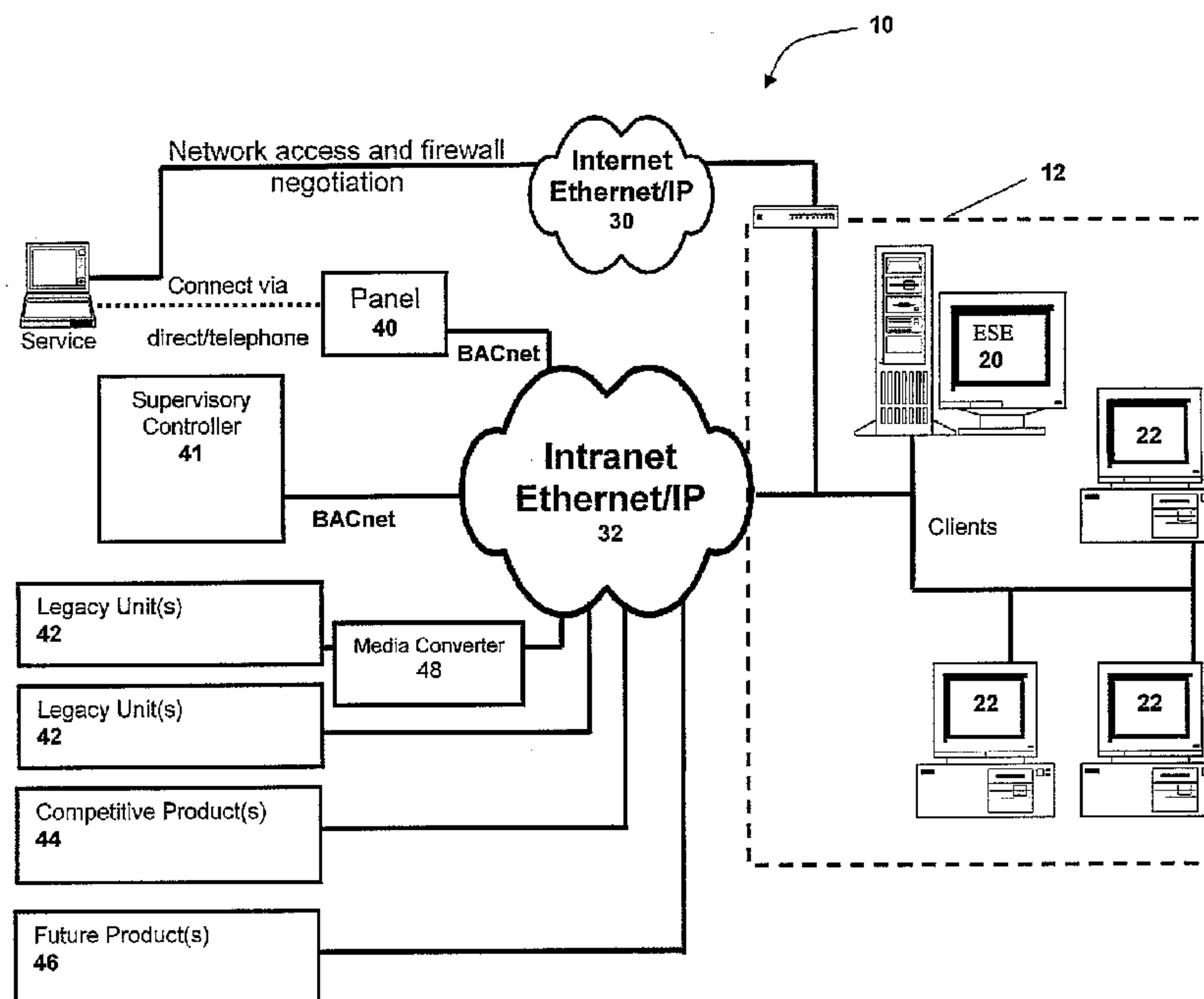




(86) Date de dépôt PCT/PCT Filing Date: 2006/08/15
(87) Date publication PCT/PCT Publication Date: 2007/03/01
(45) Date de délivrance/Issue Date: 2015/10/13
(85) Entrée phase nationale/National Entry: 2008/02/21
(86) N° demande PCT/PCT Application No.: US 2006/031863
(87) N° publication PCT/PCT Publication No.: 2007/024573
(30) Priorités/Priorities: 2005/08/22 (US11/208,773);
2005/12/22 (US11/316,699); 2005/12/22 (US11/316,687);
2005/12/22 (US11/316,695); 2005/12/22 (US11/316,410);
2005/12/22 (US11/316,697); 2005/12/22 (US11/316,703);
2005/12/22 (US11/316,698); 2005/12/22 (US11/316,702)

(51) Cl.Int./Int.Cl. *G05B 11/01* (2006.01),
G01M 1/38 (2006.01), *G05B 13/00* (2006.01),
G05B 15/00 (2006.01), *G05B 21/00* (2006.01),
G05D 23/00 (2006.01)
(72) Inventeurs/Inventors:
MCCOY, SEAN M., US;
RICHARDS, DAVID M., US
(73) Propriétaire/Owner:
TRANE INTERNATIONAL INC., US
(74) Agent: BORDEN LADNER GERVAIS LLP

(54) Titre : SYSTEME DE CONTROLE AUTOMATIQUE DE BATIMENTS A EXTENSION DYNAMIQUE ET
CONFIGURATION AUTOMATIQUE, ET ARCHITECTURE
(54) Title: DYNAMICALLY EXTENSIBLE AND AUTOMATICALLY CONFIGURABLE BUILDING AUTOMATION SYSTEM
AND ARCHITECTURE



(57) **Abrégé/Abstract:**

A building automation system (BAS) architecture is disclosed. The BAS comprises, in one embodiment, an architecture comprising a communication network and having a dynamic extensibility capability and an automatic configuration capability; an engine communicatively coupled to the communication network; and at least one control device communicatively coupled to the communication network, the control device being known or unknown to the engine. The engine can be adapted to selectively implement the dynamic extensibility capability to establish communications with and to control both known and unknown control devices. The engine can be further adapted to selectively implement the automatic configuration capability to determine at least one characteristic of both known and unknown control devices. A method of adding a control device to a building automation system (BAS) by dynamically extending and automatically configuring an architecture of the BAS is also disclosed.



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

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
1 March 2007 (01.03.2007)

PCT

(10) International Publication Number
WO 2007/024573 A2

(51) International Patent Classification:

G05B 11/01 (2006.01) *G05B 15/00* (2006.01)
G01M 1/38 (2006.01) *G05B 21/00* (2006.01)
G05B 13/00 (2006.01) *G05D 23/00* (2006.01)

(21) International Application Number:

PCT/US2006/031863

(22) International Filing Date: 15 August 2006 (15.08.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

11/208,773	22 August 2005 (22.08.2005)	US
11/316,410	22 December 2005 (22.12.2005)	US
11/316,695	22 December 2005 (22.12.2005)	US
11/316,702	22 December 2005 (22.12.2005)	US
11/316,698	22 December 2005 (22.12.2005)	US
11/316,703	22 December 2005 (22.12.2005)	US
11/316,697	22 December 2005 (22.12.2005)	US
11/316,687	22 December 2005 (22.12.2005)	US
11/316,699	22 December 2005 (22.12.2005)	US

(71) Applicant: **AMERICAN STANDARD INTERNATIONAL, INC.** [US/US]; 1370 Avenue Of The Americas, New York, NY 10019 (US).

(72) Inventors; and

(75) Inventors/Applicants (*for US only*): **MCCOY, Sean, M.** [US/US]; C/O TRANE, 12-1, 3600 Pammel Creek Road, La Crosse, WI 54601 (US). **RICHARDS, David, M.** [US/US]; C/O TRANE, 12-1, 3600 Pammel Creek Road, La Crosse, WI 54601 (US).

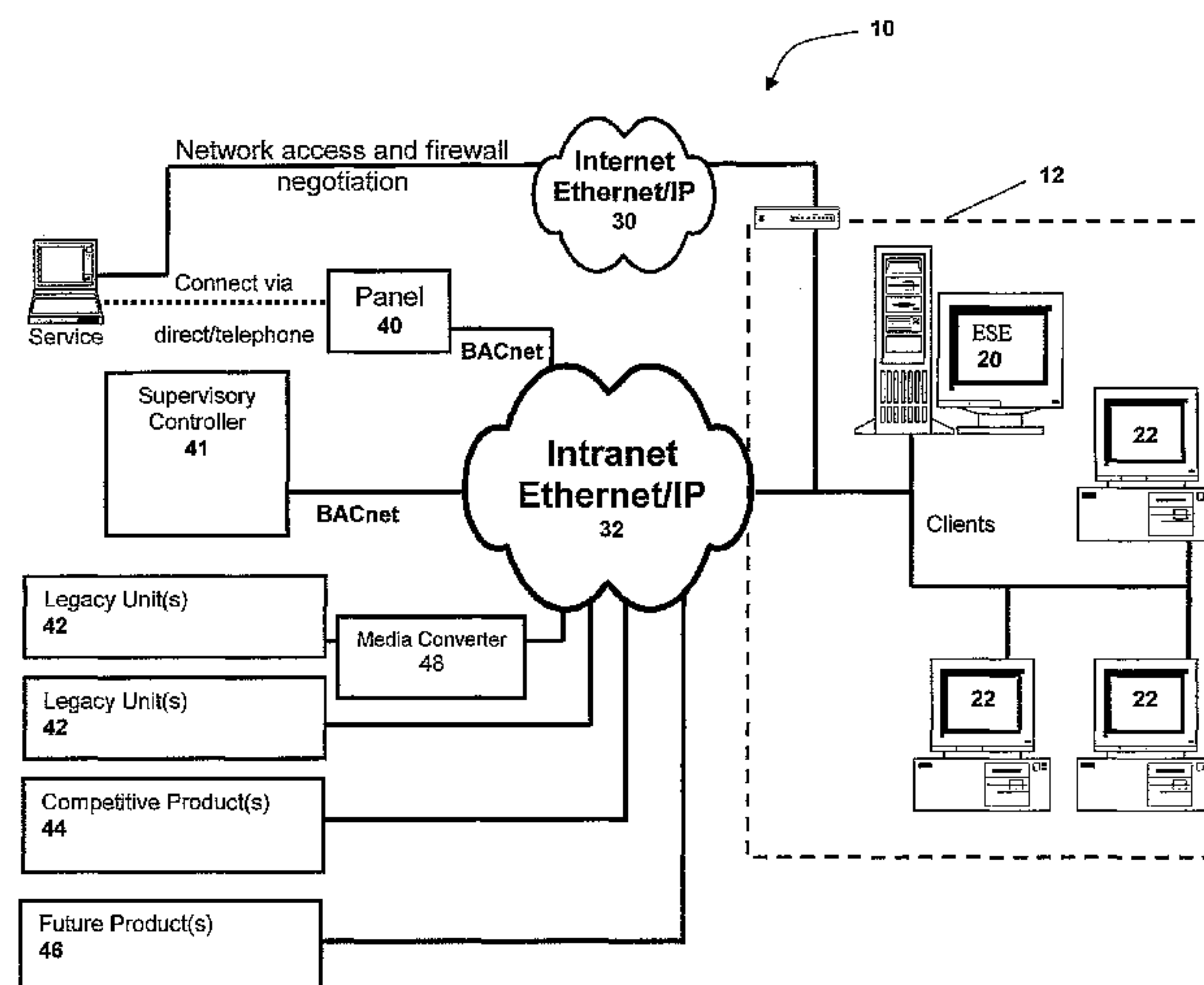
(74) Agents: **SALMELA, Amy** et al.; PATTERSON, THUENTE, SKAAR & CHRISTENSEN P.A., 4800 IDS Center, 80 South Eight Street, Minneapolis, MN 55402-2100 (US).

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

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI,

[Continued on next page]

(54) Title: DYNAMICALLY EXTENSIBLE AND AUTOMATICALLY CONFIGURABLE BUILDING AUTOMATION SYSTEM AND ARCHITECTURE



(57) Abstract: A building automation system (BAS) architecture is disclosed. The BAS comprises, in one embodiment, an architecture comprising a communication network and having a dynamic extensibility capability and an automatic configuration capability; an engine communicatively coupled to the communication network; and at least one control device communicatively coupled to the communication network, the control device being known or unknown to the engine. The engine can be adapted to selectively implement the dynamic extensibility capability to establish communications with and to control both known and unknown control devices. The engine can be further adapted to selectively implement the automatic configuration capability to determine at least one characteristic of both known and unknown control devices. A method of adding a control device to a building automation system (BAS) by dynamically extending and automatically configuring an architecture of the BAS is also disclosed.

WO 2007/024573 A2

WO 2007/024573 A2



FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT,
RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA,
GN, GQ, GW, ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:

— *without international search report and to be republished
upon receipt of that report*

DYNAMICALLY EXTENSIBLE AND AUTOMATICALLY CONFIGURABLE BUILDING AUTOMATION SYSTEM AND ARCHITECTURE

5 FIELD OF THE INVENTION

The present invention relates generally to building automation systems. More particularly, the present invention relates to building automation system architectures, communications, and configurations.

10 BACKGROUND OF THE INVENTION

Building automation systems (BAS) are used to coordinate, manage, and automate control of diverse environmental, physical, and electrical building subsystems, particularly HVAC and climate control but also including security, lighting, power, and the like. Typical existing BAS systems are hardwired or use a proprietary communication standard or protocol to
15 link the various subsystems and provide system-wide user access and control.

Hardwiring and manual programming of BAS systems can create a robust fixed system customized for a particular installation. These systems, however, often require extensive customization for each building or site. Particular manual programming and other installation elements may not be applicable to other systems, contributing to the costliness and time-
20 consuming installation associated with such systems.

Further, hardwired systems and those using proprietary communication standards and protocols are difficult or impossible to integrate with system components, panels, and other elements from different vendors or generations. For example, a campus of buildings in which an upgraded BAS is being installed may have existing previous generation (legacy) systems and
25 systems from more than one vendor. Installing a BAS and making it compatible with the existing systems in such a situation is time-consuming, requiring extensive manual service and programming to integrate the existing devices and implement the custom BAS.

With the introduction of BACnetTM, an ASHRAE (American Society of Heating, Refrigerating and Air-Conditioning Engineers) and ANSI (American National Standards
30 Institute) standard, and LonTalkTM, an integration approach developed by Echelon, some uniformity of standards and communications has been achieved in the industry. BACnetTM was intended to standardize HVAC interoperability and serve as a solution to industry-wide issues. In use, however, BACnetTM exists in multiple versions and includes various non-standard feature functions available to vendors. Many vendors dictate a particular BACnetTM version that must

be used in order to achieve system compliance, forcing BAS users to update. BACnetTM is therefore not completely interoperable across versions and features.

Additionally, BAS installation and maintenance are still generally labor-intensive custom tasks that vary with each system implementation. Upgrading, expanding, and updating or
5 removing system components and services in particular are also complex tasks, as the existing BAS may or may not support new devices and must be manually reconfigured to recognize and incorporate changes. In a common scenario, a user managing a building site with two control units operating in an existing BAS wants to add a third control unit in a newly constructed wing of the building. The user must upgrade the existing control units to the new version of the third
10 control unit in order for the system to be compliant because the system cannot accommodate multiple versions or integrate the new control unit.

Existing BASs also do not offer the accessibility, customization, and management tools desired by system users. Current BASs are difficult and communicatively cumbersome to manage on a large scale, such as by a regional or nationwide retailer or other organization.
15 Further, while Internet-based and accessible systems are presently available and in use, these systems suffer from several drawbacks. Many current Internet BASs were created as add-ons to existing BASs and thus have integrated and proprietary designs. These systems do not offer the adaptability and extensibility necessary to interface with non-native systems and sub-systems, a particular issue with respect to large-scale systems implemented in existing structures. Existing
20 system also do not provide higher-level extensibility, configurability, and customization tools. The Internet therefore presents a unique platform for which an advanced BAS could be designed, implemented, and managed.

Accordingly, a need remains for an intelligent BAS having a flexible and dynamic architecture and providing increased communication, management, and control options,
25 particularly from a user perspective.

SUMMARY OF THE INVENTION

The present invention substantially addresses the above-identified needs by providing a dynamically extendible and automatically configurable building automation system (BAS). The
30 BAS comprises, in one embodiment, an architecture comprising a communication network and having a dynamic extensibility capability and an automatic configuration capability; an engine communicatively coupled to the communication network; and at least one control device communicatively coupled to the communication network, the control device being known or unknown to the engine.

The engine of the BAS can be adapted to selectively implement the dynamic extensibility capability to establish communications with and to control both known and unknown control devices. The engine can be further adapted to selectively implement the automatic configuration capability to determine at least one characteristic of both known and unknown control devices.

5 The present invention also includes a method of adding a control device to a BAS by dynamically extending and automatically configuring an architecture of the BAS. In one embodiment, the method comprises obtaining a network address of a previously unknown control device at a site. A discovery process is then implemented to establish communications with and obtain metadata from the control device, and the site is synchronized by evaluating at
10 least one characteristic of the metadata and storing the at least one characteristic as a definition in a program of the architecture. A status of the control device is altered from known to unknown, and the architecture is dynamically extended and automatically configured by executing the program without recompilation.

15 The above summary of the invention is not intended to describe each illustrated embodiment or every implementation of the present invention. The figures and the detailed description that follow more particularly exemplify these embodiments.

BRIEF DESCRIPTION OF THE DRAWINGS

20 The invention may be more completely understood in consideration of the following detailed description of various embodiments of the invention in connection with the accompanying drawings, in which:

FIG. 1 is a building automation system (BAS) according to one embodiment of the invention.

FIG. 2 is an object diagram according to one embodiment of the invention.

25 FIG. 3 is an object model diagram according to one embodiment of the invention.

FIG. 4A is a data model block diagram according to one embodiment of the invention.

FIG. 4B is a data model block diagram according to one embodiment of the invention.

FIG. 4C is a data model example diagram according to one embodiment of the invention.

30 FIG. 5A is a simplified BAS architecture layer block diagram according to one embodiment of the invention.

FIG. 5B is a BAS architecture diagram according to one embodiment of the invention.

FIG. 6A is a start-up process flowchart according to one embodiment of the invention.

FIG. 6B is a data management sub-process flowchart according to one embodiment of the invention.

FIG. 7 is a site discovery process flowchart according to one embodiment of the invention.

FIG. 8 is a dynamic protocol support diagram according to one embodiment of the invention.

5 FIG. 9 is a site synchronization process flowchart according to one embodiment of the invention.

FIG. 10A is a site synchronization process flowchart according to one embodiment of the invention.

10 FIG. 10B is a site synchronization sub-process flowchart according to one embodiment of the invention.

FIG. 11 is a site removal process flowchart according to one embodiment of the invention.

FIG. 12 is a site synchronization process flowchart according to one embodiment of the invention.

15 FIG. 13 is an outside object data block diagram according to one embodiment of the invention.

FIG. 14 is a data block diagram according to one embodiment of the invention.

FIG. 15 is a flowchart according to one embodiment of the invention.

FIG. 16 is an alarm block diagram according to one embodiment of the invention.

20 FIG. 17 is a navigation diagram of a user interface according to one embodiment of the invention.

FIG. 18A is a user interface page according to one embodiment of the invention.

FIG. 18B is another user interface page according to one embodiment of the invention.

FIG. 19 is an attribute diagram according to one embodiment of the invention.

25 FIG. 20A is another user interface page according to one embodiment of the invention.

FIG. 20B is a detail view of the user interface page of FIG. 20A according to one embodiment of the invention.

FIG. 20C is another detail view of the user interface page of FIG. 20A according to one embodiment of the invention.

30 FIG. 20D is another user interface page according to one embodiment of the invention.

FIG. 21 is a user interface navigation diagram according to one embodiment of the invention.

FIG. 22A is a user interface page according to one embodiment of the invention.

FIG. 22B is a detail view of the user interface page of FIG. 22A according to one embodiment of the invention.

FIG. 23 is a data log block diagram according to one embodiment of the invention.

FIG. 24A is a user interface page according to one embodiment of the invention.

5 FIG. 24B is a detail view of the user interface page of FIG. 24A according to one embodiment of the invention.

FIG. 24C is a user interface page according to one embodiment of the invention.

FIG. 25 is a user interface page according to one embodiment of the invention.

FIG. 26A is a user interface page according to one embodiment of the invention.

