USER INTERFACE FOR MANAGING NETWORK DOWNLOAD AND CONFIGURATION TASKS

CROSS-REFERENCE TO RELATED APPLICATION

This application claims the benefit of U.S. Patent Application No. 12/113,038 filed April 30, 2008.

BACKGROUND

Technical Field

The present disclosure is directed to the management of networked devices, including networked electronic gaming machines.

Description of the Related Art

A system and method for downloading and configuring software for networked devices, particularly electronic gaming machines (EGM), is found in published PCT Application No. PCT US/2006/035556, which is owned by the assignee of the present application and is incorporated herein in its entirety. To briefly summarize, the disclosed network system allows a casino operator to define changes to software inventory (*e.g.*, via download) and to schedule configuration changes. For example, a casino operator can define default payouts and denominations, schedule recurring overrides for weekends, and schedule a one-time override for a holiday or casino promotion.

Ideally, changes are made without casino operator interaction or attendance to individual machines on the gaming floor each time a new configuration is needed. The configuration changes are preferably communicated during the background operation of the EGM without affecting game play, and the changes are applied at a designated convenient time. Thus, the EGM do not need to be placed in an inactive state prior to downloading configuration changes.

The network system also includes a method of recognizing when an EGM needs data downloads or configuration. These activities are coordinated through the network to avoid conflicts. For example, configuration of an EGM will be held until the downloads for the EGM are completed. In another example, the network host is configured to automatically restore data modules and to configure an EGM if the EGM has been RAM-cleared or has been offline. Thus, an operator can monitor and manage a group of EGMs from a single terminal, eliminating the need for slot technicians to collect configuration data and to manually reconfigure each EGM.

In accordance with a method of the foregoing system, a host downloads a package to the EGM that is then authenticated and either installed immediately or scheduled for later installation. Various methods of assignment conflict analysis and resolution are also provided, although conflicts typically occur for assignments of the same type (*e.g.*, download and configuration). Such conflicts are avoided by running the download process before the configuration process, with the exception that data related to the host and owner information will supersede other assignments. Conflicts can also arise as a result of an EGM being a member of different "collections" where membership may vary depending on the EGM configuration at a moment in time. For example, switching an EGM from a five cent denomination to a twenty-five cent denomination may switch the membership of the EGM to another collection (*e.g.*, all twenty-five cent EGMs). Thus, the network system includes various methods to resolve situations where EGMs are scheduled for management upgrades or updates to provide a consistent method of conflict avoidance.

Hence, scheduling assignments must be done with care to avoid unintentional conflicts or overlapping timeslot assignments that can jeopardize the configuration process for one or more EGMs. Scheduling of different tasks over a period of time in system applications has generally required a multiple step process that includes setting up the tasks in the order in which they are to

be performed, how often the tasks are to be performed, and creating a graphical user interface for creating and editing the schedule. A timer also needs to be included for use in initiating and terminating scheduled tasks. Classifications of tasks also need to be created in order to insure coordination of equipment configuration.

In a gaming system network, hundreds of EGMs can be involved, requiring constant scheduling of downloads and configurations, monitoring of the process, and resolving any conflicts that may arise due to interruptions in the schedule, breakdowns in the EGMs, and casino activities that may affect use of the EGMs, such as specials, holiday events, and the like.

BRIEF SUMMARY

In accordance with one embodiment of the present disclosure, a user interface for management of network operations, including the operation of networked devices, such as electronic gaming machines, is provided. The interface is implemented by a computer-readable medium whose contents enable a computing device to display on a display device a user interface by performing a method that includes generating a display of the user interface that includes operation assignment interfaces, each assignment interface providing information about an operation, a calendar interface that provides at least date and time information, and at least one graphical linking indicator that provides visual relational information of related operations, the user interface configured to enable scheduling of operations through the assignment interfaces on the calendar interface.

In accordance with another aspect of the present disclosure, the contents of the computer-readable medium are instructions that when executed cause the computing device to perform the method.

In accordance with another aspect of the present disclosure, the assignment interface provides information about at least one of an assignment

identifier, status of the related operation, an error indicator, and a start time. In one embodiment, the assignment interface includes a user-activated tooltip.

In accordance with another aspect of the present disclosure, the calendar interface includes a timeline displaying at least one of a date and time, and a graphical indicator interposed on the timeline to indicate at least one of current date and time.

In accordance with another embodiment of the present disclosure, a system is provided that includes a network of devices, a display device, and a processor coupled to the network of devices and the display device and configured to generate a user interface on the display device for display and user management of network operations, the user interface including assignment interfaces, each assignment interface providing information about an associated operation, a calendar interface on which the assignment interfaces are displayed, the calendar interface providing at least date and time information, and at least one graphical linking indicator that provides visual relational information of related operations.

