

### **Predicate Relationships**

[0079] The definitions of the feature interactions, in the illustrative embodiment, depend on predicate relationships. For instance, in order for a concurrency conflict to exist, a precondition for a first feature must be the negation of a precondition of another feature. In accordance with the illustrative embodiment, a precondition is said to be the negation of another precondition when the two preconditions are identified as incompatible by an administrator. In the illustrative embodiment, the identification is performed by inputting a table into feature interaction detector (FID) 300 which identifies the two conditions as incompatible.

[0080] Similarly, a postcondition is said to be the negation of another postcondition, when the two postconditions are identified as incompatible by an administrator. In the illustrative embodiment, a table entry is input into feature interaction detector (FID) 300 which identifies the two postconditions as incompatible.

[0081] Additionally, the definitions of feature conflicts depend on triggering event associations. For instance, a first feature and a second feature can have a results conflict only when a triggering condition for the first feature is equivalent to a triggering

condition for the second feature. In accordance with the illustrative embodiment, the two triggering events are equivalent only when they are the same. In some embodiments however, triggering events are considered equivalent when they are defined as such. That is, a table entry is input into feature interaction detector (FID) 300 which identifies two triggering events as equivalent.

[0082] Record 340 is an information record that identifies one or more features that have been invoked. It will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention in which record 340 identifies one or more features that have been executed.

[0083] In the illustrative embodiment, record 340 identifies features that have been invoked by both terminal 110 and terminal 150. In alternative embodiments, however, record 340 identifies features that have been invoked by terminal 110 only.

[0084] In accordance with the illustrative embodiment of the present invention, feature interaction detector (FID) 300 implements the predicate-based model described in provisional application 61/221,211. It will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention can be devised, in which other models are used to detect feature interaction. Examples of such models include the model disclosed in U.S. Application 12/392, 429, expert systems, *etc.*

[0085] **Figure 4** depicts a schematic diagram of the salient components of the illustrative embodiment of the present invention. In particular, **Figure 4** depicts different possible locations in the topology of a communication network where feature interaction detector (FID) 300 can be executed.

[0086] Network 460 is an enterprise network. Terminal 110 is part of network 460. Network 460 possesses an address space that is distinct from the address space of network 170. Network 460 is separated from network 170 by session Border Controller (SBC) 420.

[0087] In a first illustrative embodiment, feature interaction detector (FID) 300 is executed on terminal 110. In some embodiment, feature interaction detector (FID) 300, or parts of it (*e.g.*, coordinator module 310, *etc.*) is implemented as a browser plug-in. In other embodiments, feature interaction detector (FID) is implemented as a free-standing application.

[0088] An advantage of the first illustrative embodiment is that it is convenient for feature interaction detector (FID) 300 to determine what features are invoked by

terminal 110. A disadvantage, however, is that feature interaction detector (FID) 300 might not be able to obtain information about the execution of features on terminal 150. The difficulty comes from the possibility that session border controller (SBC) 420, as well as firewalls that separate network 460 from network 170, may block information that is pertinent to the operation of feature interaction detector (FID) 300 (*e.g.*, indications that features are invoked on terminal 150, indication of the existence of conflicts, *etc.*).

**[0089]** In a second illustrative embodiment of the present invention, feature interaction detector (FID) 300 is executing on session border controller (SBC) 420. Session border controller (SBC) 420 is a device used in Voice over Internet Protocol (VoIP) applications to exert control over signaling used in the establishing of communication sessions. A primary function of a session border controller (SBC) 420 is to hide domain routing and endpoint identities from external endpoints and signaling elements. In the illustrative embodiment, session border controller (SBC) 420 connects network 460 to network 170. All communication protocol signaling transmitted by terminal 110 to devices in network 170, passes through session border controller (SBC) 420.

**[0090]** Executing feature interaction detector (FID) 300 on session border controller (SBC) 420 enables feature interaction detector (FID) 300 to obtain information inside *P-Con Type* headers of messages that are transmitted by terminal 150 before this information is deleted by the other components of session border controller (SBC) 420. As is discussed, in application 61/145,700, the information in *P-Con Type* can be used in identifying call-related features that are invoked by terminal 110. Furthermore, executing feature interaction detector (FID) 300 on session border controller (SBC) 420 enables feature interaction detector (FID) 300 to receive information in *P-Con Type* headers of messages transmitted by terminal 110 before this information is deleted.

**[0091]** In a third illustrative embodiment of the present invention, feature interaction detector (FID) 300 executes on server 130. In some embodiments, feature interaction detector (FID) 300 operates as a back-to-back user agent (B2BUA) which receives signaling messages from terminal 110 and/or terminal 150. In these embodiments, feature interaction detector (FID) 300 receives one or more SIP messages that identify one or more call-related features that are or have been activated. The messages are transmitted as part of the communications session of between terminal 110 and terminal 150. That is, the messages carry an identifier that has been assigned by the SIP network (*i.e.*, network of server 130) to the communication session between terminal 110 and terminal 150.

[0092] Additionally, feature interaction detector (FID) 300 receives indications that web-enabled features are or have been invoked by terminals 110 and/or 150. The indications for the web-enabled features are received, however, in messages that are not part of the communication session between terminals 110 and 150. Examples of messages that can be used to deliver indications for the web-enabled features include HTTP messages, SOAP messages, *etc.*

[0093] In a fourth illustrative embodiment, feature interaction detector (FID) 300 is executed on terminal 150.

[0094] In a fifth illustrative embodiment of the present invention feature interaction detector (FID) 300 is executed on server 140.

[0095] In a sixth illustrative embodiment, feature interaction detector (FID) 300 is executing on a communications node (*e.g.*, switch, PBX, *etc.*). In this embodiment, feature interaction detector (FID) 300 detects at least one of the invocation, execution, and activation of call-related features by monitoring *treatments* which are transmitted on the line connecting terminal 120 and 150. As is well known, treatments are announcement tones received that are triggered by the network to handle certain conditions during a call (*e.g.*, when a call is screened, when a call is blocked, *etc.*). Potentially, there might be multiple treatments involved in a particular call.