10 FIG. 26B is a detail view of the user interface page of FIG. 26A according to one embodiment of the invention.

FIG. 26C is another detail view of the user interface page of FIG. 26A according to one embodiment of the invention.

FIG. 27 is a user interface page according to one embodiment of the invention.

15 FIG. 28 is a user interface navigation diagram according to one embodiment of the invention.

FIG. 29 is a user interface page according to one embodiment of the invention.

FIG. 30 is a user interface navigation diagram according to one embodiment of the invention.

20 FIG. 31 is a user interface page according to one embodiment of the invention.

FIG. 32 is a user interface page according to one embodiment of the invention.

FIG. 33 is a user interface page according to one embodiment of the invention.

FIG. 34 is a user interface page according to one embodiment of the invention.

25 FIG. 35A is a block diagram of a user interface page according to one embodiment of the invention.

FIG. 35B is a detail view of the user interface page of FIG. 35A according to one embodiment of the invention.

FIG. 35C is a detail view of the user interface page of FIG. 35A according to one embodiment of the invention.

30 FIG. 36 is a user interface navigation diagram according to one embodiment of the invention.

FIG. 37 is a user interface page according to one embodiment of the invention.

While the invention is amenable to various modifications and alternative forms, specifics thereof have been shown by way of example in the drawings and will be described in detail. It

should be understood, however, that the intention is not to limit the invention to the particular embodiments described. On the contrary, the intention is to cover all modifications, equivalents, and alternatives falling within the scope of the invention as defined by the appended claims.

5

DETAILED DESCRIPTION OF THE INVENTION

The systems and methods of the invention are particularly suited for a dynamically extensible and automatically configurable BAS and architecture. In one embodiment, systems and methods can effectively prioritize and manage data and information within a BAS. The

10 BAS and architecture provide an intelligent control system via a dynamically extensible and automatically configurable architecture. The system can be implemented locally or widely, from a space or building level to an enterprise level, encompassing virtually any structure, cluster, campus, and area in between,

In another embodiment, systems and methods interact with and customize a BAS. For

15 example, user customization options are presented by and accomplished through a graphical user interface. In addition to providing a portal through which users may access, manage, and customize the BAS, the user interface itself is customizable in accordance with and complimentary to the dynamic extensibility of the system. For example, in one embodiment when an enterprise server engine of the BAS discovers new objects, the user interface can be

20 customized automatically or selectively at a user's direction. The user interface also allows the user to customize the hierarchical directory of sites or buildings. The sites or buildings are searchable from the user interface and the results of searches can be used to then customize the directory. The user interface also comprises a dashboard display in one embodiment to present information about building systems at a glance. The dashboard displays include summary

25 information for buildings, for spaces within buildings, or for specific equipment in a building.

The invention can be more readily understood by reference to FIGS. 1-37 and the following description. While the invention is not necessarily limited to the specifically depicted application(s), the invention will be better appreciated using a discussion of exemplary embodiments in specific contexts.

30 A BAS according to one embodiment of the present invention comprises a dynamically extensible and automatically configurable architecture anchored by an enterprise server engine (ESE). The BAS and ESE comprise a versatile and robust processor-based control system with a communications protocol-agnostic head-end that operably supports the management of HVAC

and other subsystems in one or more buildings from a central location internal to or remote from any of the buildings.

The BAS is an automatically and intelligently scalable object-oriented system in one embodiment, providing multi-site management capabilities in a local or widely distributed geographic area. The BAS is preferably networked for user accessibility. In one embodiment, the BAS is user-accessible via either or both a computer system on an Intranet or the Internet as a web-enabled application running on a web server. The web and network applications provide operational services for HVAC and other subsystems.

In one embodiment, the BAS is capable of supporting and integrating legacy, current, and next generation components and subsystems. The BAS is further able to support common vendor or manufacturer systems as well as competitor systems by intelligently identifying the systems and/or subsystems and facilitating integration into the dynamically extensible BAS architecture. This flexibility enables the BAS architecture to support added applications and new control panel and subsystem types and versions without recompilation and reissue, and to extend, customize, and tailor the BAS to specific needs in a particular implementation. Further, dynamic extensibility enables a complex system to provide enhanced versatility and usability.

Referring to FIG. 1, a BAS 10 according to one embodiment of the invention comprises an ESE 20 preferably located at a central location 12, such as a headquarters or control station. ESE 20 comprises a single local device in one embodiment. "Central" location 12, as understood by those skilled in the art, is not necessarily a geographic center but rather a communicative or control-based location in one embodiment from which it is convenient or feasible to manage BAS 10. For example, a user may manage one or more BASs at locations nationwide or within a region from a single headquarters location.

In another embodiment, ESE 20 comprises a multiple server configuration operating in a local or distributed environment. ESE 20 may also comprise other single, multiple, and/or networked computers or microprocessors; single or multiple servers; hardware; software; firmware; software and software instructions comprising firmware; and/or any other combination of computing and storage means, and programming means, for establishing communications with and for controlling distributed points and devices within BAS 10, for selectively implementing a dynamic extensibility capability and an automatic configuration capability, and for accepting, storing, caching, searching for, requesting, serving, and/or loading data and information, as described in more detail below.

ESE 20 is preferably locally networked at location 12 and communicatively coupled to the Internet 30, Intranet 32, and/or any other compatible communication means for

communicatively coupling ESE 20 with one or more other points or devices within BAS 10 and for facilitating a dynamic extensibility capability and an automatic configuration capability. ESE 20, via communication means such as the Internet 30 and/or Intranet 32, therefore can provide access and management control from virtually any location via a computer system, internal or external to a user's computer system. ESE 20 and BAS 10 need not be web-based or communicatively coupled to the Internet 30 as shown in FIG. 1, as other compatible communication means and options known to those skilled in the art exist. Communication means such as the Internet 30 and/or Intranet Ethernet/IP 32 or another local area network (LAN) or wide area network (WAN) facilitate communications between ESE 20 and other system components and devices. Some or all communications and connections may be either wired or wireless within portions of BAS 10 as needed or desired.

Each implementation of system 10 can vary substantially by size, composition of devices, and balance of present, legacy, and future generation devices. System 10 can also vary by vendor/manufacturer, type, physical layout of building and/or campus, user needs, and other characteristics. Therefore, each implementation of system 10 and ESE 20 in particular is done on a site-by-site basis. ESE 20 can recognize, communicate with, and control a variety of system devices, including present generation and common manufacturer, legacy or previous generation, and competitor controllers and building automation panels. BAS 10, via ESE 20, can also expand to integrate next-generation devices. Accordingly, ESE 20 comprises microprocessor, computing, storage, and/or other compatible means for accepting and storing data and metadata descriptors from BAS 10 points, and microprocessor, computing, storage, and/or other compatible means for automatically requesting supplemental manually programmed data and descriptors if metadata descriptors are unavailable. Data and metadata descriptors within BAS 10 are described in more detail below.

As depicted in FIG. 1, for example, a present generation supervisory controller 41, such as a Building Control Unit manufactured by TRANE®, or a panel 40, can be directly communicatively coupled to the Internet 30 and/or Intranet 32, while legacy unit(s) 42 can be directly communicatively coupled to the Internet 30 and/or Intranet 32 or coupled via a media converter 48. Legacy unit(s) 42 can include, for example, TRACER SUMMIT and TRACKER units manufactured by TRANE®, the assignee of the present application. Media converter 48 is preferably a simple translator but may also comprise other more sophisticated devices as needed. Media converter 48 is preferably not but may also be used with competitive product(s) 44 and/or future product(s) 46 in various embodiments. Competitive products 44 are also preferably directly coupled to the Internet 30 and/or Intranet 32. The term "competitive" is used to

generally refer to products manufactured by an outside organization with respect to ESE 20. Manufacturers of building comfort and control products and systems that may comprise competitive product(s) 44 include JOHNSON CONTROLS, HONEYWELL, TRIDIUM, YORK, GENERAL ELECTRIC, CARRIER, and others.

5 ESE 20 is further able to support future product(s) 46, such as updated versions of current controllers, newly developed products, and the like. Preferably, at least a plurality of panels 40, present controllers 41, legacy units 42, competitive products 44 or future products 46 are building automation, control or HVAC products, representative examples of which include:
10 filters, and air purifiers; fire and life safety systems; security systems; electrical system monitors and controllers; lighting system monitors and controllers; ventilation system monitors and controllers; sensors, including smoke, light, occupancy, motion, humidity, and others; pumps; air handlers; fluid and air moving and handling equipment; terminal products and devices; life science and pharmacological control equipment and monitoring systems, including positive and
15 negative pressure clean rooms; industrial automation and control equipment and systems; programmable logic controllers; and others. ESE 20 is also preferably able to coexist and cooperate with other similar but previous generation control and management systems, as will be described in more detail below.

Panel 40, supervisory controller 41, legacy units 42, competitive products 44, and future
20 products 46 may be generally referred to herein as BAS end devices. In accordance with the descriptions herein of panels 40, supervisory controllers 41, legacy units 42, competitive products 44, and future products 46, BAS end devices can comprise input/output points, binary and analog devices, embedded controllers, sensors, and any other control/sensor means for measuring and communicating data about at least one of a point, a device, a space, a system, or a
25 subsystem for at least a portion of a building or campus the like. The term “end devices” is used only as a convenient, generalized reference to points within BAS 10, and the context of the term “end” in particular is not intended to be limiting or to imply a point of communicative or control termination in any given instance from the perspective of BAS 10. For example, end devices such as supervisory controllers 41 can function as intermediaries between ESE 20 and additional
30 end device-side equipment.

Further, BAS 10 can comprise non-real end devices, or points, and virtual end devices. A non-real end device, in one embodiment, is a representation of a real, actual, or physical end device instantiated by ESE 20 and associated with or related to one or more actual, real, or physical BAS end devices. A real end device is an end device as depicted and described herein

throughout, the term “real” used only to describe an end device relative to an instantiated “non-real” end device, as will be understood by those skilled in the art. Non-real end devices can be derived and instantiated by ESE 20 from algorithmic relationships among at least a plurality of real end devices, or end device points or values. One example of a non-real end device or point
5 is a building efficiency. Building efficiency is related to both input and output characteristics of BAS end devices and BAS 10 equipment. Other examples include or are related to set points and comfort settings. ESE 20 is adapted to automatically update or redefine the non-real end devices in accordance with the dynamic extensibility and automatic configurability of BAS 10.

BAS 10 can also treat a particular BAS end device differently for different applications,
10 creating a virtual end device. A virtual end device is a custom or otherwise altered definition or treatment of an actual, real, or physical BAS end device. An actual end device is an end device as depicted and described herein throughout, the term “actual” used only to describe an end device relative to a “virtual” end device, as will be understood by those skilled in the art. For context or convenience, user might select that an end device be presented as a first type, while
15 BAS 10 operates and communicates with an end device that comprises, in reality, a second type. To satisfy the user, to permit the user to view and interact with the end device as an end device the user is comfortable with, or for the sake of a consistent interface, BAS 10 can present the end device to the user as a virtual end device of the first type even though the end device is actually implemented and controlled by BAS 10 as the second type.

20 In one embodiment, a user accesses and interacts with BAS 10 through a graphical user interface (GUI or “user interface”) presented on one or more computer devices 22. Each device 22 is communicatively coupled with BAS 10. The user interface of BAS 10 may be provided by virtually any device 22 with a visual display and a communicative connection to system 10. Some examples of such devices are a personal desktop, laptop, or portable computer (PC); a
25 portable digital assistant (PDA); a cellular phone; and other similar devices. Typically, the connection between device 22 and BAS 10 is provided by the Internet 30, an Intranet system 32, and/or some other local or wide area communication network, although other means of connection and combinations of connections are also possible. For example, if an Internet-enabled cellular phone is used, the connection comprises, at least in part, a wireless cellular
30 communication network.

Each BAS end device 40, 41, 42, 44, and 46 is modeled as an object in the context of BAS 10 of the invention. In object-oriented BAS 10 and ESE 20, efficiencies are achieved by modeling common objects for recognition and application to other similar objects. An object, simply put, is an instance of a class, or an encapsulation of descriptive behaviors and

functionality of a group. A general object can then be made specific based upon rules applied to the object. Referring to BAS 10, an end device object may encompass virtually any type or piece of equipment, or any input or output point, in BAS 10, as well as any application or data structure relevant to BAS 10.

5 BAS 10 is able to reduce manual programming and integration of new devices by taking an object-oriented approach to system devices and components. BAS 10 is further able to identify and call attention to objects and object-related events that are not recognized such that manual service and attention can be delivered. Object orientation of data and metadata management within BAS 10 supports dynamic extension and automatic configuration of BAS
10 10, including the components and architecture of BAS 10 and informational and managerial representations of the structure and status of BAS 10 in the user interface. Dynamic extension and automatic configuration create a circularly recursive system with the self-descriptive objects and system use of plastic and extensible metadata from and about the objects. BAS 10 metadata is therefore multi-level, redirectable, and extensible in one embodiment. Further, the dynamic
15 extensibility of BAS 10 enables a user to utilize the user interface to customize and control BAS 10, including the user interface itself, without the need for reprogramming or recompiling code.

Accordingly, FIG. 2 is a diagram of an operating architecture of BAS 10 according to one embodiment. In dynamically extensible and scalable BAS 10, objects exist in a hierarchical or class structure. For example, data objects, site objects, and panel objects are interrelated and can
20 be relatively defined, with the objects including or associated with respective object definitions 58, such as type, version, vendor, and the like, that are stored in a database 60 and interpreted by BAS 10 within an application engine/framework 62 with ESE 20 to determine how the particular object is to be handled by BAS 10. Internal meta-object management 50, data object management 52, site management 54, and panel and communications management 56, with
25 object definitions 58, represent the kernel of ESE 20 of BAS 10 and interface application engine/framework 62 with external sources and entities to manage objects within BAS 10. The kernel preferably comprises the p-code engine and is extensible. Application engine/framework 62 with database 60 and ASP.NET applications 64 comprise graphical user interface element representations within an operating architecture of ESE 20. Database 60 is a data store or Sequel
30 server external to a graphical user interface program in one embodiment. A web server 66 then interfaces BAS 10 via application engine/framework 62 to an external interface. In one preferred but non-exclusive embodiment, the external interface comprises a GUI presented via an Internet 30 or intranet 32 system using a web browser program. Web server 66 and web browser 68 in

FIG. 2 are not client-side web server and web browser software elements but rather representations of ESE 20 operational architecture components.

The main objects and classifications used by system 10 in one embodiment are shown in FIG. 3 with reference to FIG. 2. Data object management 52 includes a data manager web engine 100 and object management 101. Data manager web engine 100 includes a data request manager 102 and a data request object 104. Data request manager 102 is an object for managing incoming XML requests, and for creating data request objects 104, associated data objects 120, and the associated URL and identification for outside clients to use as a reference. Data request manager 102 is also a cache for data request object 104 and data object 120 from the user interface and/or any client. Data request object 104 is an object that contains a collection of read requests. Object management 101 includes data object 120 and smart value 126. Data object 120 is an object that encapsulates one or more objects that exist in each panel, including both equipment and application objects. Smart value 126 is an object that encapsulates the properties that exist in the data objects and is responsible for encoding/decoding raw data into and out of any external format and for performing conversions, if needed.

Site management 54 includes a site manager 108 and site 110. Site manager 108 is an object responsible for managing all sites 110, starting, adding, and operations that transcend sites. Site 110 is an object that is central for interacting with a building, which includes at least one individual panel object 112. In one embodiment, a building is seen as a site 110 by ESE 20. A particular site 110, however, can be an individual building or a campus of more than one building. Conversely, a single building can include more than one site 110.

Referring again to FIG. 1, for example, panel 40, supervisory controller 41, legacy unit(s) 42, competitive product(s) 44, and future product(s) 46 together may comprise a single site 110, or some or each of panel 40, supervisory controller 41, legacy unit(s) 42, competitive product(s) 44, and future product(s) 46 may be located at more than one distinct site 110. ESE 20 in system 10 can default to a single building, single site view in one embodiment, which can then be customized or altered according to a user preference or a system characteristic or discovery data. In one particular example, a manufacturing facility includes a first user- and system-defined site 110 consisting of a front office area and a second user- and system-defined site 110 consisting of the manufacturing floor. This plural site definition can make it more convenient and intuitive from a facility perspective to manage disparate spaces.

General information about spaces within buildings in BAS 10 typically includes the types of equipment in the space, temperature, setpoints, and variance from setpoints. Additional conditions describing the spaces include flow rates, occupancy, modes (heating or cooling),

equipment status, and outdoor air temperature and humidity. Equipment conditions refer to the condition of specific equipment in a space. Specialized or custom equipment may provide other information.

Some or all of the general information is available for viewing in the user interface. The information displayed can also be updated to current status by activating a button. Other building spaces can be accessed through links to navigate within the user interface. The user interface can be tailored to a specific end device being represented. For example, ESE 20 and the user interface can assemble information from definitions provided to ESE 20, from self-describing end devices, from information read from end devices, and from manually programmed end devices to create user interface pages. The pages can be created from templates, with elements and information added or removed according to the assembled information.

Equipment can be viewed individually or as part of subsystem groups. Information about a subsystem group, for example, can be displayed in the user interface directly, such as in a tabular form with links to the specific equipment in the group. The collection of data can also be user customized. Other space conditions and values may be viewed by following intuitive links. In one embodiment, information pertaining to BAS 10 can be manually updated by a user through the user interface.

Meta-object management 50 includes a metadata manager 114, an objection definition 122, and a property definition 128. Metadata manager 114 is an object for parsing in metadata XML files and managing metadata definitions and is preferably cached by panel type, version, and object type in one embodiment. Object definition 122 is a metadata object that defines the properties, services, and behaviours of data object(s) 120. Property definition 128 is a metadata object that defines the attributes and behaviours for the properties of an object.

Panel and communication management 56 includes communication manager 116, panel 112, protocol stack 118 and protocol data unit (PDU) 124. Communication manager 116 is an object responsible for managing all the communication ports, threads, and protocol stacks. Panel object 112 is an object that represents the physical panel(s) and manages the version of metadata to use and services available for the protocol stack. PDU 124 is an object responsible for an encoding/decoding algorithm for the properties over the communication wire.

The main data entities are depicted in FIG. 4A, an example data model according to one embodiment is depicted in FIG. 4B, and an example model according to another embodiment is depicted in FIG. 4C. At a very basic level, each site 110 is a collection of one or more panels 112 (panel objects), and each panel 112 is a collection of one or more objects 120, which may

need extensions 130 for system operability. Site 110 can be an individual site, i.e., building, or a list of sites managed by ESE 20. Site 110 also includes information for background tasks. Panel(s) 112 is a single panel 112 or a list of panels known for site 110 and the information needed by ESE 20 to manage those particular panels. This information can include panel type, version, vendor, and ignore flags in one embodiment. Object(s) 120 is a list of objects that exist in each panel 112 and is used for navigation, display, and management. Object extension(s) 130 is information kept on ESE 20 that is specific for each object 120 as described by the metadata associated with object 120. Object extensions 130 are used to drive a user interface for determining things such as to which family a specific object belongs when an object is in a different family by the object configuration.

A data model similar to the data model of FIG. 4B exists for each individual site 110 in one embodiment. When ESE 20 discovers site 110 in this example, ESE 20 knows or can learn that that site 110A is a collection of panels 112A, 112B, and 112C. Panel 112A includes object 120A. Panel 112B includes objects 120B and 120C, and panel 112C includes objects 120D and 120E. Objects 120B and 120D require object extensions 130A and 130B, respectively. More or fewer panels 112, objects 120, and/or object extensions 130 can be used in other embodiments, the model of FIG. 4B being only one example.

ESE 20 operably reads and writes data in panels 40 and 41 and units 42, 44, and 46 (referring again generally to system 10 of FIG. 1) that support building automation standard protocols. In the context of FIG. 1 and herein, units 42, 44, and 46 can be panels but are distinguished by type in FIG. 1 to illustrate possible configurations and compositions of system 10. For example, ESE 20 and BAS 10 as a whole are generally compatible with the BACnetTM protocol and/or XML at a minimum, although physical or virtual media converters 48 may also be needed for particular devices in various embodiments. While ESE 20 is compatible with and/or configurable for a wide variety of protocols and standards, particular examples herein will refer to the BACnetTM protocol, Internet 30, and Intranet 32 systems where appropriate, in the context of one non-limiting embodiment of the invention.