In accordance with another embodiment of the present disclosure, a method of providing a user interface for management of network operations is provided, the method including presenting a time line along a time line axis with information that is indicative of at least one of dates or time, and for each of a number of operations, presenting a respective assignment icon with a start point of the assignment icon positioned along the time line axis at a point that is indicative of a time that an operation represented by the assignment icon is scheduled to start.

In accordance with another embodiment of the present disclosure, presenting a linking icon between the assignment icons that represent the at least two operations that reoccur over time includes presenting a line that extends from at least proximate a finish of a first one of the assignment icons to at least proximate a start of a second one of the assignment icons.

In accordance with another embodiment of the present disclosure, presenting a linking icon between the assignment icons that represent the at least two operations that reoccur over time includes presenting a line that extends from at least proximate a finish of a first one of the assignment icons to at least proximate a start of a second one of the assignment icons.

In accordance with another embodiment of the present disclosure, the method includes presenting a termination marker at a point along the time line axis indicative of an extended time, and presenting an extension icon proximate one of the assignment icons where a termination time of the assignment represented by the assignment icon has been extended.

As will be readily appreciated from the foregoing, the present disclosure provides an intuitive user interface tool for operators to view, monitor status, and manipulate network operations, such as assignments for download and configuration of networked devices, including EGMs.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

The foregoing and other features and advantages of the present disclosure will be more readily appreciated as the same become better understood from the following detailed description when taken in conjunction with the accompanying drawings.

In the drawings, identical reference numbers identify similar elements or acts. The size and relative positions of elements in the drawings are not necessarily drawn to scale. For example, the shapes of various elements are not drawn to scale, and some of these elements are arbitrarily enlarged and positioned to improve drawing legibility. Further, the particular shapes and the elements as drawn are not intended to convey any information regarding the actual shape of the particular elements and have been solely selected for their ease and recognition in the drawings.

Figure 1 illustrates an embodiment of a known gaming network that may be used with the user interface of the present disclosure.

Figure 2 illustrates an overview of a known download interaction.

Figure 3 illustrates an overview of EGM Startup message flow.

Figure 4 depicts the logic flow of /dev/download driver initialization.

Figure 5 is a flow diagram illustrating a routine to validate download requests and pass control to the proper handler function.

Figures 6A and 6B illustrate a user interface for managing download and configuration assignments in the EGM network; and

Figure 7 illustrates an assignment block from the user interface of Figures 6A and 6B.

DETAILED DESCRIPTION

In the following description, certain specific details are set forth in order to provide a thorough understanding of various embodiments of the disclosure. However, one skilled in the art will understand that the disclosure may be practiced without these specific details. In other instances, well-known structures associated with computers, computer networks, data structures, databases, networks such as the Internet, user interfaces and graphical user interfaces, and electronic game machines have not been described in detail to avoid unnecessarily obscuring the descriptions of the embodiments of the present disclosure.

Unless the context requires otherwise, throughout the specification and claims that follow, the word "comprise" and variations thereof, such as "comprises" and "comprising," are to be construed in an open, inclusive sense, that is, as "including, but not limited to."

Reference throughout this specification to "one embodiment" or "an embodiment" means that a particular feature, structure or characteristic described in connection with the embodiment is included in at least one embodiment. Thus, the appearances of the phrases "in one embodiment" or "in an embodiment" in various places throughout this specification are not

necessarily all referring to the same embodiment. Further more, the particular features, structures, or characteristics may be combined in any suitable manner in one or more embodiments.

As used in this specification and the appended claims, the singular forms "a," "an," and "the" include plural referents unless the content clearly dictates otherwise. It should also be noted that the term "or" is generally employed in its sense including "and/or" unless the content clearly dictates otherwise.

The headings and Abstract of the Disclosure provided herein are for convenience only and do not interpret the scope or meaning of the embodiments.

The user interface disclosed herein is described in the context of managing download and configuration tasks, jobs, assignments, and the like (referred to herein without limitation as assignments) for at least one electronic game machine (EGM) and preferably for a network of EGMs. However, it is to be understood that the user interface of the present disclosure can be adapted for use in other systems and networks that manage multiple machines, including without limitation computers, cash registers, remote teller machines, ATM machines, vending machines, toll booths, security systems, and ticket dispensing machines.

In the context of the present disclosure, various network systems and user interface methods for managing networked gaming machines are disclosed herein. The operational environment of the present interface system is described in the pending application referenced above, PCT US/2006/035556, of which a portion of the description is set forth below in conjunction with Figures 1-5 from the above-referenced application.

