



(51) International Patent Classification:
H04W 8/18 (2009.01)

(21) International Application Number:
PCT/CA2010/000644

(22) International Filing Date:
30 April 2010 (30.04.2010)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/175,024 3 May 2009 (03.05.2009) US

(71) Applicant (for all designated States except US): **RE-SEARCH IN MOTION LIMITED** [CA/CA]; 295 Phillip Street, Waterloo, Ontario, N2L 3W8 (CA).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **SCHNEIDER, Kenneth, Cyril** [CA/CA]; 75835 - 185 Columbia Street, Waterloo, Ontario N2L 525 (CA). **HON, Lenny, Kwok-Ming** [CA/CA]; 14488 - 4701 Tahoe Boulevard, Mississauga, Ontario L4W 085 (CA). **WANG, Qiusheng** [CA/CA]; 14488 - 4701 Tahoe Boulevard, Mississauga, Ontario L4W 085 (CA).

(74) Agent: **RIDOUT & MAYBEE LLP**; 225 King Street West, 10th Floor, Toronto, Ontario M5V 3M2 (CA).

(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 (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:

— with international search report (Art. 21(3))

(54) Title: SYSTEMS AND METHODS FOR MOBILITY SERVER ADMINISTRATION

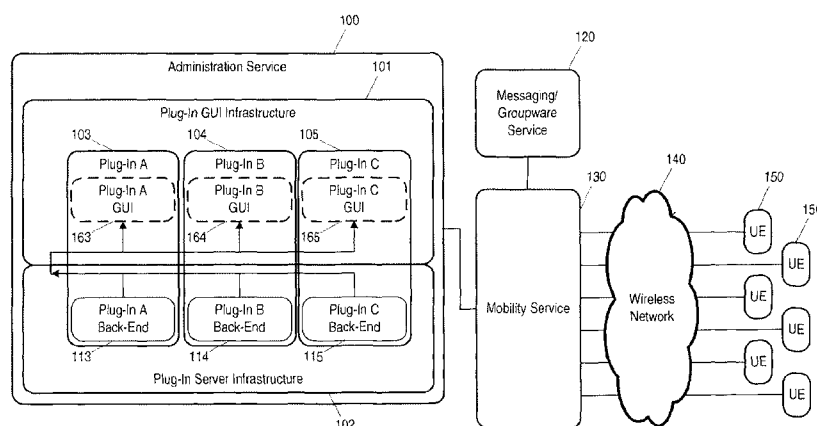


FIG. 1

(57) Abstract: An administration server of an administration service assigns attributes to objects by a plug-in of the administration service. The plug-in implements a method of a functionality set and the method is callable by the administration service to perform the assigning. Additionally or alternatively, the administration server triggers a reconciliation event by changing the assignment of an attribute of the users that comprise objects of plug-ins; determines a scope of the users and which objects are affected by changing the assignment; and reconciles conflicting assignments. Additionally or alternatively, the administration server adds tasks by the plug-ins to a job created by the plug-ins with the tasks performing the assigning; and removes tasks from the job to optimize it



SYSTEMS AND METHODS FOR MOBILITY SERVER ADMINISTRATION

Cross-Reference to Related Application

This application claims the benefit of and priority to United States Patent
5 Application No. 61/175,024 filed May 3, 2009.

The content of the above patent application is hereby expressly incorporated by reference into the detailed description hereof.

10 FIELD OF TECHNOLOGY

This disclosure relates to systems and methods for supporting and provisioning mobile devices, and in particular to systems and methods for providing administration services to servers that support mobile devices.

15 BACKGROUND

A variety of wireless mobile devices are now available, including devices which support one or more data-related applications, such as e-mail, calendar, contact management, other personal information management (PIM) applications, and other similar applications. Although there exist a variety of suitable system architectures in
20 which wireless mobile devices may be applied, an architecture which has found favor in enterprise environments involves relatively small wireless mobile devices which implement the aforementioned data-related applications (e.g., e-mail, calendar, contact management, other personal information management (PIM) applications, and other similar applications) in conjunction with enterprise-run "back-end" servers or services
25 which support these applications, and which may be accessed by a variety of client devices. For example, an enterprise may operate one or more servers that provide messaging or e-mail services to a variety of client devices. The enterprise may also operate one or more servers that provide calendar, contact management, or other PIM services to client devices. Sometimes, several of these services are integrated in a
30 "groupware" server.

In general, the enterprise messaging and PIM servers (hereafter, "application servers") have been designed to support a particular class of conventional client devices consisting of conventional personal computers (whether in notebook or other form), which are connected to the enterprise network either directly, or indirectly via a

connection method which is largely transparent to the server. Many wireless mobile devices are not conventional client devices, in that they are not conventional personal computers or are not directly or transparently connected to the enterprise network. In order to provide a high level of service to wireless mobile devices (other than

5 conventional client devices), an enterprise may deploy one or more mobility servers (along with other optional infrastructure elements) which provide intermediation services between the application servers and the wireless mobile devices. The intermediation services may include, for example, management of synchronization of contents between the application server and the wireless mobile device, rapid delivery of e-mail or other

10 content to the wireless mobile device, and adaptation of content to be delivered to the wireless mobile device in a manner consistent with the capabilities of the device and the transport media (e.g., carrier and transport networks) by which the device is connected to the application server.

A concern among information technology personnel responsible for managing an

15 enterprise fleet of wireless mobile devices and the enterprise infrastructure to support them is the orderly administration of a mobility server, and its associated wireless mobile devices, their users, the services and applications provided thereto, and the policies governing their operation.

20 **BRIEF DESCRIPTION OF THE DRAWINGS**

The disclosure will be more clearly understood by reference to the following detailed description of example embodiments and in conjunction with the accompanying drawings by way of illustration. In the drawings, like numerals describe substantially similar components throughout the several views.

25 FIG. 1 is a block diagram depicting a system constructed according to an aspect of the present disclosure for providing administration services to a mobility server;

FIG. 2 is a flow diagram depicting an example method of integration in accordance with a "plug-in" software component arrangement for use with the system of Fig. 1;

30 FIG. 3 is a flow diagram depicting an example method for reconciling objects in conjunction with the system of Fig. 1;

FIG. 4 is a flow diagram depicting an example method of creating and executing a job for use with the system of Fig. 1;

FIG. 5A and FIG. 5B are a flow diagram illustrating an example method for executing tasks and processing a finished task for use with the system of Fig. 1;

FIG. 6 is a block diagram depicting a communications system, showing an example environment in conjunction with which the system of Fig. 1 may be used;

FIG. 7A is the top portion and FIG. 7B is the bottom portion of a display diagram depicting an example display area of a graphical user interface for use in conjunction with the system of Fig. 1;

FIG. 8A and 8B are display diagrams depicting an example display area of a graphical user interface for use in conjunction with the system of Fig. 1;

FIG. 9A is the left portion and FIG. 9B is the right portion of a display diagram depicting an example display area of a graphical user interface for use in conjunction with the system of Fig. 1;

FIG. 10A is the left portion and FIG. 10B is the right portion of a display diagram depicting an example display area of a graphical user interface for use in conjunction with the system of Fig. 1; and

FIG. 11 is a display diagram depicting an example display area of a graphical user interface for use in conjunction with the system of Fig. 1.

DETAILED DESCRIPTION

According to an aspect of the present disclosure, and as best seen in Figs. 1 and 6, an administration service 100 is provided for administering a mobility service 130, including users, devices, services, and policies associated therewith. Fig. 6 is a block diagram depicting, as an example environment in which the administration service 100 may be employed, a communications system 680. As best seen in Fig. 6, communications system 680 comprises an enterprise network part 682, and a network part 684 which is generally external to the enterprise network 682. For a non-limiting example, the external network part 684 may comprise a wireless network 140 which is operatively coupled to the enterprise network 682 through a transport network 686.

As best seen in Figs. 1 and 6, an administration server 601 of administration service 100 can administer attributes of one or more users (e.g., user 651). A user can use many different services, including services provided by mobility service 130, messaging/groupware service 120, and various other applications services and servers generally shown as 623, all of which the user can access via user equipment (UE 150). In some embodiments, the mobility service 130 provides an intermediation function between UE 150 and various services and servers, e.g., 120 and 623, of enterprise network 682. The intermediation services may include, for example, management of synchronization of contents between application services, such as messaging/groupware service 120, server

623, and the UE 150; rapid delivery of e-mail or other content to the UE 150; and adaptation of content to be delivered to the UE 150 in a manner consistent with the capabilities of the UE 150 and such elements of external network 684 as wireless network 140 and transport network 686. In some embodiments, although the mobility service 130 significantly improves the user experience, the user may be unaware of its existence. In other embodiments, the mobility service 130 may directly provide user-visible services. A user 651 can be a member of one or more groups (660-661 and 663) and be serviced by one or more servers (631 and 621) of one or more services (130 and 120). Additionally, a group (e.g., 662) can comprise one or more other groups (e.g., 661).

Examples of attributes that can be assigned to a user (651) can include: information technology (IT) policies, software configurations, applications (which may be required, optional, or disallowed), device software configurations, device software bundles, native application settings, or any other application settings. The management and assignment of the number, types, and functionality of attributes can be controlled by the administration server 601. As best seen in Fig. 1, the functionality of the administration service 100 may be extended by additional software components 103, 104, and 105, referred to hereafter as "plug-ins". Although three plug-ins are shown and described, any number of plug-ins can be used. Thus, the attributes that administration server 601 can manage and assign can be continuously extended by adding plug-ins, as discussed further below in the section titled PLUG-INS.

Additionally, attributes can also be assigned to groups (660-663), services (120 and 130), servers (631-634 and 621-623), and can be assigned to domains upon which the servers (631-634 and 621-623) reside. A user (651) can inherit attributes assigned to groups, services, servers, and domains with which the user is associated. Users may inherit or be assigned an attribute that conflicts with other attributes that are assigned to groups, services, servers, or domains with which the user is associated. When various attributes that are assigned to a user conflict, a reconciliation of the various attributes can occur based on rules that indicate which assignment has priority so that the proper attribute can be assigned to the user, which is discussed below in the section titled RECONCILIATION MANAGEMENT.

Additionally, when an attribute of a group, server, service, or domain is changed, all the users associated with that group server service or domain can be affected and may need reconciliation. Also, when one or more attributes are changed, a job can be created that comprises one or more tasks that bring about the assignment of a reconciled attribute

to a user. Multiple tasks of multiple jobs may bring about similar assignments resulting in duplicate and/or superfluous tasks that can be optimized out to reduce the number of duplicate or superfluous tasks, which is discussed below in the section titled JOB MANAGEMENT.

5 The present application relates to mobile communications systems, which may be implemented using a variety of electronic and optical technologies, including but not limited to: analog electronic systems; digital electronic systems; microprocessors and other processing elements; and software and otherwise embodied collections of steps, instructions, and the like, for implementing methods, processes, or policies in conjunction
10 with such systems and processing elements. It will be appreciated that in the relevant arts, various signal leads, busses, data paths, data structures, channels, buffers, message-passing interfaces, and other communications paths may be used to implement a facility, structure, or method for conveying information or signals, and are often functionally equivalent. Accordingly, unless otherwise noted, references to apparatus or data
15 structures for conveying a signal or information are intended to refer generally to all functionally equivalent apparatus and data structures.