[0096] **Figure 5** depicts a flowchart of the salient tasks associated with the operation of the illustrative embodiment of the present invention.

[0097] At task 510, coordinator module 310 identifies a widget that is part of communication application 200. In embodiments where coordinator module 310 is implemented as a browser plug-in, coordinator module 310 interacts with the web-browser running communication application 200 to determine what widgets are constituted part of the application. In alternative embodiments, however, coordinator module 310, exchanges communications with communication application 200 in order to make this determination.

[0098] At task 520, coordinator module 310 registers with the identified widget. When coordinator module 310 has registered with the widget, the widget starts transmitting event notifications to coordinator module 310. The event notifications serve to notify coordinator module 310 that the state of a feature has changed. Examples of event notifications include indications that a feature provided by the widget is invoked, indications that a feature provided by the widget has been executed, *etc.*

**[0099]** In some of the embodiments where feature interaction detector (FID) 300 is executing on server 130, at least one of tasks 510 and 520 are executed in response to the receipt of a registration attempt by terminal 110 to register on the Session Initiation Protocol (SIP) network of server 130. In some of the embodiments where feature interaction detector (FID) 310 is executing on terminal 110, at least one of tasks 510 and 520 is executed in response to the starting of communication application 200.

**[0100]** At task 530, feature definition(s) and interaction definition(s) are stored in records 320 and 330. In the illustrative embodiments of the present invention, records 320 and 330 are populated manually by an administrator. In some embodiments, however, the feature definition(s) and interaction definition(s) are obtained from a remote server upon request by feature interaction detector (FID) 300. For instance, in some embodiments, coordinator module 310 submits a query to a remote server requesting the definition of one or more features that are identified at task 510. In response, the remote server transmits the definition to coordinator module 310 which proceeds to store it in record 320.

**[0101]** At task 540, coordinator module receives an indication that an event concerning a first feature has occurred. In the illustrative embodiment, the event is a change in the state of the first feature. Examples of events include the invocation of a feature, execution of a feature, enabling of a feature, and others.

**[0102]** In accordance with the illustrative embodiment, at task 540 coordinator module 310 receives an indication that the first feature has been executed. It will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention in which coordinator module 310 receives another indication, such as, for example, and without limitation, an indication that the first feature has been invoked, *etc.*

**[0103]** Furthermore, in the illustrative embodiment, the first feature is executed on terminal 110. In alternative embodiments, however, the feature is executed on terminal 150. Furthermore, in the illustrative embodiment, the feature is call-related. In other embodiments, however, the feature is web-enabled.

**[0104]** At task 550, coordinator module 310 stores the indication received at task 540 in record 340.

**[0105]** At task 560, coordinator module 310 receives an indication that an event concerning a second feature has occurred. In the illustrative embodiment, the event is a change in the state of the second feature. Examples of events include the invocation of a feature, execution of a feature, execution of a feature, and others.

[0106] In the illustrative embodiment, coordinator module 310 receives an indication that the second feature has been invoked. It will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention in which coordinator module 310 receives another indication, such as, for example, and without limitation, an indication that the second feature has been executed, *etc.*

[0107] The indication concerning the second feature is provided by the widget with which coordinator module 310 is registered at task 520. As discussed, the widget is configured to transmit an event notification to event coordinator 310 every time the feature which it provides is executed. Furthermore, in the illustrative embodiment, the feature is web-enabled. In other embodiments, however, the feature is call-related.

[0108] In embodiments where coordinator module 310 receives an indication that the second feature has been executed, feature interaction detector (FID) 300 does little to prevent the interaction, or conflict, from taking place. In these embodiments, however, feature interaction detector (FID) 300 transmits a message to message widget 270. The message warns the user of any unexpected behavior of communication application 200 that might be occurring as a result of the conflict.

[0109] At task 570, coordinator module 310 determines whether the second feature and the first feature interact. The manner in which coordinator module determines whether the features interact is further described in the discussion with respect to Figure 6. If the two features do not interact, coordinator module 310 proceeds to execute task 580. If the two features interact, coordinator module 310 proceeds to execute task 590.

[0110] Furthermore, although in the illustrative embodiment, the present invention uses the predicate-based rules described in the discussion with respect to Figure 3, it will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention in which other types of rules are used. In particular, those skilled in the art will recognize that in alternative embodiments of the present invention, feature interaction detector (FID) 300 operates in accordance with the rules, definitions and principles described in application 61/145,700. Furthermore, it will be clear to those skilled in the art, how to make and use alternative embodiments of the present invention, which involve other well-known in the art methods for performing feature interaction detection such as using a rule-matching engine of an expert system, a logic program, a constraint-satisfaction system, a naïve brute force search, *etc.*

**[0111]** At task 580, coordinator module 310 activates the second feature. In accordance with the illustrative embodiment, coordinator module 310 transmits a message to communication application 200 indicating that the widget can proceed with the execution of the invoked feature. It will be clear to those skilled in the art, however, that alternative embodiments of the present invention can be devised in which coordinator module 310 takes no action when the two features interact.

**[0112]** In the illustrative embodiment, the widget proceeds to execute the feature after receiving an indication that the first feature and the second feature do not interact. For instance, if the second feature is the information retrieval feature described in the discussion of Figure 2, information retrieval tool 250 would contact server 140 with a search query only after receiving permission from coordinator module 310. In other words, in the illustrative embodiment of the present invention, the widgets, or components, of communication application 200 are configured to proceed with the execution of a feature only after receiving permission from feature interaction detector (FID) 300. It will be clear to those skilled in the art, after reading this disclosure, that alternative embodiments of the present invention can be devised in which the widgets are configured to execute unless they receive a message denying permission to execute the feature.

**[0113]** Furthermore, it will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention in which feature interaction detector (FID) 300 does not directly interfere with execution of the second application. In some of these embodiments, the function of feature interaction detector (FID) 300 is limited to transmitting notifications of feature interaction notifications to message widget 270.