The disclosed system supports data downloads and configuration of gaming machines such as, but not limited to, electronic gaming machines (EGMs). With the network system, a casino operator may define a collection of EGMs, assign modules to one or more collections of EGMs, assign

configuration changes to one or more collections of EGMs, and schedule all assignments.

The system permits some or all of the software that is to be run on an EGM to be centrally stored in a software distribution library located on one or more servers. The software can be sent over a network (such as an Ethernet network) as desired for initial set up of an EGM, as updating of an EGM, game replacement, configuration, or any other desired purpose.

The system utilizes multicasting to download software packages to the EGMs in the background, even while full game play is ongoing. The server does a single multicast transfer of one or more packages to all target machines. Each EGM tracks the packages and notes missing, corrupted, or dropped packets. In one embodiment, the server receives requests from all EGMs for packets to resend and again multicasts them to all EGMs, regardless of which EGM requested which packet. The individual EGMs accept needed missing packets and ignore other re-broadcast packets. It should be noted that the system is not limited to multicast transmissions but may use HTTP, HTTPS, FTP, SFTP, and other transmission protocols.

It should be noted that the term EGM is intended to encompass any type of gaming machine, including hand-held devices used as gaming machines such as cellular based devices (e.g. phones), PDAs, or the like. The EGM can be represented by any network node that can implement a game and is not limited to cabinet based machines. The system has equal applicability to gaming machines implemented as part of video gaming consoles or handheld or other portable devices. In one embodiment, a geo-location device in the handheld or portable gaming device may be used to locate a specific player for regulatory and other purposes. Geo-location techniques that can be used include by way of example, and not by way of limitation, IP address lookup, GPS, cell phone tower location, cell DD, known Wireless Access Point location, Wi-Fi connection used, phone number, physical wire or port on client device, or by middle tier or backend server accessed. In one embodiment, GPS and

biometric devices are built within a player's client device, which in one embodiment, comprises a player's own personal computing device, or provided by the casino as an add-on device using USB, Bluetooth, IRDA, serial or other interface to the hardware to enable jurisdictionally compliant gaming, ensuring the location of play and the identity of the player. In another embodiment, the casino provides an entire personal computing device with these devices built in, such as a tablet type computing device, PDA, cell phone or other type of computing device capable of playing system games.

An embodiment of a network that may be used with the system is illustrated in Figure 1. The software distribution network consists of a top level vender distribution point 101 that contains all packages for all jurisdictions, one or more Jurisdiction distribution points 102A and 102B that contain regulator approved production signed packages used within that jurisdiction or sub-jurisdiction, one or more Software Management Points 103 A and 103B to schedule and control the downloading of packages to the EGM and a one or more Software Distribution Points 104A and 104B that contain regulator approved production signed packages only used in the gaming establishment that it supports. The Software Distribution Points (SDPs) 104A and 104B can communicate with Systems Management Points (SMPs) 105A and 105B, respectively as well as directly to one or more EGMs 106A and 106B. (The communication between SDP and SMP is optional. A benefit is to permit the SMP and the user access to the catalogue of software packages available on the SDP). The system allows for rapid and secure distribution of new games and OS's from a centralized point. It makes it possible to update and modify existing gaming machines with fixes and updates to programs as well as providing modifications to such files as screen images, video, sound, pay tables and other EGM control and support files. It provides complete control of gaming machines from a centralized control and distribution point and can minimize the need and delay of human intervention at the EGM.

In one embodiment, a 1 GB compact flash card will initially be used as the storage media in the EGMs for OS and download package storage. A second compact flash will be used for the storage of the production game as it exists today. The OS compact flash will be partitioned into production, backup, and download areas. The production area contains the active OS and game management software used to allow the playing of games on the EGM. This part of storage will have all the file integrity checking and validation performed on it. The other areas will be used to store backup software and data, new download packages, installation of the new packages and saving of NVRAM, counters, random number data and other secure information in an encrypted data format. Software downloads will be scheduled by gaming personnel via the SMPs. The SMP notifies the individual gaming machines they have been scheduled to receive a download package from a specific vendor Software Distribution Point (SDP). The EGM shall then request the SDP to start sending the download package to it.

The system solves the network bandwidth usage problem by using a multicast network protocol, which the SDP uses to broadcast the transfer to multiple EGM's simultaneously. The SDP will send a single transfer to a defined set of EGM's, thereby using a fraction of the network bandwidth when compared to the point-to-point strategy. If there are X groups of EGM's, with each group consisting of Y EGM's, then the following calculations provide a general sense of reduced network bandwidth usage:

Multicast: X * transferSize

Point-to-point: (X * Y) * transferSize

Gross Improvement: 1/Y network usage with multicast.