PLUG-INS

FIGS. 1 and 6 are a block diagrams depicting a communications system 680 that includes one or more user equipment (UE) 150, wireless network 140, mobility service
20 130, messaging/groupware service 120, and administration service 100. UE 150 can be used to send and receive messages via wireless network 140, messaging/groupware service 120, and mobility service 130. Examples of UE 150 include smart phones such as those commercially available under the BLACKBERRY brand, cellular telephones, or any other portable handheld device that can wirelessly send and receive messages.

25 Wireless network 140 provides for the communication between UE 150 and messaging/groupware service 120, other services or servers 623 (Fig. 6) and mobility service 130. The wireless network 140 may be implemented as any suitable wireless access network technology. By way of example, but not limitation, the wireless network 140 may be implemented as a wireless network that includes a number of base stations
30 (not shown), each containing radio transmitting and receiving equipment adapted to provide wireless radio-frequency (RF) network service or "coverage" to a corresponding area or cell. The wireless network 140 is typically operated by a mobile network service provider that provides subscription packages to users of wireless mobile devices or user equipment (UE) 150. The wireless network 140 may be implemented as any appropriate

wireless network, including but not limited to any to one or more of the following wireless network types: Mobitex Radio Network, DataTAC, GSM (Global System for Mobile Communication), GPRS (General Packet Radio System), TDMA (Time Division Multiple Access), CDMA (Code Division Multiple Access), CDPD (Cellular Digital Packet Data), iDEN (integrated Digital Enhanced Network), EvDO (Evolution-Data Optimized) CDMA2000, EDGE (Enhanced Data rates for GSM Evolution), UMTS (Universal Mobile Telecommunication Systems), HSPDA (High-Speed Downlink Packet Access), IEEE 802.16e (also referred to as Worldwide Interoperability for Microwave Access or "WiMAX). Other network types and technologies could also be used.

Wireless network 140 could also be implemented as a wireless local area network (WLAN) or wireless metropolitan area network (WMAN) or any combination of the aforementioned networks. As best seen in Fig. 6, a transport network may be used to couple wireless network 140 to elements of the enterprise network 682. Transport network 686 may be implemented using any suitable network technology for coupling the wireless network 140 and the enterprise network 682. For example, transport network 686 may be implemented using any combination of the public Internet, private network facilities, leased facilities, private overlays on a shared network (sometimes called a virtual private network or "VPN"), or any other suitable network technology. Enterprise network 682 may be implemented using any suitable network technology including local network technologies (LANs), wide-area network technologies (WANs), wireless network technologies, and the like. Any of the networks mentioned may employ optical, electrical, coaxial, waveguide, free-space radio or microwave, or any other suitable media, in conjunction with switches, routers, and other appropriate interconnection means.

In some embodiments, the mobility service 130 provides an intermediation function between UE 150 and various services and servers, e.g., 120 and 623, of enterprise network 682. The intermediation services may include, for example: management of synchronization of contents between application services, such as messaging/groupware service 120, server 623, and the UE 150; rapid delivery of e-mail or other content to the UE 150; and adaptation of content to be delivered to the UE 150 in a manner consistent with the capabilities of the UE 150 and such elements of external network 684 as wireless network 140 and transport network 686. For example, mobility service 130 can provide for synchronizing messages between one or more UEs 150 and messaging/groupware service 120. Mobility service 130 can comprise one or more

programs operating on one or more servers. Mobility service 130 can use multiple policies to control how and what messages and data are synchronized with UEs 150. A suite of software that implements a mobility service is commercially available from Research In Motion under the name BLACKBERRY ENTERPRISE SERVER software.

5 Messaging/groupware service 120 can send and receive messages to and from users of its service and can comprise one or more programs operating on one or more servers. Programs that operate as messaging/groupware services are commercially available from Microsoft under the name EXCHANGE software, from IBM under the name LOTUS DOMINO software, and from Novell under the name GROUWISE
10 software. Other messaging/groupware software could also be used. Types of messaging/groupware services can include email service, instant messaging service, session initiation protocol (SIP) service, or any other service that allows for sending and receiving messages to and from users.

Administration service 100 can provide a single point for all management
15 activities within the mobility service 130. Many areas of independent functionality are managed by administration service 100. Administration service 100 delegates various management details to independent components that implement specific areas of functionality. These independent components are known as “plug-in” modules (103, 104, and 105) that each comprise a respective plug-in back-end component (113, 114, and 115)
20 and can optionally comprise a respective plug-in GUI component (163, 164, and 165). Plug-in modules (plug-ins) can interact with Administration service 100 via a plug-in graphical user interface (GUI) infrastructure 101 and via a plug-in server infrastructure 102.

Plug-in GUIs 163, 164, and 165 of plug-ins 103, 104, and 105 interact with plug-
25 in GUI infrastructure 101 through one or more GUI integration points. A GUI integration point includes a place in the GUI of administration service 100 where user interaction can take place. Integration points can be defined by: a navigation location to get to a point in the GUI, a mounting point where integration is to take place; and contiguous screen area boundaries into which the plug-ins (103-105) can render information and interact with the
30 user. A GUI integration point can be any of several types including: a navigation tree node, a page, a tab, a band, and a link, and an item in a context menu (a pop-up menu), or any other GUI element.

A navigation tree node GUI integration point can comprise one or more nodes in a navigation tree. FIGS. 7A and 7B are a display diagram showing an example navigation tree in display area 700 with multiple integration points 710 outlined in black.

FIGS. 8 and 8B are display diagrams showing an example Page GUI Integration Point. A Page GUI integration point can comprise the full screen under the context title as shown by area 810 of display area 800.

FIGS. 9A and 9B are a display diagram showing an example tab GUI integration point. A tab GUI integration point can comprise a tab plus the screen area below the tab as shown by area 910 of display area 900.

FIGS. 10A and 10B are a display diagram showing an example band GUI integration point. A band GUI integration point can comprise a horizontal slice of the screen delimited by a new band as shown by areas 1010 of display area 1000.

FIG. 11 is a display diagram showing an example link GUI integration point. A link GUI integration point can comprise one or more links on the bottom of a screen as shown by area 1110 of display area 1100. The link may be clicked on to navigate to other portions of the GUI or to cause an action to be performed such as reloading the display of a user.

As best seen in Fig. 1, plug-in back-ends 113, 114, and 115 of plug-ins 103, 104, and 105, respectively, interact with plug-in server infrastructure 102 through one or more functionality sets. A functionality set includes the one or more functions or methods that are implemented by a plug-in (e.g., 103, 104, and 105) and used by the plug-in server infrastructure 102. Assigning the attributes to the users can be carried out through the execution of these methods. Any number of functionality sets can be defined and supported by administration service 100 and/or plug-in server infrastructure 102, including: a server functionality set, a user functionality set, a reconciliation functionality set, and a delivery functionality set. Also, a plug-in (e.g., plug-in A 103) can define its own functionality set which can then be implemented by other plug-ins (e.g., plug-ins B 104 and C 105). In this case, the extended functionality set is used by the plug-in that defines the extended functionality set. A functionality set allows for the implementation of functionality to be delegated to a plug-in without the plug-in server infrastructure knowing anything about the implementation of that functionality.

Administration service 100 can provide a full object data model for administration. Included within this data model are various base objects which are used within functionality sets. A plug-in which implements a functionality set that includes

base objects as parameters will extend those base objects with the plug-in's own specific attributes. These extended or polymorphic objects can be handled by the administration service 100 without the plug-in server infrastructure 102 knowing about any of the specific extensions.

5 The combination of functionality sets and polymorphic objects allows for the seamless integration of a plug-in. For example, plug-in A 103 can extend an extant user object with its own specific user attributes and, similarly, plug-in B 104 can extend the extant user object with its own specific user attributes. In other words, plug-in A can have a userA object and plug-in B can have a userB object, wherein both the userA object and the userB object extend from the extant user object. The administration service 100
10 can show all the attributes from the extant user object, the userA object, and the userB object and also allow these attributes to be edited and saved via plug-ins A 103 and B 104.

FIG. 2 is a flow diagram depicting an example method by which the integration of
15 a plug-in can take place. The steps or operations described herein are examples. There may be many variations to these steps or operations without departing from the scope of this disclosure. For instance, where appropriate, the steps may be performed in a differing order, or steps may be added, deleted, or modified. At step 201, plug-in GUI infrastructure can define a GUI integration point, such as a tab GUI integration point, for
20 plug-ins A 103 and B 104 to display and/or edit attributes.

At step 202, the plug-in GUI infrastructure 101 determines, for example, that user objects are to be displayed, and accordingly, the plug-in GUI infrastructure 101 calls a find user method defined by the plug-in server infrastructure 102.

At step 203, the find user method can discover all of the plug-ins (e.g., plug-ins A
25 103 and B 104) that are currently registered which implement the user functionality set (this is the functionality set that includes the plug-ins' implementations of the find user method). At step 204, the plug-in server infrastructure 102 can then iteratively call the find user method of plug-ins A 103 and B 104 through the user functionality sets of plug-ins A 103 and B 104.

30 At step 205, plug-in back-ends A 163 and B 164 can return an object associated with the find user method. More specifically, the result of calling, in step 204, the find user method of the user functionality set implemented by plug-in A 103 is that the plug-in back-end A 163 returns a userA object. Similarly, the result of calling, in step 204, of the

find user method of the user functionality set implemented by plug-in B 104 is that the plug-in back-end B 164 returns a userB object.

At step 206, the polymorphic user objects (the userA object and the userB object) can be gathered into an array and returned by the plug-in server infrastructure 102 to the
5 plug-in GUI infrastructure 101.

At step 207, the plug-in GUI infrastructure 102 has the full set of user attributes (i.e., the array of user, userA, and userB objects) and discovers all of the plug-ins that are registered for the tab GUI integration point.

At step 208, for each registered plug-in, plug-in GUI infrastructure 102 sets up a
10 tab and passes the polymorphic objects from the array to the appropriate plug-in GUIs to be rendered. As an example, the tab integration point for Plug-in A is called, passing the userA object, and the plug-in renders the attributes on the GUI. Similarly, the tab integration point for plug-in B is called, with the userB object, and the plug-in renders the attributes on the GUI.

At step 209 the full user object is displayed on the GUI of the administration
15 service 100, which was facilitated by both the plug-in GUI infrastructure 101 and by the plug-in server infrastructure 102 without either of them knowing any of the plug-in specific details about the user.

In addition to displaying attributes, the plug-in architecture of administration
20 service 100 also supports saving and searching the attributes. Many other functionalities can be supported by the plug-in architecture.

The plug-in architecture of administration service 100 via plug-in GUI
infrastructure 101 and plug-in server infrastructure 102 may provide many potential
benefits. One potential benefit is that the plug-in server infrastructure does not need to
25 know any of the specific details of the specialized management components (plug-ins) that implement functionality sets and provide the functionality of the administration service 100.

Additionally, plug-in modules can be upgraded independently of administration
service 100 or any other plug-in module at any time. Also, new plug-in modules can be
30 added at any time and they simply appear in the administration service 100 GUI.