**[0114]** At task 590, coordinator module 310 resolves the interaction between the two features. In accordance with the illustrative embodiment of the present invention, coordinator module transmits an indication to communication application 200 instructing it to not execute the second feature. It should be noted that the present invention is not limited to any particular way of resolving feature interactions. Those skilled in the art will readily recognize that a number of ways exist for resolving feature interactions (or conflicts) once they are executed. For example, and without limitation, such resolutions include do not execute the second feature, transmit a message warning the user of the conflict, etc.

**[0115]** Tasks 510 through 590 are described for illustrative purposes. It will be clear to those skilled in the art, after reading this disclosure, how to perform the tasks

associated with Figure 5 in a different order than represented or to perform one or more of the tasks concurrently. Furthermore, it will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention that omit one or more of the tasks.

**[0116] Figure 6** depicts a flowchart of the salient subtasks associated with the execution of task 570.

**[0117]** At task 610, coordinator module 310 determines whether a precondition for the first feature is the negation of a precondition for the second feature. In particular, coordinator module 310 identifies pairs of preconditions by consulting record 320, wherein one member of each pair is a precondition for the first feature while the other member is a precondition for the second feature. After a pair is identified, coordinator module 310 determines whether the preconditions are each other's negation by consulting a table (or another type of record) which identifies precondition negations.

**[0118]** At task 620, coordinator module 310 determines whether a postcondition for the first feature is the negation of a postcondition for the second feature. In particular, coordinator module 310 identifies pairs of postconditions by consulting record 320, wherein one member of each pair is a postcondition for the first feature while the other member is a postcondition for the second feature. After a pair is identified, coordinator module 310 determines whether the postconditions are each other's negation by consulting a table (or another type of record) which identifies postcondition negations.

**[0119]** At task 630, coordinator module 310 determines whether a triggering event for the first feature is equivalent to a triggering event for the second feature. In particular, coordinator module 310 identifies pairs of triggering events by consulting record 320, wherein one triggering event member of each pair is a triggering event for the first feature while the other pair member is a triggering event for the second feature. After each pair is identified, coordinator module 310 consults a table which identifies pairs of triggering events as equivalent.

**[0120]** At task 640, coordinator module 310 compares the information gathered at tasks 610 through 630 against the interaction definitions stored in record 330 and determines whether any of the interaction definitions is met. If yes, coordinator module 310 determines that the first feature and the second feature interact.

**[0121]** Tasks 610 through 640 are described for illustrative purposes. It will be clear to those skilled in the art, after reading this disclosure, how to perform the tasks associated with Figure 6 in a different order than represented or to perform one or more

of the tasks concurrently. Furthermore, it will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention that omit one or more of the tasks.

**[0122] Figure 7** depicts a flowchart of the salient tasks associated with the operation of another illustrative embodiment of the present invention. In accordance with this illustrative embodiment, feature interaction detector (FID) 300 is executing on a SIP server. The SIP server (*e.g.*, SIP proxy, server running a B2BUA, *etc.*).

**[0123]** At task 710, feature interaction detector (FID) 300 registers with a widget of communication application 200. In the illustrative embodiment, the registration is performed by feature interaction detector (FID) 300 contacting communication application 200 and submitting information which can be used by the widgets in transmitting event notifications to feature interaction detector (FID) 300 (*e.g.*, IP address, *etc.*). Once feature interaction detector (FID) 300 has been registered with the widget, the widget begins transmitting event notifications to feature interaction detector (FID) 300. Exemplary events include feature activation, feature execution, feature invocation, *etc*

**[0124]** In alternative embodiments, however, the widgets contact feature interaction detector (FID) 300 directly once they are loaded by communication application 200. In these embodiments, feature interaction detector (FID) 300 interacts with each widget directly.

**[0125]** At task 720, feature interaction detector (FID) 300 starts monitoring outgoing communication protocol signaling that is transmitted by terminal 110. In accordance with the illustrative embodiment, the monitored signaling is outgoing SIP messages. In particular, feature interaction detector (FID) 300 monitors whether any of the transmitted messages contain an indication that an event has occurred. Exemplary events include feature activation, feature execution, feature invocation, *etc.* The indication can be contained in the *P-Con Type* headers of the SIP messages or elsewhere.

**[0126]** At task 730, feature interaction detector (FID) 300 starts monitoring the state of the widget with which coordinator module has registered at task 720. In the illustrative embodiment, monitoring the state of the widget entails at least one of (i) determining whether a feature provided by the widget is invoked, (ii) determining whether a feature provided by the widget is executed, and (iii) identifying any other change in the status of the feature provided by the widget.

**[0127]** In accordance with the illustrative embodiment, feature interaction

detector (FID) 300 starts listening for event notification by the widget that is registered at task 710. The event notifications indicate changes in the state of the widget. Exemplary event notifications include, an indication that feature provided by a widget is invoked, indication that a feature provided by a widget is executed, *etc.*

**[0128]** In the illustrative embodiment, feature interaction detector (FID) 300 listens passively. In alternative embodiments, however, feature interaction detector (FID) 300 periodically polls the widget to determine whether change(s) in the state of the widget have occurred. The polling of the widget utilizes the widget's independent-addressing functionality.

**[0129]** Executing task 730 enables feature interaction detector (FID) 300 to catch changes in the states of features that do not involve the SIP network over which the signaling monitored at task 720 is transmitted. As discussed widgets are independently addressable. They can utilize the capabilities of the web browser executing communication application 200 and request services from servers that are not part of the communication protocol network (*e.g.*, the SIP network of the telephone network provider to which server 130 belongs, *etc.*). When services are requested independently of the communications network, there is no way for the SIP server to detect that by monitoring the SIP signaling. The execution of task 730, thus, provides the SIP server to stay apprised of changes in the state(s) of feature, even when the request/execution of these features does not involve the communication network to which the SIP server belongs in any way.

**[0130]** At task 740, feature interaction detector (FID) 300 detects that a first feature has been invoked. In accordance with the illustrative embodiment, the detection is completed by monitoring the outgoing communication protocol signaling that is started at task 720.