The system includes a recovery scheme to accommodate transmission error or reception error of the package. This recovery scheme allows each EGM to report missing pieces (packets) where either not received or received with errors (data corruption). The SDP logic can accumulate a list of

missing packets and the corresponding EGM's, and use the multicast protocol to resend the missing packets.

The specific packages that are transferred are specified by the EGM. The EGM makes requests to the SDP using a point-to-point protocol. The SDP may not service these requests immediately, so the EGM is permitted to operate while waiting for the transfer and during the transfer. The EGM is responsible for logging transfers and checking the validity of the completed transfer. This log can be used to reconcile the package content of the EGM at any given time. Additionally, the SDP will log transfer activity and provide a means to cross check the package content of an EGM.

An EGM validates transferred packages using a digital signature. This digital signature provides authentication of the package itself regardless of the specific SDP server that provided it, and is used to determine if the package incurred any transmissions errors. If an EGM is unable to validate a package, then the package will not be installed or otherwise used.

The system includes technology necessary to determine if an EGM contains packages necessary to use a new package (package dependencies). If a given package-A requires additional packages, package-B and package-C, to operate on the EGM, the dependencies can be determined from a package header contained within package-A. The SMP can then determine if the EGM already contains the dependent packages, package-B and package-C, before requesting that package-A be transferred to the EGM. Similarly, the SMP can automatically select the dependent packages to be transferred in addition to the original package-A, thus fulfilling the dependencies of package-A. This aspect of the system provides a level of assurance that the packages transferred can actually be used by the EGM.

The system includes technology necessary to determine if a package's dependent hardware components (components) are available on the EGM. If package-X requires component- Y and component-Z, the SMP can read the components of the EGM and use this information for component

dependency checking. If a package's required component dependencies are available on the EGM, then the package transfer to the EGM will be permitted; otherwise, the package transfer will not be permitted. This aspect of the system provides a level of assurance that the packages transferred can actually be used by the EGM.

The ability to transfer software directly to the EGM provides additional alternatives when installing a new EGM on the casino floor. An EGM that is manufactured and placed directly on the casino floor can be initially loaded with a boot-strap media (EPROM, CD-ROM, Compact Flash, etc.) that can boot the EGM and prepare communications before requesting new packages to install. Two boot-strap methods may be used to transfer software to an EGM. Both methods require minimal configuration of an EGM before the boot-strap process can be completed. Minimal configuration consists of the following:

1) A unique network IP Address for the EGM. This can be provided by a DHCP system, or assigned to the EGM via operator configuration.

2) A unique EGM identifier that is derived from the EGM hardware (such as network card MAC address) or assigned to the EGM via operator configuration.

3) An address of the SDP network server to make transfer requests of. This can be a hard-coded default address that may be overridden via operator configuration.

The simple boot-strap mode requires that EGM is also configured with the necessary options to locate the SMP network server. This can be a hard-coded default address that may be overridden via operator configuration. This address is used by the EGM to locate the SMP server, which can then assign new packages to the EGM for transfer. In one embodiment, the EGM and its associated peripherals can be identified using a scheme such as is described in U. S. Patent Application 11/319,034 entitled "Device Identification"

and incorporated in its entirety herein by reference. In this embodiment, during start-up, a device sends its MAC address out on the network. A local switch collects MAC and IP addresses for the devices connected to it. Periodically, the switch transmits raw Ethernet frames, USB packets, or TCP packets containing tables of devices and associated MAC/IP addresses. When a device receives information about another device, the device may attempt communication with that device. First, a verification procedure is used to validate the devices. Subsequently, communication is possible between the devices.

The advanced boot-strap mode involves the EGM requesting a default boot-strap package from the SDP. The package name for this boot-strap package is predefined and known to both the EGM and the SDP. This boot-strap package contains configuration files that identify the default software packages and configuration data. The boot-strap package can be customized for the specific installation (casino) and stored on the SDP. This way all EGM' s will request the predefined boot-strap package, the SDP will transfer the boot-strap package to the EGM, and the EGM will process the boot-strap package. When the EGM processes the boot-strap package, the EGM can read the SMP address, default EGM OS package, and other packages such as default peripheral firmware packages. At this point the EGM can request these packages from the SDP, and the SDP will begin transferring the default packages to the EGM, ultimately ending with an EGM ready for configuration.

Figure 2 illustrates an overview of download interaction between the System Management Point 105, EGM 106, and Software Distribution Point 104. The design consists of a download device driver which provides the central control logic and logic. It interfaces with the Download Class (which may be G2S or any other protocol) and a download control thread. The download control thread controls requesting and downloading packages from the download distribution point server.

