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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 April 2013 (04.04.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/049781 A1

(51) International Patent Classification:
G06F 11/30 (2006.01) *G06F 15/16* (2006.01)
G06F 11/32 (2006.01)

(21) International Application Number:
PCT/US2012/058205

(22) International Filing Date:
30 September 2012 (30.09.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/249,243 30 September 2011 (30.09.2011) US

(71) Applicant *(for all designated States except US)*: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **MANNING, Sara**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **ESTEVE BALDUCCI, Juan V.**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **PINTOS, Fabio**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). **CLAUX, David**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (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, BN, 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, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, 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, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

[Continued on next page]

(54) Title: COMMUNICATING UNEXPECTED COLLABORATION SERVER RESPONSES ON RECONNECTION

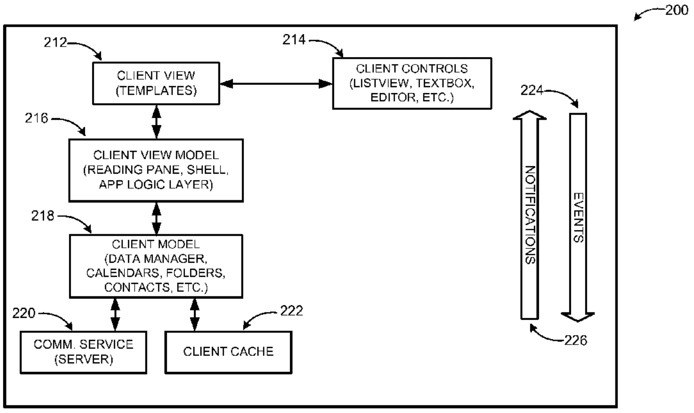

200
212
CLIENT VIEW (TEMPLATES)
214
CLIENT CONTROLS (LISTVIEW, TEXTBOX, EDITOR, ETC.)
216
CLIENT VIEW MODEL (READING PANE, SHELL, APP LOGIC LAYER)
218
CLIENT MODEL (DATA MANAGER, CALENDARS, FOLDERS, CONTACTS, ETC.)
220
COMM. SERVICE (SERVER)
222
CLIENT CACHE
224
NOTIFICATIONS
EVENTS
226

FIG. 2

(57) **Abstract**: A set of workflows are provided for supporting proper user notifications after an action is taken by the user in conjunction with an asynchronous communication service. Timing and/or type of the notifications or an action to be taken by the service is determined based on one or more of a nature of the user action that failed, a time elapsed since the action was taken, multi-action dependencies, device types, and similar characteristics enhancing user experience and reducing confusion.

Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

COMMUNICATING UNEXPECTED COLLABORATION SERVER RESPONSES ON RECONNECTION

BACKGROUND

[0001] Asynchronous communication services such as email, text messaging, and comparable ones are a big part of daily lives facilitating exchange of information for personal, business, and other purposes. Asynchronous communication services typically include one or more servers managing the communications over one or more networks, where users access the services through thick or thin clients. Thick clients are locally installed client applications that perform a majority of the functionality locally synchronizing with a server periodically or on-demand. Thin clients are typically generic applications such as web browsers that enable access to communication services through a user interface controlled by a server, where a majority of the functionality is server-based.

[0002] When network connectivity exists, thick or thin clients provide a seamless user experience. On the other hand, if network connectivity is lost, a client application may still provide some of the functionality to a user, but not be able to synchronize with the server. Thus, some of the communication services may be interrupted. For example, email messages may not be sent or retrieved, calendar items may not be updated, etc. In some implementations, a client application may act normal and perform the functions even when network connectivity is absent and replay user actions once connectivity is restored. The network interruption may not always be for a short period, however. If the interruption last a relatively long period (e.g., hours or days), replaying the user action upon restoration of connectivity, or an error in the replay of a user action upon restoration of connectivity, may lead to user confusion. For example, a scheduled appointment may be provided to the user after its scheduled time is past. Some conventional systems alert the user every time there is an interruption, which may degrade user experience. Other systems simply do not report any of the failures, which could result in worse situations such as the user losing confidence in the communication system.

SUMMARY

[0003] This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This summary is not intended to exclusively identify key features or essential features of the claimed subject matter, nor is it intended as an aid in determining the scope of the claimed subject matter.

[0004] Embodiments are directed to a set of workflows for supporting proper user notifications after an action is taken by the user in conjunction with an asynchronous communication service. Timing and/or type of the notifications or an action to be taken by the service may be determined based on one or more of a nature of the user action that failed, a time elapsed since the action was taken, multi-action dependencies, device types, and similar characteristics enhancing user experience and reducing confusion.

[0005] These and other features and advantages will be apparent from a reading of the following detailed description and a review of the associated drawings. It is to be understood that both the foregoing general description and the following detailed description are explanatory and do not restrict aspects as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] FIG. 1 illustrates an example system for facilitating asynchronous communication services, where embodiments may be implemented;

[0007] FIG. 2 illustrates an example web-based email service architecture according to embodiments;

[0008] FIG. 3 illustrates example action flows in a communication service according to embodiments;

[0009] FIG. 4 illustrates the architecture of an example system with offline support according to embodiments;

[0010] FIG. 5 illustrates an example user interface for an email service with support for offline action notifications according to embodiments;

[0011] FIG. 6 is a networked environment, where a system according to embodiments may be implemented;

[0012] FIG. 7 is a block diagram of an example computing operating environment, where embodiments may be implemented; and

[0013] FIG. 8 illustrates a logic flow diagram for a process of communicating unexpected collaboration server responses upon reconnection according to embodiments.

DETAILED DESCRIPTION

[0014] As briefly described above, proper user notifications may be provided after an action is taken by the user in conjunction with an asynchronous communication service, and in particular when that action fails to replay correctly to the server. Notifications and tasks to be performed in response to user actions when network connectivity is interrupted may be determined based on predefined criteria and at least a portion of created/modified data cached locally. Timing and/or type of the notifications or an action to be taken by the service may be determined based on one or more of a nature of the user action that failed, a time elapsed since the action was taken, multi-action dependencies, device types, and similar characteristics enhancing user experience and reducing confusion.

[0015] In the following detailed description, references are made to the accompanying drawings that form a part hereof, and in which are shown by way of illustrations specific embodiments or examples. These aspects may be combined, other aspects may be utilized, and structural changes may be made without departing from the spirit or scope of the present disclosure. The following detailed description is therefore not to be taken in a limiting sense, and the scope of the present invention is defined by the appended claims and their equivalents.

[0016] While the embodiments will be described in the general context of program modules that execute in conjunction with an application program that runs on an operating system on a computing device, those skilled in the art will recognize that aspects may also be implemented in combination with other program modules.

[0017] Generally, program modules include routines, programs, components, data structures, and other types of structures that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that embodiments may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and comparable computing devices. Embodiments may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

[0018] Embodiments may be implemented as a computer-implemented process (method), a computing system, or as an article of manufacture, such as a computer program product or computer readable media. The computer program product may be a computer storage medium readable by a computer system and encoding a computer program that comprises instructions for causing a computer or computing system to perform example process(es). The computer-readable storage medium can for example be implemented via one or more of a volatile computer memory, a non-volatile memory, a hard drive, a flash drive, a floppy disk, or a compact disk, and comparable media.