ESE 20 is structured, in one embodiment, to integrate various implementations of BACnetTM and other protocols as natively as possible. ESE 20 can operably and concurrently support multiple versions and implementations, e.g., services supported and proprietary information. This enables ESE 20 to integrate both “inside,” i.e., common vendor/manufacture or platform, and “outside,” i.e., other vendor or competitor, devices without requiring manual programming of the object. Contrast this with current methods of integration of outside objects 44 in other systems, which require time- and labor-intensive manual programming of the data

and relationship by field service technicians unique to each installation, adding to the cost and complexity of these other systems and reducing convenience.

ESE 20 operably provides an interface for system installation, setup, integration, and support. For example, ESE 20 provides an interface for device/object 40, 42, 44, and 46 setup parameters, including IP address, subnet mask, gateway, and name of server for each, where applicable. ESE 20 further provides a methodology and/or utility to setup and customize web pages, which can include both templates and individual pages, and to serve and publish graphics to web pages. System 10 and ESE 20 also allow user definition of attributes for a given site for grouping purposes. In one embodiment, at a minimum, each site 110 is associated with a geographical and a type attribute and a search function is provided to allow users to search for sites or groups of sites. ESE 20 further preferably accommodates the addition, removal, and general management of entire sites 110 within system 10.

Site management of ESE 20 is an important aspect of system 10 from an implementation perspective. Dynamic extensions, enhancements, and changes are intended to be natural, fundamental features of system 10. Further, ESE 20, as a core engine of system 10, is designed to be used as the foundation for other systems and devices, including next-generation developments. Each implementation of ESE 20 and system 10 is designed to keep site and data management services separate from a user interface and applications to ensure that the core engine aspect is not compromised.

For example, in the college campus example of FIG. 4C, sites 110 managed by ESE 20 include the various buildings on campus, such as Engineering, Library, Administration, and others. Sites 110 also include information for background tasks. Still referring to FIG. 4C, each site 110 includes a panel 112. A system controller-level single panel 112 is depicted for each site 110, although a single site 110 can include multiple panels 112. Additionally, object(s) 120 is a list of objects that exist in each panel 112 and is used for navigation, display, and management. In FIG. 4C, each panel 112 includes a plurality of objects 120, which may be equipment, sensors, receivers, machines, and other devices.

The core engine, or ESE 20 in the embodiment of FIG. 1, forms a foundation or platform for system 10. Referring to FIGS. 5A and 5B, ESE 20 supports applications 150 and user interface features and functions 160 within system 10. ESE 20 within system architecture 500 further defines and describes the whole of the engine support. A proprietary extensions layer 502 of architecture 500 includes vendor proprietary extensions that may be implemented for a specification communication protocol, for example a protocol of layer 510.

Layer 510 includes a plurality of supported and anticipated protocols. While other BASs systems may be able to communicate with a plurality of vendor devices using a plurality of protocols, the dynamic extensibility of ESE 20 in system 10 enables ESE 20 to automatically determine a vendor and appropriate protocol(s) or get support, even if a particular vendor was not originally included, without requiring recompilation and subsequent redistribution of the main program and system, or system reengineering. Variations of support within a protocol for a particular vendor panel also do not require recompilation. In one embodiment, support for such a variation may be limited to a base standard protocol support. BACnetTM 512, an implementation of the ASHRAE standard BACnetTM protocol, can include the 1998, 2001, and 2004 specifications in one embodiment, and can preferably also implement other and future specifications. BACnetTM 512 is part of protocol stack 118 and PDU 124 (refer to FIG. 3) and the implementation of panel and communications management 56 (refer to FIGS. 2 and 3). LON 514 includes the implementation of the industry standard LON protocol. LON 514 is part of protocol stack 118 and PDU 124, as well as panel and communications management 56. Protocol layers 516, 518, 520, 522, and 524 each can include implementations of various available, next-generation, proprietary, and/or emerging protocols. In one preferred embodiment, protocol layers 516, 518, 520, 522, and 524 can include supported proprietary protocols such as TRANE®'s COM4, COM3, next generation TNG/XML, and BMN, although other combinations and protocols can also be implemented. For example, one of protocol layers 516, 518, 520, 522, and 524 can include an implementation of an emerging protocol standard such as oBIXTM, or Open Building Information Exchange. The oBIXTM standard is an industry-driven protocol initiative to define XML- and web-based systems and mechanisms for building control systems. Protocol layers 516, 518, 520, and 522 are part of protocol stack 118 and PDU 124 and the implementation of panel and communications management 56.

Kernel cache 526 is a caching layer for centralizing the management of input and output to panels 112 (FIGS. 3 and 4A; refer also, for example, to FIG. 4B), more particularly panel 40 of FIG. 1, for example. Kernel cache 526 is part of site manager 108 and site 110 and of site management 54.

A communications and communications extension manager layer 530 includes logic for managing and coordinating the various communications protocols of layer 510 described above. Communications and communications extension manger layer 530 is part of communication manager 116 and the implementation of panel management 56.

A metadata management layer 532 manages metadata definitions, which include definitions and rules for managing the various objects and properties of system 10 and ESE 20.

Metadata management layer 532 includes metadata manager 114, objection definition 122, and property definition 128 and is part of the implementation of panel management 56.

5 An object management layer 534 manages in-memory objects and properties maintained by kernel 540, which is described below. Object management layer 534 includes data object 120 and smart value 126 and corresponds to object management 101 of FIG. 3.

A site management layer 536 manages all sites 110. As previously described, sites 110 can include buildings, campuses, structures, and other entities, such as individual HVAC networks. Site management layer 536 corresponds to site management 54 of FIGS. 2 and 3.

10 Direct communication interface 538 is a thin layer that provides direct access to lower-level communication services for higher-level applications. Direct communication interface 538 is part of site manager 108 and site 110 entities and is part of the implementation of site management 54.

15 In general, FIG. 3 depicts the core of data manager kernel layer 540. The kernel of system 10 and ESE 20 relies on object-oriented principles and functionality for a basic interface and framework of operability. Referring again to FIG. 5B, a data manager kernel layer 540 is used to describe and define the whole of the site, communications, object, and metadata components of system 10 and ESE 20. Kernel persistence manager layer 542 is responsible for handling persistence, or storage outside of memory, for the ESE 20 kernel. Kernel SQL interface 544 handles an interface to an SQL (structured query language) database for data
20 manager kernel 540. A test manager 546 is responsible for managing registration of low-level kernel classes for testing purposes. While an SQL database is preferred in one embodiment of the invention, other database applications can also be used in other embodiments, such as MSDE (MICROSOFT® Data Engine) and the like, as recognized by those skilled in the art.

25 The ESE 20 kernel is designed to be extensible, and kernel extension manager 550 is responsible for managing, initializing, and shutting down each extension. Various extensions in one preferred embodiment of the invention include but are not limited to site synchronization 551, alarming 552, scheduling 553, data collection 554, kernel test harness 555, start-up 556, simulation 557, and graphical programming 558.

30 Site synchronization 551 is an extension layer responsible for services needed for site synchronization. Site synchronization is described in more detail below. Alarming 552 is an extension layer responsible for services needed for handling alarms for ESE 20. Scheduling 553 is an extension layer responsible for services for managing schedules for ESE 20. Data collection 553 is an extension layer responsible for services needed for collecting data, including trended data, for ESE 20. Kernel test harness 555 is an extension layer responsible for services

needed for performing tests of the ESE 20 kernel functionality. Start-up 556 is an extension layer responsible for services needed for on-line discovery of an HVAC network for ESE 20. Simulation 557 is an extension layer responsible for services needed to run equipment simulator applications for ESE 20. Graphical programming 558 is an extension layer responsible for running graphical programming scripts for ESE 20.

Data manager web engine layer 562 brings ESE 20 to a web server to be used to support applications, such as HTML pages, built to run on web server 64. Data manager web engine layer 562 includes the implementation of data request manager 102 and data request object 104. Data manager persistence manager layer 563 manages persistence for applications built within data manager web engine 562 and is part of the implementation of application engine/framework 62. Data manager cache layer 564 manages data, including objects and properties, associated with web pages and is part of the implementation of the application engine/framework 62. Server-side test harness layer 565 is an extension layer responsible for services needed to perform tests of ESE 20 data manager server functionality. Data manager SQL interface layer 566 is responsible for handling the interface to a SQL database for the data manager of ESE 20. As previously mentioned, other database applications may be used in other embodiments of the invention, an SQL database indicative only of one embodiment of the invention. Accordingly, interface layer 566 can interface to other databases in other embodiments.

Web software framework layer 567 represents the framework used for building web applications for ESE 20 and is part of the implementation of application engine/framework 62. Applications layer 568 represents user interface 160 and applications 150 that make up ESE 20, including status, alarming, scheduling, data collection, security, administration, and the like. Client-side test harness layer 569 is responsible for performing client-side variation and verification of available tests.

Workstation software framework layer 570 represents the framework used for building non-web oriented applications 150. Workstation software persistence manager layer 572 manages persistence of applications 150 built as workstation software. Workstation software SQL interface layer 574 is responsible for handling the interface to a SQL database for workstation-based software. Similar to as mentioned above with reference to interface layer 566, interface layer 574 can also interface to other suitable database applications in other embodiments of the invention.

Simulator manager layer 575 is responsible for managing, starting, and stopping the services implemented in simulation kernel extension 557. Simulator user interface layer 576 is a user interface 160 for the simulator.

Unit test harness layer 577 is responsible for managing unit tests for each class and component within the kernel of ESE 20. Unit test harness user interface layer 578 is a user interface for running, viewing, and verifying results of unit tests.

Workstation software user interface framework layer 579 represents the framework for building workstation-based applications 150. Non-web applications layer 580 represents thick-client applications 150 to be built. Web user interface framework layer 581 is a framework that enables applications 150 built on web software framework 570 to operate on a single-user, i.e., non-web server based, machine. Applications 582 are a workstation re-use of applications 568.

With system 10 of FIG. 1 and architecture 500 of FIG. 5B as context, one preferred embodiment of ESE 20 is designed to be a self-modifying and self-adapting system integration engine, providing dynamic extensibility and scalability. Site management from the perspective of ESE 20 therefore includes the following primary system processes: system start-up; site discovery; site removal; site synchronization; and system shutdown. Each of these system processes will be described in more detail below.

Referring to FIG. 6A and 6B, and recalling start-up extension 556 of FIG. 5B, an ESE start-up process 600 begins with starting ESE 20 and trace logging services, local to ESE 20, at step 602. Next, a start-up parameters file is loaded at step 604, and information from the start-up parameters file is used to locate database 60 at step 606. Task logging services are started at step 608, followed by managers 50, 52, 54, and 56 at step 610. In one embodiment, referring also to FIG. 6B, starting managers 50, 52, 54, and 56 at step 610 includes locating metadata and a metadata server at step 610A (meta-object management 50); loading all sites at step 610B (site management 54); starting communication ports at step 610C (panel and communications management 56); and starting site state machines at step 610D (site management 54), iterating over all sites known by ESE 20.

Referring again to FIG. 6A, step 612 includes iteration and start-up of all applications using the start-up parameters. Applications includes background task/service manager and application logging services; trending services; site synchronization services; site discovery services; alarming services, including enablement of incoming alarms; and scheduling services. Next, the system is synchronized and held until services are available at step 614. In one embodiment, ESE 20 start-up is held only until critical services, such as background task/service manager and application logging services, site synchronization services, and site discovery services, are available. In another embodiment, ESE 20 start-up is held at step 614 until all services are available. At step 616, and based on start-up parameters, user interface 160 services

are started. After start-up, ESE 20 is ready for normal operations and may execute other system processes.

In one embodiment, the aforementioned site 110 or object 120 integration into system 10 is accomplished via a discovery process. For example, a new panel 40 is installed at a location and is to be incorporated into system 10. ESE 20 operably executes one or more algorithms that discover the new object 112 (panel 40) within system 10 and subsequently analyze existing programming to first determine whether panel object 112 is in fact new, or whether panel object 112 was previously discovered within system 10. Upon determining that panel object 112 is a new addition, ESE 20 subsequently obtains any relevant or necessary information, such as vendor, version, and supported protocol(s), from and about panel object 112 in order for panel 40 to be integrated into system 10 and performs on-going reconfiguration.

Newly discovered panel 40/panel object 112 is also categorized for future addressing and identification. Object data and information, including categorization, is used to manage and control individual objects, groups of objects, and the entire system in use. In the discovery and categorization process, system 10 preferably applies recognized standards and rules, for example those promulgated by ASHRAE as previously discussed, where applicable and available. Exceptions may exist, however, if system 10 discovers a panel object 112 or object 120 from a common vendor, i.e., the same vendor or manufacturer as system 10, or from an outside vendor. These objects 112, 120 can be categorized and synchronized with system 10 according to that vendor's standards and rules, which in many cases will be the same or similar to those in the applicable industry, such as the aforementioned ASHRAE standards and rules.

In the case of an outside vendor object 112, 120 discovered, default metadata definitions for outside object 112, 120 BACnetTM implementation are used, including analogs, binaries, devices, schedules, and trends, among others. If, in a particular system 10, a mix of inside and outside vendor objects 112, 120 is found, site 110 in general is treated as an outside vendor site because the inside vendor equipment is likely not the main integration tool. In this situation, a panel 40 or supervisory controller 41 is used as the integration tool by ESE 20 to interface to the outside vendor equipment. In general, ESE 20 can also assume unless otherwise programmed that an object schedule mapped for ESE 20, whether inside or outside, will manage whatever it is respectively responsible for.

If newly discovered panel object 112 cannot be categorized or does not fit any existing category, panel object 112 can be automatically flagged or otherwise marked by system 10 for manual attention. In one embodiment in which such a situation occurs, no dialog is established between new panel 40 and ESE 20 until panel 40 can be categorized and associated definitions

obtained, as panel 40 may be of a type that is not supported by system 10. While the BACnetTM protocol is used in some implementations and embodiments, LonTalkTM may be used in other implementations or embodiments. Additionally, both protocols, at different sites or at different system levels, may also be used within a single system 10. Each protocol preferably has its own
5 separate virtual bus, but each runs TCP/IP in one embodiment over the same wire to appear as different networks. In other embodiments, MSTP (Master Slave Token Passing), MODBUS, PTP (Point-to-Point), and other BACnetTM and suitable protocols may also be used.

In one embodiment, panel objects 112 and objects 120 can be identified using various standard BACnetTM services. As in the initial discovery process, ESE 20 is preferably not
10 dependent upon systems integration activities to program the specific configuration change data into system 10. If the data structures adhere to the standard data expected and recognized by ESE 20, the information is read from object 112, 120. Any specific context given to the information is also provided through input to ESE 20 without having to recompile and load another version of production code or field program the logic in system 10. In the absence of
15 information for a specific panel 40 (panel object 112) for a manufacturer, system 10 reverts to the BACnetTM standard for the description of information in object 112, 120 and operates with this fundamental information in one embodiment.

For example, system 10 can identify an object 112, 120 according to a vendor in one embodiment. After vendor identification associated with object 112,120 is determined, system
20 10 can obtain more specific information related to object 120, including product, version, and definitions of how to communicate with that object 120. System 10 algorithm(s) can then be altered and synchronized to remember how to communicate with that object 112, 120 or other like objects having similar discovered characteristics in the future.

System 10 can alternatively determine an object's vendor by systematically running
25 through available permutations or alternatively by assigning an Internet protocol (IP) address to panel object 112. Multiple options are available because response times may suffer while system 10 goes through each number or line of information. In one embodiment, a general description of the outside panel implementation of BACnetTM is provided to ESE 20 with an input file, i.e., panel metadata. ESE 20 can then discover a panel 40 at the described location, for example by
30 the panel's IP address, and obtain any information relevant to the ESE 20 application to perform its operations, such as status and setpoints, data collection, alarming, and scheduling, with the panel. If an object 112, 120 cannot be identified according to the aforementioned and other methods, object 112, 120 is labelled an exception and system 10 implements an algorithm with which to treat the exception.

Referring to FIG. 7, a site 110 discovery process 700 begins at step 702 with collecting site discovery information, such as from user input via a user interface or from a batch input file. The discovery information can include a site name, IP address/DNS name, port number to open, protocol to use, and a device identification (deviceID) to discover. The deviceID may be a
5 system default in one embodiment. The discovery information is then passed to a site management layer 536.

At step 704, a site license is validated and includes verifying that a permitted number of site licenses will not be exceeded. If the site license cannot be validated at step 704 or if the number of site licenses is not successfully verified, an error message is returned and process 700
10 is stopped. If step 704 is successfully completed, communication ports are initialized at step 706. Step 706 includes requesting, from communication manager 56, a protocol stack for the port and protocol type. In one embodiment, ports are limited to one protocol per port; accordingly, ESE 20 will only attempt to discover one type of protocol 510 at a particular IP address. If the port is already used, ESE 20 determines whether the current port was opened
15 using the requested protocol. If not, an error message is returned, discovery process 700 is stopped, and a protocol stack (if created) is deleted. If the port was opened using the requested protocol, a new protocol stack is created over the existing open socket and is initialized. Returning to the initial query, if ESE 20 determines that the port is not already in use, a new socket is opened and a new protocol stack is created and initialized. Based on the type of stack,
20 basic initialization is then performed. If initialization is not successful for any reason, an error message is returned, process 700 is stopped, and a protocol stack, if created, is deleted.

If initialization was successful, a new site object 110 is created in memory and in database 60 at step 708. The new site object 110 is flagged as a “discovering” state, wherein no user actions are yet allowed on site 110 as a site object does not yet formally exist in system 10
25 outside of the site discovery status process.

Next, at step 710, discovery metadata is wired to the site. Discovery metadata is generic, with the protocol stack at this point deferring to a temporary entity that specifies and/or references the discovery metadata and the default set of services to use. Working, or actual, metadata is discovered, wired in, and set-up at step 712 after getting a list of one or more panels
30 40 from the protocol stack. This step is dependent in part upon the type of protocol 510 and the results of previous steps and can vary according to inside vs. outside panels 40, including previous discovery and an available device list from a site layout object and a general broadcast algorithm to request responses from objects 112, 120. Iterating for each device 40, low-level communications bindings/tables are set up for panel 40, including IP address, MAC address,

deviceID, and the like. If a metadata version for panel 40 is found, appropriate metadata for panel 40 is wired in, a list of supported services is read from panel 40, and panel object 112 is created. Panel object creation also includes setting all internal values and storing in database 60. If a metadata version for panel 40 is not found, the panel state is set to “not available,” requiring
5 user attention to resolve. After iterating for each device found, a site state is set to an “okay to synchronize” state.

At step 714, site 110, panels 40 (112), and metadata are validated. Validation initially includes verifying that supporting metadata for each panel 40 is available to enable the communication manager 56 and data management 52 services to properly operate, and
10 determining whether a sufficient number of panels 40 are supported. In one embodiment, this second aspect of validation is successful if only one working panel 40 is found. In other embodiments, more working panels 40 are required. If validation is not successful, discovery process 700 fails and a protocol stack, if created, is deleted.

If validation is successful, a transition decision occurs at step 716, wherein if
15 communications with at least one panel 40 at site 110 can be established at a high enough level, discovery 700 continues. Transition decision 716 is followed by a first site synchronization at step 718. Upon successful completion of the first site synchronization, site 110 is transitioned to an operational state and incoming alarming and trending notifications are allowed at upload transition site step 720.

20 With respect to establishing communications with at least one panel at a sufficient or high enough level, ESE 20 operably provides dynamic protocol support. Referring to FIG. 8, a representative and example dynamic protocol support algorithm table 800 illustrates various “levels” of identification and communication that can be established with a panel 40 or other object in system 10. For example, protocol support table 800 includes at least one available
25 protocol 802, or PROTOCOLa/ in FIG. 8. PROTOCOLa/ may be a BACnetTM protocol or another suitable protocol as previously described. PROTOCOLa/ then more specifically includes at least one vendor 804. VENDOR0 may be a default vendor, VENDOR1 may be ASHRAE, VENDOR2 may be TRANE®, and so on, these particularly vendors used only for one example. At least one product 806 may then be associated with each vendor 804, and each
30 product 806 may include at least one type or version 808. When establishing communications with a panel 40, then, ESE 20 preferably obtains metadata to identify panel 40 as specifically as possible to establish higher level communications. If ESE 20 is able to identify a first panel 40 to a vendor level 804 and second panel 40 to a type level 808, for example, ESE 20 will be able

to establish higher level communications with second panel 40 because ESE 20 will have more detailed and specific information.

