Various systems and methods for identifying functions and data with respect to a service and a subscriber and service management system employing the same.

Inventors: Neil Dholakia, Austin, TX (US); Edward S. Pelley, Cedar Park, TX (US); Thomas A. Palkot, Austin, TX (US); Bryan P. Tacker, Austin, TX (US)

Assignee: Motive, Incorporated, Austin, TX (US)

Filed: Nov. 21, 2008

Related U.S. Application Data

Provisional application No. 60/989,730, filed on Nov. 21, 2007.

Abstract

Various systems and methods for identifying functions and data with respect to a service. One aspect provides a method of exposing a service-specific function associated with a subscriber. In one embodiment, the method includes: (1) determining services to which the subscriber has subscriptions, (2) retrieving service descriptions associated with the services and (3) retrieving at least one function associated with the service descriptions as a whole.
FIG. 5

FIG. 6
COLLECT A SERVICE LEVEL VIEW OF THE DEVICES IN A SERVICE

OBTAIN THE LIST OF SYSTEMS AND DEVICES ASSOCIATED WITH THE SUBSCRIBER

OBTAIN THE SERVICE DESCRIPTION FROM THE SERVICE REPOSITORY

FOR EACH DEVICE AND SYSTEM LOOP

MAP THE CURRENT DEVICE INTO THE CURRENT SERVICE DESCRIPTION TO DETERMINE WHETHER IT FITS A ROLE IN THAT SERVICE

RETRIEVE THE KEY/VALUE PAIRS FOR THE CURRENT DEVICE OR SYSTEM BASED ON ITS ROLE IN THE SERVICE

CALL INTO THE DEVICE NORMALIZATION API TO RETRIEVE THE KEY/VALUE PAIRS

RETRIEVE RESULTS FROM THE DIFFERENT FUNCTION CALLS AND RETURN TO THE CALLER

FIG. 9
SYSTEM AND METHOD FOR IDENTIFYING FUNCTIONS AND DATA WITH RESPECT TO A SERVICE AND A SUBSCRIBER AND SERVICE MANAGEMENT SYSTEM EMPLOYING THE SAME

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of U.S. Provisional Application Ser. No. 60/989,730, filed by Dholakia, et al., on Nov. 21, 2007, entitled “Method and System for Remote Device Management,” commonly assigned with this application and incorporated herein by reference. This application is also related to the following U.S. patent applications, which are filed on even date herewith, commonly assigned with this application and incorporated herein by reference:

<table>
<thead>
<tr>
<th>Ser. No.</th>
<th>Inventors</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>804144-</td>
<td>US-NP(10)</td>
<td></td>
</tr>
<tr>
<td>[Attorney Dholakia, et al.</td>
<td>“System and Method for Identifying and Calling a Function of a Service With Respect to a Subscriber And Service Management System Employing the Same”</td>
<td></td>
</tr>
<tr>
<td>804144-</td>
<td>US-NP(12)</td>
<td></td>
</tr>
<tr>
<td>[Attorney Pelley, et al.</td>
<td>“Normalization Engine and Method of Requesting a Key Or Performing an Operation Pertainiing to an End Point”</td>
<td></td>
</tr>
<tr>
<td>804144-</td>
<td>US-NP(13)</td>
<td></td>
</tr>
<tr>
<td>804144-</td>
<td>US-NP(15)</td>
<td></td>
</tr>
</tbody>
</table>

TECHNICAL FIELD

[0002] This application relates to remote management of fixed-line and mobile devices, and, more particularly, to activation, provisioning, support, management and assurance of consumer and business services spanning one or more fixed-line devices and one or more mobile devices.

BACKGROUND

[0003] Network service providers are called upon to support a large variety of networked devices, including devices coupled to home networks (e.g., residential gateways, set-top boxes and voice-over-IP, or VoIP, adapters) and cellular networks (e.g., smart phones and pocket computers). Given the proliferation of such devices and the distributed nature of the networks involved, remote management of such devices is highly desirable.

[0004] For example, demand for smart phones and other advanced handsets is growing faster than anticipated as users look for new ways to increase their personal and professional productivity. In 2005, year-over-year growth in the smart phone market exceeded 70%, and the industry experts expect that trend to continue for the next several years. In fact by 2009, it is estimated that smart phones will represent almost 30% of all new handsets sold—up from less than three percent in 2004.

[0005] As smart phones and services for smart phones boom, so do the challenges. Today, the complexity often associated with smart phones is driving customer service costs up and serves as a potential inhibitor as mobile network operators strive to achieve mass-market adoption with these sophisticated devices. In fact, consumers are finding mobile services increasingly confusing and issues around ease-of-use are holding them back from buying and using third generation (3G) handsets and services.

[0006] Wireless service providers who sell and support smart phones and their associated data services face the prospect of rising customer support costs due to the complexity associated with these devices and services. In 2007, the sup-
port costs for smartphones will surpass that of feature phones. The following are few of the top reasons for this support cost.

[0007] Multiple contacts are made to a helpdesk to solve a single problem.

[0008] 34% of users have never solved a problem with a single contact to the helpdesk.

[0009] Calls last two to three times longer than calls from users of feature phones.

[0010] It is common practice to escalate care from a helpdesk (Tier 1) to expensive technicians (Tier 2 and Tier 3).

[0011] FMC (Fixed-Mobile Convergence) will add to the support burden. 89% of early adopters are more likely to go to CE vendors for support. Mainstream consumers are three times more likely to look to their service provider for support.