In operation, the System Management Point 105 sends package commands and requests to the Download Control 206 of EGM 106 to tell it to

add and delete packages, install packages and request information about packages and installed modules. The Download Control 206 will filter these requests and pass then to the Download Driver 207 for processing. Requests to add packages are then given to the Download Receiver Process 208 which then opens communications with the Software Distribution Point 104. The Download Receiver Process 208 receives the package data from the Software Distribution Point 104 via a multicast (or any other defined transmission protocol) link 204 in multiple packets and assembles the package in compact flash. Once all packets of the package have been received, the package data is verified using a SHA-I verification string passed in the package header. The appropriate status and package state information shall be passed back to the Download Driver 207 which passes it to the Download Control 206 for logging and passing back to the System Management Point.

Download Driver 207

The download driver is installed when the EGM system is started. It's primary functions are to:

1 - Initialize package and control information as stored on the compact flash.

2 - Process commands to add packages and install rules from the Download Control 206.

3 - Supply the Download Control 206 with list and status information and download and install events.

4 - Queue add package requests for the Download Receiver process 208 and pass them to the Download Receiver 208 via a read 211 to the driver.

5 - Update package the status and error conditions of packages and install rules and pass them back to the Download Control Process 206.

Download Control Thread 206

The Download Control Process 206 is either a thread running within the game manager context or a standalone process when there is no game manager running on the system. It's primary tasks are:

1 - Receives download commands

2 - Passes the commands that it will not process on to the download driver 207 for delivery to the Download Receiver process 208 and the Download Installer process 209.

3 - Receive download and install status from the driver and log in the appropriate log files on the EGM 106.

4 - Supply the log information and package, install rule and module lists to the System Management Point 105 when requested.

Download Receiver Process 208

The Download Receiver Process 208 is responsible for communications between the EGM 106 and the Software Distribution Point 104. It's primary functions are:

1 - Receive add package requests from the Download Driver 207.

2 - Open a TCPIP link to the Software Distribution Point 104 and request the package defined within the add package request be sent to the EGM 106.

3 - Open a multicast communications port specified by the Software Distribution Point 104 and receive the requested package in multiple packet format.

4 - Assemble the packets of the package into a single file.

5 - Request any missed or missing packets be resent.

6 - Verify the package data by calculating a SHA-I value and matching that against the SHA-I validation string passed in the package header.

7 - Send back status and error information of the package receive process to the Download Control Process 206 via the Download Driver 207.

Download Package Installer Process 209

The Download Package Installer Process 209 is responsible for installing the packages that are downloaded from the Software Distribution Point 104. It is notified of when to start installing a downloaded package when a set install rule command is sent from the Systems Management Point 105. It receives this command via a read to the Download Driver 207. The basic functions of the Download Package Installer Process 109 are:

1 - Parse the set install rules to determine how to process the package.

2 - Disable the EGM 106 based on the instructions in the set install rule command.

3 - Install the package using the command file sent as part of the download package.

4 - Clear NVRAM as instructed by the install rules. [0089] 5 - Reboot the EGM 106 as instructed by the set install rules. [0090] 6 - Pass back status and error information via the Download Driver 207. [0091] EGM Initialization

Figure 3 illustrates an overview of EGM startup message flow. EGM 106 sends a message 302 to SDP 104 requesting a package. SDP 104 replies with a message 303 indicating package size and the multicast IP/Socket that will be used for transmission. Message 304 from SDP 104 to EGM 106 is the package data itself. If necessary, EGM 106 can request missed packets by sending a message 305.

As noted above, the system uses multicasting to send packages to multiple EGMs. When packets are missed and requested by an EGM, SDP 104 collects a number of requests and sends out multicasts of all missed packets to all of the EGMs. If a particular EGM does not need one of these re-broadcast packets, it is ignored. Otherwise, the EGM accepts the re- broadcast packet and uses it to complete the package transmission.

Download Driver

In the interface between the download driver 207 and the download processes, all set status, set error information and command information is passed to the driver 207 via an IOCTL call. The processes use the read interface to get processing instructions from the driver 207. The Download Receiver 208 uses the read interface to get add package requests, the Download Installer Process 209 uses the read to obtain set install rule requests, and the Download Control Process 206 uses the read interface to receive the status and error information from the Receiver and Install processes.

/dev/download is a loadable module, "device driver", that is loaded as the system comes up. (It is a Linux loadable module in one embodiment). DI_Init_module() is the first routine to gain control when the module is loaded. It will:

1 - Allocate the download buffer used to store Download Class messages, (e.g. the G2S or other protocol)

2 - Register the download support as a Linux kernel device.

3 - Initialize the read queues fro the processes that communicate with the driver.

4 - Initialize the storage area for package and set install rules that may be received.