System 10 further operates, by way of example, as an infinite state machine. Current embedded systems are state machines with a finite number of operating states. An infinite state machine, however, can provide so-called “plug-and-play” operability by discovering a panel object 40, synchronizing panel object 40, recompiling ESE 20 for integration or re-integration, and changing state while running. For a system integration platform as in system 10 having an infinite number of states, each state of system 10 must be discovered and anticipated, in contrast to a danger/safe system in which the system must know all possible states and potentially be reengineered to recognize additional or updated states.

ESE 20 comprises a plurality of background administrative state machines that keep ESE 20 operational and up-to-date. These state machines, and each implementation of ESE 20 generally, vary from site to site. In one embodiment, ESE 20 provides an intuitive interface for device set-up parameters, including but not limited to an IP address, subnet mask, gateway, and server name, and provides means for setting up, customizing, and publishing both template and individual web pages. For either templates or individual pages, ESE 20 can present dynamically generated content as the pages are served. ESE 20 further provides an interface to make administrative functions available through a web browser for configuration of system 10 and applications. Functions and applications that may require administrative configuration include site management, customization, user security, alarms, scheduling, trending, and the like, and can vary according to an object, panel, building, or other component or characteristic of system 10.

ESE 20 is preferably not dependent upon systems integration activities to program specific data into system 10, in contrast with current methods of field programming. If panel 40 data structures adhere to the applicable standard data recognized by ESE 20, the information can be automatically read from panel 40. Applicable standards in various embodiments include those defined in ASHRAE 135-2004 or future standard protocols such as OBIXTM, as well as others. Any specific context given to the information, such as that created by the panel vendor/manufacture, can be provided through input to ESE 20. This eliminates the need to recompile and load subsequent versions of production code or have a field organization program the logic in system 10.

Additionally, ESE 20 can detect configuration changes after initial panel 40 discovery (700) and automatically adjust to the detected changes. In one embodiment, this is accomplished

by identifying all of the objects 120 on each panel 112 and then performing a synchronizing process periodically after initial discovery as previously described.

The synchronizing process preferably runs on a configurable timer in one embodiment. System 10 compares a version running with detected building or location activity. If any
5 synchronization is needed, system 10 next determines whether the synchronization can be handled via an available algorithm. If yes, system 10 proceeds to execute the algorithm. If no, system 10 can send a request for manual service.

Synchronization can be automatic, scheduled, or forced in one embodiment. System 10 can automatically discover and synchronize a new panel object 112, as described above.
10 System-wide synchronization can also be periodically scheduled, such as at midnight each day or at some other time or interval. Synchronization can also be forced on-demand. User interface 160 can include a “synchronize now” feature by which a user can selectively synchronize system 10 on demand. This feature can be particularly useful in situations in which service has been performed, such as in response to a comfort complaint or for some other purpose, and system 10
15 can be subsequently synchronized close-in-time to quickly incorporate the update(s).

Referring to FIG. 9, a site synchronization process 900 can be triggered or initiated by several different events, including a site addition, a recurring schedule, and a user-initiated “synchronize now” event. Process 900 is followed for each site 110 and begins at step 902 by verifying that the IP address/DNS name for site 110 has not changed. If the address or name
20 have changed, do not match, or otherwise conflict, site 110 is flagged and logged.

Next, a list of all panels 40 known to ESE 20, and having verified IP addresses and DNS names, is obtained at step 904 as a list of panels 40 to be synchronized. Panels 40 that have already been synchronized or those that are not in a proper operating state are identified and skipped in the subsequent synchronization steps; remaining panels 40 are flagged as
25 unsynchronized and all associated objects are also flagged as not synchronized at step 906. At step 908, a list of all panels 40 “on the wire” is obtained for site 110.

For each panel 40 on the wire, panel synchronization tasks are then performed at step 910. Referring to FIG. 10A, step 1002 is determining whether panel 40 is new. If panel 40 is new, step 1004 is determining whether panel 40 is supported, i.e., is metadata available. If yes,
30 sub-process 1001 is implemented as shown in FIG. 10B: appropriate metadata for panel 40 is wired in 1001A; the list of supported services for panel 40 is read 1001B; panel object 112 is created, and internal values are set and stored in the database 1001C; and objects 120 are uploaded from panel 40 and appropriate tables are updated 1001D. At step 1006, any

unsynchronized objects are deleted and the synchronized panel is labelled as such and updated with the latest synchronization date/time at step 1008.

Returning to step 1004, if panel 40 is not supported, the panel state is set to "metadata not available" at step 1010 and process 1000 returns to step 1006. Returning to step 1002, if panel 5 40 is not new and, at step 1012, the vendor or version of panel 40 has not changed, objects 120 are uploaded from panel 40 and tables are updated at step 1014 before returning to step 1006. If the panel 40 vendor or version is found to have changed at step 1012, step 1016 determines whether panel 40 is supported. If panel 40 is not supported, process 1000 advances to step 1010. If panel 40 is supported, process 1000 advances to step 1018, wherein existing panel information 10 (metadata) is replaced with new or updated information. In one embodiment, this is accomplished by making a copy of a row in a panel table and any associated rows in object and object_extension tables. Process 1000 subsequently advances to sub-process 1001. In process 1000 to determine whether a panel is new, has changed, is supported, and the like, sub-processes similar to discovery process 700, in particular steps 706-716, are generally used in 15 communications between ESE 20 and panel(s) 40.

Similarly, and referring to FIG. 11, a BAS end device synchronization begins with step 181, determining whether a BAS end device is new. This process is similar to that shown in FIG. 10A for determining whether panel 40 is new. If the device is new, step 182 is determining whether the BAS end device is supported, i.e., is metadata available. If yes, appropriate 20 metadata for the BAS end device is wired in; the list of supported services for the BAS end device is read; a BAS end device object is created, and internal values are set and stored in the database; and objects are uploaded from the BAS end device and appropriate tables are updated. At step 183, any unsynchronized objects are deleted and the synchronized panel is labelled as such and updated with the latest synchronization date/time at step 184.

Returning to step 182, if a BAS end device is not supported, the end device state is set to "metadata not available" at step 185 and process 180 returns to step 183. Returning to step 181, if a BAS end device is not new and, at step 186, the vendor or version of the BAS end device has not changed, objects are uploaded from the BAS end device and tables are updated at step 187 before returning to step 183. If the BAS end device vendor or version is found to have changed 30 at step 186, step 188 determines whether the BAS end device is supported. If the BAS end device is not supported, process 180 advances to step 185. If the BAS end device is supported, process 180 advances to step 189, wherein existing BAS end device information (metadata) is replaced with new or updated information. In one embodiment, this is accomplished by making a copy of a row in a device table and any associated rows in object and object_extension tables.

Object or panel removal from system 10 is typically more complex than object addition through discovery process 700 as previously described. For example, interdependencies related to the object to be removed must be resolved or corrected. Further, system 10 generally cannot automatically recognize an object removal in the same way a new object can be discovered because an object removal can appear to be a fault or error related to the object, indistinguishable from a legitimate removal. Accordingly, object removals may require manual service or updating to accomplish.

Referring to FIG. 11, a site removal process 1100 begins with flagging a site 110 as deleted at step 1102. Synchronization is interrupted if running at step 1104, and incoming alarms are shutdown at step 1106. Other site 110 tasks are shut down at step 1108, and communications to site 110 are subsequently shut down at step 1110. The site object is deleted from memory at step 1112, and, at step 1114, site 110 is deleted from database 50.

In use, ESE 20 and system 10 provide summary tables by equipment type or some other attribute, per site. The summary tables are preferably based upon system- or user-defined attributes, wherein user-defined attributes are the most intuitive for management from a user perspective. Some attributes, however, may be system-defined, such as a system identifier, an object type, and the like. In one embodiment, summary tables include site and object names or other identifiers, space temperatures, setpoints, and diagnostic status.

Another aspect of one embodiment of ESE 20 and system 10 of the invention is related to alarming. System 10 and various objects therein will, by their very function and purpose, occasionally or systematically generate alarms. The alarms may be related to an operating state of the object, a service need status, a detected object or system characteristic, or some other indicator or condition. ESE 20 operably receives alarms from objects and, according to the invention, triages, manages, or otherwise appropriately handles the alarms. ESE 20 can also store or archive alarms and display an alarm log for a user.

Referring to FIG. 13, ESE 20 provides extensible support to outside object 201 according to object data 205 and object metadata 207. In one embodiment, ESE 20 discovers object 201 at a location. The discovery can be user-initiated, such as by providing a network address of object 201 to ESE 20 via the user interface in one embodiment, or automatic on behalf of ESE 20 in another embodiment. To integrate object 201, ESE 20 utilizes object metadata 207 to obtain a general description of object 201 based upon a communications implementation of the outside vendor of object 201. In one embodiment, object metadata 207 is data description code about object 201 and object data 205. The communications implementation may include, for example, a specific revision and version. ESE 20 of BAS 10 also accommodates changes in BAS 10 over

time, including BAS end device additions, removal, or changes, including changes to particular points. ESE 20 further handles versioning and dynamics over time, in contrast to other systems that assume a homogenous system and protocol.

5 Upon discovery of object 201, ESE 20 determines all available information relevant to operation of object 201 in system 10, including status and setpoints, data collection, alarming, scheduling, and the like, to establish communications with object 201. ESE 20 is not dependent on systems integration activities to program specific data and information; rather, if the information conforms to standard data structures, ESE 20 reads object data 205 directly from object 201. In other words, system objects, including outside object 201, are preferably self-
10 describing as discussed herein and are interrogated for object metadata 207 without programming intervention, such as manual mapping of points. Any specific context given to data 205 according to the vendor of object 201 can be provided by input to ESE 20 without recompilation of production code or field programming of logic.

ESE 20 operably provides an interface for system installation, setup, integration, and
15 support. For example, ESE 20 provides an interface for BAS end devices 40, 41, 42, 44, and 46 setup parameters, including IP address, subnet mask, gateway, and name of server for each, where applicable. ESE 20 further provides a methodology and/or utility to set up and customize web pages, which can include both templates and individual pages, and to serve and publish graphics to web pages. System 10 and ESE 20 also allow user definition of attributes for a given
20 site for grouping purposes. In one embodiment, at a minimum, each site 110 is associated with a geographical and a type attribute and a search function is provided to allow users to search for sites or groups of sites. ESE 20 further preferably accommodates the addition, removal, and general management of entire sites 110 within BAS 10.

ESE 20 efficiently handles data and information to enable operation of BAS 10 and
25 support external interactions with BAS 10. In particular, ESE 20 utilizes data management techniques to enhance communicative performance of BAS 10. In one embodiment, ESE 20 minimizes communication and data transfer related burdens on system 10 and components of system 10 through data caching. The user interface of BAS 10 provides static and dynamic information regarding the status and operation of BAS 10. Dynamic, real-time data from objects
30 in system 10 is presented in the user interface and can be updated according to a defined refresh rate or manually on-demand by a user. Unscheduled real-time data events can also occur at any time, for example as an alarm. BAS 10 can efficiently handle scheduled updates and presentation of dynamic real-time data in order to accommodate unscheduled data requests and events.

Referring to FIG. 14, ESE 20 and applications 150 implement refresh cache and multi-step delivery processes in one embodiment for responding to user interface requests, including HTTP requests for user interface web-based pages that represent the building automation equipment in system 10. These algorithms enable users to navigate through user interface 160, and request and view both static and dynamic data and information about BAS 10, with as minimal an impact on performance as possible. The refresh cache and multi-step delivery processes implemented by ESE 20 remove the burden from the panels and objects 203, which have much slower information communication performance characteristics. In particular, panels and objects 203 are typically embedded controllers with limited buffers. ESE 20 can sample and refresh data to relieve panels and objects 203 and improve the performance of BAS 10. A refresh or reinitiation rate can be based upon a characteristic of BAS 10 or of a portion of BAS 10. In one embodiment, a refresh rate is related to an end device (panels and objects 203) characteristic, such as a type, version, location, status, user preference, availability, and the like. A refresh rate can also be based upon the data characteristic, such as a data type, a rate of change, a metadata descriptor, a user preference or attribute, and the like. The refresh rate may be related to a user specification or a default set for BAS 10. The refresh rate can also be based upon a logical combination, synthesis, or amalgamation of one or more refresh rates by ESE 20. For example, an overall refresh or reinitiation rate for an end device may conflict with the refresh rate of a particular end device element or a refresh rate based on a data rate of change. ESE 20 can resolve any such conflict, which in one embodiment will be to select the most frequent refresh rate. In other embodiments, the resolution may be a logical combination, a system default, or some other selection or combination of a refresh or reinitiation rate or frequency.

Referring to FIGS. 14 and 15, applications 150 use object metadata 205 to determine object information and data 207 discovered from object 205 to be maintained in database 60 in one embodiment. ESE 20 then receives and stores data 207 in database 60. According to process 209, when a user requests a page related to object 203 in user interface 160 at step 211, applications 150 initiate two processes. In a first process, ESE 20 and application 150 determine the page and content based upon object metadata 205 and information 207 stored in database 60 at step 213. A page is then returned to the user with the information available from database 60 at step 215. The initial page returned can include static information related to object 203, BAS 10 in general, or some other object or information.

Concurrent to steps 213 and 215, to obtain the dynamic, real-time, or other information for the requested page that is only available directly from the panel, a read request is generated and processed to go over the wire to the panel at step 217. Due to the typical performance

constraints of the specific panels, a read request may take some time to be returned to the user interface page and the information made available to the user. Accordingly, the page initially displayed at step 215 includes as much static and dynamic information as is available, typically that from the database received at step 213 and initial but incomplete responses from the panel at
5 step 219. In one embodiment, the user interface page automatically and periodically refreshes at step 223 to provide additional dynamic information as it becomes available from the panels at step 219 until the page is complete at step 221.

To reduce the performance impact on BAS 10 of a user navigating off the requested page and then returning, which would require repetition of steps 211-221, ESE 20 can maintain the
10 page, complete or otherwise, in cache memory at step 225. In addition to caching the page itself, ESE 20 can also cache the dynamic input/output data received from the BAS end devices at step 219. ESE 20 can periodically refresh the dynamic data for the page for a period of time, even if the page is not currently requested or viewed. The cache also handles situations in which a single object is relevant to multiple pages. Data associated with that object can be requested for
15 a first page, then cached and accessed as necessary from the cache to load subsequent pages that include the some or all of the same data. A cache session can correspond to a user session in one embodiment. In other embodiments, cache session maintenance can be time, object, or system related.

ESE 20 implements a dual-stage periodic refresh in one embodiment of the invention. A
20 first stage is a system (BAS 10) stage and comprises three refresh levels in one embodiment. A first level is a one-time refresh. A one-time refresh typically occurs only a single time, such as when a page is first requested and loaded. Data having a one-time refresh metadata descriptor or tag includes configuration data, for example. A second level is permanent expiration. Some page data and content expires immediately upon request and load because the data is live and
25 real-time, such as a current temperature. Permanent expiration metadata tagged data and content is refreshed each time a page is requested or loaded, the finest refresh granularity. A third refresh level is intermediate the one-time refresh and the permanent expiration and is periodic expiration. Some content, including some real-time data, changes at a slow rate, making permanent expiration inappropriate. A periodic expiration may be refreshed, for example, every
30 ten minutes in one embodiment. Other periods may also be set or may vary according to a metadata descriptor or tag, system-wide setting, or other criteria in other embodiments.

In one embodiment, the cache is transaction-based, keeping the page for a fixed period, for example about fifteen minutes, as long as page hits continue. If a user returns to the page within the period of time, the page and its data are still available and could be immediately

presented in user interface 160, instead of having to repeat the BAS end device read request of step 217 and wait for the complete response at step 219.

In another embodiment, the cache is location-based, which is a variation on aging. In a location-based cache, ESE 20 will effect a proactive data fetch time-stamp configured based upon a particular location. ESE 20 utilizes object metadata 205 to determine when data for that object (location) is expired. While the entire page is periodically refreshed according to this scheme, the burden on the object (BAS end device) is reduced because ESE 20 only read requests the data on the page that has expired or that is changing more frequently according to metadata BAS end devices, which may begin to drop commands if barraged with read requests, rather than treating the BAS end devices as servers of data within system 10 from the perspective of user interface 160.

Site management of ESE 20 is an important aspect of BAS 10 from an implementation perspective. Dynamic extensions, enhancements, and changes are intended to be natural, fundamental features of building automation system 10. Further, ESE 20, as a core engine of BAS 10, is designed to be used as the foundation for other systems and devices, including next-generation developments. Each implementation of ESE 20 and BAS 10 is designed to keep site and data management services separate from user interface 160 and applications 150 to ensure that the core engine aspect is not compromised by building ESE 20 and user interface 160 in separate modules.

Data management services, user interface 160, and applications 150, however, intersect and cooperate in the ordinary operation of BAS 10 and ESE 20. For example, an important aspect of system 10 and ESE 20 is related to alarming, as mentioned above. Referring to FIG. 16, system 10 and various objects 203 therein will, by their very function and purpose, occasionally or systematically generate alarms 251. Alarms 251 may be related to an operating state of object 203, a service need status, a detected object or system characteristic, or some other indicator or condition. ESE 20 and alarm applications 253 operably receive alarms 251 from objects 203 and, according to the invention, triage, manage, or otherwise appropriately handle alarms 251. ESE 20 can also store or archive alarms 251 and display an alarm log in user interface 160.

In one embodiment, relevant to alarm triage, ESE 20 can automatically analyze alarm 251 to notify and/or request service or otherwise ensure that the alarm will receive the attention it warrants. Alarm triage, sorting, and filtering can be provided based upon an alarm and/or site attribute and alarm rules 255. By way of example, it can be appreciated that an alarm 251 related to a particular area or object 203 within a facility can have a much greater significance than an

alarm related to another area within the same facility. Similarly, one type of alarm may require a more rapid response than another type of alarm. Therefore, ESE 20 can automatically assess an incoming alarm according to alarm rules 255 related to an alarm type, source, and/or relevant object attribute and then handle alarm 251 appropriately.

5 For example, ESE 20 can forward a higher priority alarm via email 261 after ascertaining the relative importance of the alarm indicator according to alarm rules 255. Within system 10, alarm forwarding via email is a user interface 160 customization feature implemented as an administrative function and enables a user to specify to whom or what the notification should be sent. ESE 20 can also simply catalog lower priority alarms for later review by a user in a
10 viewable alarm log.

ESE 20 provides alarm message assessment and diagnostics with respect to alarms received from within system 10 to develop alarm triage algorithms 259. Algorithms 259 can be developed in compliance with rules 255 and applied to match alarm patterns and analyze alarm timings in future events and consolidate messages or provide automated actions. ESE 20 can
15 then intelligently identify patterns, sequences, and/or occurrences of alarms 251 to diagnose a common source and respond appropriately and automatically. Preferred embodiments of ESE 20 can identify, sort, sequence, and trend alarms 251 in order to identify a common link, if any, and reduce the number of alarm notifications 259 sent to a user for manual attention.

For example, a loss of power for a given circuit in a building can create multiple
20 diagnostics. ESE 20 can assess the pattern of diagnostics within BAS 10 and report only the loss of power and not the redundant and source-related alarm messages. ESE 20 can also send only a single alarm notice 259 including information about the common fault to a user in a user-identifiable format. Rather than sending a plurality of alarm notices 259 or complex system-driven information, ESE 20 can report the identified common fault in user-identifiable and
25 defined terms for context. The user can then deal with the single source of the alarms expeditiously, rather than attempting to clear each of the plurality of alarm notices.

ESE 20 can also maintain one or more alarm logs 263 and can catalog or archive alarms in an appropriate log 263. A user can then review log 263 and acknowledge or delete the alarms as desired. ESE 20 can also automatically and periodically purge alarm log(s) 263 as needed or
30 as defined by a user or administrator of BAS 10. Alarms are typically time-stamp recorded and/or sorted by some characteristic, such as object or type.

In one embodiment, alarms 251 are preferably received and handled by ESE 20 in real time. In another embodiment, such as one incorporating legacy panels and devices, ESE 20

optionally collects alarms 251 from objects on a periodic basis, such as hourly, daily, or more or less frequently.

In addition to automatically handling and triaging alarms, BAS 10 and more particularly ESE 20 can trend alarms and other data. Trending within BAS 10 is an intuitive and efficient management and diagnostic tool. In one embodiment, trend data is collected by ESE 20 from one or more objects 40, 42, 44, and/or 46 at a maximum frequency of once per minute or at another lower frequency or on a specific scheduled basis as defined by a user or administrator. Trend data can then be stored in a database and, in one embodiment, is available for sharing with network peers.