**[0131]** In accordance with the illustrative embodiment, feature interaction detector (FID) 300 detects feature invocation. It will be clear to those skilled in the art, however, that alternative embodiments of the present invention can be devised in which feature interaction detector (FID) 300 in which feature interaction detector (FID) 300 detects that a feature has been executed.

**[0132]** At task 750, feature interaction detector (FID) 300 detects that a second feature has been invoked. In accordance with the illustrative embodiment, the detection is completed by the listening (and/or polling) started at task 740.

**[0133]** In accordance with the illustrative embodiment, feature interaction detector (FID) 300 detects feature invocation. It will be clear to those skilled in the art,

however, that alternative embodiments of the present invention can be devised in which feature interaction detector (FID) 300 in which feature interaction detector (FID) 300 detects that a feature has not just been just invoked, but rather that the feature has been executed.

[0134] At task 760, feature interaction detector (FID) 300 determines whether the second feature and the first feature interact. If the two features do not interact, feature interaction detector (FID) 300 proceeds to execute task 770. If the two features interact, feature interaction detector (FID) 300 proceeds to execute task 780. Task 760 is identical to task 570.

[0135] At task 770, feature interaction detector (FID) 300 activates the second feature. Task 770 is identical to task 580.

[0136] At task 780, feature interaction detector (FID) 300 resolves the interaction between the two features. Task 780 is identical to task 590.

[0137] It is to be understood that although the illustrative embodiment of Figure Z is discussed in the context of a SIP network and SIP node executing feature interaction detector (FID) 300, the present invention is not limited to this type communication protocol. It will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention in which another type of network and another type of communication protocol are used (e.g., GSM network, H.323, Skype, etc.). More specifically, it will be clear to those skilled in the art, after reading this disclosure, that alternative embodiments of the present invention can be devised in which the node executing feature interaction detector (FID) 300 is another type of network node, such as, for example and without limitation, PBX switch, GSM base station, Skype server, etc.

[0138] Tasks 710 through 780 are described for illustrative purposes. It will be clear to those skilled in the art, after reading this disclosure, how to perform the tasks associated with Figure 7 in a different order than represented or to perform one or more of the tasks concurrently. Furthermore, it will be clear to those skilled in the art, after reading this disclosure, how to make and use alternative embodiments of the present invention that omit one or more of the tasks.

[0139] It is to be understood that the disclosure teaches just one example of the illustrative embodiment and that many variations of the invention can easily be devised by those skilled in the art after reading this disclosure and that the scope of the present invention is to be determined by the following claims.

What is claimed is:

**1.** A method comprising determining whether a first feature and a second feature interact, wherein the first feature involves a first server in a first administrative domain and the second feature involves a second server in a second administrative domain.

**2.** The method of claim 1 wherein the first feature is a web service.

**3.** The method of claim 2 wherein the second feature is also a web service.

**4.** The method of claim 2 wherein the second feature is not a web service.

**5.** The method of claim 1 wherein the first feature and the second feature are converged services.

**6.** A method comprising determining in an Internet browser whether a first feature and a second feature interact, wherein the first feature involves a first server in a first administrative domain and the second feature involves second server in a second administrative domain.

**7.** The method of claim 6 wherein the determining is performed by one or more widgets in a web page at the Internet browser.

**8.** The method of claim 6 wherein the determining is based on the state of one or more widgets in a web page at the Internet browser.

**9.** The method of claim 9 wherein the first feature is a web service.

**10.** The method of claim 9 wherein the second feature is also a web service.

**11.** The method of claim 9 wherein the second feature is not a web service.

**12.** The method of claim 6 wherein the first feature and the second feature are converged services.

**13.** A method comprising:

detecting, at a communication network node, an invocation or execution of a first feature by monitoring communication protocol signaling that is transmitted by a communication application;

detecting, at the communication network node, an invocation of a second feature by monitoring the state of a component of the communication application; and

determining, at the communication network node, whether the first feature and the second feature interact.

**14.** The method of claim 13 wherein the communication network node is a Session Initiation Protocol (SIP) server.

**15.** The method of claim 13 comprising registering the communication network node with the component of the communication application.

**16.** The method of claim 13, wherein the monitoring of the state of the component of the communication application comprises receiving an indication of a state change.

**17.** The method of claim 13 wherein the monitoring of the state of the component of the communication application comprises periodically polling the component.

**18.** The method of claim 13 wherein the communication application comprises a message component, comprising transmitting an indication to the message component when the first feature and the second feature are determined to interact.

**19.** The method of claim 13 wherein the communication protocol signaling is transmitted as part of an ongoing communication session that is taking place between the communication application and another party.

**20.** The method of claim 13 wherein the determination of whether the first feature and the second feature interact is made by the communication network node based on a rule which relates at least one of (i) precondition, postcondition, and triggering event associated with the first feature with at least one of (ii) precondition, postcondition, and triggering event associated with the second feature.

**21.** A method comprising:

registering, at a communication terminal that is executing browser software, a browser plug-in with a widget of a browser-based application, wherein:

- (i) the widget is operable to provide a first feature,
- (ii) the browser plug-in is operable to determine whether the first feature interacts with another feature, and
- (iii) both the communication application and the browser plug-in are executed by the browser software; and

determining, at the communication terminal, whether the first feature interacts with a second feature, wherein the determination is made by the browser plug-in.

**22.** The method of claim 21 comprising executing the first feature only when permission from the browser plug-in is received by the widget.

**23.** A method comprising providing a widget of a browser-based communication application that is operable to transmit an indication to a feature interaction detector (FID) when a feature provided by the widget is invoked.

**24.** The method of claim 23 wherein the feature interaction detector (FID) is a software module operable to determine when the feature provided by the widget interacts with another feature.

**25.** The method of claim 23 wherein the feature interaction detector (FID) and the communication application are executing on the same communication terminal.

**26.** The method of claim 23 wherein the feature interaction detector (FID) and the communication application are executing on different network nodes.