Another potential benefit is that plug-in modules can be developed independently, including by external third parties, and still be seamlessly incorporated into administration service 100 and look like a natural part of administration service 100.

RECONCILIATION MANAGEMENT

Administration service 100 can manage the assignment of attributes to users, groups, servers, services, and domains. These attributes can include such objects as applications, data, access rights, device configurations, software configurations, and policies. Sometimes, one or more plug-ins may try to set an attribute of a group (or user,
5 service, server, or domain) that results in a conflict with one or more attributes that are directly or indirectly assigned to or inherited by a user. Administration service 100 can resolve conflicts that arise from reconciliation objects being assigned to the users, groups, servers, services, or domains.

Examples of reconciliation objects can include: IT Policies, software
10 configurations, applications (which may be required, optional, or disallowed), application control policy (ACP) for unlisted applications, device software configurations, device software bundles, native application settings or any other application settings. The reconciliation objects are plug-in polymorphic objects, wherein the plug-in server infrastructure need not know the details of the reconciliation objects, but does know that
15 they can be assigned to users, groups, services, servers, and domains. A plug-in can define new reconciliation objects at any time and extend the functionality of the entire system from a reconciliation point of view.

The following set of conditions is an example illustrating a situation in which reconciliation may be useful. A user is defined to a member of group A and group A is
20 defined to be a member of group B. (As mentioned earlier, a group may contain other groups.) Subsequently, an attribute (IT Policy 1) is assigned directly to the user and another attribute (IT Policy 2) is assigned to group B. This means that the user would be assigned both IT Policy 1 and IT Policy 2. However, a constraint of communication system 680 is that only a single IT Policy may be active on any user's device (UE 150).
25 According to an aspect of the present disclosure, the conflict resulting from the apparent assignment of two different IT Policies to a user (and thus the user's UE 150) can be reconciled through the use of reconciliation objects and reconciliation functionality sets.

A reconciliation functionality set allows the plug-in server infrastructure 102 to perform reconciliation without knowing exactly what the objects being reconciled are.
30 One of the capabilities of a reconciliation functionality set is the ability to furnish to the plug-in server infrastructure a list of the reconciliation objects supported by a plug-in such as 103, 104, or 105. There are also several options with regard to a reconciliation operation that are made available through the functionality set. Specific business rules for resolving conflicts are implemented by each of the plug-ins 103, 104, or 105. The plug-in

server infrastructure 102 can resolve conflicts by passing a set of reconciliation objects (of the specific types supported by the corresponding plug-in 103, 104, or 105) through the functionality set method to have the plug-in choose the reconciled value.

Any number of administration operations can cause the need for reconciliation to take place. Each specific situation that requires reconciliation can be represented by a reconciliation event. In other words, reconciliation events may be caused or generated by a number of administration operations, which can include:

- an object has been assigned to a user, group, service, server, or domain or unassigned from a user, group, service, server, or domain;
- 10 a user or group has been added as a member of a group or removed as a member from a group;
- a user has been added to a service or removed from a service;
- a user has been added to a server or removed from a server;
- a user has been added to a domain or removed from a domain;
- 15 a ranking of a set of reconciliation objects has changed;
- a reconciliation object has been deleted;
- a reconciliation object has been modified;
- a user has activated a device;
- a user has changed from one device to another;
- 20 a personal identification number (PIN) for a user's device has been set;
- a user has moved to a different server within mobility service 130; and
- a UE 150 software (operating system (OS) bundle) has changed.

Any number of rules may be used by a plug-in to reconcile a conflict based on the constraints of the administration service 100 and communications system 680, which may include the business logic and security model needed to administer the mobility service 130 and associated UE 150. The rules may be extensive and complex. Different groups of reconciliation rules may be applied depending on the category of the reconciliation object. Example sets of rules for various types of reconciliation objects are provided below for reconciliation objects of the following types: User/Group Attributes; IT

30 Policies; Applications; Application Control Policies; Device Software Bundles; and Native Application Settings.

User/Group Attribute Conflict Resolution Rule

(a) Whenever a reconciliation object is assigned directly to a user, any reconciliation objects (of the same type) assigned to groups of which the user is a member of are ignored.

IT Policy Conflict Resolution Rules

5 (a) Higher Ranked IT Policy Takes Precedence; and

(b) Each IT Policy in the domain is ranked against each of the other IT Policies in the domain. When multiple IT Policies are applied to a user, the highest ranked IT Policy is the reconciled IT Policy.

Application Conflict Resolution Rules

10 (a) Only applications that are supported by the user's current device (UE 150) and software are considered for reconciliation;

(b) Applications in a user assigned software configuration take precedence over a group assigned software configuration;

(c) An application with a higher version number takes precedence over a lower
15 version number;

(d) Reconciling an application requires first reconciling the ACP for the application; in this case the disposition of an ACP belonging to an application in a user attached software configuration takes precedence over a group attached software configuration; the disposition of required takes precedence over optional which takes
20 precedence over disallowed; the ACP settings are those from the highest ranked ACP that has the reconciled disposition;

(e) The delivery mode of a reconciled application is defined by a mode of push taking precedence over pull after considering a user attached software configuration taking precedence over a group attached software configuration;

25 (f) The delivery transport of a reconciled application is defined by a transport of wireless and wired if both are specified after considering a user attached software configuration taking precedence over a group attached software configuration;

(g) Wireless delivery transport is only reconciled if the user's device supports wireless application loading;

30 (h) Applications are only reconciled if all of the application and module dependencies are resolved by reconciled applications;

(i) Applications are only reconciled if there are no conflicting modules already present on the user's device (optional rule);

(j) Applications are only reconciled for wireless delivery transport if there are no circular dependencies; and

(k) Applications are only reconciled if they will fit into the device memory; memory checking is performed for applications in the following order (stopping when memory is full): applications with a disposition of required, a transport delivery of wireless, and a transport mode of push; applications with a disposition of required, a transport delivery wired, a transport mode of push; applications with a disposition of optional, a transport delivery wireless, and a transport mode of push; applications with a disposition of optional, a transport delivery wired, and a transport mode of push; and applications with a disposition of optional, a transport delivery wireless, and a transport mode of pull.

Application Control Policy (ACP) For Unlisted Applications Conflict Resolution Rules

(a) The disposition of an ACP for unlisted applications in a user attached software configuration takes precedence over a group attached software configuration;

(b) A disposition of disallowed takes precedence over optional; and

(c) The ACP settings are those from the highest ranked ACP that has the reconciled disposition.

Device Software Bundle Conflict Resolution Rules

(a) Device software bundles in a user attached software configuration takes precedence over a group attached software configuration; and

(b) The highest ranked device software bundle that is supported by the user's device and carrier is reconciled provided it has a higher rank than the device software bundle that is already installed on the user's device.

Native Application Setting Conflict Resolution Rules

(a) Native application settings are reconciled on a setting by setting basis;

(b) Native application settings belonging to a device software configuration in a user attached software configuration takes precedence over a group attached software configuration;

(c) For the "calendar initial view" setting the reconciled value is always the minimum number of days specified; values ordered from min to max are: day, week, month, agenda, last;

(d) For the "calendar keep appointments" setting the reconciled value is always the maximum number of days;

(e) For the "email confirm delete" setting the reconciled value of yes takes precedence over no;

(f) For the "email hide sent messages" setting the reconciled value of no takes precedence over yes;

5 (g) For the "email save copy in sent folder" setting the reconciled value of yes takes precedence over no;

(h) For the "address book sort by" setting the reconciled value of "first name" takes precedence over "last name" which takes precedence over "company"; and

(i) Each of the above settings also have an attribute which defines if the setting is
10 visible or not on the user's device and whether or not the user can change the value of the setting; the precedence order of the attribute for a setting is: locked and visible takes precedence over locked and hidden which takes precedence over unlocked and visible.

The reconciliation rules listed above are merely examples of reconciliation rules for a particular example embodiment of an administration service 100 for use in
15 conjunction with an example communications system 680 and various associated requirements and constraints. Additional or varied rules may be needed for the listed categories of reconciliation objects and additional rules likely would be needed for any categories of reconciliation objects that might be added through reconciliation functionally sets that may be added. One of skill in the art will appreciate how these rules
20 may be modified for use with an administration service 100 subject to different requirements and constraints.

FIG. 3 is a flow diagram depicting a method of reconciliation that resolves the conflicts using the reconciliation objects that are triggered by the reconciliation events that can be used by administration service 100. The steps or operations described herein
25 are examples. There may be many variations to these steps or operations without departing from the scope of this disclosure. For instance, where appropriate, the steps may be performed in a differing order, or steps may be added, deleted, or modified.

At step 301, administration service 100 gets a reconciliation event that was triggered. At step 302, based on the reconciliation event, administration service 100 can
30 determine a scope of change. The scope of change can have the following dimensions: users who are affected by the change; and reconciliation object types that are affected by the change.

At step 303, administration service 100 can reconcile each element of change. In other words, for each reconciliation object for a specific user, administration service 100

can determine the reconciled view of that object type, wherein the reconciled view is determined by the set of rules related to the object. The rules required to reconcile a set of objects are implemented by a plug-in (such as 103, 104 or 105) and performed by the plug-in for the plug-in server infrastructure 102 using a reconciliation functionality set.

5 At step 304, based on the reconciled view of each object type for each user, administration service 100 can prepare for the delivery of the reconciled object. Preparing for the delivery of the reconciled object for a user can result in a series of job tasks (all belonging to a single job for the reconciliation event) being created to perform the actual delivery. The tasks include appropriate dependencies when ordering of the
10 deliveries is important. Jobs and tasks are described below.

 At step 305, after processing for the reconciliation event is complete, any post-reconciliation processing can be performed. Some events can require post-reconciliation processing. This processing is done through the reconciliation functionality set. An example of post-reconciliation processing is the physical delete of a logically deleted
15 reconciliation object. The detailed reconciliation process can involve several queues to support scalable asynchronous processing of reconciliation events.

JOB MANAGEMENT

 A plug-in such as 103, 104, and 105 or administration service 100 can use jobs
20 and tasks to accomplish their functionality (e.g., assigning attributes to a user). A plug-in 103, 104, or 105 or the administration service 100 itself can create any number of jobs within the administration service 100 and any plug-in 103, 104, or 105 or administration service 100 can create any number of tasks within any of the jobs within administration service 100. Tasks can operate to install or remove software (i.e., via an attribute for a
25 software configuration) onto or from UEs 150. Several types of tasks are defined, including INSTALL, UNINSTALL, UPGRADE, and DOWNGRADE task types that correspond to installing, uninstalling, upgrading, and downgrading software on UEs 150.

 FIG. 4 is a flow diagram depicting an example method of creating and executing a job, effectively showing the life cycle of a job. The steps or operations described herein
30 are examples. There may be many variations to these steps or operations without departing from the scope of this disclosure. For instance, where appropriate, the steps may be performed in a differing order, or steps may be added, deleted, or modified.