[0019] Throughout this specification, the term “platform” may be a combination of software and hardware components for providing asynchronous communication services such as email or text messaging. Examples of platforms include, but are not limited to, a hosted service executed over a plurality of servers, an application executed on a single computing device, and comparable systems. The term “server” generally refers to a computing device executing one or more software programs typically in a networked environment. However, a server may also be implemented as a virtual server (software programs) executed on one or more computing devices viewed as a server on the network. More detail on these technologies and example operations is provided below. As used herein, the term “action” refers to user interaction with a communication application such as selecting/activating a user interface element or providing input through the user interface. The term “task” refers to operations performed by the system in response to “actions” or other events in providing communication services to the users.

[0020] FIG. 1 illustrates, in diagram 100, an example system for facilitating asynchronous communication services, where embodiments may be implemented. A communication service 102 may facilitate services asynchronous communications such as email exchange, text message exchange, and similar ones, along with ancillary features such as scheduling services (calendar), contacts, task lists, and comparable ones. Communication service 102 may interact with client applications 104, 106 over network(s) 110 to facilitate exchange of communications.

[0021] Client applications 104, 106 may be thick of thin clients. Thick clients are locally installed client applications that perform a majority of the functionality locally synchronizing with a server periodically or on-demand. Thin clients are typically generic applications such as web browsers that enable access to communication services through a user interface controlled by a server, where a majority of the functionality is server-based. In either case, functionality of the client applications is dependent on connectivity to the communication service 102.

[0022] A client application may be disconnected from network(s) 110 when providing email, calendar or contacts features of an email exchange service, for example, particularly on a portable device. The actions that a user takes may be replayed once connectivity is restored without user intervention. The disconnect period may, however, be seconds, minutes, hours or days. If an action fails to replay wholly or partially, the challenge is whether and how to indicate the replay failure to the user.

[0023] In a system according to embodiments, failure notifications may be provided and tasks associated with the failed actions performed based on a number of predefined criteria such as a type or nature of the failure, a time of failure, a type of application or device, etc.

The actions may include, but are not limited to, sending / deleting / modifying a message, creating / deleting / modifying a calendar or contact item, or creating / deleting / modifying properties for an item. Thus, in response to detecting a failure to replay an action, a system according to embodiments may apply one or more criteria and perform the tasks including providing a notification (type, timing, format of notification). Some of the created / modified data may also be cached locally based on the criteria.

[0024] FIG. 2 illustrates an example web-based email service architecture according to embodiments in diagram 200.

[0025] According to some embodiments, if a user chooses the ability to work offline in their client application, the application may create a local data store (client cache 222) to store items associated with the application, for example, email, calendar and contacts in an email application to support functionality while the application is disconnected. When the application is connected, the user interface may read and write user actions from and to the local data store (for both performance and to keep it up-to-date), and write actions to both the local data store and to the server (communication service 220) to maintain synchronization. For any actions that may be too resource-intensive to be written locally, a data synchronization component may copy incremental server changes to the local data store directly.

[0026] Other major functional components in a system according to embodiments may include client controls 214 managing elements such as listview, textbox, editor, etc.; client view 212, which is a collection of templates that are a composition of primitive controls; client view model 216, which includes the application logic; and client model 218, which includes data manager, calendars, folders, contacts, etc.

[0027] As mentioned above, client view model 216 may include user interface specific application logic and maintain application state. According to some embodiments, client view model 216 may have no knowledge of the user interface and the states may be exposed through properties and command bindings. Responsibility of responding to a user action with changes in client model 218 may belong in this layer.

[0028] Client view 212 may bind to user experience interface and act like a reflection of client view model 216 in the user interface. Data binding is the main mechanism for communication between client view 212 and lower layers. Client view 212, which is a collection of templates comprising primitive controls may have a 1:1 mapping to client view model 216. The configuration employing data binding between client view 212 and client view model 216 may enable creation of the entire view for the view model using templates. In a system as the one shown in diagram 200, notifications 226 may flow from communication service 220 toward client controls 214, while event flow from client controls 214 toward communication service 220 or client cache 222.

[0029] FIG. 3 illustrates example action flows in a communication service according to embodiments in diagram 300. In a system according to embodiments, actions 334 may be obtained through view models 332 as discussed above.

[0030] When a communication application is connected, the user interface may read and write user actions from and to the local data store (client store proxy 336) and write actions to both the local data store and to the server to maintain synchronization. When the application loses connectivity, actions 334 may be committed locally until connectivity is restored. To have these actions reflected on the server once connectivity is restored, the actions 334 may be stored in a queue (queued proxy 344) in a local web database table or native device database table at the same time that they are committed locally. In some embodiments, an aggregated proxy 340 may be employed between client store proxy 336 and the server's online proxy 342. The flow may be directly from aggregated proxy 340 to online proxy 342 or via queued proxy 344. Whenever the device regains connectivity (as determined by a separate process, such as a connectivity manager), the actions may be played back to the server (to online proxy 342).

[0031] If the server is not able to process the action, it may return a failure response. The failure response may be passed to the response processor 330 in the action manager architecture. The response processor 330 may provide information to enable the user interface code to choose if and how to surface the failure to the user based on a number of factors. Data about each of these factors may be returned with the failure response to the response processor. The factors may include, but are not limited to, a type of failure, whether other actions depend on this action's success, time since action was taken, a device on which the action was taken, a user profile and/or usage, a communication method (e.g., email vs. text message exchange), or a nature of the action. For example, creation / modification / deletion of a calendar item that involves a user's own time may be less critical than when the calendar item involves other users (e.g., a meeting) as well. Short term disconnects may be transparent to a user (e.g., for calendar or contact item related actions), while long term disconnects may have significance in terms of planning, communication, analysis, etc.

[0032] In some example implementations, the type of failure may include creation of high investment custom content, such as a new message or meeting request, creation of low investment custom content, such as adding or changing the phone number on a contact, changing item-wide properties, such as deleting a message or moving to a different folder, and changing single properties within an item, such as setting a flag or marking the item as read.

[0033] The dependency of other actions on a particular action's success may be, for example, a message having a recipient whose address does not exist in the organization. Send may succeed, but address name resolution for that single recipient may fail. In case of single failures on which other actions are dependent a rollback of all related actions may be performed or other actions may be allowed to proceed and the failure of only this action returned. Thus, response processor 330 can invoke code to react accordingly to a success or failure response, and dependent on success or failure responses of related or dependent action, from the server whenever connectivity is restored, at which time the action may be replayed, re-queued, or purged (ignored).

[0034] FIG. 4 illustrates the architecture of an example system with offline support according to embodiments in diagram 400.

[0035] User interface 452 is the top layer of the system enabling interaction with users and including view and view model objects. Actions 458 such as reply to a message, change an item property, delete a property, read a message, etc. received from the user interface layer 452 may be used for user interface data bound objects 456 that feed back to the user interface layer 452. Actions 458 and data bound objects 456 are in model layer 454 (client-side).

[0036] On the server side, a persisted data access layer 460 may include action dispatcher 464 and sync manager 462. Action dispatcher 464 may receive the actions (e.g., using an action request call). Actions may be provided to and tasks in response to the actions (for the user interface) received from online and offline business logic and data stores. Sync manager 462 may facilitate synchronization of actions and failure responses between online and offline states.

[0037] Online business logic 468 may include folders, messages, attachments, calendars, contacts, and similar items associated with the communication service. Online business logic 468 may also include the views for those items, as well as folder synchronization logic, notifications logic, and property schema logic. Online store 470 may store folders, messages, and attachments as well as views for those. Online store 470 may further include interface synchronization, the notifications, and property schema. A physical layer abstraction for 472 for the online business logic may include indices, tables, and columns, and enable interaction with databases such as relational databases or others.