[0012] Similarly, network providers that are coupled to home networks (e.g., Digital Subscriber Line, or DSL, and cable) find those networks coupled to a variety of customer premises equipment (CPE) within the homes that are gradually becoming more and more sophisticated. Customer issues with such devices are no less taxing upon support staff and support infrastructure.

[0013] The Open Mobile Alliance (OMA) is currently defining a number of standards for managing functionality on mobile devices. These include protocols for device management (OMA-DM), client provisioning (OMA-CP), firmware updates, data synchronization (OMA-DS) and the like. Devices that support at least some of these protocols are becoming prevalent. A support solution that utilizes these protocols and provides a usable console for customer support is the only way network providers and mobile carriers can handle support for the increasing number of devices in the market.

[0014] It is therefore desirable to provide a support solution that allows centralized management and control of remotely networked devices such as smartphones and CPE using protocols established for device management, updates, data synchronization and the like.

SUMMARY

[0015] Various embodiments of a method and system for providing customer support with centralized management and control of mobile phones and customer premises equipment in order to aid users of such equipment with problems related to that equipment. In one embodiment, a user interface driven mechanism is provided to allow customer support representatives to manipulate remote devices in, for example, the following manners: access information about the remote devices and the users thereof, including history of issues with a particular device, device provisioning, access to diagnostics of a device, ability to upgrade firmware/software of a device, synchronization of data, enablement of security features, remote control of devices, service and application provisioning, defining and following policies related to service management for a variety of devices and resetting devices. Such functionality can be provided, for example, through the use of a device management server that uses a variety of appropriate protocols to communicate with the remote devices.

[0016] Another aspect provides a method of exposing a service-specific function associated with a subscriber. In one embodiment, the method includes: (1) determining services to which the subscriber has subscriptions, (2) retrieving service descriptions associated with the services and (3) retrieving at least one function associated with the service descriptions as a whole.

[0017] Yet another aspect provides a method of exposing service-specific functions associated with a subscriber. In one embodiment, the method includes: (1) identifying at least one end point associated with the subscriber, (2) identifying at least one service to which the subscriber has a subscription, (3) retrieving at least one service description associated with the at least one service, (4) mapping the at least one end point to at least one role in the at least one service description and (5) collecting at least one function for at least one of the roles.

[0018] Still another aspect provides a method of collecting key/value pairs for a service. In one embodiment, the method includes: (1) identifying end points associated with the service, (2) collecting the key/value pairs from the end points and (3) categorizing the key/value pairs by role with respect to the service.

[0019] Still yet another aspect provides a method of collecting a service-level view of an end point relative to a service. In one embodiment, the method includes: (1) obtaining a list of end points associated with a particular subscriber, (2) obtaining a service description associated with the subscriber from a service repository and (3) repeating, for each end point in the service description: (3a) mapping the end point into the service description to determine whether the end point fits a role therein, (3b) retrieving at least one key/value pair for the current end point based on the role and (3c) retrieving key/value pairs relating to the current end point.

BRIEF DESCRIPTION

[0020] Reference is now made to the following descriptions taken in conjunction with the accompanying drawings, in which:

[0021] FIG. 1 is a block diagram illustrating a network environment in which commercial transaction processing according to embodiments of the invention may be practiced;

[0022] FIG. 2 is a block diagram illustrating a computer system suitable for implementing embodiments of the invention;

[0023] FIG. 3 is a block diagram illustrating the interconnection of the computer system of FIG. 2 to client and host systems;

[0024] FIG. 4 is a diagram illustrating relationships that may exist among a subscriber, a service and various devices and systems;

[0025] FIG. 5 is a diagram of one embodiment of a service description;

[0026] FIG. 6 is a diagram illustrating relationships that may exist among management operations, roles, capabilities and attributes;

[0027] FIG. 7 is a high-level block diagram of one embodiment of a service management system;

[0028] FIG. 8 is a block diagram of one embodiment of a service normalization block of FIG. 7; and

[0029] FIG. 9 is a flow diagram of one embodiment of a method of collecting a service-level view of a particular device in a particular service.

DETAILED DESCRIPTION

[0030] The following is intended to provide a detailed description of an example of the invention and should not be taken to be limiting of the invention itself. Rather, any number
of variations may fall within the scope of the invention which is defined in the claims following the description. The use of the same reference symbols in different drawings indicates similar or identical items.

[0031] Introduction

[0032] Described herein are various embodiments of a management system that allows users to create, define and maintain services by defining the roles of its constituent devices and systems. Certain of the embodiments have the ability to map a given set of device and systems into roles. The roles may then be used to select key/value pairs, alerts and management function from each device. Certain of the embodiments allow relationships to be specified between roles and between other services. Using the roles and relationships as a lens upon the service’s constituent devices, service-wide key/value pairs, alerts and management functions may be created.

[0033] In various embodiments to be described and illustrated herein, a method, apparatus and process are disclosed that allow for activation, provisioning, support (by call center, functions or self), management (by call center, functions or self) and assurance of consumer and business services spanning one or more fixed-line devices and one or more mobile devices, such as PCs, AAM servers, email servers, web servers and devices of every kind. Before describing the embodiments, an example computing network environment within which the embodiments may operate will be described.