Once the driver has been installed and initialized, all further actions will controlled by the ioctl and read function entry points into the driver. The ioctl entry point, DL_ioctl(), will process all the System Management Point 105 requests passed through the Download Control Process received from the download class. The DLjreadQ entry point services all the read requests from:

- The Download Control Process 206 for packages status and error,
- The Download Receiver process 208 to pass add package requests.
- The Download Installer 209 to receive set install rule request.

Figure 4 depicts the logic flow of /dev/download driver initialization. At step 401 the Read Queue areas and semaphores are initialized. At step 402 the package is initialized and the Install Rule Data Structures are set. At step 403 the driver is registered with the system and at step 404 the system returns.

Once the /dev/download driver is installed and initialized, the dl_ioctl() driver 207 serves as the interface to the Download Control Process 206, and the Receiver 208 and Install 209 processes. All requests from System Management Point 105 are passed through Download Control Process 206 and either passed onto the driver 207 to be processed, or in the case of module and log requests, processed within the Download Control Process 206 itself. The dl_ioctl() is a command ID to determine what to do with the specific request. Figure 5 is a flow diagram illustrating a routine to validate download requests and pass control to the proper handler function.

At step 501 shown in Figure 5, the particular switch that is set on the Command ID is examined so that the request from the SMP 105 can be routed to the correct handler. Add and delete Package commands are provide to DL_Packages 502. Set and Delete Install rules are passed to DL_Install 503. Set Status, Retrieve Status, Retrieve List commands are sent to DL_Status 504. Register processes are sent to block 505 where the Pid of the Process being registered is saved. Unknown commands return an error event at 506.

The scheduling and management of the various operations associated with the foregoing download and configuration processes is accomplished via the user interface (UI) shown in Figures 6A and 6B. Generally the UI 610 is adapted for display on a display device, such as a display screen, CRT, computer screen, television, handheld displays, PDA, and the like. In one embodiment the UI 610 is implemented from instructions or executable code stored on a computer readable medium, such as a processor described above in connection with Figures 1-5. The UI 610 enables a user to display information about the operation of selected devices coupled to the network,

particularly with respect to scheduled operations presented on the user interface, and to control the scheduling of the operations, such as start time and cancellation thereof.

In one embodiment, a system is provided that generally includes a network of devices, such as EGMs, a display device, and a processor coupled to the network of devices and the display device and configured to generate a UI 610 (shown in Figures 6A and 6B) on the display device for display and user management of network operations. The UI 610 includes assignment interfaces, with each assignment interface providing information about an associated operation, a calendar interface on which the assignment interfaces are displayed, the calendar interface providing at least date and time information.

The UI 610 as shown in Figures 6A and 6B is encompassed in a display area 612 that includes a calendar interface 614 and a plurality of assignment interfaces 616 displayed thereon. Ideally the calendar interface 614 includes at least one timeline header 618 that presents at least one of a date or a time or both date and time. In the illustrated embodiment the timeline 618 is presented horizontally across the top of the calendar interface 614 with vertical lines 619 presented across the interface 614 in alignment with the hours depicted in the timeline 618.

The display area 612 also includes a columnar menu 620 on the left side, a Windows-style tool bar 622 across the top, and a horizontal status bar 624 across the bottom, thus partially framing the calendar interface 614.

Immediately below the first tool bar 622 is a UI tool bar 626 presented horizontally above the timeline 618. Included in this UI tool bar 626 are three filters 628, 630, 632 and a cancel jobs button 634.

The first filter 628 is a Zoom function presented as a window that allows an operator to control how much detail is shown on the timeline 618. Choices can include: Hour, Day, and Week views.

The second filter 630 is Date Range function, also presented as a window, which allows the user to specify how much time before and after the current time the data should be displayed. Choices include: Today Only, Plus or Minus One Day, Plus or Minus One Week, Plus or Minus One Month. The user would use a report look at activity in the past that is outside of the one month range. Reports can be generated for display and printed on paper as well for activities within the date ranges set forth above.

The third filter 632 is a Job Status function, again presented as a window, that allows the user to only show jobs or assignments matching the status criteria. The choices here include: All, Active, Running, Pending

The Cancel Jobs button 634 activates a function that enables a user to select assignments for cancellation that have not yet been completed.

Each assignment interface 616 is presented on the calendar interface 614 in a manner that shows the start time for each associated operation. Ideally, there is a vertical bar 637 of a distinguishing color, such as lime green, that indicates the current time as a reference. The vertical bar 637 is perpendicular to the timeline 618 as shown in Figures 6A and 6B. In the depicted embodiment, each assignment interface 616 is represented as a rectangular box with the left vertical line of the box positioned in alignment with the vertical position of the start time shown on the timeline 618. The interior 636 of the assignment interface 616 includes an assignment identifier 638 associated with an operation, as shown in the enlarged view of Figure 7.