Building automation system 10 is therefore an object-oriented system designed with algorithms that work with self-describing panels 40 or objects. Algorithms implemented as part of BAS 10 communicate with objects to determine whether the objects are operating with algorithms by which they can be identified and integrated. If BAS 10 cannot determine whether an object is operating with an algorithm, BAS 10 intelligently and automatically defines the object as an exception. Building automation system 10 is universally self-describing in that BAS 10 applies concepts and captures algorithms based on object self-descriptions. The algorithms are then translated to accomplish associated mechanical aspects of the objects and BAS 10.

The present invention further provides the ability to alter definitions of objects in ESE 20 without having to recompile the production code. This provides for ease of maintenance and product support. Altered or updated definitions can then be input files to ESE 20, and complete or more complex updates can be made separately. Contrast this update process of the present invention with current methods, in which in order to get an update to object definitions to the end user or customer, production code needs to be rebuilt, tested, and updated for an installation. This increases the amount of time required by an on-site technician and the risk of failed installations. A further benefit provided by ESE 20 and system 10 of the invention is an automatic maintenance application. The automatic maintenance application may relate to updates, upgrades, and other regular or semi-regular tasks. In general, three types of updates will most frequently apply to system 10: simple updates; manageable updates; and complex updates.

Simple updates include minor changes and/or module additions to system 10. Simple updates can typically be implemented “on-the-fly” without bringing down any other applications or services provided and/or managed by system 10.

Manageable updates can include simple updates but may also require a service to be paused or memory caches flushed in order to apply the necessary or desired change. Unlike simple updates, manageable updates generally require system user notification because of the

service interruption. In some circumstances, simple updates may become manageable updates, or even complex updates as described below, because of consequential operations of the system and circumstances that develop during the update process.

Complex updates will generally require that servers and systems be brought down to accomplish the update. Complex updates may also or alternatively require that servers be restarted upon installation of the update. Updates to ESE 20, database changes, and other major updates are all included in complex updates. Additionally, simple and manageable updates may become complex updates because of unintended circumstances and events that occur during the update process.

In one embodiment, there is no distinction between a manageable update and a complex update from a user perspective, as each is implemented in the same manner and requires that servers and systems be brought down.

System 10 is therefore an object-oriented system designed with algorithms that work with self-describing panels 40 or objects. System 10 algorithms communicate with objects to determine whether the objects are operating with algorithms by which they can be identified and integrated. If system 10 cannot determine whether an object is operating with an algorithm, system 10 intelligently and automatically defines the object as an exception. System 10 is universally self-describing in that system 10 applies concepts and captures algorithms based on object self-descriptions. The algorithms are then translated to accomplish associated mechanical aspects of the objects and system 10.

In one embodiment, a building automation system (BAS) according to the invention comprises a plurality of end devices each associated with at least one of a space, a system, or a subsystem for at least a portion of a building or a campus; at least one communication network communicatively coupling at least a portion of the plurality of end devices and supporting a plurality of communication protocols; and a protocol-independent server engine communicatively coupled to the at least one communication network. The server engine includes programming means for selectively implementing a dynamic extensibility capability for the BAS that establishes communications with and control of the plurality of end devices over the plurality of communication protocols; and programming means for selectively implementing an automatic configuration capability for the BAS that supports addition of end devices to the plurality of end devices by determining at least one characteristic of each end device, the at least one characteristic being selected from the set consisting of a self-describing status and a non-self-describing status. For an end device having a self-describing status, the server engine includes programming means for accepting and storing data and metadata descriptors

communicated from the end device. For an end device having a non-self-describing status, the server engine includes programming means for searching a database of data and metadata descriptors for end devices maintained by the server engine for data and metadata descriptors based on the non-self-describing status of the end device and automatically requesting
5 supplemental manually programmed data and metadata descriptors for the end device if the non-self-describing status of the device is not sufficient to retrieve data and metadata descriptors for the end device from the database.

In another embodiment, a method of establishing communications with unknown end devices in a building automation system (BAS) based upon metadata descriptors provided by
10 known and unknown end devices comprises discovering an unknown end device on a communication network, the unknown end device associated with at least one of a point, a space, a system, or a subsystem for at least a portion of a building or campus. The unknown end device is queried for a communication protocol metadata descriptor and classified as a self-describing end device if the unknown end device provides a communication protocol metadata descriptor in
15 response to the query and selecting a communication protocol that corresponds to the communication protocol metadata descriptor for the unknown end device. The unknown end device is classified as a non-self-describing end device if the unknown end device does not provide a communication protocol metadata descriptor in response to the query and automatically requesting supplemental manually programmed communication protocol
20 descriptors.

Referring now to FIGS. 17, 18A, and 18B, one embodiment of user interface 160 includes a home or main page 200 from which a plurality of additional pages can be accessed. Information displayed on the pages accessible via user interface 160 generally relates to certain broad categories, such as data points relevant to the status of spaces and equipment. Information
25 can also be organized by priority in pages that show various alarms triggered by space and equipment conditions that vary from predetermined standards. The information and pages include, at a high level within BAS 10, content such as a hierarchical building index or relational directory 230 and a find buildings feature 228, and a set of navigation tables 202. Directory 230 is a navigable directory of pages within user interface 160. Navigation tabs 202 are not unique
30 to home page 200 and will generally be provided on most pages within user interface 160 to enable quick and efficient navigation. Tabs 202 include a user interface home page tab 204, an enterprise alarms tab 206, a user preferences tab 208, and an administration tab 210. In accordance with the user customizability of user interface 160, in other embodiments home page 200 can include additional, fewer, and/or other tabs as a user desires. Other options related to

the general customization of the display and behavior of user interface 160 itself are also available.

The portions of user interface 160 accessible via tabs 222, 224, and 226 are relevant to the overall navigation and functionality of user interface 160. Home page tab 204 provides a convenient link back to home page 200 during user navigation within user interface 160. Alarms tab 206 corresponds with alarm portion 222, preference tab 208 corresponds with preferences portion 224, and administration tab 210 corresponds with administration portion 226, which are described in more detail below. Through these portions, pages, tabs, and user interface 160 in general, a user can navigate within interface 160 and can add, edit, categorize, customize, and control BAS 10 by executing commands, often initiated by command buttons or links within pages. Activating these buttons navigates the user within one or more pages through which the user may carry out tasks and affect a wide variety of customizations. A user can also customize the behaviors and operation of user interface 160 itself.

The features described above may be illustrated by reference to the following examples of how user interface 160 may be navigated and utilized to customize and control BAS 10. In one embodiment, after connecting with BAS 10 and completing any required security routines, such as logging in, password input, and authentication of credentials, a user reaches home page 200.

Home page 200 has many general features in common with other pages of user interface 160, including linking, manipulating data onscreen, and providing an interface through which BAS 10 may be customized. Home page 200, like other pages seen by a user, includes both content, such as buildings index 230, and navigation tools, such as tabs 202. In the particular case of index 230, content and navigation tools are integrated, as buildings within index 230 are displayed as hyperlinks that direct a user to a building summary page for the selected building. Building summary pages are described in more detail below.

Instead of or in addition to the hyperlinks of index 230, home page 200 may include a navigable customized graphic, such as building map 231 depicted in FIG. 18B. Map 231 can be integrated with index 230, depicted on a distinct page reachable by hyperlink from home page 200 (as shown in FIGS. 18A and 18B), or depicted on home page 200 instead of textual index 230 in various embodiments of the invention. Map 231 is preferably navigable, wherein a user may select a particular building to be directed to that building's page.

Home page 200 includes a search input field 228 for executing searches of building directory 230 and its subdirectories. Interface 160, by web server 66 and browser 68 in cooperation with a database, can cache user visits to an interactions with specific pages and

directories and provide a list 238 of this cached information to enable a user to quickly return to a frequently visited page. Interface 160 also permits a user to import custom page links by a link 242 on home page 200. Custom links can also be removed through link 244.

Building index 230 is a dynamically extensible and customizable content and navigation feature of home page 200. Index 230 is preferably organized hierarchically or in some other manner intuitive to or specified by a user. For example, user interface 160 by default can list buildings alphabetically. With minimal information from a user, however, user interface 160 can group buildings geographically, such as by ZIP code. A user may also customize index 230 by specifying another attribute by which to arrange the buildings, such as a name, term, or building number. In a school district, buildings can be arranged by user-specified type, such as primary, elementary, and high school. A user can then easily locate and select a building from the directory by clicking a link to that building, either directly or after expanding the index directory until the building of interest is found.

Alternatively, a user may use find field 228 to search for and locate system buildings in a searchable database. In one embodiment, if a user enters a search term or string in field 228 and an exact match is found, user interface 160 will display a building summary page for the match. Building summary pages are described in more detail below.

In one embodiment when a building link within directory 230 or map 231 has been selected, a user is directed to a building-specific summary page 250 as shown in FIG. 20A. Similar to home page 200, building summary page 250 includes content and navigation tools. Navigation tools include building information tabs 252. Building information tabs 252 include a summary tab 254, which links to summary page 250; an alarms tab 256; a spaces tab 258; an equipment tab 260; a subsystems tab 262; a schedules tab 264; a data logs tab 266; and an advanced tab 268. These tabs and the information to which each links within user interface 160 will be described in more detail below, but generally are presented on a plurality of pages within user interface 160 that include similar content in order to provide a consistent, easily navigable format. Building information tabs 250 are preferably dynamic, based upon the data and information discovered and/or available for the particular building featured on summary page 250, or the particular space or equipment on other pages of user interface 160.

In general, the content of building summary page 250 relates to building equipment and building spaces. Building equipment includes panels, HVAC units, and other electrical and mechanical systems related to operations within the building. Building spaces are rooms, floors, or other areas within the building that are managed, controlled, or affected by the equipment.

Both spaces and equipment are relevant to the operation of BAS 10 and are of interest to users of user interface 160.

In particular, the content of building summary page 250, and other pages of user interface 160, includes status critical information. The content of summary page 250 therefore includes an alarm summary portion 310 and a spaces summary portion 330 in one embodiment to quickly synopsise events and provide status items of note to users. User interface 160 intuitively presents certain status critical information in proximity to other related or important information.

Referring to FIG. 20B, alarm summary portion 310 includes alarms associated with the building summarized on page 250 of FIG. 20A. Summary portion 310 provides a tabular organization of more detailed information related to each alarm that improves a user's ability to assess and respond to alarms, including an alarm severity level 312, occurrence time 314, type 316, alarm details 318, and an alarm source 320. Within alarm summary portion 310, relevant equipment and information can be automatically hyperlinked to other portions of user interface 160. For example, alarm source 320 can be hyperlinked to an equipment summary page (refer, for example, to FIGS. 24A and 24B and the related discussion below) for the equipment from which the alarm is originating.

Referring to FIG. 20C, spaces summary portion 330 includes information related to spaces within the building of summary page 250 to enable to a user to view current settings, view current status and operations, and quickly link to more detailed information about a space. Spaces summary portion 330 includes a user customizable space name 332, an equipment type identifier 334, a sensed space temperature 336, a current space temperature setpoint 338, a calculated variance 340, and an operational mode 342. Calculated variance 340 is a difference between current space setpoint 338 and sensed space temperature 336.

In FIG. 20C, spaces summary portion lists the twenty-five spaces having the greatest variance between setpoint and sensed temperature (not all twenty-five are visible in the screenshot of FIGS. 20A and 20C). The degree of variance from setpoint directs the user's attention to space conditions most likely to require attention. A user can specify that more or fewer spaces be included in spaces summary portion 330 by selecting a link 346. In other embodiments of the invention, and according to the user customization features of user interface 160, a user can specify which particular spaces are summarized in spaces summary portion 330, rather than including spaces based upon a setpoint variance.

Referring to FIGS. 20A-D, the content of page 250 can include both static and dynamic information related to the building. Static information includes a building's location and contact information 270 in one embodiment. Static and dynamic information can also be integrated on a

building floor plan reachable through floor plan link 272. In FIG. 20D, a building floor plan page 274 comprises a static building layout diagram 276 that includes dynamic building space status information in one embodiment. For example, occupied rooms 278 can be depicted in a first color, while a room 280 associated with an alarm or registering a comfort complaint within BAS 10 can be shown in a second color. Another room 282 for which a status has been altered or a comfort complaint remedied can be highlighted in yet another color by which a user can quickly locate a relevant space and determine a current status at a glance. A particular room may then be selected to reach a space page, equipment page, or alarm log, for example.

Dynamic information included on summary page 250, floor plan page 274, and other pages within user interface 160 can be updated in a plurality of ways. On page 250, alarm summary portion 310 and space summary portion 330 include dynamic information which can be automatically updated, or refreshed, periodically. In one embodiment, the dynamic information can be updated by BAS 10 and ESE 20 every ten minutes, although a more or less frequent refresh may occur in other embodiments or may be user-defined within set parameters. More frequent updates place a higher burden on BAS 10 and therefore, in one embodiment, a user may select from refresh rates predetermined not to have a detrimental affect on BAS 10 performance.

Dynamic information may be updated on demand. An on-demand refresh of alarm summary portion 310 can be user initiated by activating a refresh alarms link 322, and an on-demand refresh of spaces summary portion 330 can be initiated by activating refresh spaces link 344. Certain priority features of user interface 160, like alarms, which are important to the performance, safety, and integrity of BAS 10 operation, are associated with an automatic and dynamic prompt, such as new alarms prompt 324. To minimize impact on BAS 10 bandwidth performance, ESE 20 can provide prompt 314, alerting a user that a manual refresh may be helpful, rather than frequently updating alarm summary portion 310 even if no updated information is available.

From summary page 250, a user may access yet more detailed information about the spaces of the selected building. FIG. 21 shows schematically how a user can navigate from summary page 250 in one embodiment of the invention. From summary page 250 and other pages displayed by user interface 160, vertical tabs alarms 256, spaces 258, equipment 260, subsystems 262, schedules 264, data logs 266, and advanced 268 provide quick navigation links. As previously mentioned, a user can navigate from within alarm summary portion 310 and spaces summary portion 230 of building summary page 250. For example, selecting a space (332) from spaces summary portion 230 takes a user to a spaces page associated with the

selected space. An example spaces page 350 is depicted in FIG. 22A. Spaces page 350 includes a spaces status table 352, shown in detail in FIG. 22B, which lists information about the subject space.

Within BAS 10, spaces can be grouped and defined according to BAS 10 default rules or user customized rules, and this information can be presented at-a-glance proximate status critical and other important information regarding equipment, or a building, space, system, or subsystem. Refer also to the discussion above regarding home page 200 and building index 230, and to page 380 of FIG. 24A and page 394 of FIG. 25. For example, a nation-wide retailer may choose to group its stores by geographic or sales region, by store type or format, by time zone, or by some other characteristic. BAS 10 may default to grouping spaces by geographic location. In one embodiment, a current group to which the subject space of page 350 belongs is provided as a group link 353 next to the heading "Member Of." Group link 353 can also be used to determine what or who is responsible for a particular building, space, equipment, or system. Link 353 also provides navigation to other group members and information. Group information may be dynamically discovered by ESE 20 if group, parent, and/or child information is provided, shown, or exposed during set-up or discovery. A user is therefore able to quickly ascertain the relevant group to which a space belongs and, by selecting group link 353, retrieve a page of user interface 160 that provides a group summary and group editing capabilities. By way of example, the national retailer previously mentioned could edit the setpoints related to all of its locations in a particular time zone to accommodate an earlier opening time for a holiday sale by changing the values group-wide on a single page, rather than editing the information individually for each group member.

A user can also dynamically create and edit groups, as the group assignments are not fixed and do not require customized programming. Referring also to FIGS. 1 and 19, ESE 20 operating application 70 discovers buildings 72. Through the discovery process, ESE 20 learns standard attributes 74 about the building and its panels and equipment. Standard attributes 74 are stored in database 60. ESE 20 and applications 70 then can formulate a default building index 230 based upon standard attributes. As mentioned above, a user can provide custom attributes 75 to ESE 20 and applications 70 via user interface 160. Custom attributes are also stored in database 60. A user can then specify at any time a custom building index 230 based on custom attributes 75 or a combination of standard attributes 74 and custom attributes 75. Index 230 and related groups can be changed and immediately implemented by ESE 20 for display in user interface 160 should an attribute 74, 75 be edited or updated or a new building discovered.

Further, ESE 20 can dynamically and automatically update groups and index 230 for newly discovered buildings if common standard or custom attributes 74, 75 are found.

User interface 160 can also relay information regarding a space occupancy status. An occupancy indicator 354, a schedule indicator 355, and a next event indicator 356 are provided on page 350. This information can be helpful for maintenance, scheduling, and/or value alteration purposes. For example, a user may not want to edit certain setpoints while a space is occupied but rather wait until the space is unoccupied. Or, a user may desire to determine or update scheduling information related to occupancy.

Schedule indicator 355 also provides a user with at-a-glance control information, such as whether equipment is controlled by a main schedule or is under a special schedule. A main schedule is a primary set of operating characteristics for entities operating within BAS 10, such as buildings, spaces, equipment, devices, systems, and subsystems. In one embodiment, a main schedule controls basic operations and set points. Special schedules may be implemented to accommodate limited run or short term changes, such as for a holiday, to accommodate maintenance or a special event, or for some other reason. Special schedules are preferably used for short term or temporary changes overriding the main schedule to prevent special schedules from being left active unintentionally. Special schedules also provide a way to schedule temporary events or occurrences without having to alter the main schedule. Next event indicator 356 provides a brief schedule preview of the next event scheduled for the equipment. Providing group member, occupancy, control, and event information proximately and on page 350 enables a user to quickly determine current and impending status information for equipment without having to access multiple pages or navigate to find desired information.

Referring to FIGS. 22A and 22B, spaces table 352 includes a space condition portion 358 and a system status portion 359. Spaces table 352 thus includes information likely to be status critical and of greatest interest to a user first accessing user interface 160 to spaces page 350. Spaces table 352 presents space condition portion 358 and system status portion 359 proximate each other, enabling a user to quickly assess a space status, access additional information, and edit set points, if desired.

Space condition portion 358 includes available space conditions 360, current sensed conditions 362, new value fields 364, and data log selectors 366 in one embodiment. System status portion 359 includes similar information. Current sensed conditions 362 can include temperature, humidity, and other real-time sensed values. In one embodiment, spaces table 352 includes a real-time sensed temperature value and displays a current active setpoint. A user can alter a desired heating or cooling temperature setpoint easily and conveniently within user

interface 160 by entering the desired value in a corresponding new value field 364 and instructing BAS 10 to apply the new values by selecting button 368. BAS 10 can incorporate the update immediately without system interruption or recompilation.

Regarding data log selectors 366, the manner in which data is collected can be use customized using a “set up data logs” sequence. By checking a log data box 316 corresponding to specific equipment and activating a set up data logs button 326, the user can set data collection intervals and adjust the time period for the collections. Instead of a date range as the time period for collections, the user may alternatively select a fixed number of samples for collection. An example data log sequence is depicted in FIG. 23.

An equipment summary page 380 is depicted in FIG. 24A. Pages of user interface 160 relevant to specific equipment, similar to building summary page 250 and space summary page 350 described above, are accessible when the user selects equipment tab 260 or otherwise navigates within user interface 160. Similar to as described above with respect to selecting a space from space summary portion 330 of building summary page 250 to navigate to space summary page 350, selecting an alarm source (320) from alarm summary portion 310 of FIGS. 20A and 20B also directs a user to an equipment summary page 380. It will be appreciated by those skilled in the art that there typically are several ways to navigate to any given page in user interface 160; certain navigation paths are described herein in order to define an overall organization, layout, and flow of one embodiment of user interface 160.

On page 380, various categories of available equipment appear as subheadings 382 below tab 260, such as “Chiller,” “Air Handler,” and “Programmable Controller.” Selecting a desired subheading 382 directs the user to a list of the specific units within each category from which a particular equipment unit can be selected to display an equipment status page.

As previously described above with reference to FIGS. 22A and 22B, current status values and setpoints are displayed on page 380 in equipment status summary portion 384, as well group data 353, 354 and links 386 to additional information regarding the subject equipment. For example, equipment status page 340 relates to a chiller. Referring to FIGS. 24A and 24B, chiller status summary portion 384 is divided into a chiller condition portion 388 and a status portion 390 and lists static and real-time dynamic information about the specific chiller, such as current values 392 for various aspects of the chiller’s status and performance. An equipment graphic page 381, reachable via an equipment graphic link 386, is depicted in FIG. 24C. Equipment graphic page 381 also includes static and dynamic graphics and text related to particular equipment and systems and spaces associated with the equipment.