**27.** The method of claim 23 wherein the widget is operable to execute the feature only when permission from the feature interaction detector (FID) is received by the widget.

**28.** The method of claim 23 wherein:

the feature interaction detector (FID) is executing on a first server; and  
the feature is provided by a second server.

**29.** The method of claim 28 wherein the first server and the second server are in different administrative domains.

**30.** The method of claim 29 wherein:

the browser-based communication application is connected to the first server via

a first network interface; and

the browser-based communication application is connected to the second server via a second network interface.

**31.** The method of claim 30 wherein the first interface is a mobile telephony interface and the second communication interface is a Wi-Fi interface.

**32.** The method of claim 30 wherein the indication is transmitted via the first network interface.

**33.** A method comprising:

providing, at time =  $t_1$ , a communication application comprising a plurality of components;

wherein, at time =  $t_1$ , a first component of the plurality is not operable to transmit an indication to a feature interaction detector (FID) when a feature provided by the first component is invoked;

wherein, at time =  $t_1$ , a second component of the plurality is not operable to transmit an indication to the feature interaction detector (FID) when a feature provided by the second component is invoked;

registering, at time =  $t_2$ , the feature interaction detector (FID) with the first component;

wherein when the registration of the feature interaction detector (FID) with the first component is complete, the first component becomes operable to transmit an indication to the feature interaction detector (FID) when a feature provided by the first component is invoked;

wherein, at time =  $t_2$ , the second component is not operable to transmit an indication to the feature interaction detector (FID) when a feature provided by the second component is invoked;

wherein the feature interaction detector (FID) is a software module operable to determine when a feature provided by one of the first component and the second component interacts with another feature; and

wherein  $t_1 < t_2$ .

**34.** The method of claim 33 comprising:

registering, at time =  $t_3$ , the feature interaction detector (FID) with the second component;

wherein when the registration of the feature interaction detector (FID) with the second component is complete, the second component becomes operable to transmit an indication to the feature interaction detector (FID) when a feature provided by the second component is invoked; and

wherein  $t_1 < t_2 < t_3$ .