The entire content of the calendar interface 614 can be scrolled in any direction, however when scrolling vertically, the header timeline 618 showing the dates and hours will always remain visible.

The assignment interfaces 616 are preferably color coded to indicate their status. The start time is indicated by a vertical black bar 640 on the left ending in the diamond 642. If the assignment is an override, then the duration of the override is indicated by the horizontal line 644 extending to the

terminator with the second diamond 646. In this sample shown in Figure 7, there is a 3 hour configuration override that resulted in an error.

Hovering a mouse cursor over an assignment interface 616 will bring up a tooltip 648 as shown in Figures 6A and 6B. The tooltip provides additional details about the operation associated with the assignment interface 616. More detail can be provided in the tooltip than shown in this screen shot of Figure 7, such as a summary of the EGMs affected by the operation associated with this assignment interface 616.

Two unique concepts are displayed in the screen shot of Figure 7. The first is the linking of related operations, such as recurring scheduled assignments and the second is assignment interfaces that run multiple operations based on templates.

The first row of horizontally presented assignment interfaces 616 shown in Figures 6A and 6B represent a recurring schedule of operations. Each horizontal curved linking line 652 is a visual relational indicator or icon that links a previous assignment interface 654 with the next occurring assignment interface 656. Here, the linking line 652 extends from an end or finish of the first assignment interface 654 to the beginning or start of the following assignment interface 656.

A Multi-Job Assignment interface template 652 shown at about 3:30 am on Monday represents an assignment that was set up to run two operations in sequence. In this case it is showing two downloads, but it might be a download followed by a configuration, or any combination of two or more operations. A vertically oriented curved linking line 653 connects the related assignment templates 652, 655.

Assignment interfaces 616 can be configured so that they schedule one or more operation templates describing what to do. In this way the actual instructions about what to configure or download can be shared at different times or for different EGMs without doing a clumsy copy step.

Double-Clicking or Opening an assignment interface 616 will open a detail window much like an expandable grid, but focused only on the operation in question. An operator or user will be able to see the individual status for each EGM and, in the case of configurations, for each option. Two or more of these can be opened at once to aid in comparing results.

Assignment interfaces 616 are located or positioned on the calendar interface 614 in a manner that keeps the presentation compact while at the same time avoiding overlaps. Ideally, all recurring operations scheduled during the time period of interest will be displayed near the top of the calendar interface 614 as shown so that they are horizontally aligned. Each operation will then be taken in start time order and displayed beginning at the top left corner of the calendar interface 614. Each subsequent assignment interface 616 is positioned one row below the ones already displayed until the start time has shifted forward far enough that the next subsequent assignment interface 616 can be presented in a new column without overwriting the previous ones. At this point, the current row is reset to the top and the next column of assignment interfaces 616 begins again.

The various embodiments described above can be combined to provide further embodiments. All of the U.S. patents, U.S. patent application publications, U.S. patent applications, foreign patents, foreign patent applications and non-patent publications referred to in this specification and/or listed in the Application Data Sheet, including but not limited to [insert list], are incorporated herein by reference, in their entirety. Aspects of the embodiments can be modified, if necessary to employ concepts of the various patents, applications and publications to provide yet further embodiments.

For example, each assignment interface rectangle 616 can include one or more progress indicators that will show in real time how much of the operation has been completed. Because an operation could involve more than one EGM, the percentage shown could be the average percent complete for all EGMs, or the percent complete for the one that is least complete (in order

to give operator some idea of how much longer it will take to complete the operation), or some other useful measure of completion. The UI 610 will also provide some obvious indicator that an operation has had some kind of error and the operator or user will be able to select the indicator, such as clicking with a mouse device on that indicator, to see details of the error.

These and other changes can be made to the embodiments in light of the above-detailed description. In general, in the following claims, the terms used should not be construed to limit the claims to the specific embodiments disclosed in the specification and the claims, but should be construed to include all possible embodiments along with the full scope of equivalents to which such claims are entitled. Accordingly, the claims are not limited by the disclosure.

CLAIMS

1. A system, comprising:

a network of devices;

a display device; and

a processor coupled to the network of devices and the display device and configured to generate a user interface on the display device for display and user management of network operations, the user interface including assignment interfaces, each assignment interface providing information about an associated operation, a calendar interface on which the assignment interfaces are displayed, the calendar interface providing at least date and time information, and at least one graphical linking indicator that provides visual relational information of related operations.

2. The system of claim 1 wherein the assignment interface provides information about at least one of an assignment identifier, an operation status, an error indicator, and a start time.