 At step 401 a plug-in, such as 103, 104, or 105 or administration service 100 can create a job that is open for tasks to be added. At step 402, the plug-in 103, 104, or 105,

or the administration service 100 can create tasks are created one at a time and add the tasks to any open job. During the creation, each task receives a number associated with the time the task was created such that tasks with lower numbers are can be identified as being created before tasks with higher numbers. One task can represent the delivery or
5 removal of a reconciled view of an object type to or from a UE 150. At step 403, the plug-in 103, 104, or 105, or the administration service 100 can close the job is so that further tasks cannot be added and so that the job can be prepared for execution. A number of operations are carried out during job closure, which is described in more detail below. At step 404, administration service 100 can add default job scheduling and
10 throttling parameters to the job. These job scheduling and throttling parameters can be changed at any by the plug-in 103, 104, or 105, or administration service 100. At step 405, administration service 100 can make the job ready for execution. At step 406, administration service 100 can execute the tasks of the job, possibly with multiple automatic retries when certain failure conditions occur. At step 407, when all tasks of the
15 job have completed (successfully or with failure), administration service 100 can mark the job as completed.

The job closure process makes a job ready to be executed. During a job close process, the job is optimized according to predefined optimization rules to improve system performance. Initially all tasks within the job being closed are optimized with
20 respect to just those tasks in the job being closed. An example of optimization rules for this case can include:

When a lower number INSTALL task (T1) already exists, the higher number INSTALL task (T2) is optimized out by task T1.

When a lower number INSTALL task (T1) already exists, the higher number
25 UNINSTALL task (T2) is optimized out by task T1.

When a lower number INSTALL task (T1) already exists, the higher number UPGRADE task (T2) is optimized out by task T1.

When a lower number INSTALL task (T1) already exists, the higher number DOWNGRADE task (T2) is optimized out by task T1.

30 When a lower number UNINSTALL task (T1) already exists, the higher number INSTALL task (T2) is optimized out by task T1.

When a lower number UNINSTALL task (T1) already exists, the higher number UNINSTALL task (T2) is optimized out by task T1.

When a lower number UNINSTALL task (T1) already exists, the higher number UPGRADE task (T2) is optimized out by task T1.

When a lower number UNINSTALL task (T1) already exists, the higher number DOWNGRADE task (T2) is optimized out by task T1.

- 5 When a lower number UPGRADE task (T1) already exists, the higher number UNINSTALL task (T2) is optimized out by task T1.

When a lower number UPGRADE task (T1) already exists, the higher number UPGRADE task (T2) is optimized out by task T1.

- 10 When a lower number UPGRADE task (T1) already exists, the higher number DOWNGRADE task (T2) is optimized out by task T1.

When a lower number DOWNGRADE task (T1) already exists, the higher number UNINSTALL task (T2) is optimized out by task T1.

When a lower number DOWNGRADE task (T1) already exists, the higher number UPGRADE task (T2) is optimized out by task T1.

- 15 When a lower number DOWNGRADE task (T1) already exists, the higher number DOWNGRADE task (T2) is optimized out by task T1.

The above-listed rules may be generalized as follows: INSTALL and UNINSTALL tasks with lower numbers optimize out INSTALL, UNINSTALL, UPGRADE, and DOWNGRADE tasks with higher numbers. Also, UPGRADE and DOWNGRADE tasks with lower numbers optimize out UNINSTALL, UPGRADE, and DOWNGRADE tasks with higher numbers.

Additionally, all tasks within the job being closed can be optimized with respect to all tasks in all other jobs. The optimization rules in this case are as follows:

- 25 When an INSTALL task (T1) already exists and a new INSTALL task (T2) is being closed, task T1 is optimized out by task T2.

When an INSTALL task (T1) already exists and a new UNINSTALL task (T2) is being closed, task T1 is optimized out by task T2.

When an UNINSTALL task (T1) already exists and a new INSTALL task (T2) is being closed, task T1 is optimized out by task T2.

- 30 When an UNINSTALL task (T1) already exists and a new UNINSTALL task (T2) is being closed, task T1 is optimized out by task T2.

When an UPGRADE task (T1) already exists and a new INSTALL task (T2) is being closed, task T1 is optimized out by task T2.

When an UPGRADE task (T1) already exists and a new UNINSTALL task (T2) is being closed, task T1 is optimized out by task T2.

When an UPGRADE task (T1) already exists and a new UPGRADE task (T2) is being closed, task T1 is optimized out by task T2.

- 5 When an UPGRADE task (T1) already exists and a new DOWNGRADE task (T2) is being closed, task T1 is optimized out by task T2.

When a DOWNGRADE task (T1) already exists and a new INSTALL task (T2) is being closed, task T1 is optimized out by task T2.

- 10 When a DOWNGRADE task (T1) already exists and a new UNINSTALL task (T2) is being closed, task T1 is optimized out by task T2.

When a DOWNGRADE task (T1) already exists and a new UPGRADE task (T2) is being closed, task T1 is optimized out by task T2.

When a DOWNGRADE task (T1) already exists and a new DOWNGRADE task (T2) is being closed, task T1 is optimized out by task T2.

- 15 The above-listed rules may be generalized as follows: Existing INSTALL tasks can be optimized out by new INSTALL or UNINSTALL tasks that are being closed. Existing UNINSTALL tasks can be optimized out by new INSTALL or UNINSTALL tasks that are being closed. Existing UPGRADE and DOWNGRADE tasks can be optimized out by new INSTALL, UNINSTALL, UPGRADE, or DOWNGRADE tasks
20 that are being closed.

Additionally or alternatively, a task can be added to a job that has already been closed and optimized. The job can be briefly reopened and then closed to add the new task. In this case, the optimization process can be skipped so that the job will not be optimized with respect to the new task.

- 25 A job also takes on scheduling and throttling parameters that can be changed at any time until the job is complete. Each object type can have its own set of scheduling and throttling parameters that apply to all of a job's tasks that pertain to the specific object type. Scheduling parameters for job tasks can schedule the execution of the tasks to occur at any of one or more times on any of one or more days of a week. Additionally,
30 a scheduling parameter can schedule the execution of a job to begin at some future date. Job throttling parameters, which can quantify a number of job slots, can limit the execution of a task of a job and can be specified on a job basis by the number of tasks for a specific object type executing simultaneously and/or by the number of tasks executing in a scheduling window by limiting the number of job slots available to a job. A

scheduling window is a contiguous period of time during a particular day during which a task may execute as defined by its scheduling parameters. System throttling parameters, which can quantify a number of system slots, can also be specified to limit the total number of tasks executing simultaneously across all jobs of the administration service

5 100 for a specific object type by limiting the number of system slots available to the administration service 100.

Additionally, a specific object type can be defined such that only one task may be executing at any instant of time that is related to a particular user of mobility service 130, regardless of the particular instance of the object. In such a case, a running dependency is
10 created among all the tasks for all jobs that pertain to that specific object type for a user. This effectively single-threads this type of reconciliation object type for a user. The running dependencies between all such tasks are adjusted dynamically as required as the various tasks execute.

FIG. 5A and FIG. 5B are a flow diagram illustrating an example method 500 for
15 executing tasks and processing a finished task. The steps or operations described herein are examples. There may be many variations to these steps or operations without departing from the scope of this disclosure. For instance, where appropriate, the steps may be performed in a differing order, or steps may be added, deleted, or modified.

When a job timer wakes up, the system will execute ready tasks from jobs based
20 on their priority of execution. System throttling and job throttling parameters determine how many tasks can be executed at the same time. A task can only be executed if it has been granted a job slot (for job throttling) and a system slot (for system throttling). If a system slot is not available to execute a task, then information is recorded regarding what kind of task is waiting for a system slot in order to be executed. Once both a job slot and
25 a system slot become available, the job manager will pick a task to execute based on its priority among the pending tasks.

The execution of a task can involve delivering an object to the UE 150. The details of how to do the delivery can be known by the plug-in that supports the delivery functionality set for that particular reconciliation object type. Hence delivery is initiated
30 by the job manager of the administration service 100, even though it is carried out via a plug-in that is registered the implements the delivery functionality set for the object type.

When a task is completed successfully, then tasks that depend on the successfully completed task can be marked ready to deliver. If there are ready tasks due to the completed task, then the job manager will deliver these ready tasks subject to the system

and job throttling parameters at that time. If there are no dependent tasks, the job manager will try to find another ready task in the job for next execution. When a task is not completed successfully, the job manager can mark the dependent tasks of the unsuccessfully completed task as failed and find a ready task in the job that can be
5 executed next.

A particular UE 150 is serviced via a specific mobility server of mobility service 130. Tasks are executed such that the load of task execution is balanced across the various servers of mobility service 130.

Additionally, in the case of a job task that has not completed within a system
10 allowed time threshold, the job manager will set the task to be timed out, so that the job and system slots can be used to execute another task.

In step 510, a portion of method 500 for executing tasks begins responsive to a "wake-up" expiration of a job timer. In step 512, the administration service 100 finds the highest priority job associated with the timer of which the processing window is open. In
15 step 514, if no job is found, the method exits. In step 516, the administration service 100 obtains the number of ready tasks N1 in the job (or the predefined batch size, whichever is smaller). In step 518, if no ready tasks are found, the method exits.

In step 520, the administration service 100 requests a number N1 job slots for the job. The result of the request is a number N2 of job slots granted. In step 522, a check is
20 made to determine if system slots were granted. If so execution continues in step 524, in which the system slots pending count is decremented by the number N2 of granted job slots, and execution continues in step 534. Otherwise, execution continues in step 526. In step 526 (no system slots granted), the administration service 100 requests a number of system slots equal to the number N2 of granted job slots. The result of the request is a
25 number N3 of system slots granted.

In step 528, a check is made to determine if the number N2 of granted job slots is equal to the number N3 of granted system slots. If so, execution continues with step 534. Otherwise, execution continues with step 530, in which the job system slots pending count is incremented by the difference between the number N2 of granted job slots and
30 the number N3 of granted system slots (i.e., $N2 - N3$). In step 532, in which a check is made to determine if the number N3 of granted system slots is zero (meaning that no system slots were granted). If no system slots were granted, no task can be executed, and the method exits. However, in step 534, if system slots were granted earlier in one of steps 522 or 530, then a number of tasks equal to the number N3 of granted system slots

are delivered. Generally, "delivery" means that an object or other content is delivered to the UE 150.

A portion of method 550 for processing a finished task (which is a member, for example, of a job F), begins in step 550, when the work of the task is complete. In step
5 552, the administration service 100 finds each task that depends on the successfully "finished" task and if such task has no other unmet dependencies, administration service 100 marks such task as ready for execution. Such tasks may be grouped as a task set D.

In step 554, a check is made to determine if any such tasks have been found. If so, execution continues in step 556, in which each of the tasks of task set D are selected
10 for execution, and then in steps 520, etc. Otherwise, execution continues in step 558, in which the administration service 100 finds the highest priority job pending for system slots which has one job slot available and its processing window open (meaning that the job has the same object type as that of recently-finished task).

In step 560, a check is made to determine if an eligible job has been found. If so,
15 execution continues in step 562 in which the job slot is returned to the pool of job slots, in step 564 in which the task is prepared for execution, and then in steps 520, etc. If no eligible job was found in step 560, then execution continues in step 566, to find a ready task of job F of the same object type as that of the recently-finished task.