[0034] An Example Computing and Network Environment

[0035] FIG. 1 is a block diagram illustrating a network environment in which a system according to the invention may be practiced. As is illustrated in FIG. 1, network 100, such as a private wide area network (WAN) or the Internet, includes a number of networked servers 110(1)-(N) that are accessible by client computers 120(1)-(N).

[0036] Communication between the client computers 120(1)-(N) and the servers 110(1)-(N) typically occurs over a publicly accessible network, such as a public switched telephone network (PSTN), a DSL connection, a cable modem connection or large bandwidth trunks (e.g., communications channels providing T1 or OC3 service). The client computers 120(1)-(N) access the servers 110(1)-(N) through, for example, a service provider. This might be, for example, an Internet Service Provider (ISP) such as America On-Line™, Prodigy™, CompuServe™ or the like. Access is typically had by executing application specific software (e.g., network connection software and a browser) on the given one of the client computers 120(1)-(N).

[0037] One or more of the client computers 120(1)-(N) and/or one or more of the servers 110(1)-(N) may be, for example, a computer system of any appropriate design, in general, including a mainframe, a mini-computer or a personal computer system. Such a computer system typically includes a system unit having a system processor and associated volatile and non-volatile memory, one or more display monitors and keyboards, one or more diskette drives, one or more fixed disk storage devices and one or more printers. These computer systems are typically information handling systems which are designed to provide computing power to one or more users, either locally or remotely. Such a computer system may also include one or a plurality of I/O devices (i.e., peripheral devices) which are coupled to the system processor and which perform specialized functions. Examples of I/O devices include modems, sound and video devices and specialized communication devices. Mass storage devices such as hard disks, CD-ROM drives and magneto-optical drives may also be provided, either as an integrated or peripheral device. One such example, computer system, discussed in terms of the client computers 120(1)-(N), is shown in detail in FIG. 2.

[0038] It will be noted that the variable identifier “N” is used in several instances in FIG. 1 to more simply designate the final element (e.g., the servers 110(1)-(N) and the client computers 120(1)-(N)) of a series of related or similar elements (e.g., servers and client computers). The repeated use of such variable identifiers is not meant to imply a correlation between the sizes of such series elements, although such correlation may exist. The use of such variable identifiers does not require that each series of elements has the same number of elements as another series delimited by the same variable identifier. Rather, in each instance of use, the variable identified by “N” may hold the same or a different value than other instances of the same variable identifier.

[0039] FIG. 2 depicts a block diagram of a computer system 210 suitable for implementing the invention and example of one or more of the client computers 120(1)-(N). A computer system 210 includes a bus 212 which interconnects major subsystems of the computer system 210 such as a central processor 214, a system memory 216 (typically random-access memory, or RAM, but which may also include read-only memory, or ROM, flash RAM, or the like), an input/output controller 218, an external audio device such as a speaker system 220 via an audio output interface 222, an external device such as a display screen 224 via a display adapter 226, serial ports 228, 230, a keyboard 232 (interfaced with a keyboard controller 233), a storage interface 234, a floppy disk drive 236 operative to receive a floppy disk 238 and a CD-ROM drive 240 operative to receive a CD-ROM 242. Also included are a mouse 246 (or other point-and-click device, coupled to the bus 212 via the serial port 228), a modem 247 (coupled to the bus 212 via the serial port 230) and a network interface 248 (coupled directly to the bus 212).

[0040] The bus 212 allows data communication between a central processor 214 and a system memory 216, which may include RAM, ROM or flash memory, as previously noted. The RAM is generally the main memory into which the operating system and application programs are loaded and typically affords at least 16 megabytes of memory space. The ROM or flash memory may contain, among other code, the Basic Input-Output system (BIOS) which controls basic hardware operation such as the interaction with peripheral components. Applications resident with the computer system 210 are generally stored on and accessed via a computer readable medium, such as a hard disk drive (e.g., fixed disk 244), an optical drive (e.g., CD-ROM drive 240), a floppy disk unit 236 or other storage medium. Additionally, applications may be in the form of electronic signals modulated in accordance with the application and data communication technology when accessed via a network modem 247 or an interface 248.

[0041] The storage interface 234, as with the other storage interfaces of the computer system 210, may connect to a standard computer readable medium for storage and/or retrieval of information, such as a fixed disk drive 244. The fixed disk drive 244 may be a part of computer system 210 or may be separate and accessed through other interface systems. Many other devices can be connected, such as a mouse 246 connected to the bus 212 via serial port 228, a modem 247
connected to the bus 212 via serial port 230 and a network interface 248 connected directly to the bus 212. The modem
247 may provide a direct connection to a remote server via a telephone link or to the Internet via an internet service
provider (ISP). The network interface 248 may provide a direct connection to a remote server via a direct network link to
the Internet via a POP (point of presence). The network interface 248 may provide such connection using wireless
techniques, including digital cellular telephone connection, Cellular Digital Packet Data (CDPD) connection, digital satellite
data connection or the like.

[0042] Many other devices or subsystems (not shown) may be connected in a similar manner (e.g., bar code readers,
document scanners, digital cameras and so on).