Pages and links similar to those described above are also provided for other categories of equipment in BAS 10. In FIG. 24A, chiller status page 380 was depicted, but other status pages for other equipment, such as air handlers, are also included in user interface 160. An example air handler status page 394 of user interface 160 is depicted in FIG. 25. Similar customized status pages within user interface 160 may also be used for any other equipment controlled or managed by BAS 10. For example, certain new values can be substituted by the user, including setpoints, heat/cool mode, and discharge air temperature, as previously described, for programmable equipment controlled by BAS 10. BAS 10 can accept new values on-the-fly, without requiring code recompilation, system restart, or some other disruption or suspension of normal activity of BAS 10.

Referring briefly to FIG. 20A, subsystems tab 262 provides links to portions and pages of user interface 160 that display information related to equipment systems and subsystems of BAS 10. For example, FIGS. 26A-C depict an example subsystem summary page 400 of user interface 160 related to a chiller plant. Although page 400 relates specifically to a chiller plant, the equipment choice of this example is arbitrary and the general features of page 400 are generally relevant within user interface 160 and BAS 10 to virtually any equipment system or subsystem. Different values and information will be relevant to different equipment systems; accordingly, some variation from the particular examples depicted in and described with reference to FIGS. 26A-C will exist on other equipment system pages. As previously described with respect to other pages of user interface 160, page 400 includes status tables related to equipment subsystems, including subsystem status portion 402. Subsystem summary page 400 also includes an equipment status portion 404.

Custom screens and pages such as page 400 of are presented in user interface 160 to simplify the information presented regarding complex systems and subsystems. Raw data and information not edited and tailored for presentation via user-intuitive page 400 could be overwhelming and therefore not useful to the average user. From page 400, however, a user may view status critical information and access more detailed data and information about sophisticated systems and subsystems as needed.

Subsystem status portion 402 as depicted in FIGS. 26A and 26B includes information about a chiller plant, which is one or more chiller units operating as a group. The information includes current static and dynamic values 406 relating to chiller plant conditions and current operational information 408. A static value 406 is, for example, a current setpoint, while a dynamic value 406 in table 362 is a return or supply water temperature. Operational information 408 provides scheduling and maintenance information and user control features. For the chiller

plant of page 400, chiller rotation 410, addition 412, and subtraction 414 can be scheduled, with current schedules 416 displayed on page 400. In general, schedules define relationships between objects in BAS 10, time, and/or other objects in BAS 10. A user can define or alter schedules related to an object in one embodiment of the invention. A user can also manually implement or
5 force rotation, addition, or subtraction as needed or desired through page 400 and through other pages of user interface 160. Basic operational status information is also provided.

Page 400 further displays equipment status portion 404, which includes equipment identifier links 418 for each chiller as depicted in FIGS. 26A and 26C. Identifier links 418 can be user customized names or default system values obtained during a discovery or integration
10 process. In the example of page 400, three chillers of BAS 10 are identified as Chiller 1, Chiller 2, and Chiller 3. The equipment identifiers are hyperlinked (418) to other portions and pages of user interface 160. For the chillers listed in equipment status portion 404 of page 400, equipment identifier links 418 direct a user back to an equipment page 380 (refer to FIG. 26A) for the individual chillers. A user may also manually control values and settings for selected
15 chillers through status portion 404, for example by selecting a button or link 420, marking a selector field 422, making a selection from a drop-down menu 424, or by other information editing or entry means. For the chiller plant of page 400, a user can apply new values 426, initiate a failure reset 428, or make a particular chiller within the chiller plant available or unavailable 430.

Similar sets of pages are provided for other equipment subsystems of BAS 10, such as a heat pump loop and variable air subsystems. The pages for these equipment subsystems may also be configured to display information specific to a particular subsystem. For example, as shown in FIG. 27, a variable air page 440 includes tabular information 442 identified and sorted according to user customized information 444. In this case, the information is sorted by a
25 person's name. The person may be a maintenance or managerial person responsible in some way for a space, or the person can be associated with the space in another manner, such as by physical office or workspace assignment. A user can therefore customize page 440 and other pages with associations, identifiers, and references that are familiar, making BAS 10 more easily understood and manageable via user interface 160.

The descriptions and depictions of the above-described specific pages set forth by example the general functionality and operation of BAS 10, in particular user interface 160, and are a context within which the following description of use of user interface 160 can be understood according to one embodiment. As previously mentioned, a variety of user customization and control features are provided by user interface 160. Through links on pages
30

displaying space and equipment information, the user can change setpoints, control data logging, and create custom pages.

Administration link 210 (refer to FIG. 21) directs the user to an administration portion 226 of user interface 160 comprising a series of pages that are organized as shown in FIG. 28 in one embodiment. Administration portion 226 of user interface 160 provides administrative customization and control of BAS 10, which generally relates to the addition or removal of the data shown on the various equipment pages, the ability to manage users of the system, install new buildings, manage alarm routing, and view system tasks, as previously described.

In one embodiment, BAS 10 provides user access at more than one level. High-level users can manage the level of access granted to other users in administration portion 226 through managing users portion 226A. Other user management options may also be available. Level access can be controlled via a user login, password, and/or other user identification processes. A user is generally a person whose duties typically relate to monitoring or controlling BAS 10 without having to engage in programming or the recompilation of software code. A high-level, or administrative, user is typically a user who generally has a higher level of access to system controls and customization functions. For example, an administrative user may be provided with a login code that authorizes access to pages that a general user could not access. Despite this greater access, however, an administrative user, like a general user, will not normally be expected to engage in reprogramming or recompiling to customize user interface 160 or BAS 10. Administration portion 226 of user interface 160 can be wholly or partially available to users based upon their level of administrative access. Administration portion 226 of user interface 160 also includes utilities for installing buildings 226B, managing alarm routing 226C, viewing system tasks 226D, and performing advanced tasks 226E, such as configuration of system parameters, creating and managing custom attributes, creating schedules, and customizing pages viewable in user interface 160.

An example building installation page 828 is depicted in FIG. 29. The dynamic extensibility of BAS 10 provides for the periodic discovery, addition, or uploading of new buildings or panels. Page 828 includes a progress and status portion 829 for each of the buildings undergoing the installation process, which includes information regarding the status of the communications between ESE 20 and the new building (or panel) and the number of panels loaded out of the total number of panels.

A user can also view and manage system tasks 226D. Advanced tasks 226E are shown in more detail in FIG. 30. Advanced tasks 226E include the customization of system pages 812, the

management of custom attributes of buildings 814, the management of alarm settings 816, and the management of schedule application settings 818.

Customization of system pages 812 includes both content and layout control options. Referring to FIG. 31, a system customization page 820 includes user selectable options for customizing the display of buildings in the index on home page 200. Customization can be carried out by a user by selecting the desired number of index levels 822 and assigning a corresponding number of general and special grouping attributes 824, 826. With one index level, the buildings are grouped according to one general attribute. When two index levels are used, the buildings are grouped by a special attribute within a general attribute. The grouping attributes 824, 826 are relevant to the display and arrangement of buildings in building index 230 (see FIG. 18A) and are included as group link 353 (see FIG. 22A), for example. The number of index levels determines how the special and general attributes affect the arrangement and appearance of index 230. With no index levels, all buildings are displayed together in alphabetical order in index 230 according to a default setting of BAS 10 and user interface 160 in one embodiment.

Other page customizations relate to the links available on home page 200 and the addition or removal of data shown on equipment pages and equipment subsystem pages. Referring to FIG. 32, a customize system page 830 is depicted. Page 830 specifically relates to adding or removing data to be shown on equipment or subsystem pages (refer, for example, to FIGS. 24A, 25, and 26A described above), but the format is exemplary of and generally relevant to adding or removing data on home page 200 or on other pages of user interface 160. A user first selects the equipment or subsystem data to be customized at 832. Then, the user is directed to a table showing all currently displayed data points, as well as default display settings, for that equipment or subsystem, such as on page 834 in FIG. 33. The user may customize the display by selecting or deselecting data points in selection table 836 as desired.

A user may also add or remove links to home page 200 and to other pages of user interface 160 by importing and removing custom links. The custom links added may be to other pages of user interface 160, or internal links. The links added may also be external, such as to web or Intranet pages. A user may desire to link to a news or weather site publicly available on the Internet. A user may also link to non-public pages or information. For example, if BAS 10 relates to a college campus, a user may link to an internal campus event calendar or information page, such as a staff and faculty directory. The custom links can be added to pages of user interface 160 to integrate virtually any information a user deems helpful to management of BAS 10.

User interface 160 therefore provides ways in which the user can streamline interface 160 to include only those links relevant to the user's task. Further, BAS 10 allows each user of interface 160 to customize the pages and links according to their preferences and tasks in one embodiment. Therefore, users responsible for different tasks or having varying duties can create
5 their own custom user interface 160. BAS 10 serves and loads the proper custom user interface 160 by saving and associating the customizations with a user identifier, such as by a logon routine. In another embodiment, only an administrative user can customize interface 160 in this manner, providing a single user interface 160 for standard level users.

A related customization function accessible via through advanced tasks 226E concerns
10 custom attributes of buildings 814. Referring to FIG. 34, a custom building attributes management page 840 allows a user to create and manage the custom attributes both for buildings currently managed by BAS 10 and for buildings yet to be discovered. In one embodiment, four types of attributes 842 may be used. A first type of attribute 844 has two choices; two mutually exclusive values are defined that require a choice. A second type 846 is a
15 fixed list, where a choice is made either from the list or entered manually. A third type 848 is an expandable list, which is an initial list that may be supplemented by the user. A fourth type 850 is a free form value, which is a unique value for a building that a user can create. When creating custom building attributes, the user may assign a new attribute as a default value for all newly discovered buildings and may also select the attribute for use in customizing the building index
20 on home page 200.

Referring to FIGS. 35A-C, an important aspect of BAS management and control is the efficient receipt and handling of system alarms 816. The management of and response to alarms in BAS 10 may be user customized according to one embodiment of the invention, generally available through alarm tab 206 (refer to FIG. 18A, for example).

Navigation within user interface 160 to the alarm mapping pages according to one
25 embodiment is represented in FIG. 36. Alarms tab 256 has a map priorities sub-tab 257 that directs a user to an alarm management page 860, such as that shown in FIGS. 35A-C, on which the user may select a panel type 862, view a list 863 of panels of different types, and map alarm priorities 864 based on panel priorities. Page 860 also allows the user to add new panel types
30 866, as shown in FIG. 37.

Alarm mapping refers to the assignment of priority levels to the panels based on panel type. In one embodiment, a user can specify alarm priorities to be assigned both to system panels and panels that have not yet been discovered by the system. By assigning alarm priorities for panels not yet discovered, user interface 160 gives the user control over how the dynamic

extensibility of BAS 10 will be implemented in the context of future panel or building additions to the BAS.

Alarms are generated by BAS 10 in a variety of situations, such as when variances in temperature and other variances from predetermined setpoints are recorded. In one embodiment, BAS 10 alarm handling can be user customized. For example, alarm notifications may automatically be sent to one or more designated email or text message accounts. Audio or other text and visual notifications can also be automatically sent by BAS 10, such as to pagers, cellular phones, network broadcast messages, and the like. Within user interface 160, and in addition to or instead of email messages, alarms can also be displayed in tabular or list form on a building summary page.

According to alarm routing 226C, a user can also route email notifications for certain panel types that may be discovered by ESE 20 in the future. The routing and display of alarms may be customized by matching alarm attributes to one or more specific email recipients. Alarm attributes can relate to the type of alarm, time of alarm, alarm trigger, alarm location, occurrence or repetition of multiple alarms, patterns of alarms, or some other characteristic or combination of characteristics. Thus, in one example, a user who is the manager of a particular building at a site within BAS 10 can be designated within BAS 10 to receive alarm notifications for each alarm related to that building. In another example, a site manager and each member of the electrical maintenance staff of a building can be designated to receive alarm notifications related to electrical faults in that building. In yet another example, different alarm notification recipients and formats can be user customized based on the time of day or the day of the week. During the day when a user is generally interacting with user interface 160 directly via device 22, a tabular display on a building summary page can be specified. After hours, email and/or paging notifications can be used instead of or in addition to alarm notifications in user interface 160.

Alarm handling and priorities may also be customized in advance for panels to be discovered by ESE 20. Future panels may be assigned alarm priority status based on user preferences. A user may assign a general alarm priority or response based upon panels currently known in and any to be discovered in a particular building. Alarm priorities can also be assigned based upon panel characteristics. If a panel having a characteristic is later discovered, BAS 10 can automatically assign priority or handle alarms based upon the user-selected characteristic.

In another embodiment, BAS 10 can assign priority and manage alarms by default, by associating newly discovered panels as the same as or similar to known system panels and assigning like management features. For example, a user customizes the handling of alarms for a particular panel by specifying a response procedure. In the future if a new panel is discovered by

BAS 10 and if the newly discovered panel shares characteristics with the panel for which a response has been set, BAS 10, in the absence of instructions or customization relating to the newly discovered panel, can similarly handle alarms for the newly discovered panel.

5 The scope of the claims should not be limited by particular embodiments set forth herein,
but should be construed in a manner consistent with the specification as a whole.

CLAIMS:

1. A building automation system (BAS) comprising:
 - an architecture comprising a communication network and having a dynamic extensibility capability and an automatic configuration capability;
 - an engine communicatively coupled to the communication network; and
 - at least one control device communicatively coupled to the communication network, the at least one control device being known or unknown to the engine,wherein the engine is adapted to selectively implement the dynamic extensibility capability to establish communications with and to control the at least one control devices, and wherein the engine is adapted to selectively implement the automatic configuration capability to determine at least one characteristic of the at least one control devices.
2. The BAS of claim 1, wherein the at least one characteristic is selected from the group consisting of a communication protocol, a communication protocol version, a vendor, a product, a type, and a version.
3. The BAS of claims 1 or 2, wherein the characteristic is referenced as a definition and the definition is stored by the engine in a program.
4. The BAS of claim 3, wherein the engine need not recompile the program after storing a definition.
5. The BAS of claims 3, wherein the dynamic extensibility capability comprises a discovery routine to integrate an unknown control device into the architecture as a known control device by adding or altering a definition in the program.
6. The BAS of claim 5, wherein the engine need not recompile the program after adding or altering a definition.

7. The BAS of claim 3, further comprising a database communicatively coupled to the communication network and controlled by the engine, wherein the program is stored in the database.
8. The BAS of claim 3, wherein the engine is capable of applying the definition to identify an unknown control device by the dynamic extensibility capability and the automatic configuration capability.
9. The BAS of claim 8, wherein the engine comprises a generic communication protocol compatible with known and unknown control devices.
10. The BAS of claim 3, wherein a first control device comprises a first communication protocol compatibility and a second control device comprises a second communication protocol compatibility different from the first communication protocol compatibility, and wherein the communication system and the engine comprises both the first and second communication protocol compatibilities.
11. The BAS of claim 10, wherein the first and second communication protocol compatibilities differ according to a communication standard, a version, or any combination thereof.
12. The BAS of claim 10, wherein the engine is capable of concurrently implementing both the first and second protocol compatibilities.
13. The BAS of claim 1, wherein the communication network comprises an Intranet network, an Internet network, or any combination thereof.
14. The BAS of claim 13, wherein the engine is assigned a single address and the control device is assigned a network address.

15. The BAS of claim 14, wherein the engine discovers the network address according to the dynamic extensibility capability.

16. The BAS of claim 14, wherein the engine receives the network address from an external source.

17. A method of adding a control device to a building automation system (BAS) by dynamically extending and automatically configuring an architecture of the BAS, the method comprising the steps of:

obtaining a network address of a previously unknown control device at a site;

implementing a discovery process to attempt to automatically establish communications with and obtain metadata from the control device utilizing the network address;

recognizing the control device as a portion of the BAS by synchronizing the site with the architecture of the BAS by evaluating at least one characteristic of the metadata and storing the at least one characteristic as a definition utilized in a program of the architecture and dynamically extending and automatically configuring the architecture of the BAS by executing the program without recompilation if communications can be established with the control device; and

requesting manual programming for the control device if communications cannot be automatically established with the control device.

18. The method of claim 17, wherein the step of requesting manual programming further comprises the steps of:

automatically requesting manual programming;

manually creating a control device definition utilized in the program; and

recognizing the control device as a portion of the BAS after the manual programming is entered.

19. The method of claim 18, further comprising the step of: recompiling the program after manual creation of the control device definition.

20. The method of claim 18 or 19, wherein the step of recognizing the control device as a portion of the BAS further comprises attempting to automatically determine a communication protocol compatible with the control device.

21. The method of claim 20, wherein the step of attempting to automatically determine a communication protocol compatible with the control device further comprises analyzing the metadata from the control device to determine a compatible communication protocol.

22. The method of claim 21, further comprising the steps of:

- determining if a vendor characteristic of the communication protocol can be specified by the control device, and

- if a vendor characteristic cannot be specified, selecting a basic communication protocol;

- if a vendor characteristic can be specified, determining if a product characteristic of the communication protocol can be specified by the control device, and

- if a product characteristic cannot be specified, selecting a communication protocol compatible with the vendor characteristic;

- if a product characteristic can be specified, determining if a control device type characteristic can be specified by the control device, and

- if a control device type characteristic cannot be specified by the control device, selecting a communication protocol compatible with the vendor characteristic and product characteristic; and

- if a control device type characteristic can be specified, selecting a communication protocol compatible with the vendor characteristic, the product characteristic, and the control device type characteristic.

23. A server engine for a building automation system (BAS), the server engine comprising:

means for obtaining a network address of a previously unknown control device at a site communicatively coupled to the BAS;

means for implementing a discovery process to attempt to automatically establish communications with and obtain metadata from the control device utilizing the network address;

means for synchronizing the site with the BAS by evaluating at least one characteristic of the metadata and storing the at least one characteristic as a definition utilized in a software program of the server engine;

means for altering a status of the control device from unknown to known; and

means for dynamically extending and automatically configuring the BAS by executing the software program without recompilation.

24. The server engine of claim 23, further comprising:

means for determining if a vendor characteristic of the communication protocol can be specified by the control device;

means for selecting a basic communication protocol if a vendor characteristic cannot be specified; means for determining if a product characteristic of the communication protocol can be specified by the control device if a vendor characteristic can be specified;

means for selecting a communication protocol compatible with the vendor characteristic if a product characteristic cannot be specified;

means for determining if a control device type characteristic can be specified by the control device if a product characteristic can be specified;

means for selecting a communication protocol compatible with the vendor characteristic and product characteristic if a control device type characteristic cannot be specified by the control device; and

means for selecting a communication protocol compatible with the vendor characteristic, the product characteristic, and the control device type characteristic if a control device type characteristic can be specified.