3. The system of claim 1 wherein the assignment interface provides information about selected devices coupled to the network.

4. The system of claim 1 wherein the assignment interface comprises a user-activated tooltip.

5. The system of claim 1 wherein the calendar interface comprises a timeline displaying at least one of a date and time, and a graphical indicator interposed on the timeline to indicate at least one of current date and time.

6. The system of claim 5 wherein the calendar interface, assignment interface, and graphical linking indicator can be scrolled while the timeline remains visible during scrolling.

7. The system of claim 1 wherein the user interface further comprises a tool bar displaying at least one filter of one from among display detail, range of displayed date or time, and operation status of an assignment.

8. The system of claim 7 wherein the tool bar further comprises a cancel selection that enables a user to cancel selected assignments.

9. The system of claim 1 wherein assignment interfaces for recurring operations are displayed in horizontal alignment at a top region of the calendar interface and assignment interfaces for non-recurring operations are displayed below the assignment interfaces for recurring operations in a column format.

10. The system of claim 1 wherein the processor comprises a computer-readable medium and the computer-readable medium has instructions stored thereon that when executed cause the processor to generate the user interface.

11. A method of providing a user interface for management of network operations, the method comprising:

presenting a time line along a time line axis with information that is indicative of at least one of dates or time; and

for each of a number of operations, presenting a respective assignment icon with a start point of the assignment icon positioned along the time line axis at a point that is indicative of a time that an operation represented by the assignment icon is scheduled to start.

12. The method of claim 11, further comprising:

for at least two of the operations that reoccur over time, presenting a linking icon between at least two assignment icons that visually represent a relational connection between the at least two operations associated with the at least two assignment icons that reoccur over time.

13. The method of claim 12 wherein presenting a linking icon between the assignment icons that represent the at least two operations that reoccur over time includes presenting a line that extends from at least proximate a finish of a first one of the assignment icons to at least proximate a start of a second one of the assignment icons.

14. The method of claim 11, further comprising:

for at least two of the operations that are logically connected, presenting a multi-operation linking icon between the assignment icons that represent the at least two operations that are logically connected.

15. The method of claim 14 wherein presenting a multi-operation linking icon between the assignment icons that represent the at least two operations that are logically connected includes presenting a line that extends from at least proximate a start of a first one of the assignment icons to at least proximate a start of a second one of the assignment icons.

16. The method of claim 11 wherein presenting a respective assignment icon includes presenting the respective assignment icon with a start marker extending perpendicular to the time line axis and with a termination marker extending perpendicular to the time line axis.

17. The method of claim 16, further comprising:

presenting the termination marker at a point along the time line axis indicative of an extended time; and

presenting an extension icon proximate one of the assignment icons where a termination time of the assignment represented by the assignment icon has been extended.

18. The method of claim 17 wherein presenting an extension icon includes presenting a line extending between the start marker and the termination marker.

19. The method of claim 11, further comprising, in response to a positioning of a cursor over a selected assignment icon, presenting detailed information about at least one aspect associated with the selected assignment icon.

20. The method of claim 19 wherein presenting detailed information about at least one aspect associated with the selected assignment icon includes presenting a list of devices affected by the operation represented by the selected assignment icon.

21. The method of claim 11, further comprising:

scrolling across a display of the assignment icons in a direction perpendicular to the time line axis in response to a user input while maintaining the display of the time line axis.

22. The method of claim 11, further comprising:

presenting a current time icon positioned along the time line axis at a point that is indicative of at least one of a current date and a current time.

23. The method of claim 22 wherein presenting a current time icon includes presenting a line extending perpendicular to the time line axis.

24. The method of claim 11, further comprising:

controlling a resolution of the time line that is presented based on a user selected input indicative of a level of presentation detail.

25. The method of claim 11, further comprising:

controlling a length of the time line that is presented in response to a user selected input indicative of a range of time line length.

26. The method of claim 11, further comprising:

in response to a user input indicative of operations to present that correspond to a user identified operation status, selecting a subset of all of the operations for the presenting of a respective assignment icon in which the associated operation is in the subset are each operation currently logically associated with the status.

27. The method of claim 26 wherein presenting a respective assignment icon includes presenting a human-readable title that is logically associated with the operation.

28. The method of claim 11, further comprising:

color coding the assignment icons based on a respective assignment type logically associated with the operation represented by the assignment icon.

29. The method of claim 28, further comprising:

determining the assignment type logically associated with the operation based at least in part on a title logically associated with the assignment icon.

30. The method of claim 28, further comprising:

stopping the presenting of a selected one of the assignment icons in response to a user input indicative of deletion of the operation represented by the selected assignment icon associated with the operation.