[0038] Offline business logic and store combination 466 may include items feeding into folders, messages, attachments, calendars, contacts, and similar ones, as well as views for the same. Offline business logic and store combination 466 may further include notifications, views, and persist-and-upload actions. Physical layer abstraction for 474 for the offline business logic may also include indices, tables, and columns, and enable interaction with databases such as web relational databases, indexed databases, or others.

[0039] The example systems in FIG. 1 through 4 have been described with specific configurations, applications, and interactions. Embodiments are not limited to systems according to these examples. A system for providing an adaptive monitoring system for monitoring the performance of a cloud service may be implemented in configurations employing fewer or additional components and performing other tasks. Furthermore, specific protocols and/or interfaces may be implemented in a similar manner using the principles described herein.

[0040] FIG. 5 illustrates an example user interface for an email service with support for offline action notifications according to embodiments. User notifications about failures may be provided in a variety of ways through any communication application user interface. the example user interface 500 in FIG. 5 is for illustration purposes only.

[0041] In a system according to embodiments, user experience may be enhanced through a number of features according to some embodiments. For example, the code to render the user experience may decide if and how to surface the errors to the user based on evaluation of the factors discussed above and returned to a response processor by the action manager architecture. The user experience may choose one of the approaches below based on the data provided by the response processor, in order from most to least invasive and attention-grabbing (which may correspond to most to least critical failures, respectively).

[0042] Notifications may be surfaced through sounds or vibration similar to the user experience when receiving a text message or calendar notification (in devices which support surfacing this type of notification). Information may be displayed on the main screen of the device or on a main view of a web-based application user interface (e.g., a web-based email application). The information may also be detailed in a message added to the user’s inbox.

[0043] Example user interface 500 presents a “Drafts” folder of an email application. A user interface element 582 enables a user to return to a listing of all folders. A brief status update 584 indicates to the user the last time the local version was updated. As discussed above, a visual notification 586 may emphasize to the user a nature of the failure. The visual notification 586 may be accompanied by an audio notification 588 such as a specific sound. In other embodiments, more detailed information about the failure such as a communication associated with the failure, a time of the failure, user options to mitigate the failure, etc. may also be displayed depending on user interface configuration, available screen space, device type, etc.

[0044] FIG. 6 is an example networked environment, where embodiments may be implemented. A system for communicating unexpected collaboration server responses upon reconnect may be implemented via software executed over one or more servers 614 such as a hosted service. The platform may communicate with client applications on individual computing devices such as a smart phone 613, a laptop computer 612, or desktop computer 611 (‘client devices’) through network(s) 610.

[0045] Client applications executed on any of the client devices 611-613 may facilitate communications via application(s) executed by servers 616, or on individual server 616. A communication application executed on one of the servers may facilitate providing failure notifications to users and performance of appropriate tasks based on predefined criteria as discussed above. The application may retrieve relevant data from data store(s) 619 directly or through database server 618, and provide requested services to the user(s) through client devices 611-613.

[0046] Network(s) 610 may comprise any topology of servers, clients, Internet service providers, and communication media. A system according to embodiments may have a static or dynamic topology. Network(s) 610 may include secure networks such as an enterprise network, an unsecure network such as a wireless open network, or the Internet. Network(s) 610 may also coordinate communication over other networks such as Public Switched Telephone Network (PSTN) or cellular networks. Furthermore, network(s) 610 may include short range wireless networks such as Bluetooth or similar ones. Network(s) 610 provide communication between the nodes described herein. By way of example, and not limitation, network(s) 610 may include wireless media such as acoustic, RF, infrared and other wireless media.

[0047] Many other configurations of computing devices, applications, data sources, and data distribution systems may be employed to implement a platform for providing communication services with failure notifications due to network connectivity interruption. Furthermore, the networked environments discussed in FIG. 5 are for illustration purposes only. Embodiments are not limited to the example applications, modules, or processes.

[0048] FIG. 7 and the associated discussion are intended to provide a brief, general description of a suitable computing environment in which embodiments may be implemented. With reference to FIG. 7, a block diagram of an example computing operating environment for an application according to embodiments is illustrated, such as computing device 700. In a basic configuration, computing device 700 may be any computing device such as a server executing a communication application providing connectivity related failure notification according to embodiments and include at least one processing unit 702 and system memory 704. Computing device 700 may also include a plurality of processing units that cooperate in executing programs. Depending on the exact configuration and type of computing device, the system memory 704 may be volatile (such as RAM), non-volatile (such as ROM, flash memory, etc.) or some combination of the two. System memory 704 typically includes an operating system 705 suitable for controlling the operation of the platform, such as the WINDOWS ® operating systems from MICROSOFT CORPORATION of Redmond, Washington. The system memory 704 may also include one or more software applications such as communication service 724 and a sync manager module 726.

[0049] When communication application 724 loses connectivity, actions may be committed locally until connectivity is restored. Whenever the client device regains connectivity, the actions may be played back to a server. If the server is not able to process the action, it may return a failure response. Sync manager module 726 may determine how to surface the failure to the user based on a number of factors. The factors may include, but are not limited to, a type of failure, whether other actions depend on this action's success, time since action was taken, or device on which the action was taken. This basic configuration is illustrated in FIG. 7 by those components within dashed line 708.

[0050] Computing device 700 may have additional features or functionality. For example, the computing device 700 may also include additional data storage devices (removable and/or non-removable) such as, for example, magnetic disks, optical disks, or tape. Such additional storage is illustrated in FIG. 7 by removable storage 709 and non-removable storage 710. Computer readable storage media may include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information, such as computer readable instructions, data structures, program modules, or other data. System memory 704, removable storage 709 and non-removable storage 710 are all examples of computer readable storage media. Computer readable storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computing device 700. Any such computer readable storage media may be part of computing device 700. Computing device 700 may also have input device(s) 712 such as keyboard, mouse, pen, voice input device, touch input device, and comparable input devices. Output device(s) 714 such as a display, speakers, printer, and other types of output devices may also be included. These devices are well known in the art and need not be discussed at length here.

[0051] Computing device 700 may also contain communication connections 716 that allow the device to communicate with other devices 718, such as over a wired or wireless network in a distributed computing environment, a satellite link, a cellular link, a short range network, and comparable mechanisms. Other devices 718 may include computer device(s) that execute communication applications, web servers, and comparable devices. Communication connection(s) 716 is one example of communication media. Communication media can include therein computer readable instructions, data structures, program modules, or other data. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media.

[0052] Example embodiments also include methods. These methods can be implemented in any number of ways, including the structures described in this document. One such way is by machine operations, of devices of the type described in this document.

[0053] Another optional way is for one or more of the individual operations of the methods to be performed in conjunction with one or more human operators performing some. These human operators need not be collocated with each other, but each can be only with a machine that performs a portion of the program.

[0054] FIG. 8 illustrates a logic flow diagram for a process of communicating unexpected collaboration server responses upon reconnection according to embodiments.

Process 800 may be implemented in a communication system providing failure responses to users when network connectivity is lost.

[0055] Process 800 begins with operation 810, where a user action such as creation / deletion / modification of an item (message, appointment, contact, property) may be detected through the interaction of a user with the user interface. The user interface may be complemented by view and view model components interacting through data binding in detecting actions and providing responses to the user.

[0056] At optional operation 820, the actions may be provided to a server and stored at a local cache as backup in case of network connectivity loss. When the application loses connectivity, actions may be committed locally until connectivity is restored. To have these actions reflected on the server once connectivity is restored, the actions may be stored in a queue in a local web database table or native device database table at the same time that they are committed locally

[0057] At operation 830, a response processor may determine when and how to surface a failure response to the user (through the user interface) based on one or more factors. If the server is not able to process the action, it may return a failure response. The failure response may be passed to the response processor in the action manager architecture. The response processor may provide information to enable the user interface code to choose if and how to surface the failure to the user based on a number of factors. Data about each of these factors may be returned with the failure response to the response processor. The factors may include, but are not limited to, a type of failure, whether other actions depend on this action's success, time since action was taken, or device on which the action was taken.