In step 568, a check is made to determine if a ready task was found. If so, the
20 task, which reuses recently-finished task job and system slots, is executed in step 534, etc. Otherwise, execution continues in step 570, in which the job and system slots reserved for the tasks of this job are returned to their respective pools. In step 572, the administration service 100 finds a ready task, which may be from a different job. If a ready task is found, then the number N1 of requested job slots is set to 1, and execution continues at
25 steps 520, etc. If no ready task is found, the method exits at step 576.

Embodiments described herein may be realized as a computer program product, comprising a computer readable medium, computer instructions stored on the computer readable medium, the computer instructions being executable by one or more processors in a mobile communications device for implementing any of the methods described in
30 this application.

Although example embodiments of this disclosure have been depicted and described in detail herein, it will be apparent to those skilled in the relevant art that various modifications, additions, substitutions, and the like can be made without departing from the concepts and embodiments disclosed herein, and these are therefore

considered to be within the scope of the present subject matter as defined in the following claims.

CLAIMS

What is claimed is:

1. A method of an administration server comprising:
assigning a first attribute to one or more objects by a plug-in of an
5 administration service of the administration server;
wherein the plug-in implements a method of a functionality set, the method
being callable by the administration service to perform the assigning.
2. The method of claim 1 wherein the assigning is in response to the first attribute
10 being assigned to one of a group, a server, a service, and a domain.
3. The method of claim 1, further comprising:
displaying the first attribute on a graphical user interface (GUI) of the
administration service at an integration point defined by the administration
15 service and identified by the plug-in.
4. The method of claim 1, wherein the first attribute is one of a policy, a software
configuration, a required application, an optional application, a disallowed
application, a device software configuration, a device software bundle, a native
20 application setting, or an application setting.
5. The method of claim 2, wherein the one or more objects inherit a second attribute
from the one of the group, the server, the service, and the domain.
- 25 6. The method of claim 5, further comprising reconciling the assignment of the first
attribute with the assignment of the second attribute when the first attribute
conflicts with the second attribute.
7. The method of claim 5, further comprising reducing a number of duplicate tasks
30 that assign the first attribute and the second attribute to the one or more objects when
the first attribute is similar to the second attribute and is assigned to a same one of the
one or more objects.

8. A method of an administration server comprising:
- 5 triggering a reconciliation event by changing an assignment of an attribute of one or more users that each comprise one or more objects of one or more plug-ins;
- determining a scope of the one or more users affected by changing the assignment;
- determining one or more objects of the one or more plug-ins that are affected by changing the assignment;
- 10 reconciling the assignment with any conflicting assignments via one or more reconciliation objects, the conflicting assignments determined from one of the scope and the one or more objects that are affected.
9. The method of claim 8, wherein changing the assignment of the attribute is via
- 15 changing the association of a user to at least one of a group, service, server, or domain.
10. The method of claim 8, wherein the changing the assignment of the attribute is via directly changing the attribute of at least one of a user, group, service, server, and
- 20 domain.
11. The method of claim 8, wherein the attribute is one of a policy, a software configuration, a required application, an optional application, a disallowed application, a device software configuration, a device software bundle, a native
- 25 application setting, or an application setting.
12. The method of claim 8, wherein the triggering a reconciliation event is caused by changing a ranking of the one or more reconciliation objects.
- 30 13. The method of claim 8, wherein one of the one or more reconciliation objects is one of deleted and modified.
14. The method of claim 8, wherein the triggering a reconciliation event is caused by one of activating a first wireless mobile device, changing from the first wireless

mobile device to a second wireless mobile device, setting a personal identification number (PIN) for the wireless mobile device, moving to a different server of a mobility service, and changing an operating system (OS) bundle of the first wireless mobile device

5

15. A method of an administration server comprising:

assigning an attribute to an object of one or more users via an administration service;

10 adding one or more tasks by one or more plug-ins to a job created by one of the one or more plug-ins, the one or more tasks performing the assigning;

removing one or more of the one or more tasks from the job to optimize the job based on a set of rules;

performing the assigning by the plug-in.

15 16. The method of claim 15, further comprising:

closing the job before the removing;

reopening the job after the removing;

adding at least one additional task to the job;

closing the job after the adding.

20

17. The method of claim 15, wherein the attribute is one of a policy, a software configuration, a required application, an optional application, a disallowed application, a device software configuration, a device software bundle, a native application setting, or an application setting.

25

18. The method of claim 15, wherein the job and the one or more tasks each comprise a scheduling parameter and a throttling parameter, the scheduling parameter schedules an execution date, and the throttling parameter quantifies a number of job slots used to limit the execution of a task of a job.

30

19. The method of claim 18, wherein a plurality of system throttling parameter and the job throttling parameters of the job and the one or more tasks determine a number of tasks that can be executed at a same time.

20. The method of claim 19, wherein each task can only be executed when it has been granted a system slot and a job slot, the plurality of system parameters quantify a number of system slots to limit a total number of tasks executing simultaneously all jobs of the administration service, and the job throttling parameters quantify a number of job slots to limit the execution of a task of a job.
21. The method as claimed in any of claims 1 to 8 and 15 to 20, further comprising:
triggering a reconciliation event by changing an assignment of the first attribute of one or more users comprising the one or more objects of one or more plug-ins;
determining a scope of the one or more users affected by changing the assignment;
determining one or more objects of the one or more plug-ins that are affected by changing the assignment;
reconciling the assignment with any conflicting assignments via one or more reconciliation objects, the conflicting assignments determined from one of the scope and the one or more objects determined to be affected.
22. The method as claimed in any of claims 1 to 8 and 15 to 21, wherein changing the assignment of the first attribute is via changing the association of a user to at least one of a group, service, server, or domain.
23. The method as claimed in any of claims 1 to 8 and 15 to 22, wherein the changing the assignment of the first attribute is via directly changing the first of at least one of a user, group, service, server, and domain.
24. The method as claimed in any of claims 1 to 8 and 15 to 23, wherein the triggering a reconciliation event is caused by changing a ranking of the one or more reconciliation objects.
25. The method as claimed in any of claims 1 to 8 and 15 to 24, wherein one of the one or more reconciliation objects is one of deleted and modified.

26. The method as claimed in any of claims 1 to 8 and 15 to 25, wherein the triggering a reconciliation event is caused by one of activating a first wireless mobile device, changing from the first wireless mobile device to a second wireless mobile device, setting a personal identification number (PIN) for the wireless mobile device,
5 moving to a different server of a mobility service, and changing an operating system (OS) bundle of the first wireless mobile device
27. The method as claimed in any of claims 1 to 14, further comprising:
adding one or more tasks by one or more plug-ins to a job created by one of
10 the one or more plug-ins, the one or more tasks performing the assigning;
removing one or more of the one or more tasks from the job to optimize the job based on a set of rules;
performing the assigning by the plug-in.
28. The method as claimed in any of claims 1 to 14 and 27, further comprising:
closing the job before the removing;
reopening the job after the removing;
adding at least one additional task to the job;
closing the job after the adding.
20
29. The method as claimed in any of claims 1 to 14 and 27 to 28, wherein the job and the one or more tasks each comprise a scheduling parameter and a throttling parameter, the scheduling parameter schedules an execution date, and the throttling parameter quantifies a number of job slots used to limit the execution of a task of a
25 job.
30. The method as claimed in claim 29, wherein a plurality of system throttling parameter and the job throttling parameters of the job and the one or more tasks determine a number of tasks that can be executed at a same time.
31. The method as claimed in claim 30, wherein each task can only be executed when it has been granted a system slot and a job slot, the plurality of system parameters quantify a number of system slots to limit a total number of tasks executing
30

simultaneously all jobs of the administration service, and the job throttling parameters quantify a number of job slots to limit the execution of a task of a job.