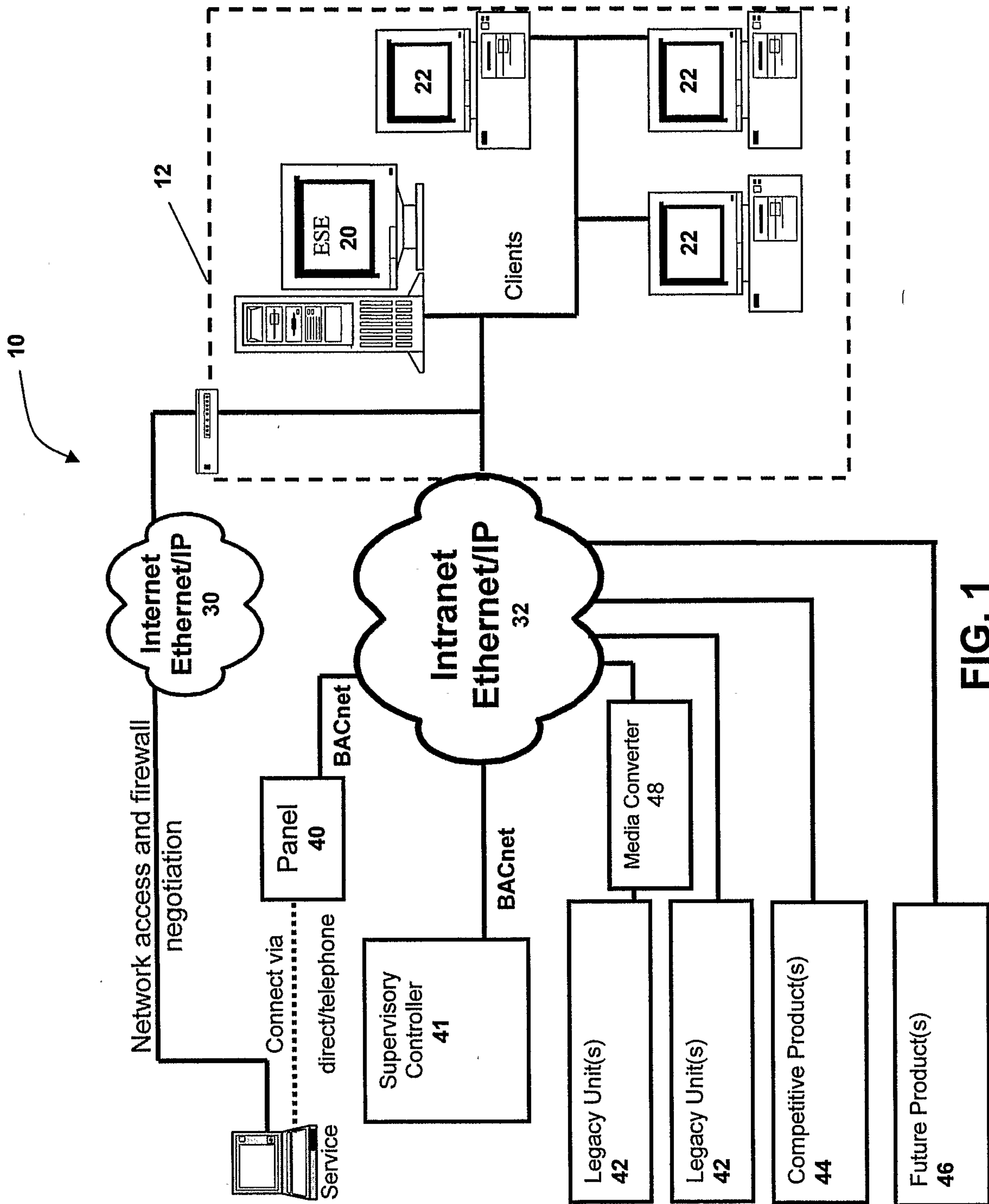
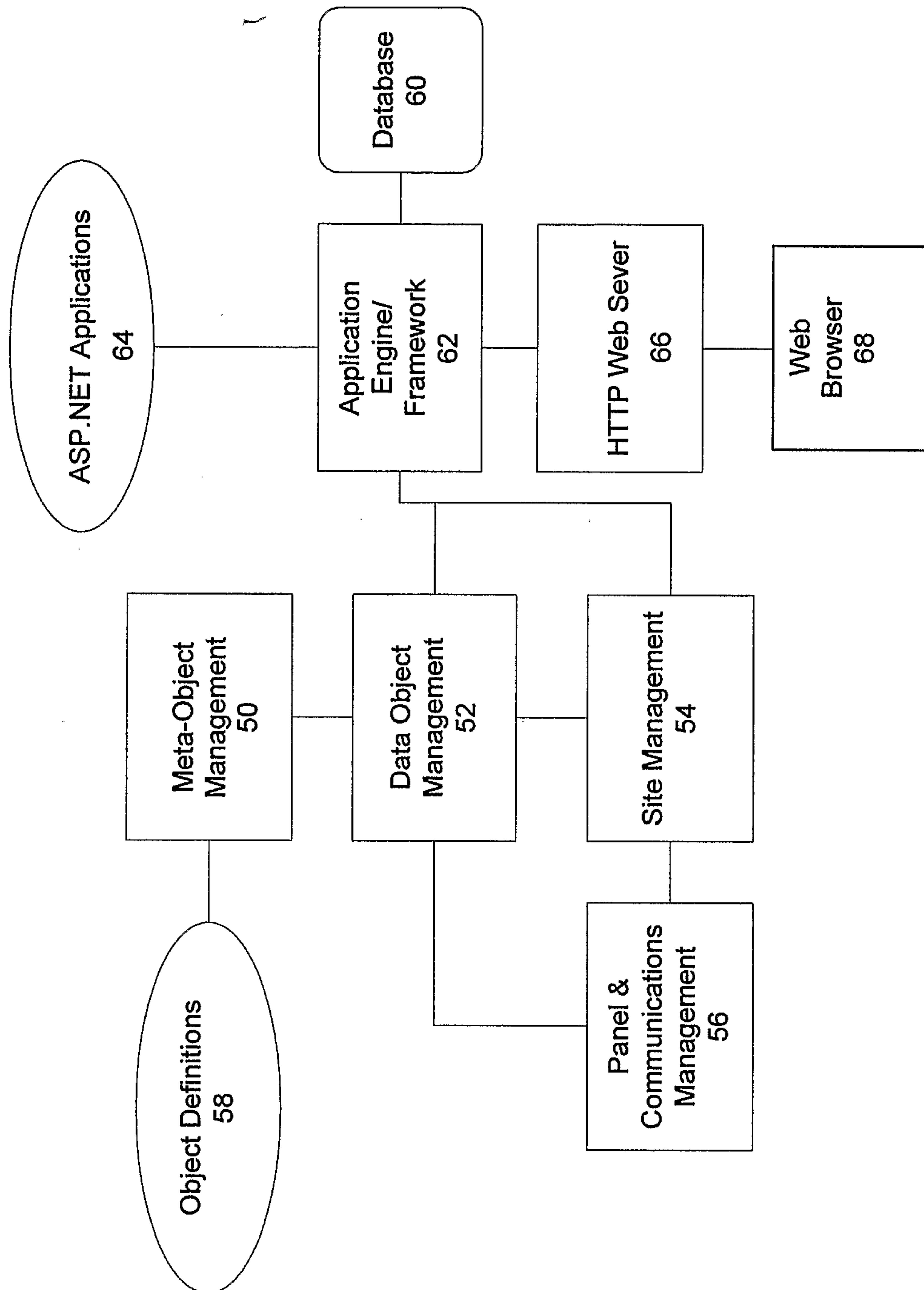
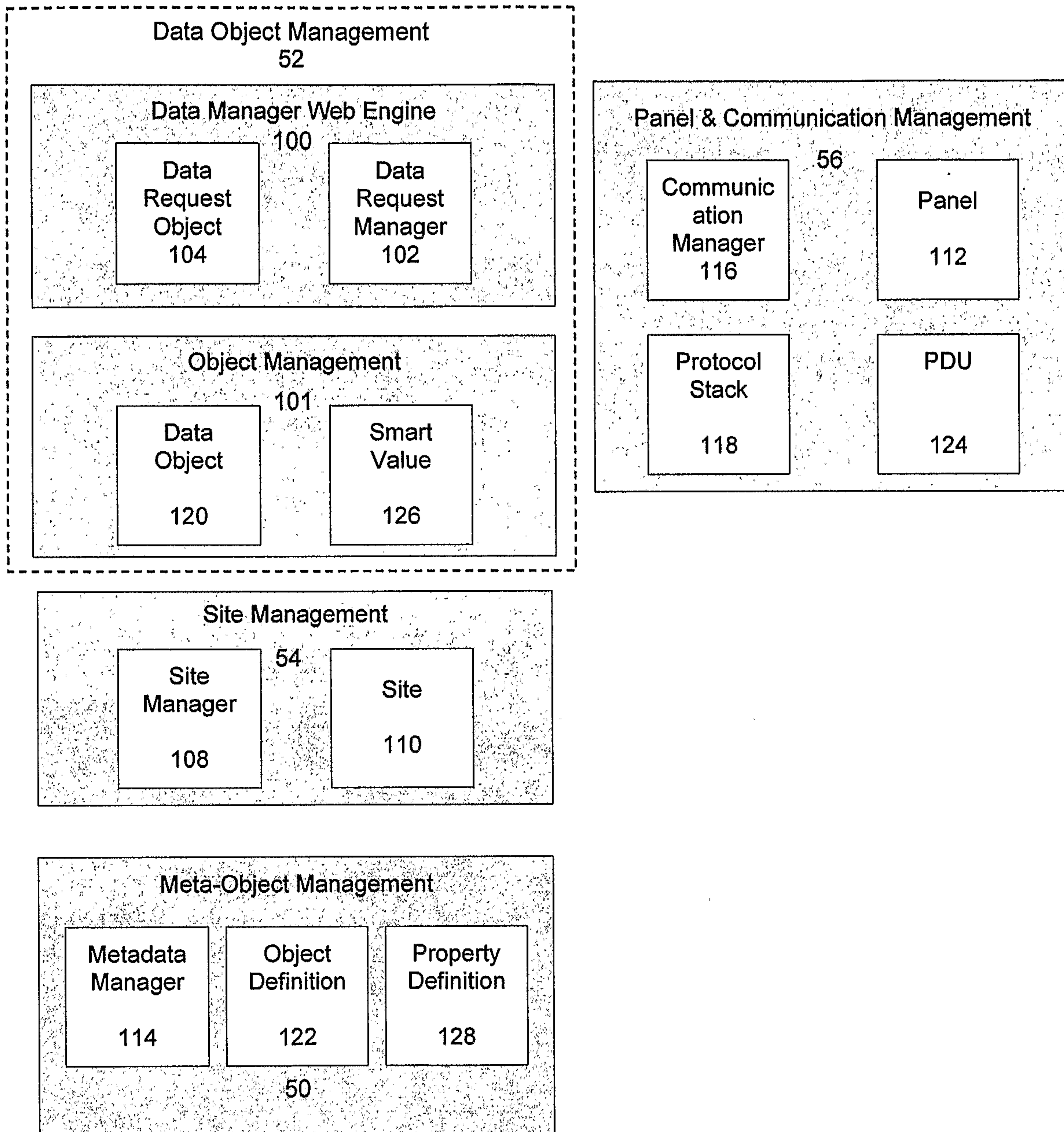
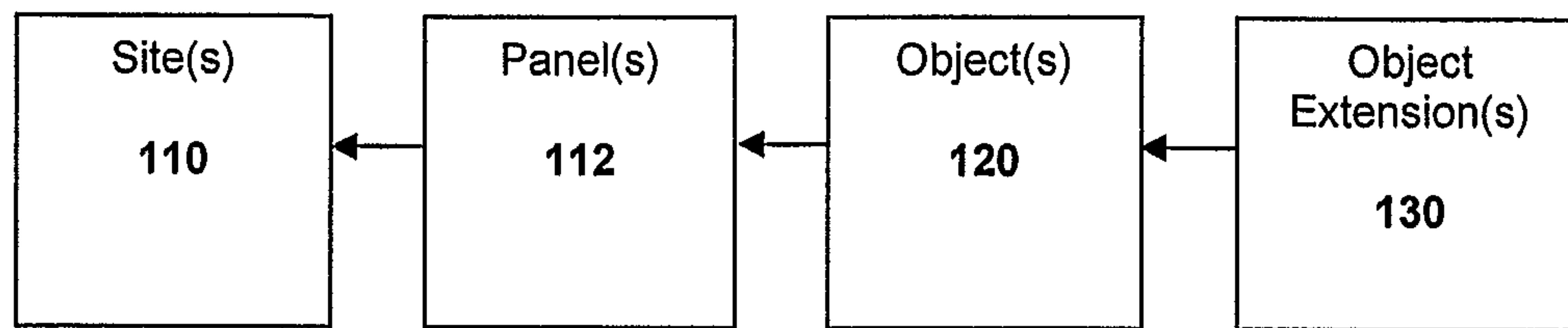
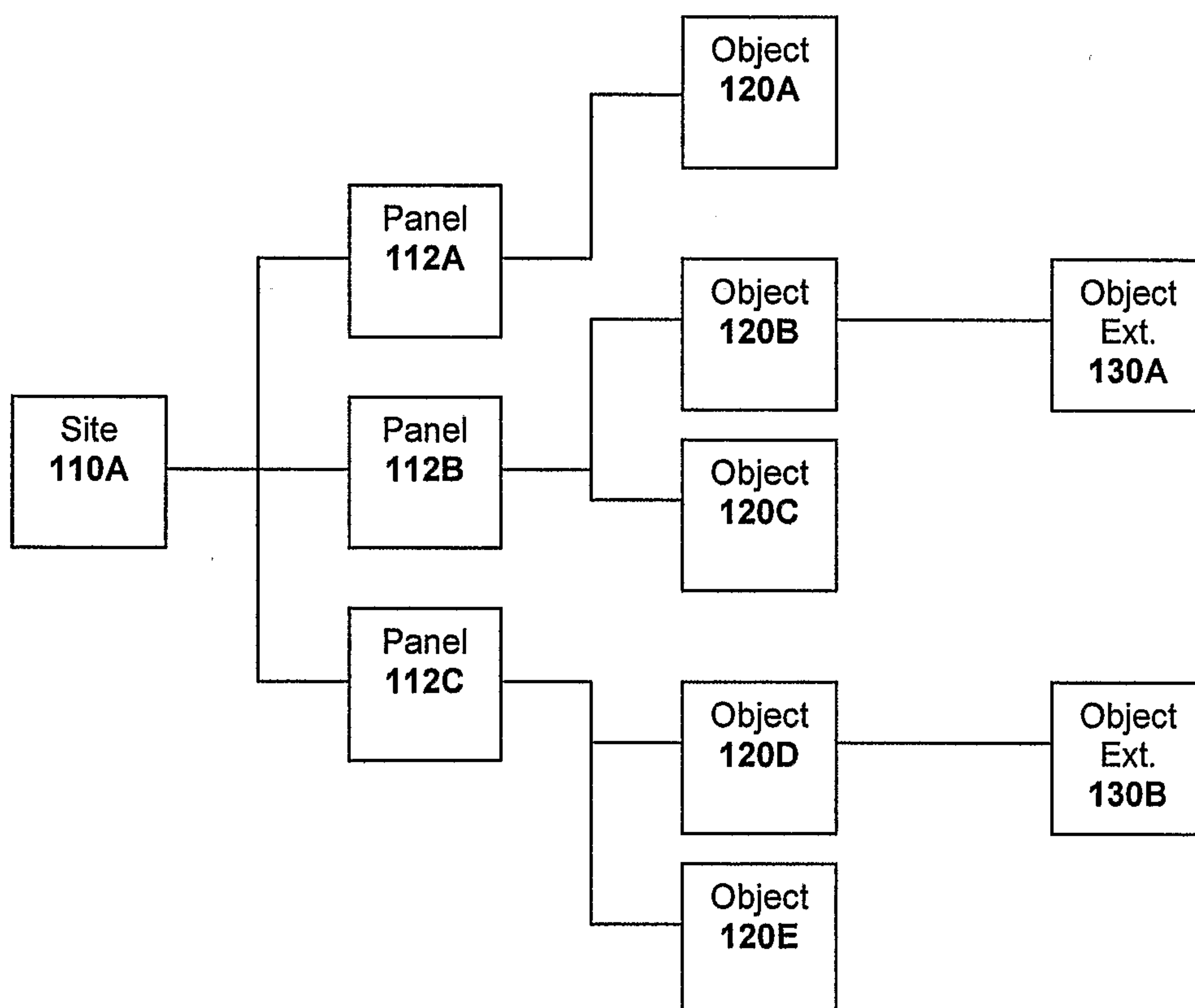


FIG. 1

**FIG. 2**

**FIG. 3**

**FIG. 4A****FIG. 4B**

College Campus Example

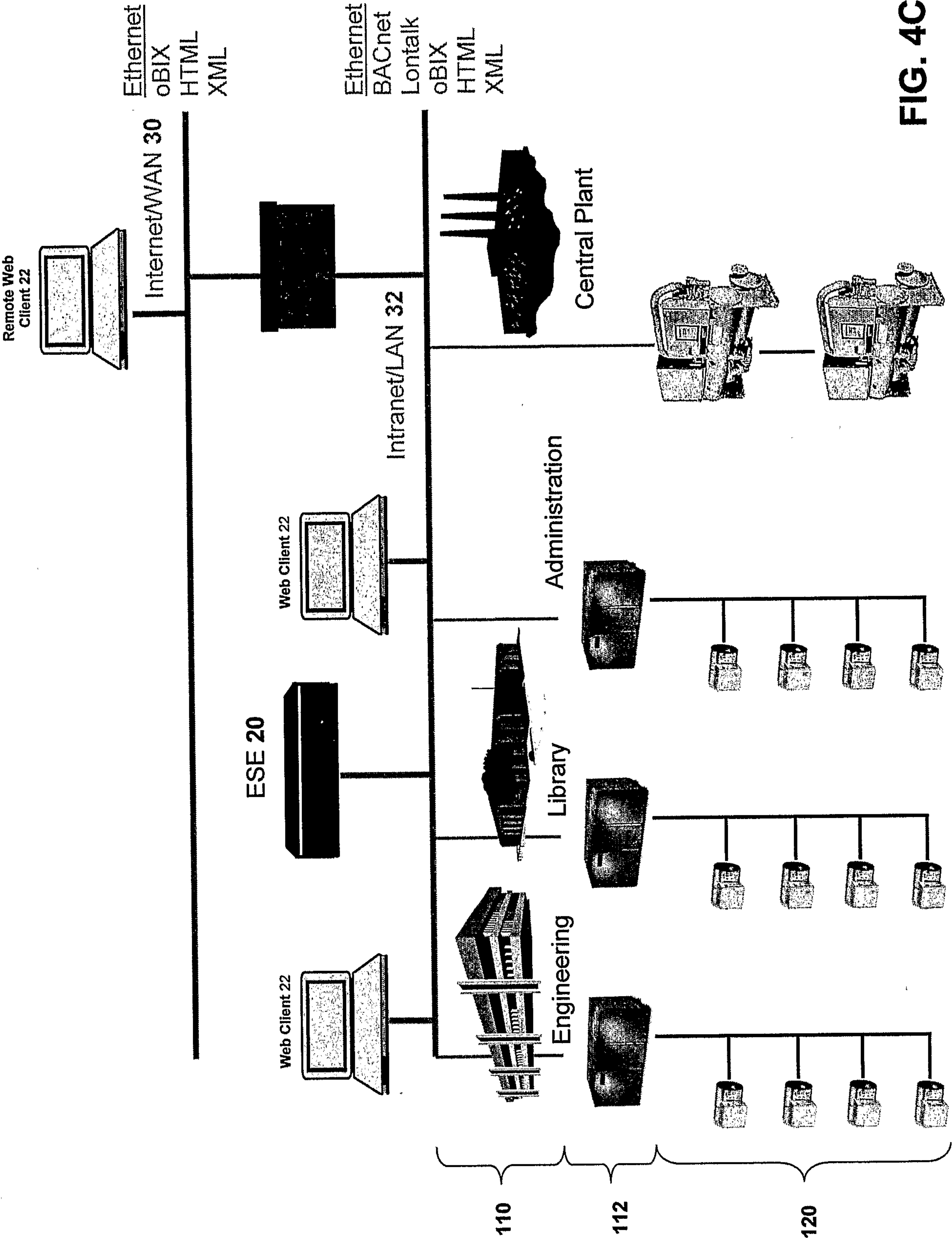
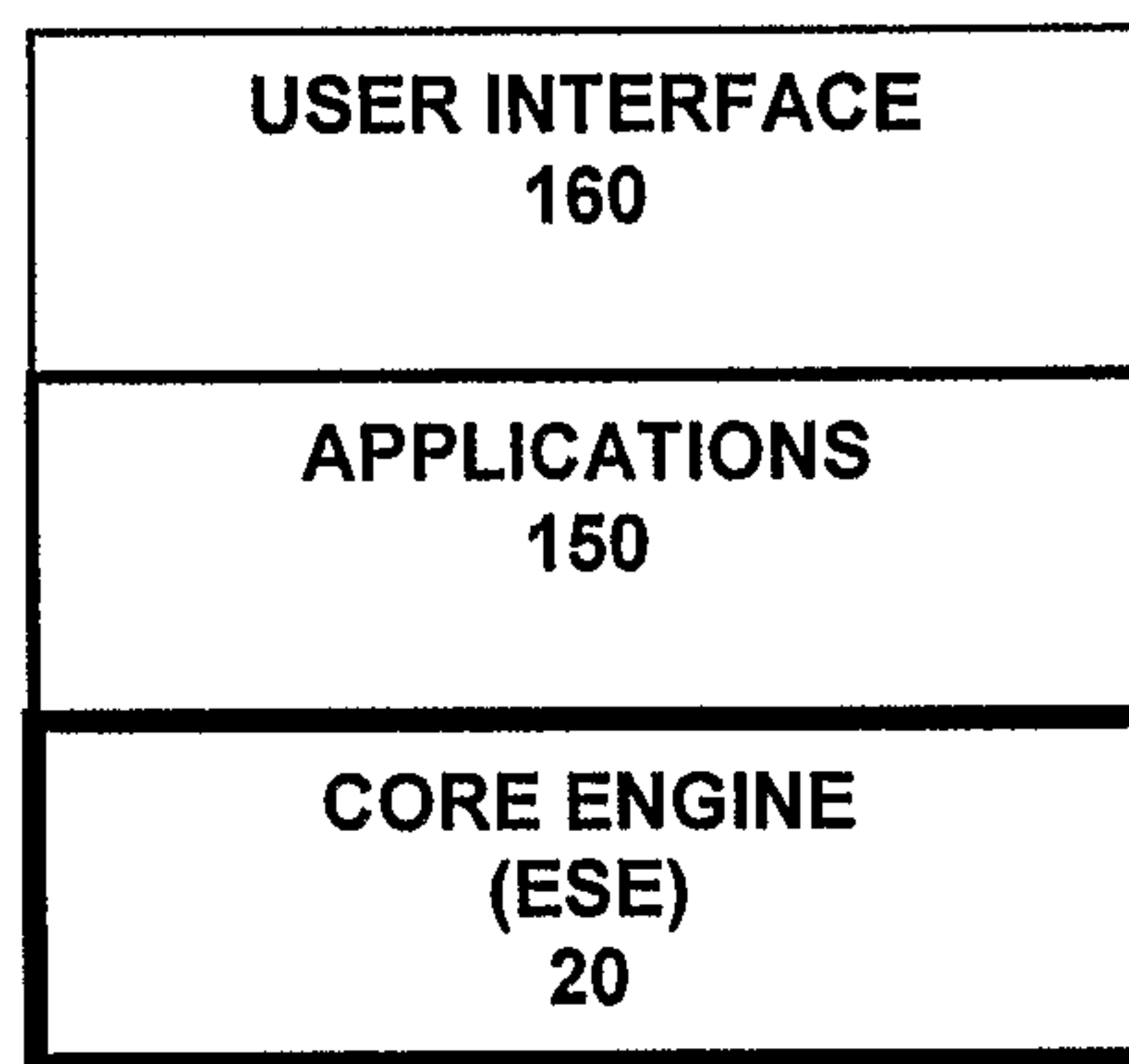