[0058] At optional operation 840, the actions may be replayed upon restoration of network connectivity. When the application is connected, the user interface may read and write user actions from and to the local data store (for both performance and to keep it up-to-date), and write actions to both the local data store and to the server to maintain synchronization.

[0059] The operations included in process 800 are for illustration purposes. A process for communicating unexpected collaborative server responses upon reconnect may be implemented by similar processes with fewer or additional steps, as well as in different order of operations using the principles described herein.

[0060] The above specification, examples and data provide a complete description of the manufacture and use of the composition of the embodiments. Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims and embodiments.

CLAIMS

WHAT IS CLAIMED IS:

1. A method at least partially executed on a computing device for providing failure response due to connectivity loss in a communication system, the method comprising:

   receiving a user action from a user interface of a communication application;

   determining a failure to provide the received action to a server due to connectivity loss; and

   determining a timing and a type of surfacing of a failure notification based on at least one from a set of: a type of the failure, whether other actions depend on the received action's success, a time since the action was received, a device on which the action was received, a user profile, a usage, a type of the action, and a type of facilitated communication.

2. The method of claim 1, further comprising:

   storing the received action in a local cache.

3. The method of claim 2, further comprising:

   replaying the stored action to the server upon reestablishing connectivity.

4. The method of claim 1, further comprising:

   surfacing the failure notification through at least one from a set of: a sound alert, a vibration alert, a visual notification on a main view of the user interface, an informative email message and an event to be handled by the system.

5. The method of claim 1, further comprising:

   displaying detailed information about the failure through one of the user interface and via a message, the information including at least one from a set of: a communication associated with the failure, a time of the failure, one or more user options to mitigate the failure.

6. The method of claim 1, wherein the user interface is enabled to read and write user actions from and to a local data store, and write actions to the server to maintain synchronization when connectivity is reestablished.

7. A communication system for providing failure response due to connectivity loss, the system comprising:

a client controls module managing a user interface configured to receive a user action and to surface a failure notification;

a connection manager module configured to determine a failure to provide the received action to a server due to connectivity loss;

a client view module that is a collection of templates for the user interface;

a client view model that includes application logic for the user interface, wherein the application logic is configured to determine a timing and a type of surfacing of a failure notification based on a plurality of factors including a type of the failure, whether other actions depend on the received action's success, a time since the action was received, a device on which the action was received, a user profile, a usage, a type of the action, and a type of facilitated communication; and

a local cache for storing the received user action.

8. The system of claim 7, wherein the client view model maintains application states.

9. The system of claim 7, wherein the application logic is further configured to:

store received actions in a queue in one of a local web database table and a native device database table when the actions are committed locally.

10. A computer-readable memory device with instructions stored thereon for providing failure response due to connectivity loss in a communication system, the instructions comprising:

receiving a user action from a user interface of a communication application;

determining a failure to provide the received action to a server due to connectivity loss;

storing the received action in a local cache;

determining a timing and a type of surfacing of a failure notification based on a plurality of factors including a type of the failure, whether other actions depend on the received action's success, a time since the action was received, a device on which the action was received, a user profile, a usage, a type of the action, and a type of facilitated communication; and

replaying the stored action to the server upon reestablishing connectivity.