32. A server configured to perform the method of any of claims 1 to 31.

5

33. A computer readable medium storing a set of instructions configured to perform the steps of any of claims 1 to 31.

1/16

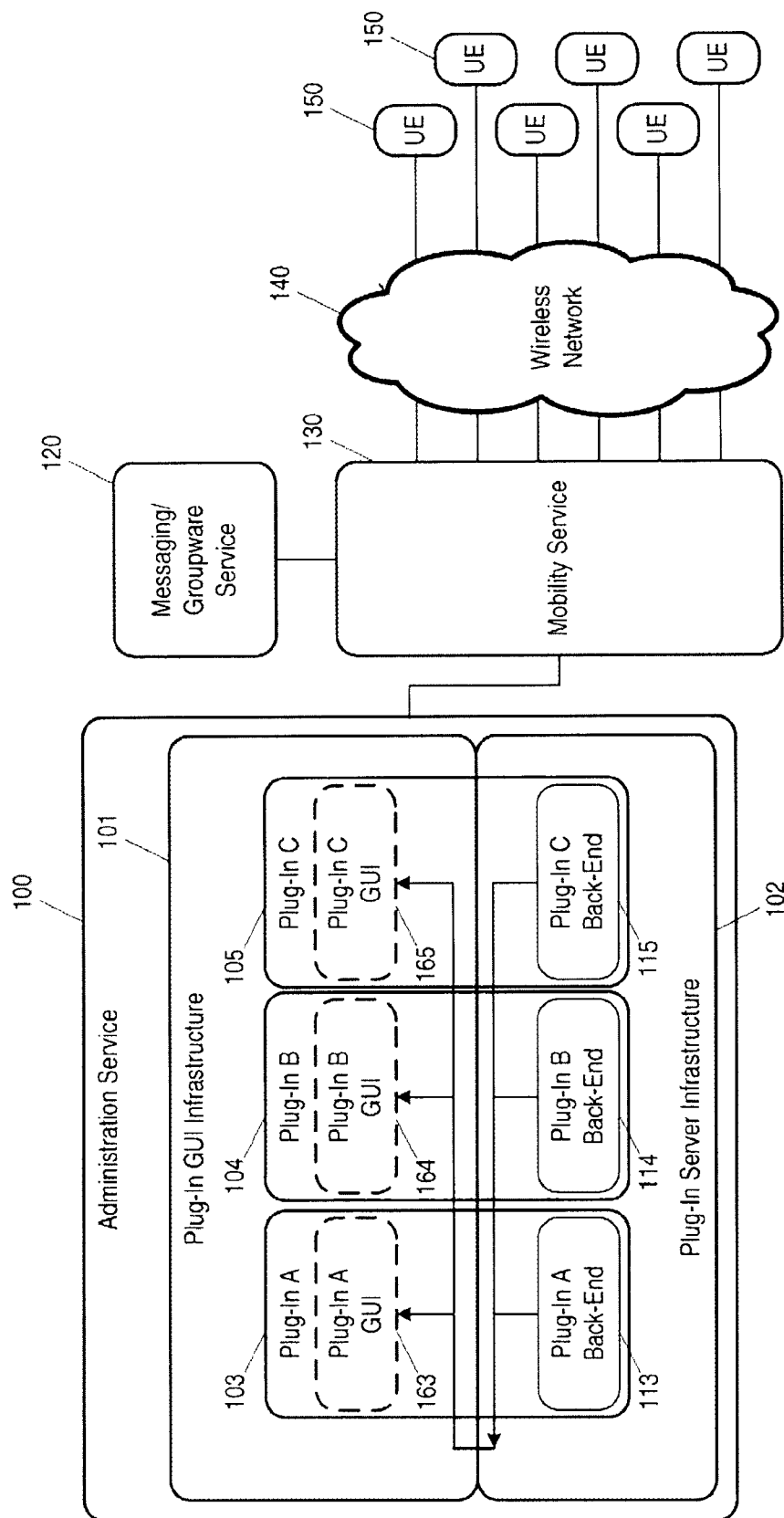


FIG. 1

2/16

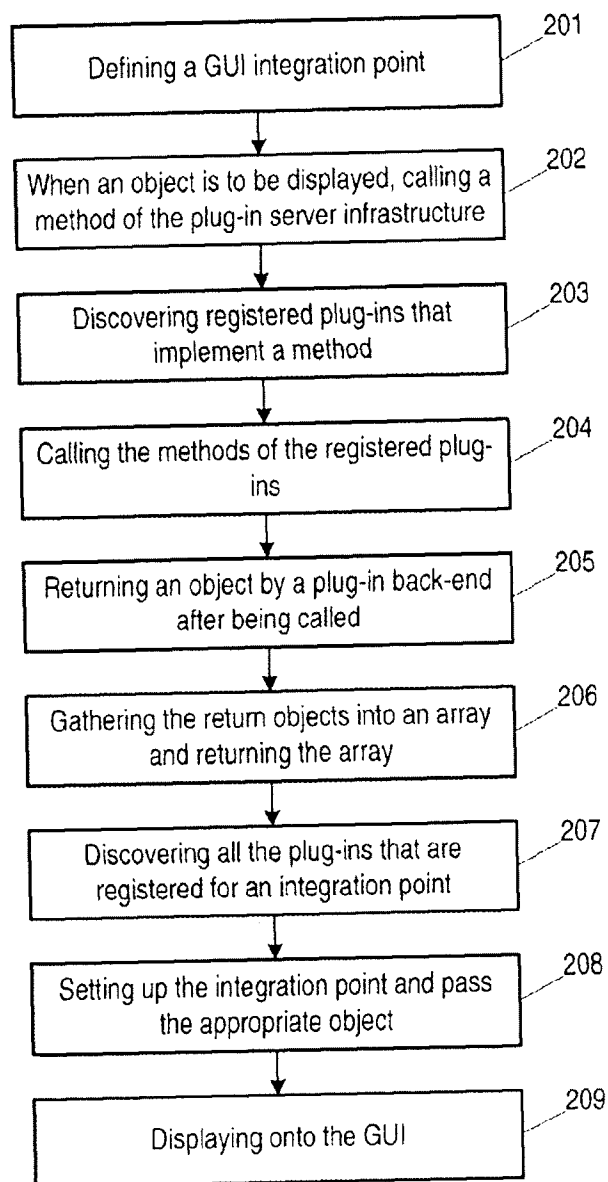


FIG. 2

3/16

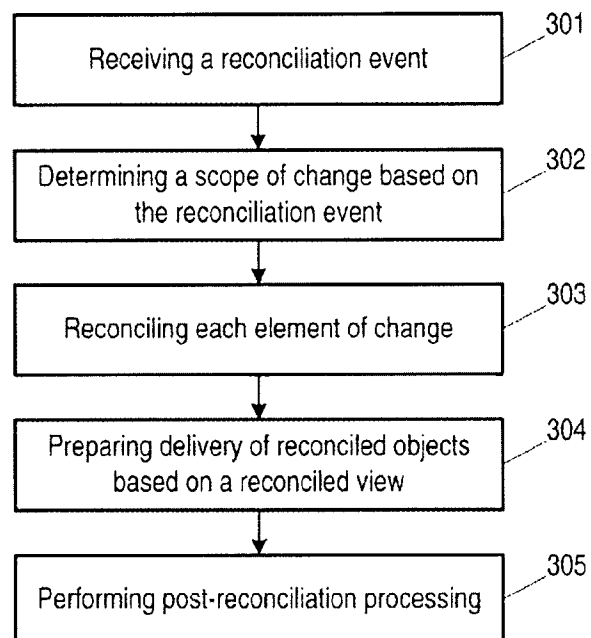


FIG. 3

4/16

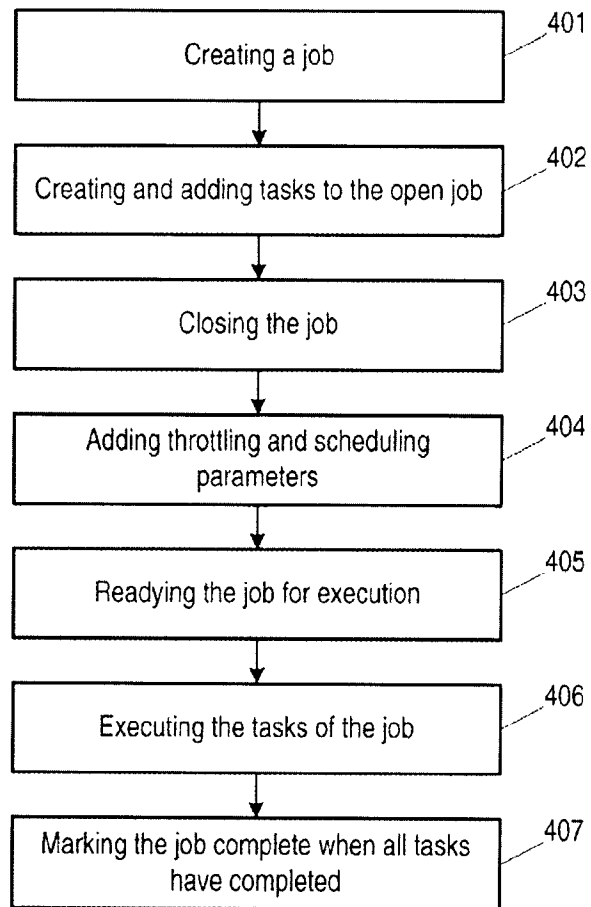
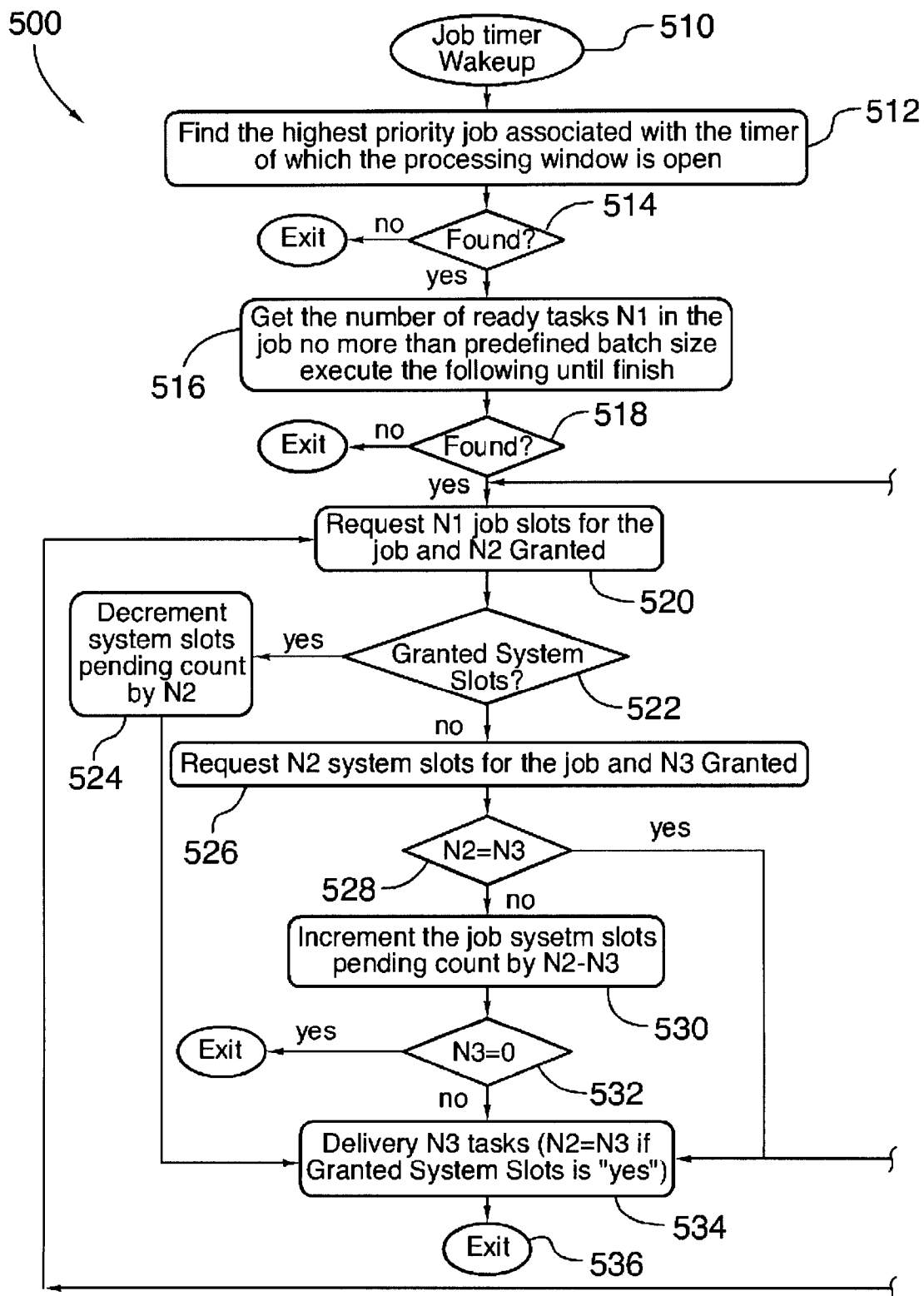