[0043] Conversely, it is not necessary for all of the devices shown in FIG. 2 to be present to practice the invention. The
devices and subsystems may be interconnected in different ways from that shown in FIG. 2. The operation of a computer
system such as that shown in FIG. 2 is readily known in the art and is not discussed in detail in this application. Code to
implement the invention may be stored in computer-readable storage media such as one or more of the system memory
216, the fixed disk 244, the CD-ROM 242 or the floppy disk 238. Additionally, the computer system 210 may be any kind
of computing device and so includes personal data assistants (PDAs), network appliance, X-window terminal or other such
computing device. The operating system provided on the computer system 210 may be MS-DOS®, MS-Windows®,
OS/2®, UNIX®, Linux® or other known operating system. The computer system 210 may also support a number of
Internet access tools, including, for example, a Hypertext Transfer Protocol (HTTP)-compliant web browser having a
JavaScript interpreter, such as Netscape Navigator® 3.0, Microsoft Explorer® 3.0 and the like.

[0044] The foregoing described embodiment wherein the different components are contained within different other
components (e.g., the various elements shown as components of the computer system 210). It is to be understood that such
depicted architectures are merely examples, and that in fact, many other architectures can be implemented which achieve
the same functionality. In an abstract, but still definite sense, any arrangement of components to achieve the same
functionality is effectively “associated” such that the desired functionality is achieved. Hence, any two components herein
combined to achieve a particular functionality can be seen as “associated with” each other such that the desired function-
ality is achieved, irrespective of architectures or intermediate components. Likewise, any two components so associated
can also be viewed as being “operably connected,” or “oper-
ably coupled,” to each other to achieve the desired function-
ality.

[0045] FIG. 3 is a block diagram depicting a network 300 in
which the computer system 210 is coupled to an internetwork
310, which is coupled, in turn, to client systems 320, 330, as
well as a server 340. An internetwork 310 (e.g., the Internet or
a wide-area network, or WAN) is also capable of coupling the
client systems 320, 330 and the server 340 to one another.
With reference to the computer system 210, the modem 247,
the network interface 248 or some other method can be used
to provide connectivity from the computer system 210 to the
internetwork 310. The computer system 210, the client sys-
tem 320 and the client system 330 are able to access informa-
tion on the server 340 using, for example, a web browser
(not shown). Such a web browser allows the computer system
210, as well as the client systems 320, 330, to access data on
the server 340 representing the pages of a website hosted on
the server 340. Protocols for exchanging data via the Internet
are well known to those skilled in the art. Although FIG. 3
depicts the use of the Internet for exchanging data, the inven-
tion is not limited to the Internet or any particular
networked environment.

[0046] Referring to FIGS. 1, 2 and 3, a browser running on
the computer system 210 employs a TCP/IP connection to
pass a request to the server 340, which can run an HTTP
“service” (e.g., under the WINDOWS® operating system) or
a “daemon” (e.g., under the UNIX® operating system), for
example. Such a request can be processed, for example, by
contacting an HTTP server employing a protocol that can be
used to communicate between the HTTP server and the client
computer. The HTTP server then responds to the protocol,
typically by sending a “web page” formatted as an HTML
file. The browser interprets the HTML file and may form a
visual representation of the same using local resources (e.g.,
fonts and colors).

[0047] Example Embodiments of a Service Management System

[0048] The functions referred to herein may be modules or
portions of modules (e.g., software, firmware or hardware
modules). For example, although the described embodiment
includes software modules and/or includes manually entered
user commands, the various example modules may be appli-
cation specific hardware modules. The software modules
discussed herein may include script, batch or other executable
files, or combinations and/or portions of such files. The soft-
ware modules may include a computer program or subrou-
tines thereof encoded on computer-readable media.

[0049] Additionally, those skilled in the art will recognize
that the boundaries between modules are merely illustrative
and alternative embodiments may merge modules or impose
an alternative decomposition of functionality of modules. For
example, the modules discussed herein may be decomposed
into sub-modules to be executed as multiple computer pro-
cesses and, optionally, on multiple computers. Moreover,
alternative embodiments may combine multiple instances of
a particular module or sub-module. Furthermore, those
skilled in the art will recognize that the functions described in
example embodiment are for illustration only. Operations
may be combined or the functionality of the functions may be
distributed in additional functions in accordance with the
invention.

[0050] Alternatively, such actions may be embodied in the
structure of circuitry that implements such functionality, such
as the micro-code of a complex instruction set computer
(CISC), firmware programmed into programmable or eras-
able/programmable devices, the configuration of a field-pro-
grammable gate array (FPGA), the design of a gate array or
full-custom application-specific integrated circuit (ASIC), or
the like.

[0051] Each of the blocks of the flow diagram may be
executed by a module (e.g., a software module) or a portion
of a module or a computer system user using, for example, a
computer system such as the computer system 210. Thus, the
above described method, the functions thereof and modules
therefore may be executed on a computer system configured
to execute the functions of the method and/or may be
executed from computer-readable media. The method may
be embodied in a machine-readable and/or computer-readable
medium for configuring a computer system to execute the
method. Thus, the software modules may be stored within and/or transmitted to a computer system memory to configure the computer system to perform the functions of the module.

[0052] Such a computer system normally processes information according to a program (a list of internally stored instructions such as a particular application program and/or an operating system) and produces resultant output information via I/O devices. A computer process typically includes an executing (running) program or portion of a program, current program values and state information and the resources used by the operating system to manage the execution of the process. A parent process may spawn other, child processes to help perform the overall functionality of the parent process. Because the parent process specifically spawns the child processes to perform a portion of the overall functionality of the parent process, the functions performed by child processes (and grandchild processes, etc.) may sometimes be described as being performed by the parent process.