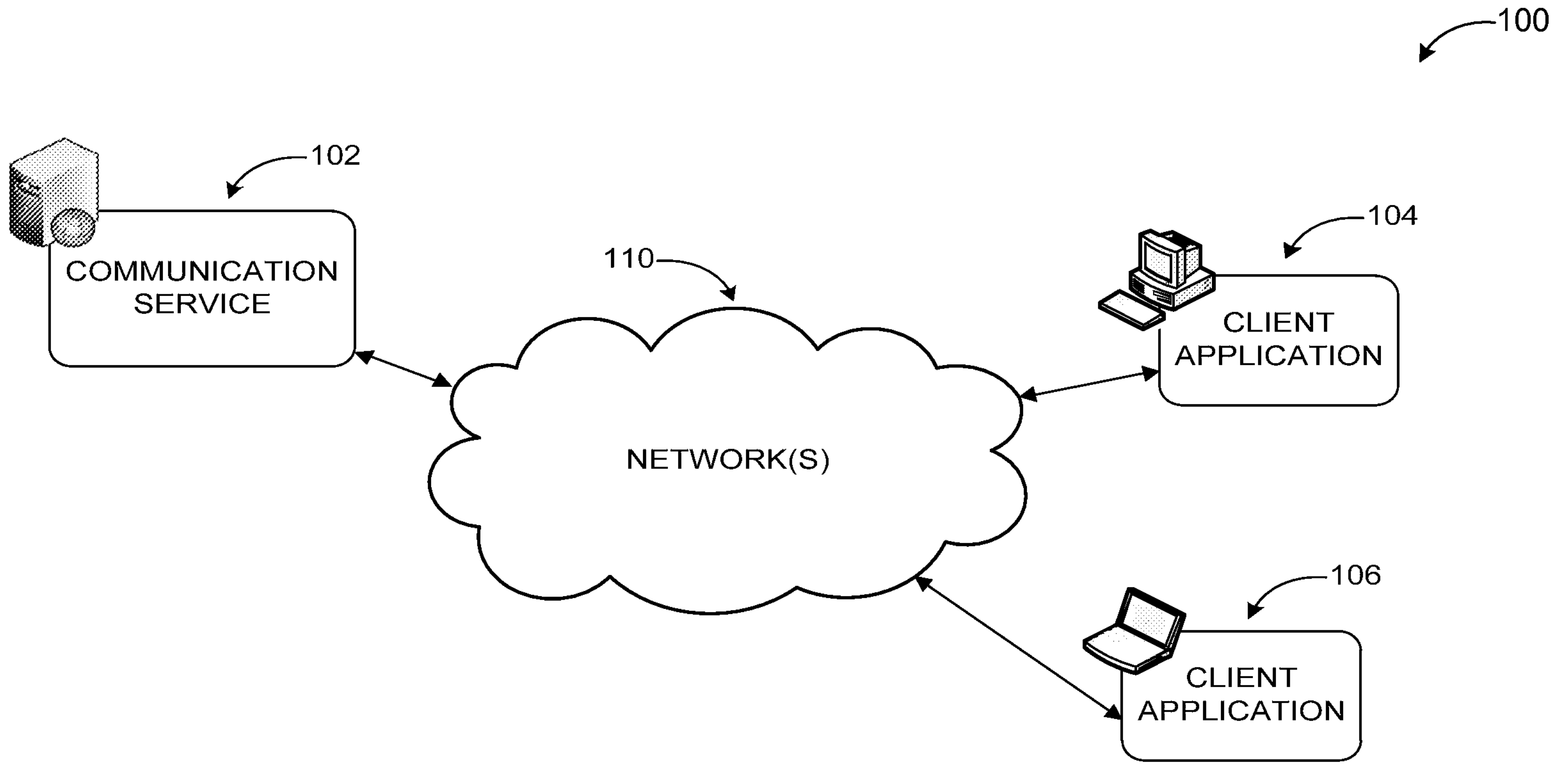

100
102
COMMUNICATION SERVICE
110
NETWORK(S)
104
CLIENT APPLICATION
106
CLIENT APPLICATION

FIG. 1

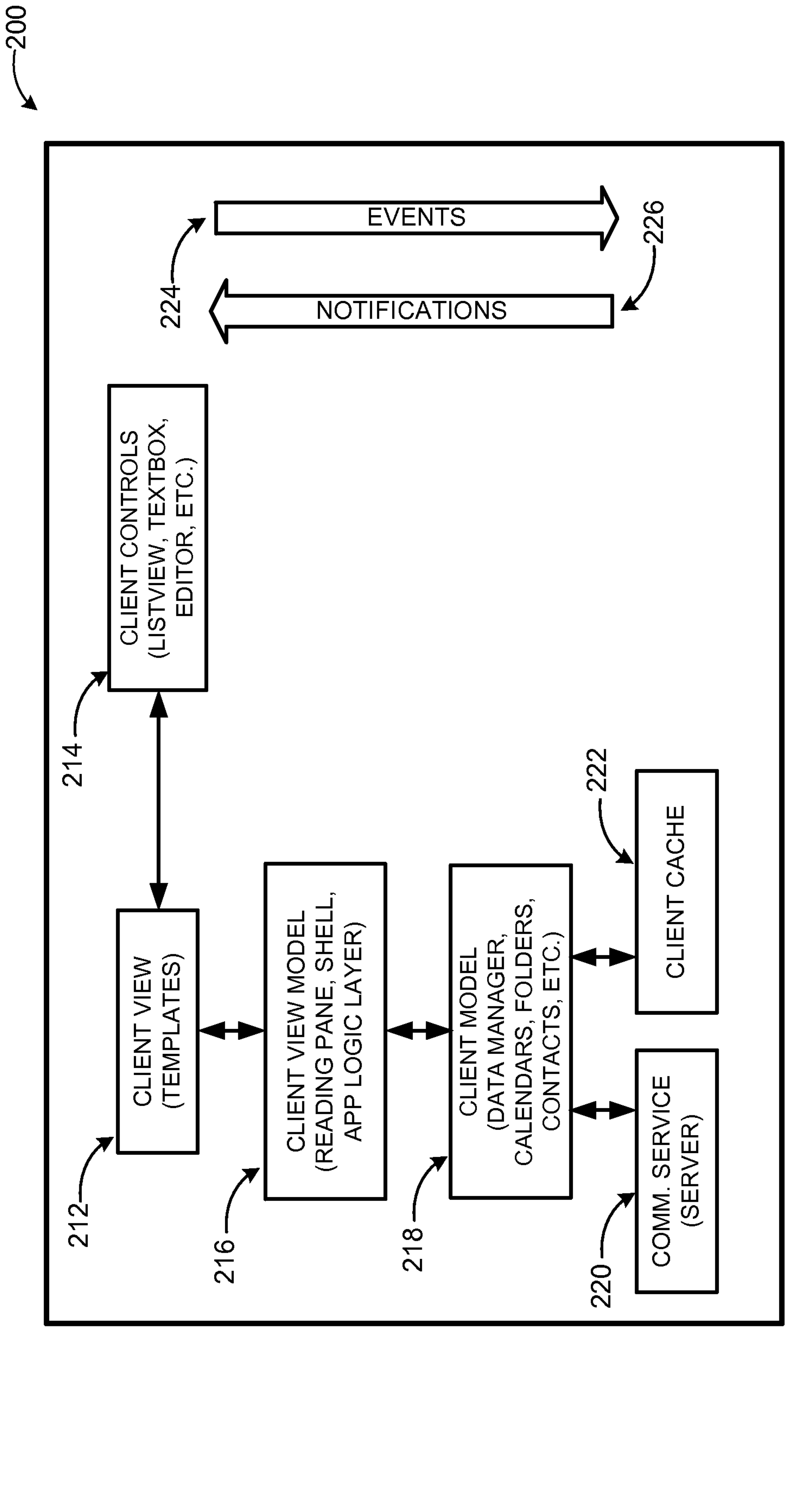

200
EVENTS
226
224
NOTIFICATIONS
CLIENT CONTROLS (LISTVIEW, TEXTBOX, EDITOR, ETC.)
214
CLIENT VIEW (TEMPLATES)
212
CLIENT VIEW MODEL (READING PANE, SHELL, APP LOGIC LAYER)
216
CLIENT MODEL (DATA MANAGER, CALENDARS, FOLDERS, CONTACTS, ETC.)
218
CLIENT CACHE
222
COMM. SERVICE (SERVER)
220

FIG. 2

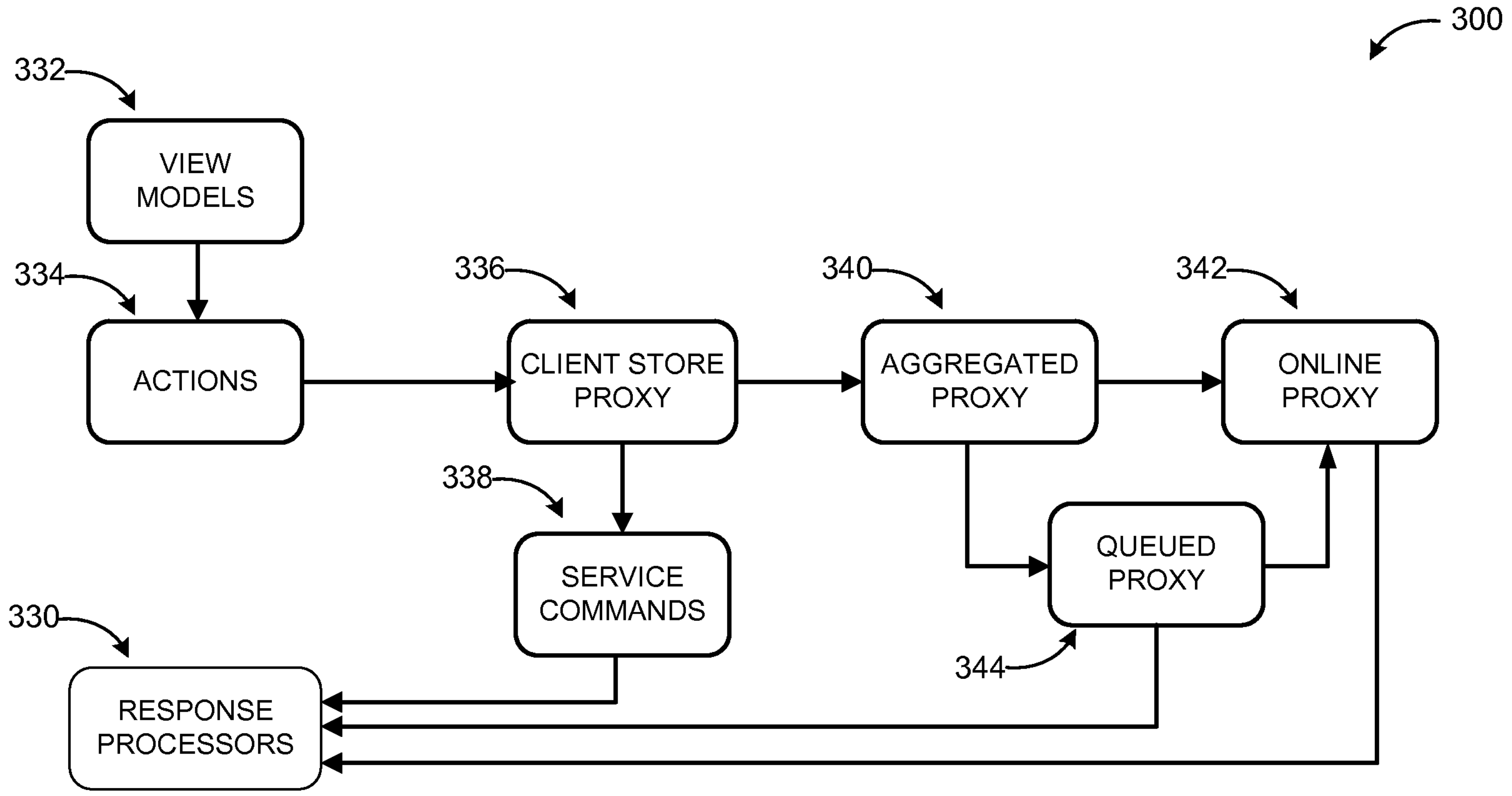

300
332
VIEW MODELS
334
ACTIONS
336
CLIENT STORE PROXY
340
AGGREGATED PROXY
342
ONLINE PROXY
338
SERVICE COMMANDS
QUEUED PROXY
344
330
RESPONSE PROCESSORS