FIG. 4

5/16



6/16

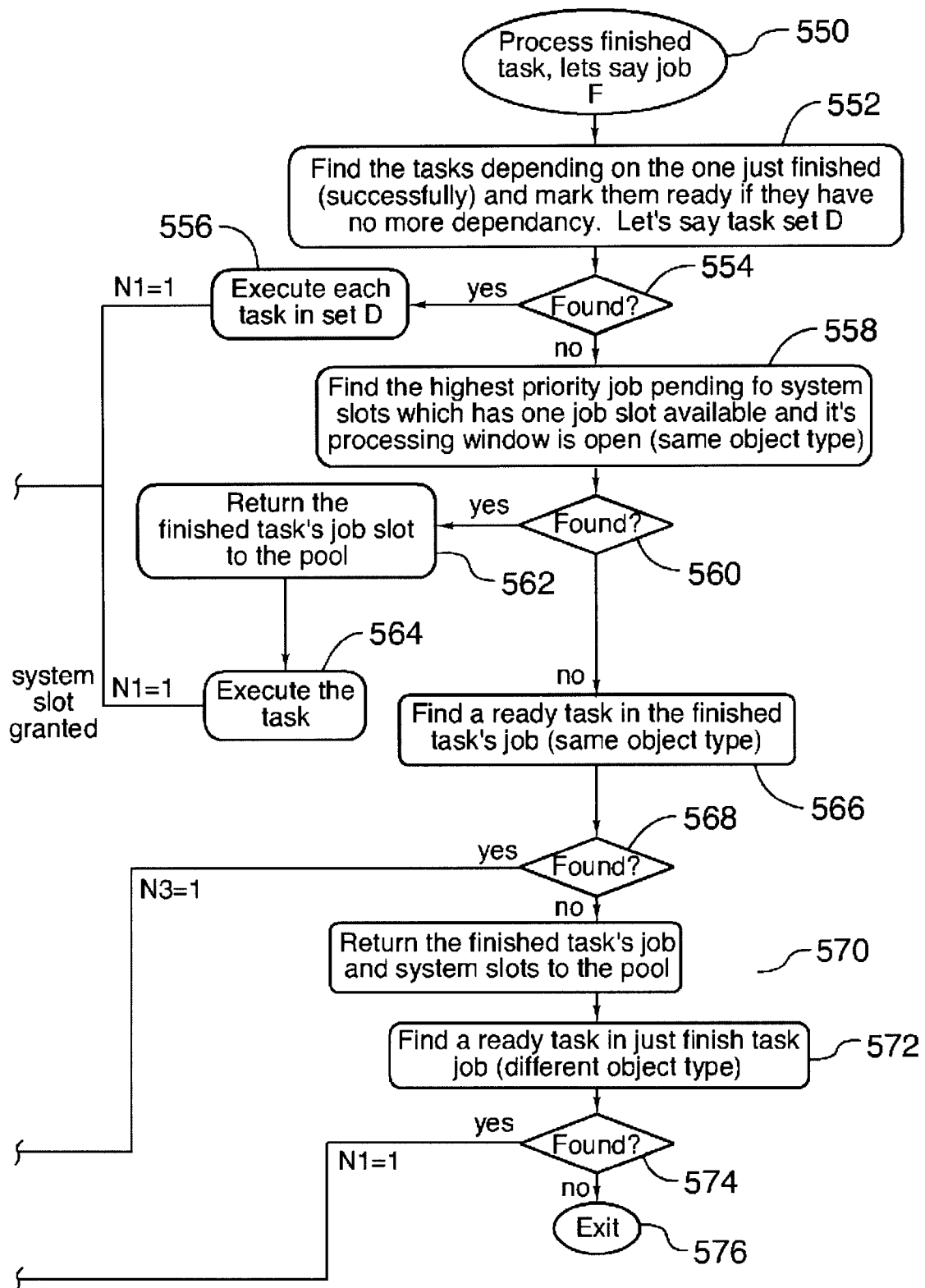


FIG.5B

7/16

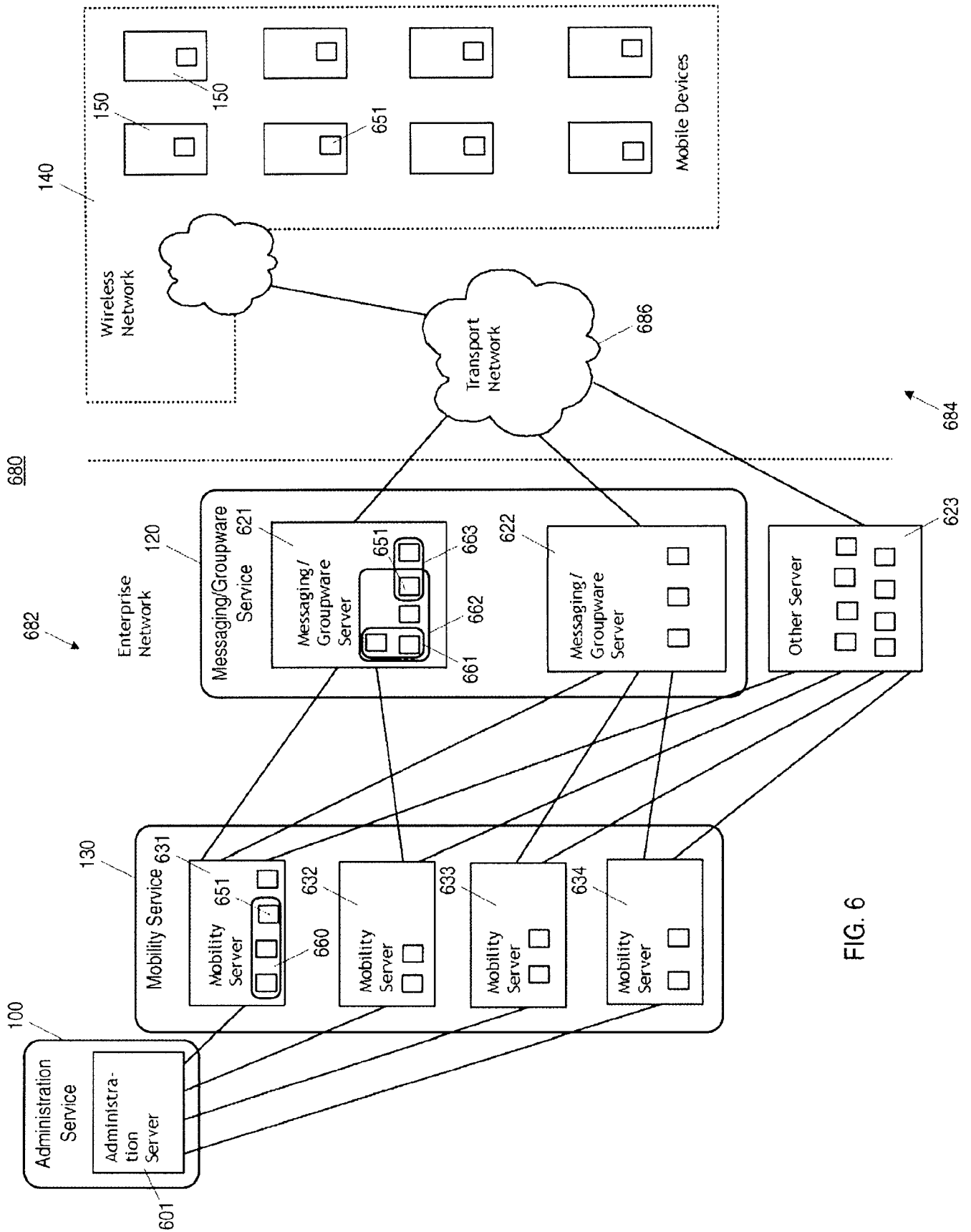


FIG. 6

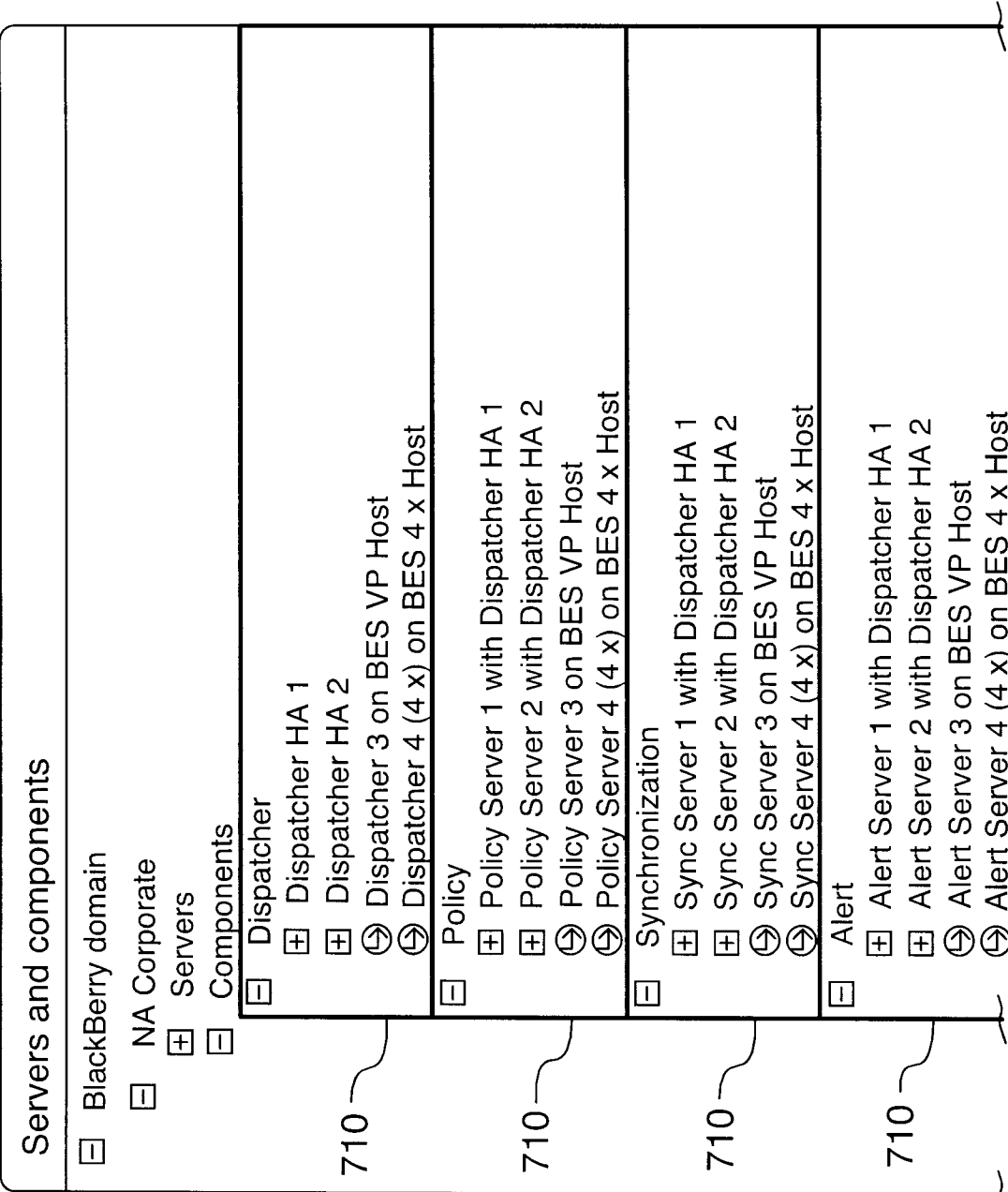


FIG.7A

9/16

710	☐ Email (Exchange) ☒ Exchange 1 with Dispatcher HA 1 ☒ Exchange 2 with Dispatcher HA 2 ☒ Exchange 3 on BES VP Host ☒ Exchange 4 (4 x) on BES 4 x Host
710	☒ Email (Domino) ☒ Email (GroupWise)
710	☐ Attachment ☒ Connector ☒ Server
710	☐ Collaboration ☒ Same Time Instant Messaging ☒ GroupWise Instant Messaging ☒ Office Instant Messaging
710	☐ MDS-IS ☒ IS 5.0 (BMS-02,rim.net) ☒ MDS-IS 2 Node ☒ IS VP on BEX VP Host ☒ IS Remote on MDS Server G
710	☐ MDS-CS ☒ MDS-CS (BMS-02,rim.net) on MDS Server ☒ CS 2 on BES VP Host ☒ CS 4 x on MDS Server G
710	☐ BAS ☒ BAS System