[0053] Such a computer system typically includes multiple computer processes executing “concurrently?” Often, a computer system includes a single processing unit which is capable of supporting many active processes alternately. Although multiple processes may appear to be executing concurrently, at any given point in time only one process is actually executed by the single processing unit. By rapidly changing the process executing, a computer system gives the appearance of concurrent process execution. The ability of a computer system to multiplex the computer system’s resources among multiple processes in various stages of execution is called multitasking. Systems with multiple processing units, which by definition can support true concurrent processing, are called multiprocessor system. Active processes are often referred to as executing concurrently when such processes are executed in a multitasking and/or a multiprocessor environment.

[0054] The software modules described herein may be received by such a computer system, for example, from computer readable media. The computer readable media may be permanently, removably or remotely coupled to the computer system. The computer readable media may non-exclusively include, for example, any number of the following: magnetic storage media including disk and tape storage media, optical storage media such as compact disk media (e.g., CD-ROM, CD-R, etc.) and digital video disk storage media, nonvolatile memory storage memory including semiconductor-based memory units such as flash memory, EEPROM, EPROM, ROM or application-specific integrated circuits (ASICs), volatile storage media including registers, buffers or caches, main memory, RAM and the like, and data transmission media including computer network, point-to-point telecommunication and carrier wave transmission media. In a UNIX-based embodiment, the software modules may be embodied in a file which may be a device, a terminal, a local or remote file, a socket, a network connection, a signal, or other expedient of communication or state change. Other new and various types of computer-readable media may be used to store and/or transmit the software modules discussed herein.

[0055] Before describing various embodiments of management systems constructed according to the principles of the invention, some use cases or interactions will be described that provide a framework for understanding the management systems. Various of the embodiments of the management system are directed to addressing the following categories of use cases or interactions: service activation, service management, service interruption and resumption and service offering. Service activation refers to all the use cases that involve creating (provisioning) and deleting (unprovisioning) a new service instance, or “subscription.” Service management refers to day-to-day management tasks involving a given service or subscription. Service interruption and resumption may be thought of as special types of service management that involve the loss and restoration of service. Service offering refers to new services that may be offered to a subscriber.

[0056] The categories of use cases that set forth above may be illustrated with reference to FIG. 4. FIG. 4 is a diagram illustrating relationships that may exist among a subscriber 410, a service 420 and various devices and systems 430. A service provider, such as a cellular phone company, an Internet service provider, a cable television company or a combination of these, offers one or more services to subscribers that involve devices and systems such as cell phones, set-top boxes, routers, cell towers, email servers, DSLAMs, landline phones and other mobile and customer premises equipment and network infrastructure. In the context of FIG. 4, the subscriber 410 takes out a subscription 440 to a service 420 offered by a service provider. The subscription contains state and other information that is both relative to the subscriber 410 and the service 420. The subscription calls for or one or more associations 450 to be made between the subscriber 410 and various devices and systems 430. The subscriber 410 may own or lease one or more devices 430. The subscriber 410 may also be associated with one or more systems 430. Once the associations 450 are made, the devices and systems 430 assume roles 460 in delivering the service 420 to the subscriber 410. The roles describe to the service management system how the device should be managed.

[0057] FIG. 4 may be employed to illustrate two example use cases: activating a device for a subscriber and managing and provisioning a subscription.

[0058] To activate a device (one of the devices and systems 430) for a subscriber 410, the following steps may be taken. First, for a given device 410, the subscriber 410 associated with that device 430 is found. This is done by employing the associations 450. Once the associated subscriber 410 has been identified, a corresponding subscription 440 may then be used to determine the service or services 420 that should be provisioned on the device 430. For each service 420 that needs to be activated on the device 430, two sets of administrative actions may be taken. Based upon the role the device 430 plays with respect to the service 420, settings on the device 430 may be set to provision the device. Alternatively or additionally, based on the roles of other devices and systems 430 relative to the service 420, settings on the other devices and systems may be set to provision them for the new device’s presence.

[0059] Managing and provisioning a subscription involves either adding a new service to a subscriber or managing an existing service. To manage and provision a subscription 440 for a subscriber 410, the following steps may be taken. First, a service 420 is added to a subscriber 410, or an existing service 420 is modified. Devices and systems 430 associated with the subscriber are collected. Then, each device associated with the service 420 is mapped into the different roles 460 in the targeted service. This reveals what actions should be taken with respect to each device or system to provision the service 420.

[0060] Another use case that is a variant on the one above is a bulk change to an existing service for all subscribers. In this
use case, existing subscriptions are retrieved to obtain a list of subscribers. Then, a list of devices and systems 430 is assembled for each subscriber. Using roles, changes are then applied to the devices and systems 430 to enable the bulk change.

[0061] Having described various use cases, one manner in which interaction may occur among roles, devices and systems, and the service level management interfaces will now be described. FIG. 5 is a diagram of one embodiment of a service description 500. FIG. 5 shows that the service description 500 includes service actions 505, functions 510 and key/value pairs 515. Roles are associated with the service alerts 505, the functions 510 and the key/value pairs 515. The roles, designated Role A 520, Role B 525 and Role C 530 are associated as shown with the service alerts 505, the functions 510 and the key/value pairs 515 as various arrows show. Meta data 535 is also associated with the key/value pairs 515. Devices or systems 540, 545 are associated with the roles 520, 525, 535 as various arrows show.