FIG. 3

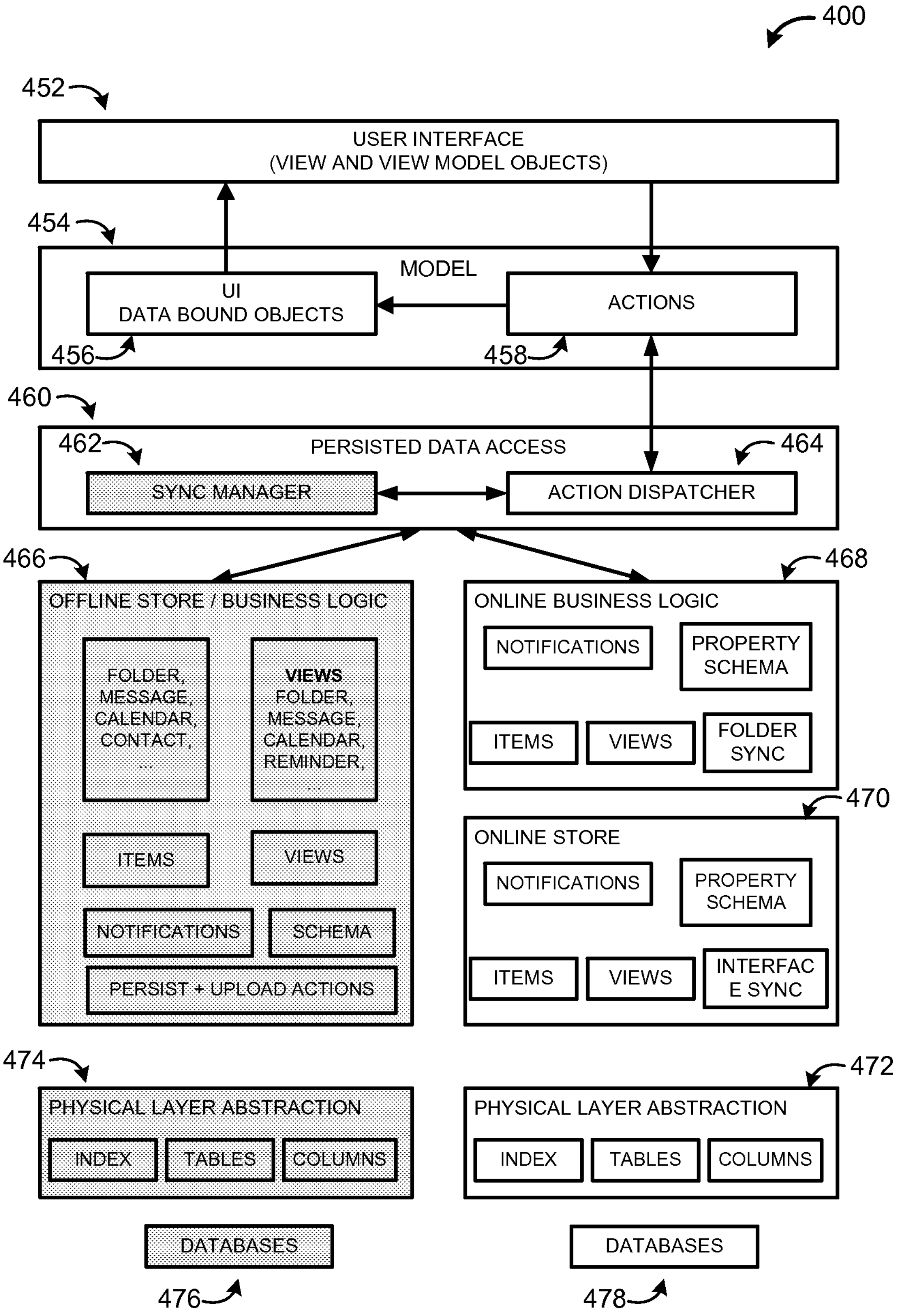
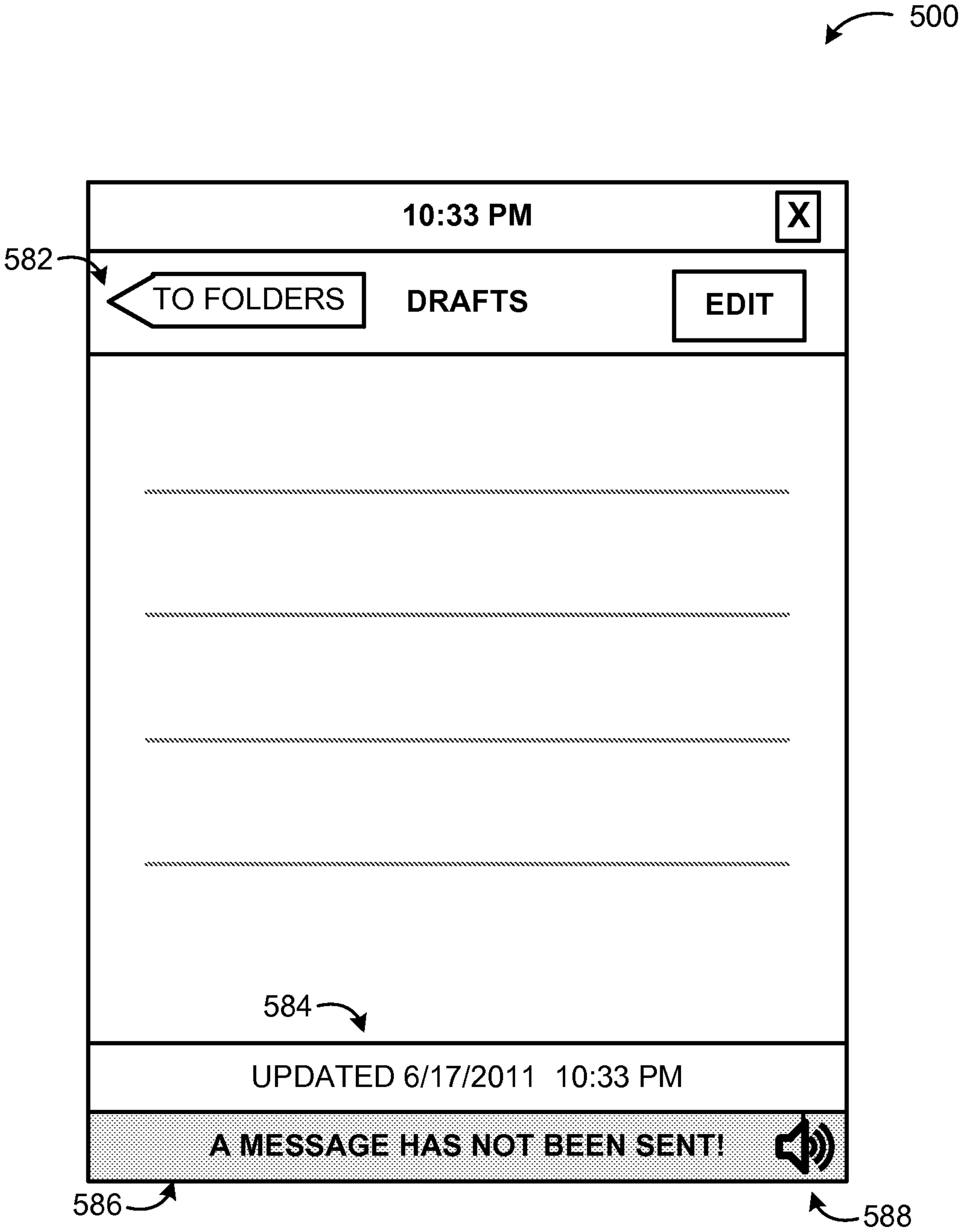