FIG.7B

10/16

800

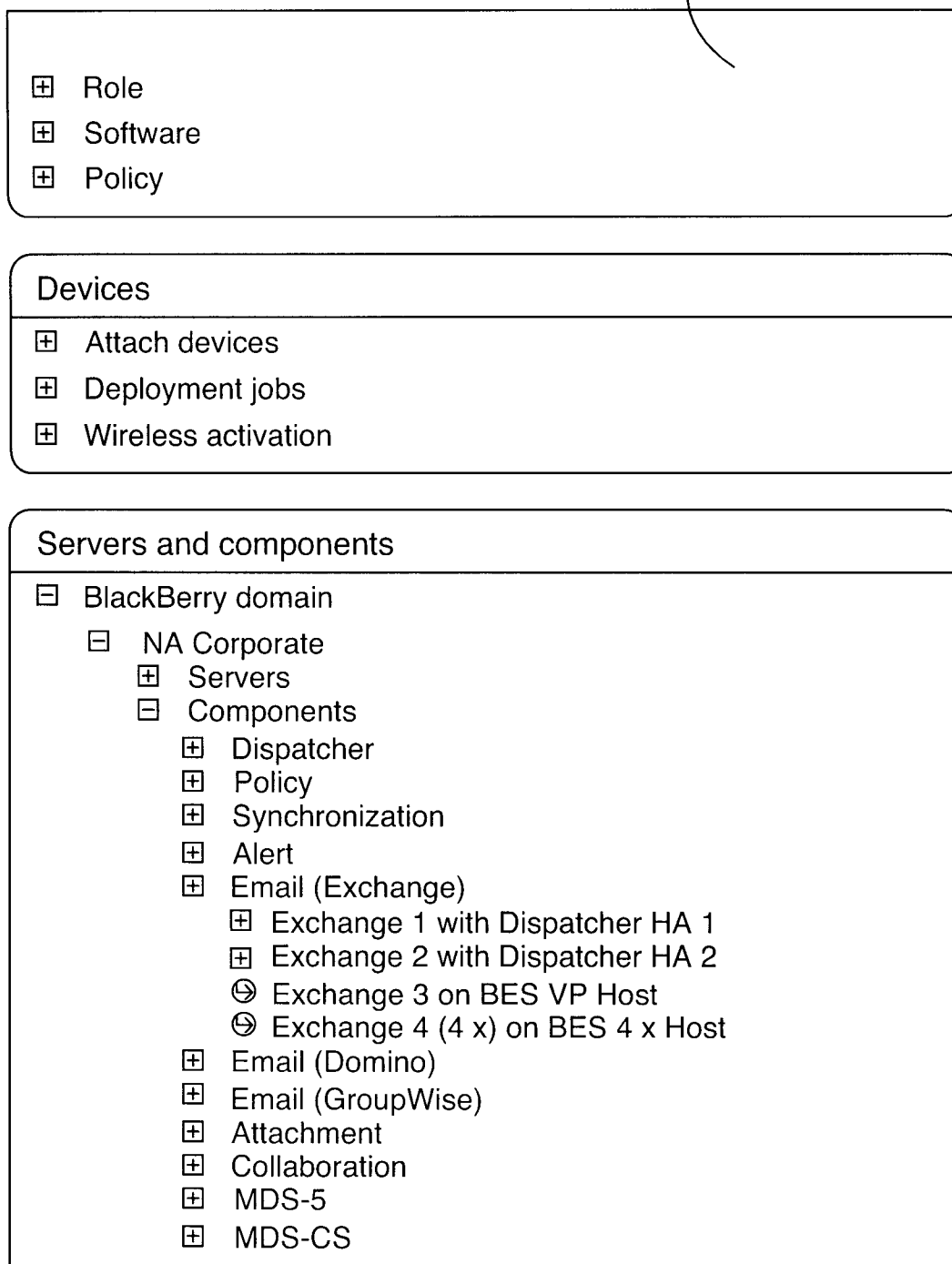


FIG.8A

810

Describe Host Email Instance

Host Instance Information	Messaging	Email message filters	Extension Plug-ins
---------------------------	-----------	-----------------------	--------------------

Host Instance Information

Host Service name: RD_EMAIL_1

Friendly Description: ServiceEmail (Exchange)

Instance statistics

Forwarded Messages: 1

Pending Data Packets: 0

Expired Messages: 6

General

BlackBerry Email Services Version: 5.0.0.0

Supported Email (Exchange) Instances name Desc

Exchange 1A

Exchange 1B

Edit host instance

Clear statistics (disp msg. "cleared")

View services list

FIG.8B

900

BlackBerry Domains > NA Corporate > Components > Dispatcher Category > Dispatcher HA 1...

Instances

Describe Dispatcher...

Instance information	Supported Samesite Instant Messaging Instances
Supported Office Instant Messaging Instances	Supported MDS-CS Instances
Name	
MDS-CS (BMS-02.rim.net)	
CS 2	
Description	

Manual Failover

Set Primary to Standby

Edit instance

View services list

FIG.9A

...> BES Main Host > View Instance Dispatcher 1A

Supported GroupWise Instant Messaging Instances	
910	
Server Name	Availability State
MDS Server	Primary
BES VP Host	Standby

FIG.9B

14/16

User Management > Manage Users

1000

Manage Users

Describe Manage users

Search Users	
1010	User Criteria
	Display Name:
	First Name:
	User Name:
	Device Criteria
	PIN:
	IT Policy:
	Last Contact Date:
	Phone Number:
	Software Configuration Criteria
OS Bundle:	
Configuration Set:	
Group Criteria	
Specific Group:	
BlackBerry Enterprise Server Criteria	
BlackBerry Enterprise Server	
Email Criteria	
Email Address:	
State:	
Queued Removal of BES Services:	
Sort Criteria	
Sort by:	
 Basic:	

FIG.10A

15/16

	User ID:
	Last Name:
	Model:
v	IT Policy Status:
v	Carrier:
	Application:
None selected v	
None selected v	
v	
	Mail Server:
False v	
Display Name v	
☒ A to Z ☐ Z to A	

FIG.10B

User Management > Manage Users > View User
Manage users
Describe Manage users....

1100

User information	Groups	Roles	Software configurations	Policies	WLAN configurations
Component information	Associated device	Message settings	SIP Configuration		

Name
Display Name:
First Name:

Authentication Type
BlackBerry Administration Service

User Name

- ☒ Edit user
- ☐ Send message to user
- ☒ Back to search results
- BlackBerry Enterprise Server status**
- ☒ Switch BlackBerry user to different BES
- ☒ Disable as BlackBerry user

1110

User status

☒ Reload User

FIG.11

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CA2010/000644

A. CLASSIFICATION OF SUBJECT MATTER

IPC: **H04W 8/18** (2009.01)

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)

IPC: H04W 8/18 (2009.01)

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

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

Epoque, TotalPatent and Canadian Patent Database

Keywords used: administration; server; plug-in; attribute or policy; assign; conflict; reconciliation; resolve or resolution; job; task; optimize

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	US 5,797,128 A (BIRNBAUM) 18 August 1998 (18.08.1998) The whole document	1 to 7, 15 to 20, 27 to 31, 32 and 33 21 to 26
X Y	US 5,889,953 A (THEBAUT et al.) 30 March 1999 (30.03.1999) The whole document	8 to 14 21 to 26
A	US 2006/0195508 A1 (BERNARDIN et al.) 31 August 2006 (31.08.2006) The whole document	15 to 31
A	US 2005/0172015 A1 (RANA et al.) 4 August 2005 (04.08.2005) The whole document	8 to 14 and 21 to 26

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents :	"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
"A" document defining the general state of the art which is not considered to be of particular relevance	"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
"E" earlier application or patent but published on or after the international filing date	"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
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&" document member of the same patent family
"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	

Date of the actual completion of the international search

22 July 2010 (22-07-2010)

Date of mailing of the international search report

27 July 2010 (27-07-2010)

Name and mailing address of the ISA/CA
Canadian Intellectual Property Office
Place du Portage I, C114 - 1st Floor, Box PCT
50 Victoria Street
Gatineau, Quebec K1A 0C9
Facsimile No.: 001-819-953-2476

Authorized officer

Cathy Ma (819) 994-7510

INTERNATIONAL SEARCH REPORTInternational application No.
PCT/CA2010/000644**Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of the first sheet)**

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons :

1. ☐ Claim Nos. :
because they relate to subject matter not required to be searched by this Authority, namely :
2. ☐ Claim Nos. :
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically :
3. ☐ Claim Nos. :
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows :

Group A - Claims 1 to 7, 15 to 32 and 33 are directed to a method of administration server by assigning an attribute to one or more objects by a plug-in of the server, and a server and computer readable medium for implementing the method; and

Group B - Claims 8 to 14 are directed to a method of administration server by reconciling the conflict attributes.

Furthermore, even if the applicant may argue that the Group B also involves the assignment of attributes, this feature is not considered to be a linking inventive concept, since prior art US 5,797,128 A teaches this feature (abstract; column 6, lines 61 and 62).

1. ☒ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claim Nos. :
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claim Nos. :

Remark on Protest ☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☒ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CA2010/000644

Patent Document Cited in Search Report	Publication Date	Patent Family Member(s)	Publication Date
US5797128A	18-08-1998	DE69601149D1	28-01-1999
		DE69601149T2	05-08-1999
		EP0752652A2	08-01-1997
		EP0752652A3	22-01-1997
		EP0752652B1	16-12-1998
		JP9069077A	11-03-1997
US5889953A	30-03-1999	AU702607B2	25-02-1999
		AU719918B2	18-05-2000
		AU720871B2	15-06-2000
		AU2556997A	22-10-1997
		AU4702996A	21-08-1996
		AU5183796A	11-09-1996
		AU7374398A	08-12-1998
		EP0807349A1	19-11-1997
		EP0811284A1	10-12-1997
		EP0890240A2	13-01-1999
		EP0981877A1	01-03-2000
		US5706436A	06-01-1998
		US5832503A	03-11-1998
		US5872928A	16-02-1999
		US6209033B1	27-03-2001
		US6243747B1	05-06-2001
		US6381639B1	30-04-2002
		US6421719B1	16-07-2002
		US2001047409A1	29-11-2001
		WO9624211A1	08-08-1996
		WO9626588A1	29-08-1996
		WO9737477A2	09-10-1997
		WO9737477A3	12-02-1998
		WO9852322A1	19-11-1998
US2006195508A1	31-08-2006	AU6305601A	26-11-2001
		US6757730B1	29-06-2004
		US2003191795A1	09-10-2003
		US7093004B2	15-08-2006
		US2005021594A1	27-01-2005
		US7130891B2	31-10-2006
		US2002023117A1	21-02-2002
		US2003154284A1	14-08-2003
		US2008313345A1	18-12-2008
		WO0188708A2	22-11-2001
		WO0188708A3	07-08-2003
		WO02063479A1	15-08-2002
US2005172015A1	04-08-2005	CA2479573A1	09-10-2003
		DE60325341D1	29-01-2009
		EP1349316A1	01-10-2003
		EP1495580A1	12-01-2005
		EP1495580B1	17-12-2008
		US7716313B2	11-05-2010
		WO03084130A1	09-10-2003