[0062] FIG. 5 illustrates, in the context of the service description 500, how a set of roles (e.g., the roles 520, 525, 535) can be mapped onto a set of devices and systems (e.g., the devices or systems 540, 545). The roles define the functions, alerts and functions that are of interest from each device or system. In the illustrated embodiment, the meta data 535 contains both service-wide and subscriber/subscription data that is specific to the service description. Items like level of service and current state of activation would be a part of the meta data. The alerts, functions, and key/value pairs 505, 510, 515 together constitute services supported by the devices and systems 540, 545 exposed through the roles 520, 525, 530. The service description 500 is configured to contain arbitrary, named relationships that can be used to affect a service. The service description 500 may also be configured to contain one or more references to other service with which it has relationships or upon which it has dependencies. Accordingly, a further service description 550 is associated with the meta data 535 as an arrow shows. Similar to the relationships between roles, relationships between services are exposed to the logic in the service description and to external logic associated with the service description.

[0063] To make a role useful, the role is matched with a device. FIG. 6 is a diagram illustrating relationships that may exist between devices (i.e., values, alerts, and functions 505, 510, 515), roles (e.g., 460), capabilities and attributes 610 and devices and systems (e.g., 430). In the embodiment of FIG. 6, the mechanism for doing this is twofold. First, a role may be matched to a device or a system based on the known attributes of the device or system. Second, a role may be matched to a device or a system based upon the known capabilities of the device or system.

[0064] Device attributes are known aspects of the device, e.g., the type, serial number, MAC address, manufacture date, make, model, service tags, device ID or operating system of the device or system. Other attributes may include the firmware version, hardware version, embedded device, locale (language), and physical location. Device attributes, in the simplest form, could be a list of known key/value pairs associated to the device.

[0065] Capabilities are similar to device attributes. In this case, instead of a list of key/value pairs, these are a list of values (without the key) of known capabilities about the device, e.g., generic email client, Microsoft Outlook® email client, phone, router or IPTV device. Other examples include network attached storage, media server, media renderer, camera, MMS client, SMS client, wireless access provider, wireless access client, printer, GPS, vibrate, Bluetooth, USB, Wi-Fi, clock, browser, QVGA, flight mode, caller ID, touch screen, or fax.

[0066] Both the capabilities and the attributes can be provided to or retrieved from an external system, deduced or derived from known attributes and capabilities, queried directly from the device or system or a combination of these. For example, prior knowledge may exist that any device that has a serial number from a given manufacturer starting with the letter W- has built-in Wi-Fi capabilities, or that Windows® Mobile phone supports OMA-DM.

[0067] It can be determined whether a given device or system matches a role by matching its attributes and capabilities (derived, discovered, or known) against the required attributes and capabilities of a given role. Each role defines a set of key/value pairs, alerts, and functions that are relevant for devices of that role in the service description.

[0068] It should be noted that roles do not imply device type, model or brand. Direct mappings between devices and roles may occur in practice, but the mappings are flexible such that they may change as device attributes or capabilities change. For example, newer devices may support more roles than do older devices. One example of a role is a phone capable of functioning as an email client may play an “Email-Client” role in a service description associated with an email service. Other roles in an email service may include “SMTPServer,” “POP3Server” and “IMAPServer.” Roles in a data connectivity service may include “Host,” “Router,” “Wireless Access Point,” “Head End,” “Border Gateway” and “Authentication, Authorization, Account Server.”

[0069] FIG. 7 is a high-level block diagram of one embodiment of a service management system. The service management system includes a service normalization block 705 that will be described in greater detail in conjunction with FIG. 8 below. The service normalization block interacts with an optimal settings block 710, a device (and/or system) normalization block 715, a diagnostic engine block 720 and a content repository block 725. A segmentation block 730 and a knowledge base 735 interact with the optimal settings block 710, the device normalization block 715, the diagnostic engine block 720 and the content repository block 725 as shown.

[0070] The service normalization block 705 employs an application programming interface (API), allowing it to exchange information with an interactive voice response (IVR) system 745, a console 750 for a customer service representative (CSR), a Self-Service Management (SSM) application module 755 and other applications 760, 765 as may be found advantageous in a particular environment.

[0071] The optimal settings block 710 is a repository of predefined known good values used for the purposes of comparing key/value pairs to determine diagnostic and state information. The key/value pairs are also used during provisioning to set up a system or a device. The optimal settings block 710 may be regarded as a configuration repository that contains meta data about the configuration so that applications and other systems can look up known good values for the purpose of configuring (provisioning), diagnostic, and repair. The illustrated embodiment of the optimal settings block 710 is configured to define optimal values for any given key/value pair based on the context of the device, subscriber, customer, or any other segmentation scheme that could be use to define different values for the same attribute (key). These values
may be used by both the script engine 830 and directly by the service management engine 805 to determine whether or not a given key/value pair is “optimal.” Optimal values may fall into three categories: (1) values predefined in the context of the service as being correct, (2) values defined based on a call to an extrinsic system or by subscriber input as being correct and (3) absolute values (which is often built into the logic of the script or service description logic and not stored externally). An example of a predefined optimal value is a POP server. The subscriber is aware of its identity, and it is the same for all subscribers. An example of an absolute value is “connectivity good.” An example of a value defined by a subscriber as being correct is a password, something that the subscriber chooses and is not defined before the subscriber chooses it.