400
452
USER INTERFACE
(VIEW AND VIEW MODEL OBJECTS)
454
MODEL
UI
DATA BOUND OBJECTS
ACTIONS
456
458
460
PERSISTED DATA ACCESS
462
464
SYNC MANAGER
ACTION DISPATCHER
466
468
OFFLINE STORE / BUSINESS LOGIC
FOLDER,
MESSAGE,
CALENDAR,
CONTACT,
...
VIEWS
FOLDER,
MESSAGE,
CALENDAR,
REMINDER,
...
ITEMS
VIEWS
NOTIFICATIONS
SCHEMA
PERSIST + UPLOAD ACTIONS
ONLINE BUSINESS LOGIC
NOTIFICATIONS
PROPERTY SCHEMA
ITEMS
VIEWS
FOLDER SYNC
470
ONLINE STORE
NOTIFICATIONS
PROPERTY SCHEMA
ITEMS
VIEWS
INTERFAC E SYNC
474
472
PHYSICAL LAYER ABSTRACTION
INDEX
TABLES
COLUMNS
PHYSICAL LAYER ABSTRACTION
INDEX
TABLES
COLUMNS
DATABASES
DATABASES
476
478

FIG. 4



UPDATED 6/17/2011 10:33 PM

A MESSAGE HAS NOT BEEN SENT!