**35.** The method of claim 33 wherein the components of the plurality are widgets.

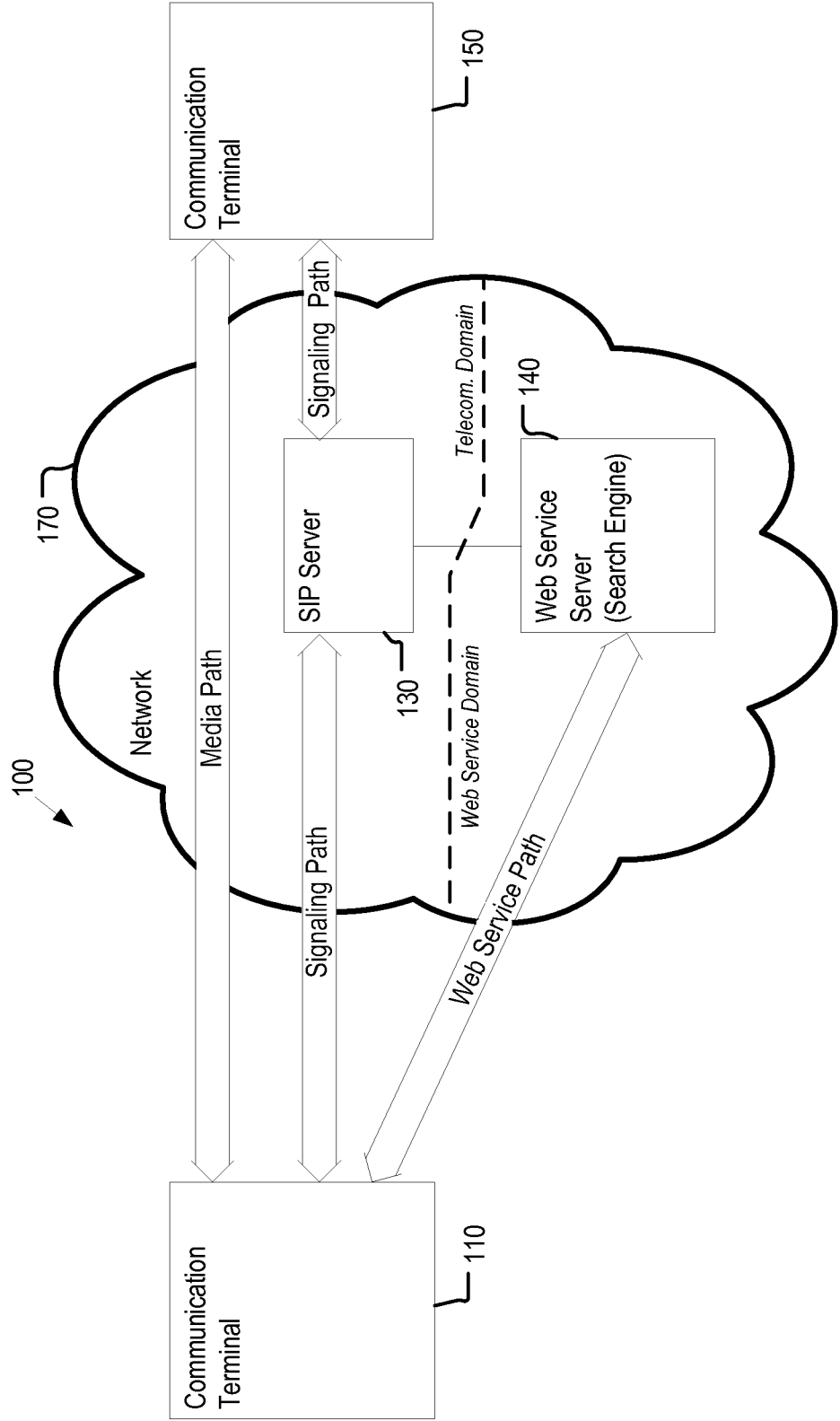


Figure 1

Figure 2

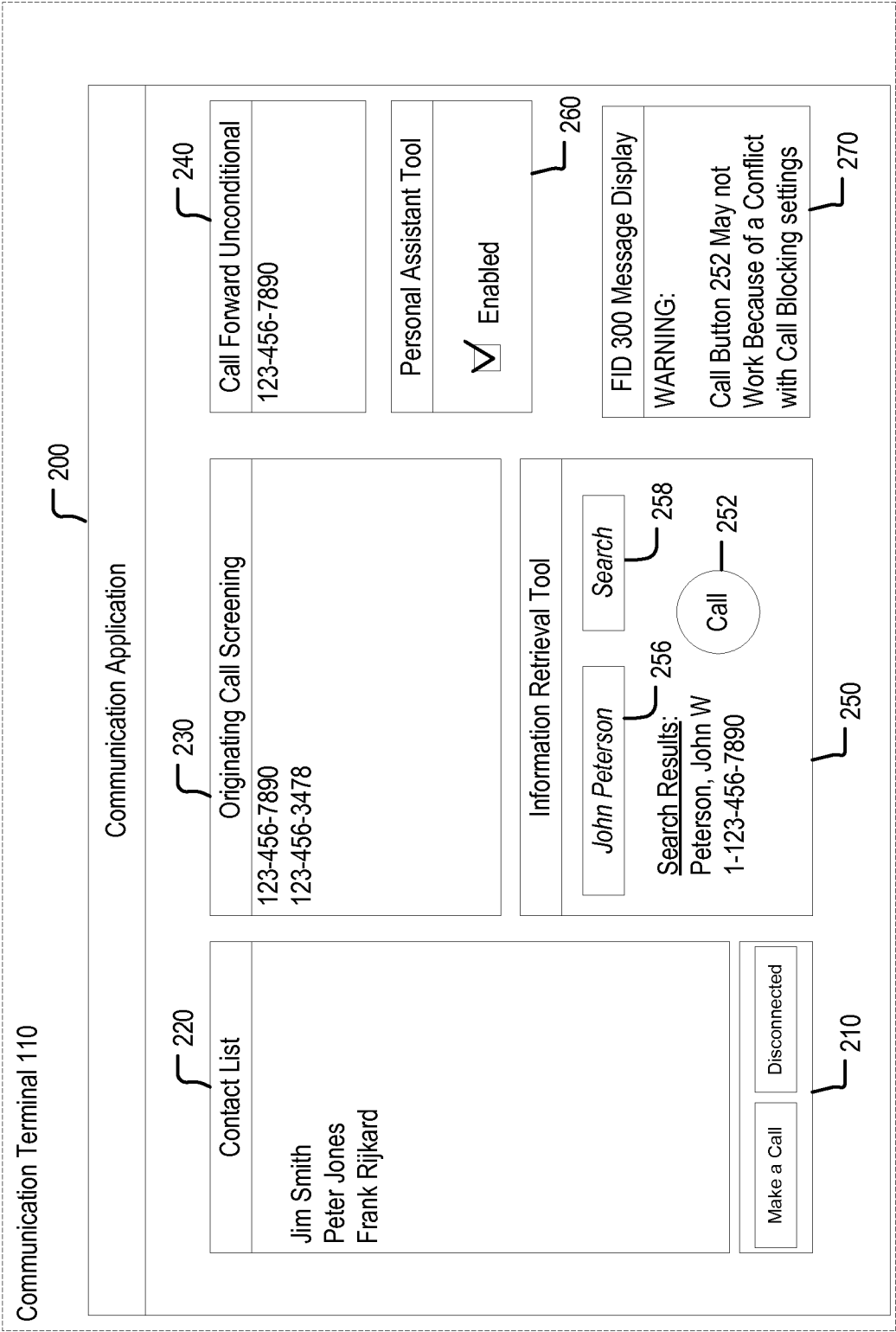
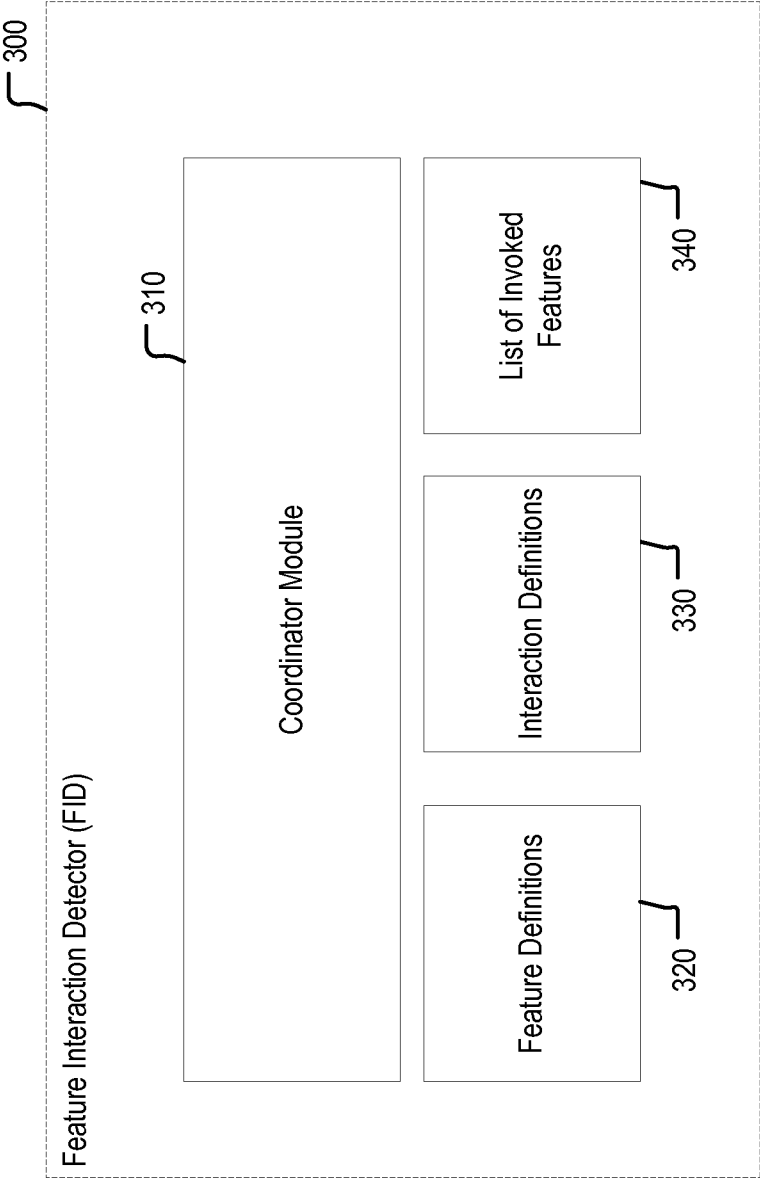