[0072] The device normalization block 715 is configured to map normalized key/value pairs to device-specific or system-specific key value pair sets. Mapping may be performed by transformation, executing a script, or undertaking any other appropriate normalization mechanism.

[0073] The diagnostic engine 720 is configured to contain diagnostic rules and cause diagnostic rules to be executed in order to identify, characterize and present potential solutions to problems that may exist with devices or systems. The content repository is configured to provide a channel-independent mechanism for associating bearer (e.g., IVR voice flows, self-service portal web content and customer service articles) to diagnose a problem.

[0074] The segmentation block 730 and knowledge base 735 likewise employ a data sources abstraction layer 770, allowing it to communicate with systems 775, 790 and provisioning servers and device managers 780, 785.

[0075] Different subscribers subscribe to different levels of service and live in different locations and under different circumstances. The segmentation block 730 is configured to enable other portions of the service management system to tailor responses to a subscriber based on his level of service, location and/or circumstance.

[0076] The knowledge base 735 is configured to contain articles associated with known device, system and/or service problems. When the diagnostic engine 720 identifies a problem area or a specified problem, it may provide articles from the knowledge base 735 to an application to provide the article in turn to a subscriber or other user for informational purposes.

[0077] The data sources abstraction layer 770 is configured to operate as a protocol implementation and adaptation layer, allowing generic logic to interact with specific devices and systems without having to employ a device-specific or system-specific protocol.

[0078] The systems 775, 790 are typically added by a particular service provider and interact with the service management system as needed. The provisioning servers and device managers 780, 785 support various devices 795 intended for use by subscribers. In the illustrated embodiment, the provisioning servers and device managers 780, 785 are management systems that manage large groups of devices that typically share the same protocol (e.g., a mobile device manager that manages 10 million phones using the OMA-DM protocol). The devices 795 are just CPE, such as phones and routers.

[0079] FIG. 8 is a block diagram of one embodiment of the service normalization block 705. The API 740 provides a mechanism by which the service normalization block 705 may be called from within and without the service management system. By means of the API 740, subscribers may have services added (provisioned), removed (unprovisioned), modified or otherwise managed. Devices of subscribers may also be managed in the context of a particular service. Management access to key/value pairs of constituent devices and systems may also be dynamically determined and managed based on their roles in providing particular services.

[0080] A service management engine 805 enables a service provider to implement and manage services by means of the service normalization block 705 according to the various use cases described above. The illustrated embodiment of the service management engine 805 functions in two primary ways. First, the service management engine 805 manages functions defined by service descriptions. Second, the service management engine 805 provides a dynamic view of a given service.

[0081] The management of functions allows service descriptions to define named functions that can be called with contextual data derived from analyzing constituent devices and systems in their associated roles of a service.

[0082] The provision of a dynamic view of a given service enables a service description to associate key/value pairs (data) with different roles and dynamically to gather data from the devices and systems so that data can be presented without the need for intrinsic knowledge of the data that is being collected. For example, a service-view dashboard capable of creating a map of devices with their associated data of interest (each categorized by their role in the service) may employ dynamic views of services. In the illustrated embodiment, the data itself is self-describing and typically presented in tabular form.

[0083] In an alternative embodiment, the service management engine 805 is also capable of providing a view of a given service, in which case the application does have prior intrinsic knowledge regarding the data that is being collected.

[0084] The first step in managing the service is to collect a list of the devices and systems that are associated to the subscriber of the service. A device repository 835 serves this purpose. In the illustrated embodiment, the device repository is external to the service normalization block 705.

[0085] The service management engine 805 employs a capabilities repository to obtain an expanded view on the capabilities of a device so that it may map it into a role. Often the only pieces of information obtained from extrinsic systems are the unique identifier of the device, e.g., its make and model. A device may be viewed as a list of attributes (e.g., further key/value pairs). From those attributes, the capabilities of the device may be extrapolated. Extrapolation may involve expanding the known attributes of a device by deriving new attributes of a device based on the original attributes. For example, once the make and model of a system or device is obtained by means of a query, built-in rules may then be able to determine whether or not it has Wi-Fi capability.

[0086] A service description repository contains service descriptions. As described above, a service description includes at least some of: functions, key/value pairs, alerts, roles (along with their associated key/value pairs, alerts and actions) and relationships. Functions, at the service description level, can be actions exposed by a device, a script that can be executed, or a process (a series of scripts executed in a state engine).

[0087] A script engine 830 is configured to execute service-level functions. As described previously, a service-level func-
tion could be a script, a process or action (a series of scripts) or any other type of computer program. In the illustrated embodiment, the service management engine 805 retrieves a named script from a service description based on an event or request from a consumer of the service and passes it along with a set of parameters, to the script engine 805 for execution. In the illustrated embodiment, the set of parameters includes references to the constituent devices (categorized by role) and the service description. The script, once started, has access to the optimal values, the devices and systems (abstracted by device normalization or directly) and the service management system in general.

[0088] The service normalization block 705 has access to a capabilities repository 820. The capabilities repository 820 is configured to derive new attributes and capabilities through rules based on existing, known attributes. For example, it is known that Windows® Mobile cell phones have Internet browsers.