FIG. 5

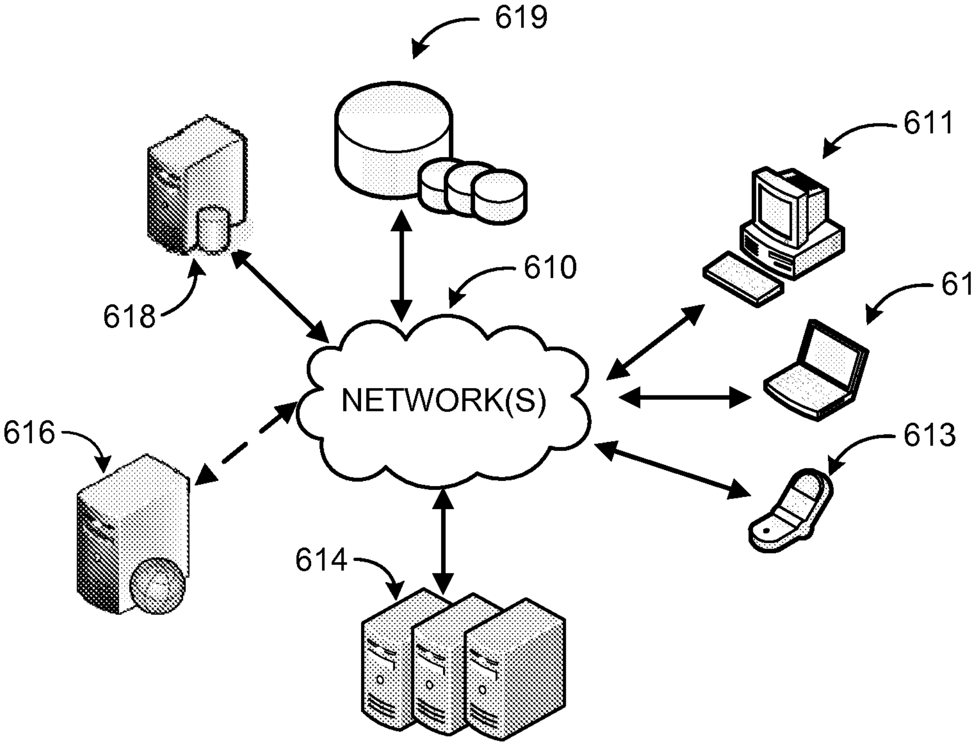


FIG. 6

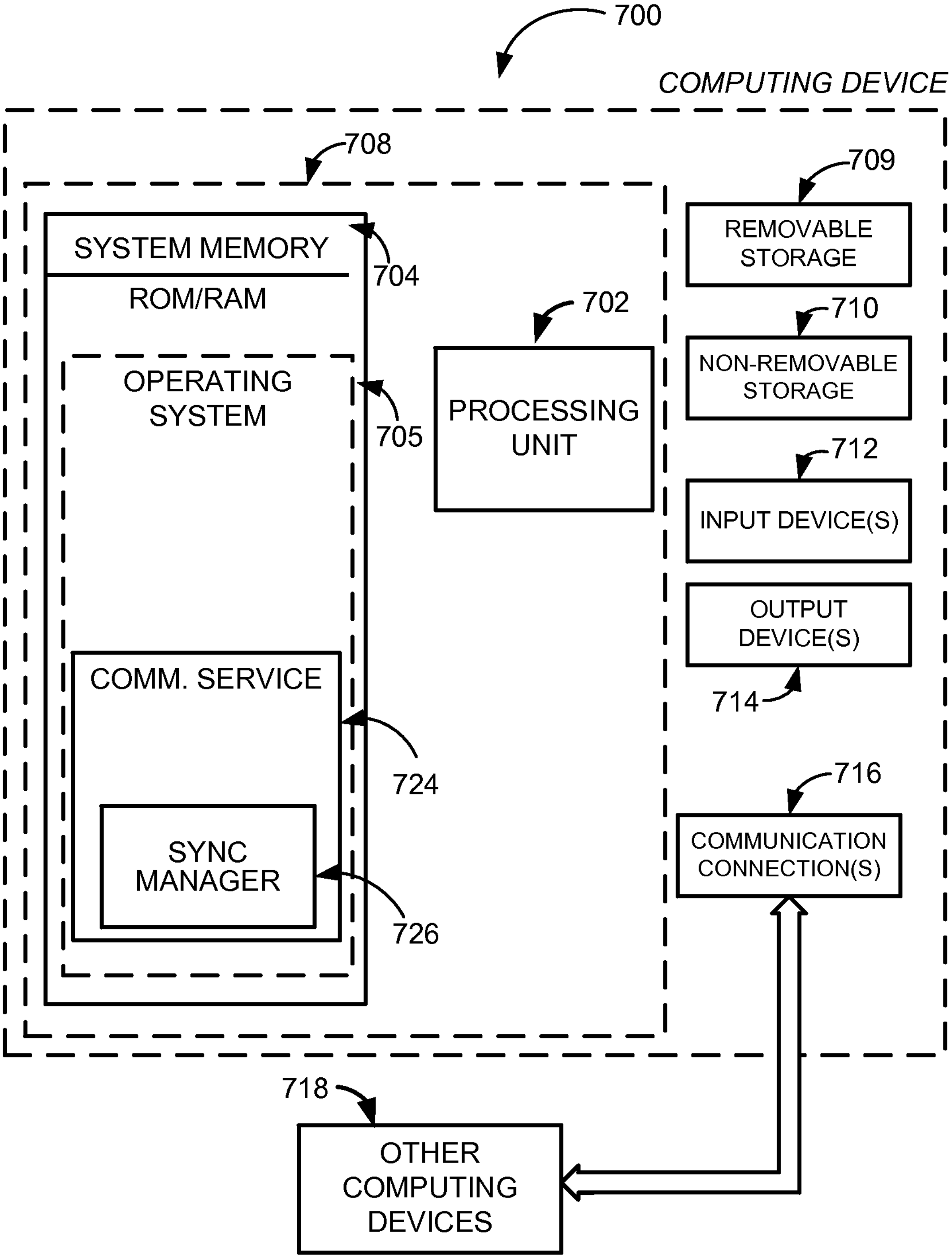

700
COMPUTING DEVICE
708
709
REMOVABLE STORAGE
SYSTEM MEMORY
704
ROM/RAM
702
710
NON-REMOVABLE STORAGE
OPERATING SYSTEM
705
PROCESSING UNIT
712
INPUT DEVICE(S)
OUTPUT DEVICE(S)
714
COMM. SERVICE
724
716
COMMUNICATION CONNECTION(S)
SYNC MANAGER
726
718
OTHER COMPUTING DEVICES

FIG. 7

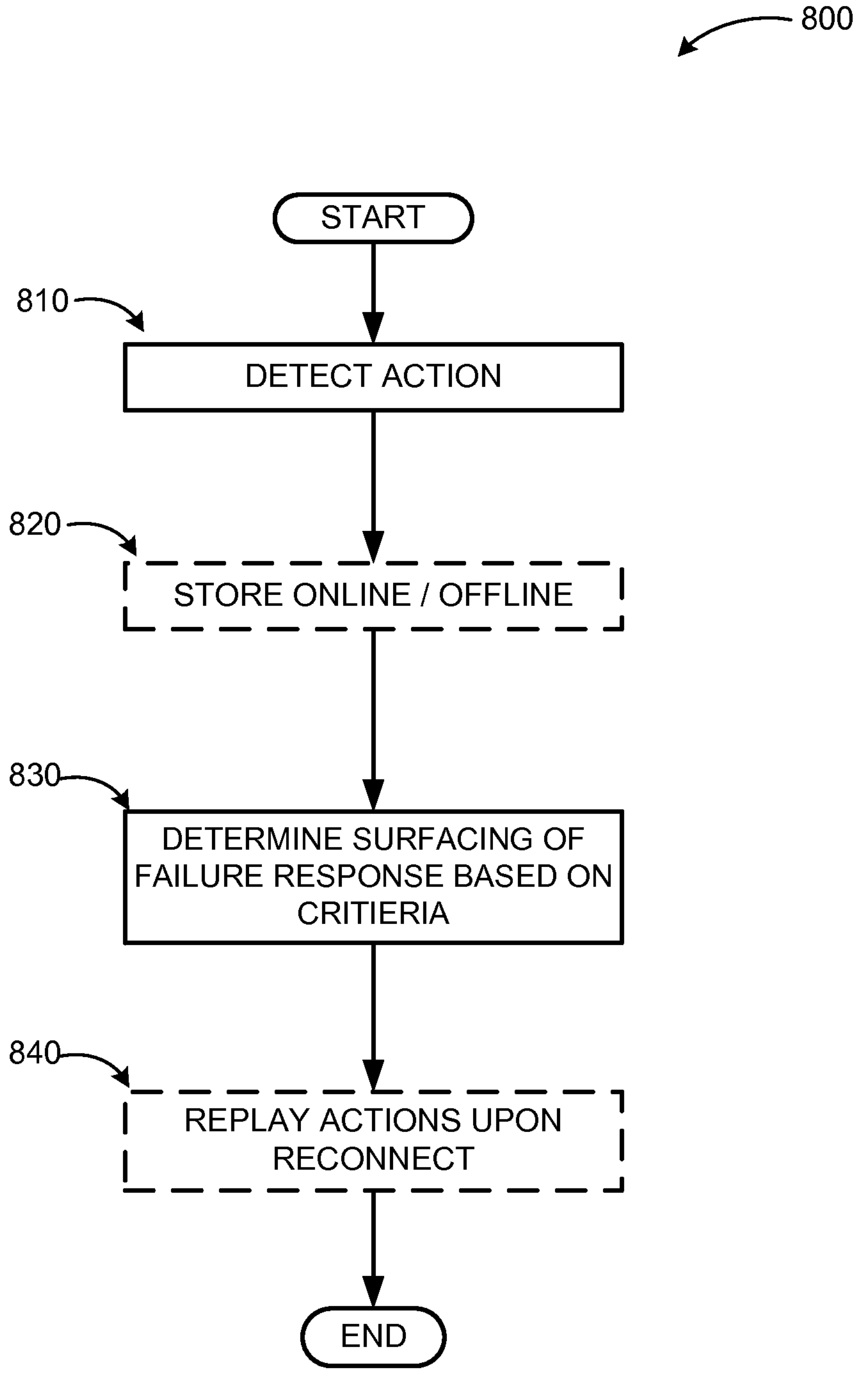

800
START
810
DETECT ACTION
820
STORE ONLINE / OFFLINE
830
DETERMINE SURFACING OF FAILURE RESPONSE BASED ON CRITIERIA
840
REPLAY ACTIONS UPON RECONNECT
END

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2012/058205

A. CLASSIFICATION OF SUBJECT MATTER

G06F 11/30(2006.01)i, G06F 11/32(2006.01)i, G06F 15/16(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 11/30; H04L 1/22; G06F 11/07; G06F 15/167; G06F 11/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

eKOMPASS(KIPO internal) & Keywords: user action, communication failure, timing, type and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y A	US 2004-0117453 A1 (MICHAEL CHENG et al.) 17 June 2004 See paragraphs [0034]-[0042]; claims 1,6; and figures 2,5.	1,4-5,7,10 2-3,6,8-9
Y	US 7111205 B1 (JEFFREY M. JAHN et al.) 19 September 2006 See column 3, line 3 - column 6, line 64; claims 1,5; and figures 2-4.	1,4-5,7,10
A	US 6665820 B1 (RICHARD L. FROWEIN et al.) 16 December 2003 See column 7, line 31 - column 8, line 52; and figures 4-5.	1-10
A	US 2003-0196136 A1 (LEON E. HAYNES et al.) 16 October 2003 See paragraphs [0037]-[0043],[0062]-[0067];and figures 2,4-5.	1-10
A	US 2010-0083032 A1 (BYRON A. ALCORN et al.) 01 April 2010 See paragraphs [0038]-[0042]; claims 1-4; and figure 4.	1-10

□ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search 29 JANUARY 2013 (29.01.2013)	Date of mailing of the international search report **30 JANUARY 2013 (30.01.2013)**
Name and mailing address of the ISA/KR Facsimile No. 82-42-472-7140	Authorized officer BYUN, SUNG CHEAL 변성철인 Telephone No. 82-42-481-8262

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2012/058205

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2004-0117453 A1	17.06.2004	None	
US 7111205 B1	19.09.2006	None	
US 6665820 B1	16.12.2003	FR 2803041 A1 FR 2803041 B1 JP 2001-237835 A	29.06.2001 21.09.2007 31.08.2001
US 2003-0196136 A1	16.10.2003	US 6993681 B2	31.01.2006
US 2010-0083032 A1	01.04.2010	None	