Figure 3



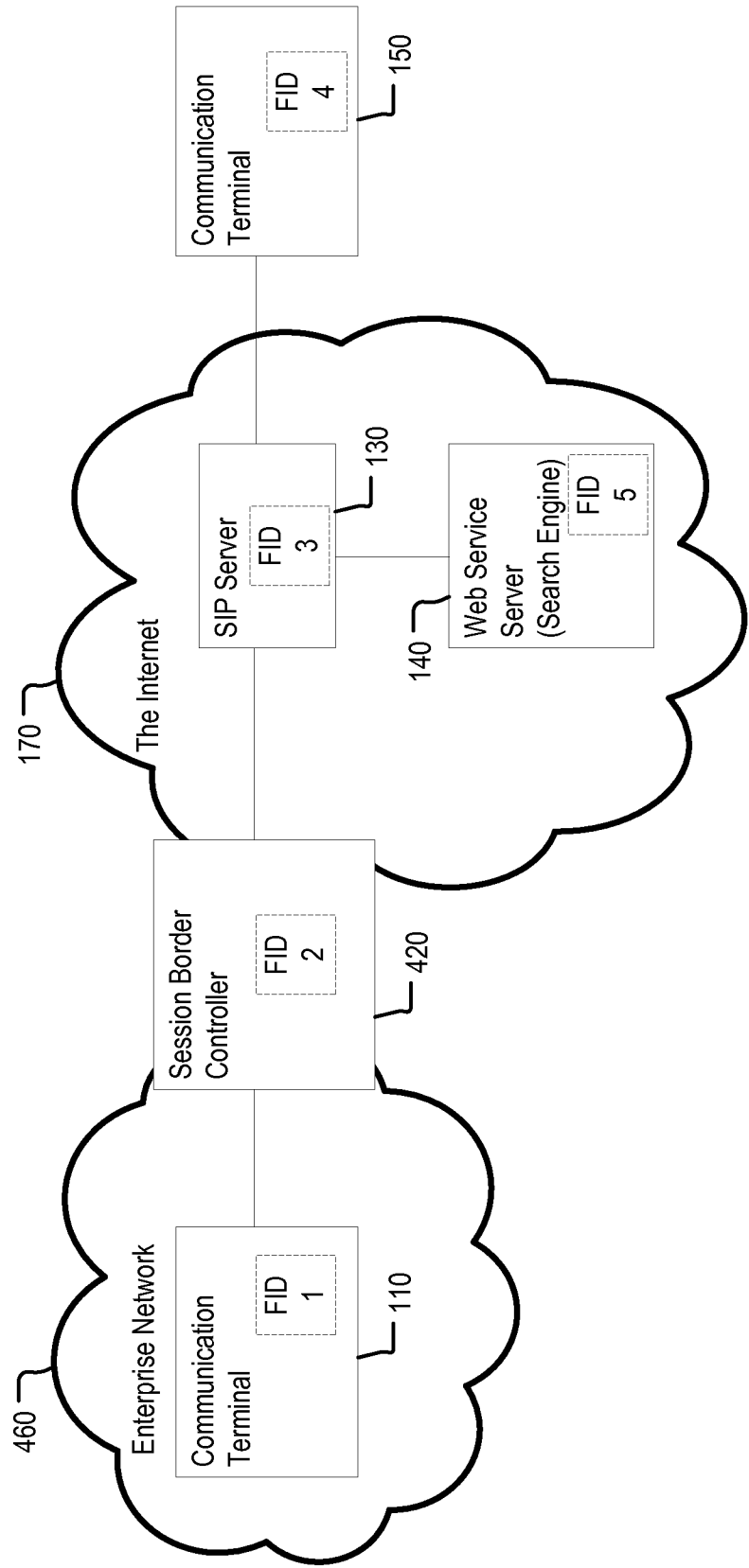


Figure 4

5/7

Figure 5