[0089] The service normalization block 705 employs the device normalization engine 715 configured to create an abstraction that provides a normal view of extrinsic device and systems. This allows the service description to be defined generically for devices and systems of the same class without having to include logic and cases for each device. For example, if one device is managed by OMA-DM and has an email client while another device that also has an email client is managed by Digital Subscriber Line (DSL) Forum standard TR069, both devices will have a Simple Mail Transfer Protocol (SMTP) server. However, the manner in which values are obtained can differ by protocol (OMA-DM versus TR069) and by key (the name for the value).

[0090] At least some of the systems and devices 775, 795 have the capability to generate alerts or events. The alert/event engine 815 is configured to receive these alerts or events and apply them against each service description to determine whether the alert applies to that service description. If a particular alert or event does apply to a particular service, the service management system is configured to obtain a corresponding action, script or process from the service description that may be executed to respond to the alert or event.

[0091] Identification of Functions and Data with Respect to a Service and a Subscriber

[0092] Described hereinafter are various systems and methods for identifying functions and data. One aspect provides a method of exposing service-specific functions based upon a subscriber. One embodiment of the method begins by determining the services to which a subscriber has subscriptions. Service descriptions associated with those services are then retrieved. Then, functions associated with the service descriptions as a whole are retrieved and executed. It should be noted that the functions being retrieved are not associated with particular roles of the services, but the services overall.

[0093] For example, a function associated with a service description as a whole is a function that determines whether a service has the necessary roles fulfilled by devices or systems in order to be provided to subscribers.

[0094] Another aspect provides a method of exposing service-specific functions associated with a particular subscriber. In one embodiment, the method begins by determining the devices and systems that are associated with the subscriber. Then, the devices and systems are mapped to roles in services to which the subscriber has a subscription. Next, functions for each role in the service fulfilled by each device are collected.

[0095] For example, a service-specific function associated with a particular subscriber is a function that, given the devices a subscriber owns and a particular service, provides functions that can be called with respect to the service.

[0096] Yet another aspect provides a method of collecting and exposing data for a service in the form of key/value pairs from devices and systems associated with a subscriber in the context of the service where the data exposed is categorized by each system or device's role in the service.

[0097] FIG. 9 is a flow diagram of one embodiment of a method of collecting a service-level view of a particular target device or system relative to a particular service. The method begins in a start step 905, when it is desired to collect a service-level view of the target device or system. In a step 910, a list of systems and devices associated with a particular subscriber is obtained. In a step 915, a service description associated with the particular subscriber is obtained from the service repository. In a step 920, an analytical process is repeated for each device or system in the service description.

[0098] One step in the analytical process, a step 925, the current device or system is mapped into the service description to determine whether it fits a role in the service description. In a step 930, key/value pairs for the current device or system are retrieved based on the role in the service. In a step 935, a call is made into the device normalization API to retrieve key/value pairs relating to the current device or system.

[0099] Once the analytical process of the steps 925 through 935 is repeated for each device and system, a step 940 calls for the results from the different function calls to be retrieved and returned. In the illustrated embodiment, roles and relationships between the devices and systems are included in the results.

[0100] Those skilled in the art to which this application relates will appreciate that other and further additions, deletions, substitutions and modifications may be made to the described embodiments.

What is claimed is:

1. A method of exposing a service-specific function associated with a subscriber, comprising:
   determining services to which said subscriber has subscriptions;
   retrieving service descriptions associated with said services; and
   retrieving at least one function associated with said service descriptions as a whole.

2. The method as recited in claim 1 further comprising causing said at least one function to be executed.

3. The method as recited in claim 1 wherein said at least one function is dissociated from particular roles in said service descriptions.

4. The method as recited in claim 1 wherein said at least one function includes a function configured to determine whether a service has necessary roles fulfilled by end points.

5. The method as recited in claim 1 further comprising carrying out said retrieving by making a call into a device normalization application programming interface.

6. A method of exposing service-specific functions associated with a subscriber, comprising:
   identifying at least one end point associated with said subscriber;
   identifying at least one service to which said subscriber has a subscription;
retrieving at least one service description associated with said at least one service; mapping said at least one end point to at least one role in said at least one service description; and collecting at least one function for at least one of said roles.

7. The method as recited in claim 6 further comprising causing said at least one function to be executed.

8. The method as recited in claim 6 wherein said collecting comprises collecting said at least one function for all of said roles.

9. The method as recited in claim 6 wherein said at least one function includes a function configured to provide functions that can be called with respect to a particular service.

10. A method of collecting key/value pairs for a service, comprising: identifying end points associated with said service; collecting said key/value pairs from said end points; and categorizing said key/value pairs by role with respect to said service.

11. The method as recited in claim 10 further comprising carrying out said retrieving by making a call into a device normalization application programming interface.

12. A method of collecting a service-level view of an end point relative to a service, comprising: obtaining end points associated with a particular subscriber; obtaining a service description associated with said subscriber from a service repository; and repeating, for each end point in said service description: mapping said end point into said service description to determine whether said end point fits a role therein, retrieving at least one key/value pair for said current end point based on said role, and retrieving key/value pairs relating to said current end point.

13. The method as recited in claim 12 further comprising collecting optimal values of key/value pairs for said end points.

14. The method as recited in claim 13 further comprising returning said optimal values to an application.

15. The method as recited in claim 14 wherein said returning further comprises returning roles and relationships between said end points.

16. The method as recited in claim 12 wherein said end points are selected from the group consisting of: devices, and systems.

17. The method as recited in claim 12 further comprising carrying out said retrieving by making a call into a device normalization application programming interface.

* * * * *