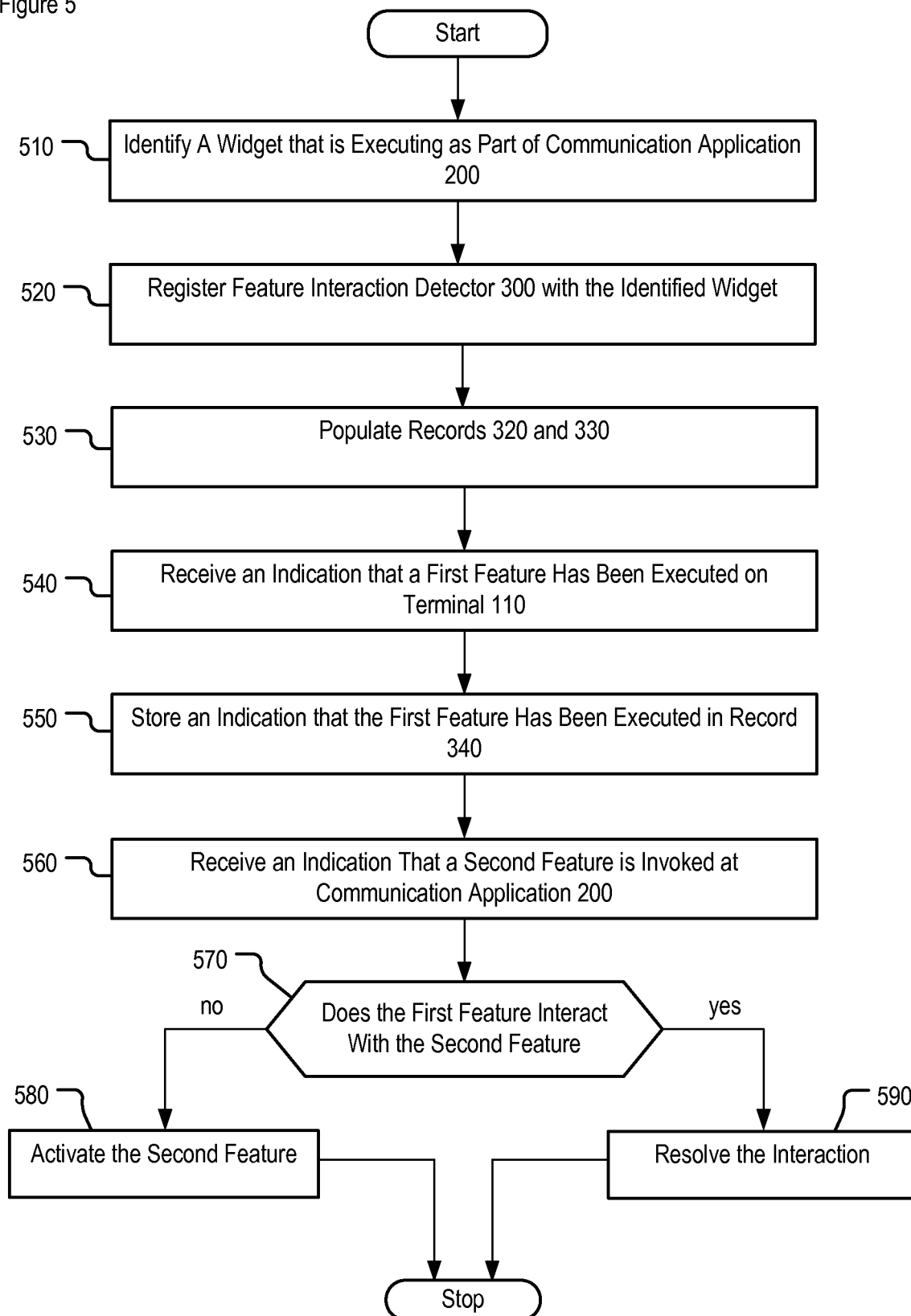
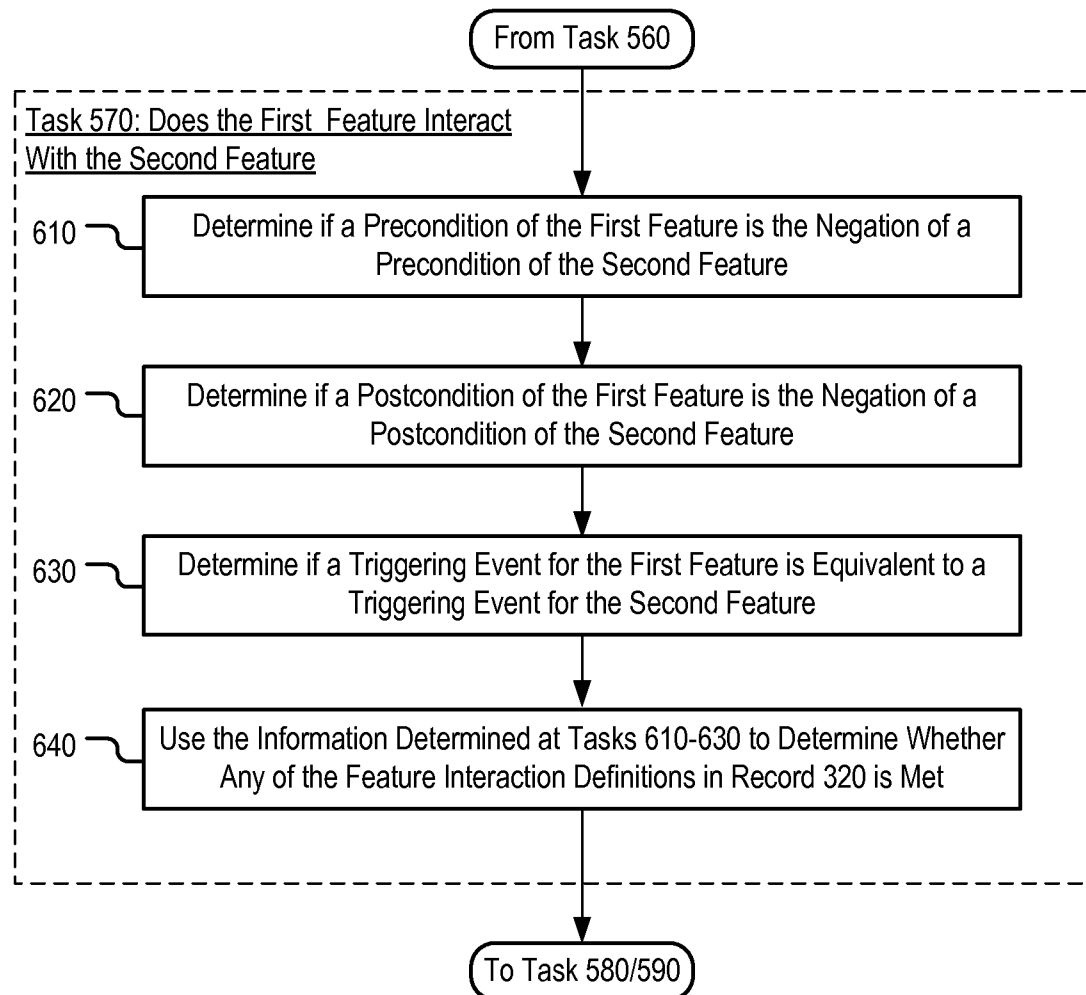


Figure 6



7/7

Figure 7