FIG. 4C

**FIG. 5A**

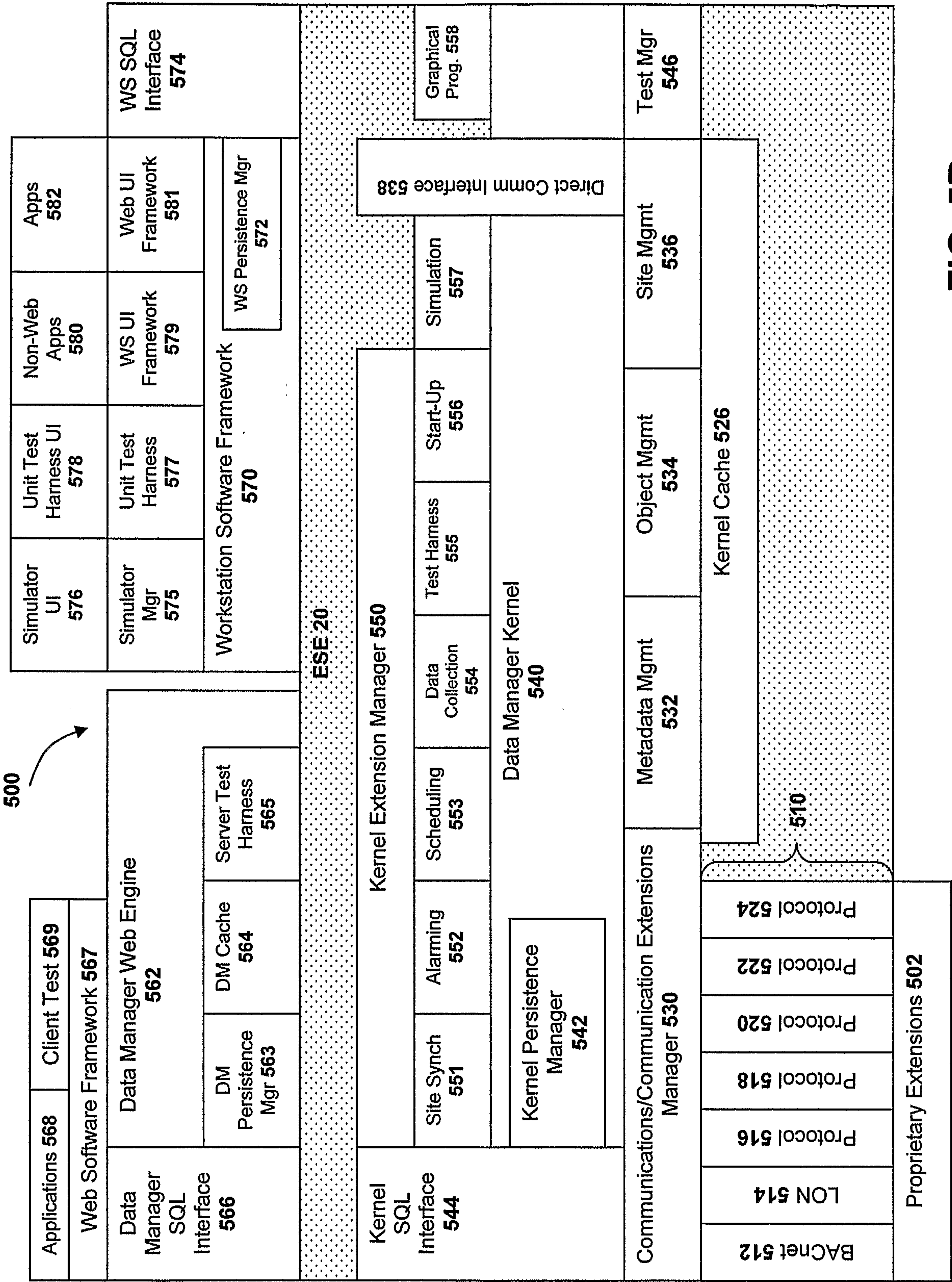
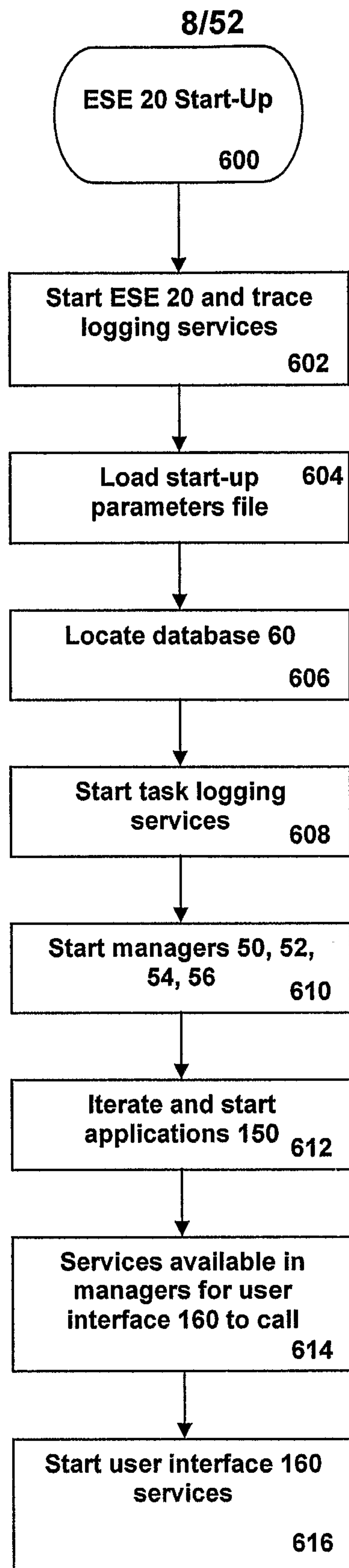
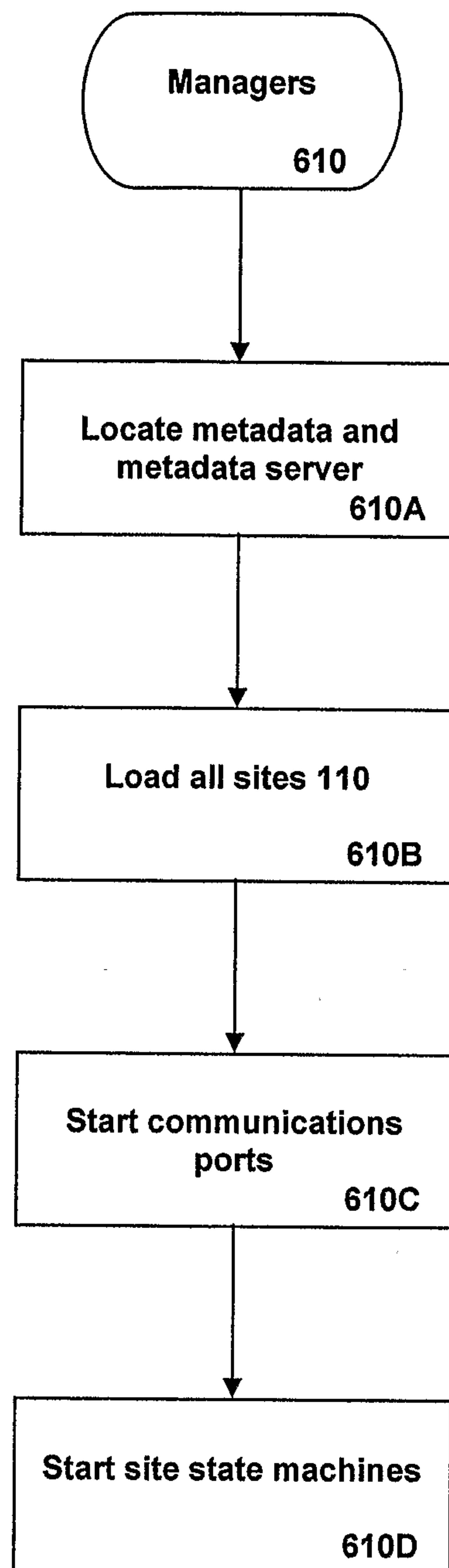


FIG. 5B

**FIG. 6A**

**FIG. 6B**

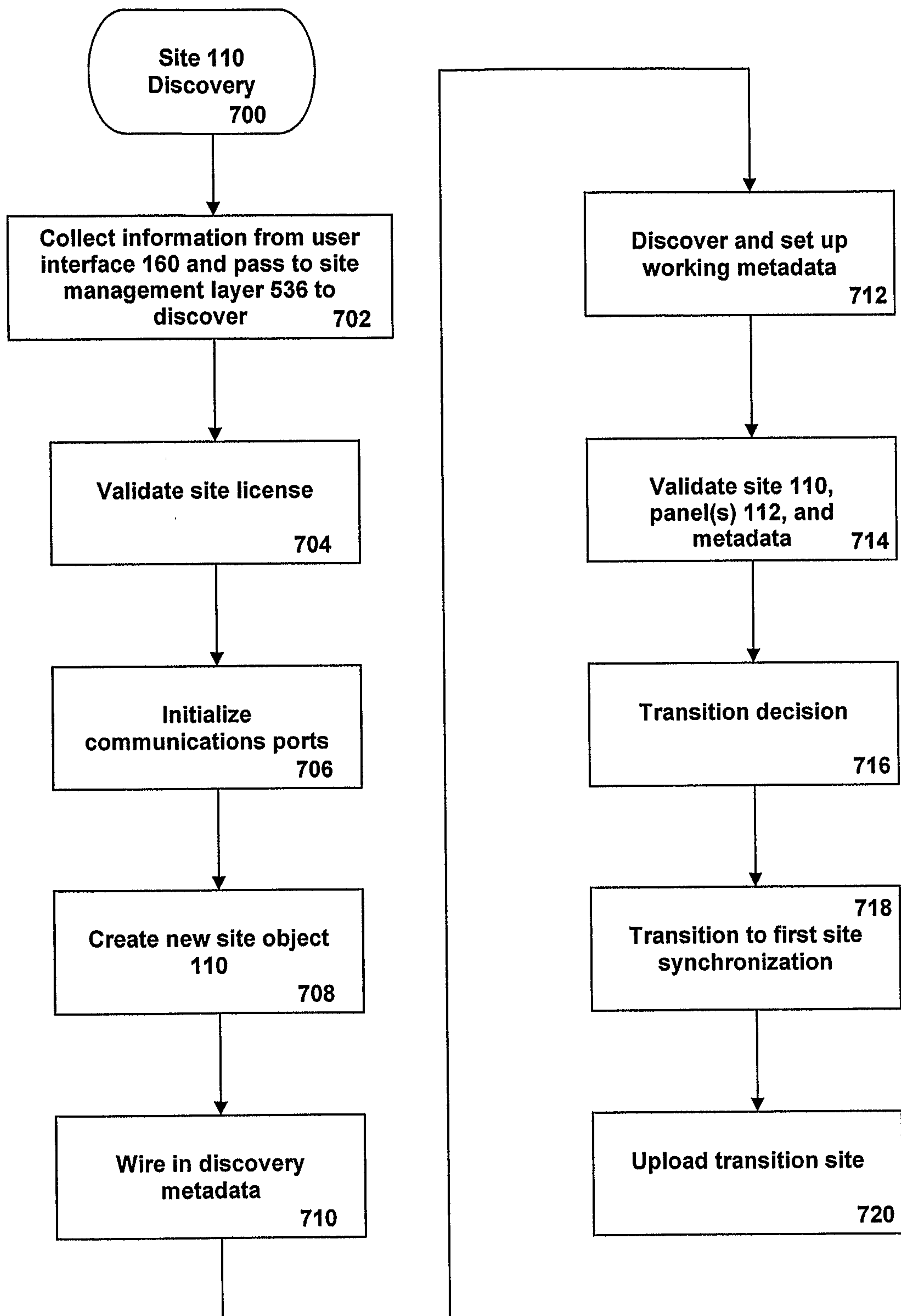


FIG. 7

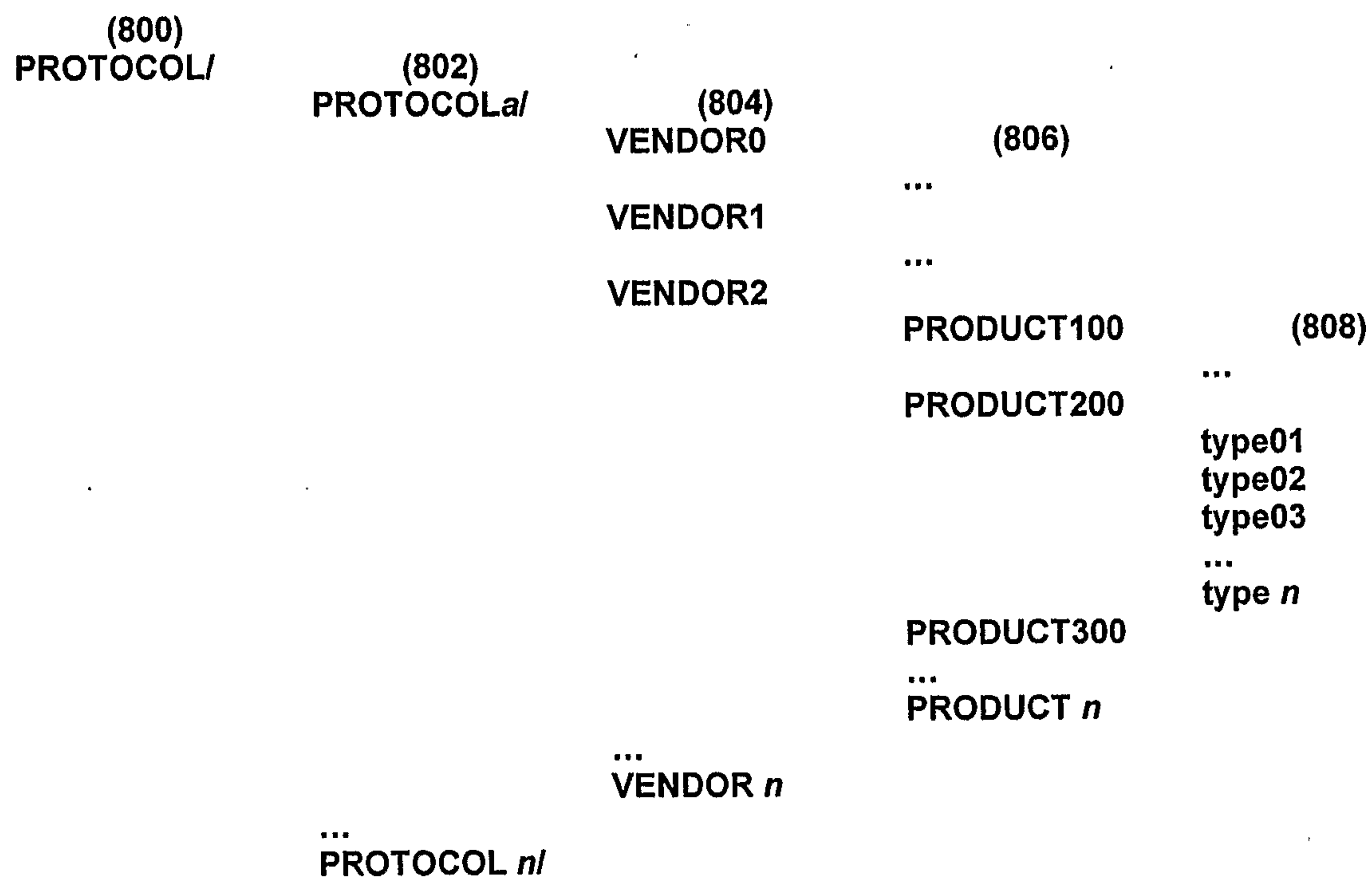


FIG. 8

12/52

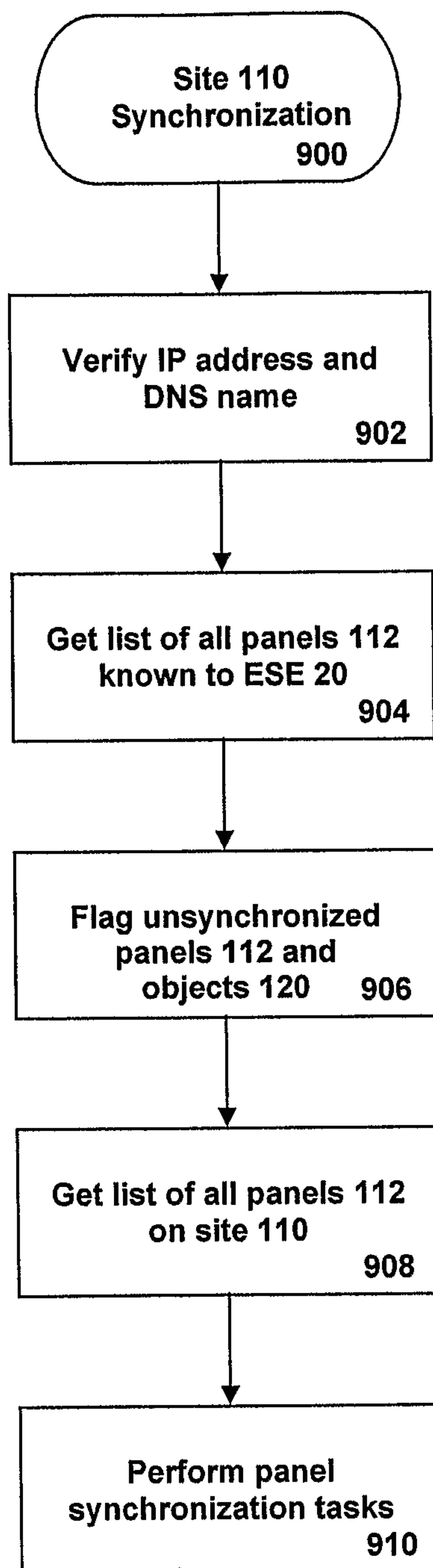


FIG. 9

13/52

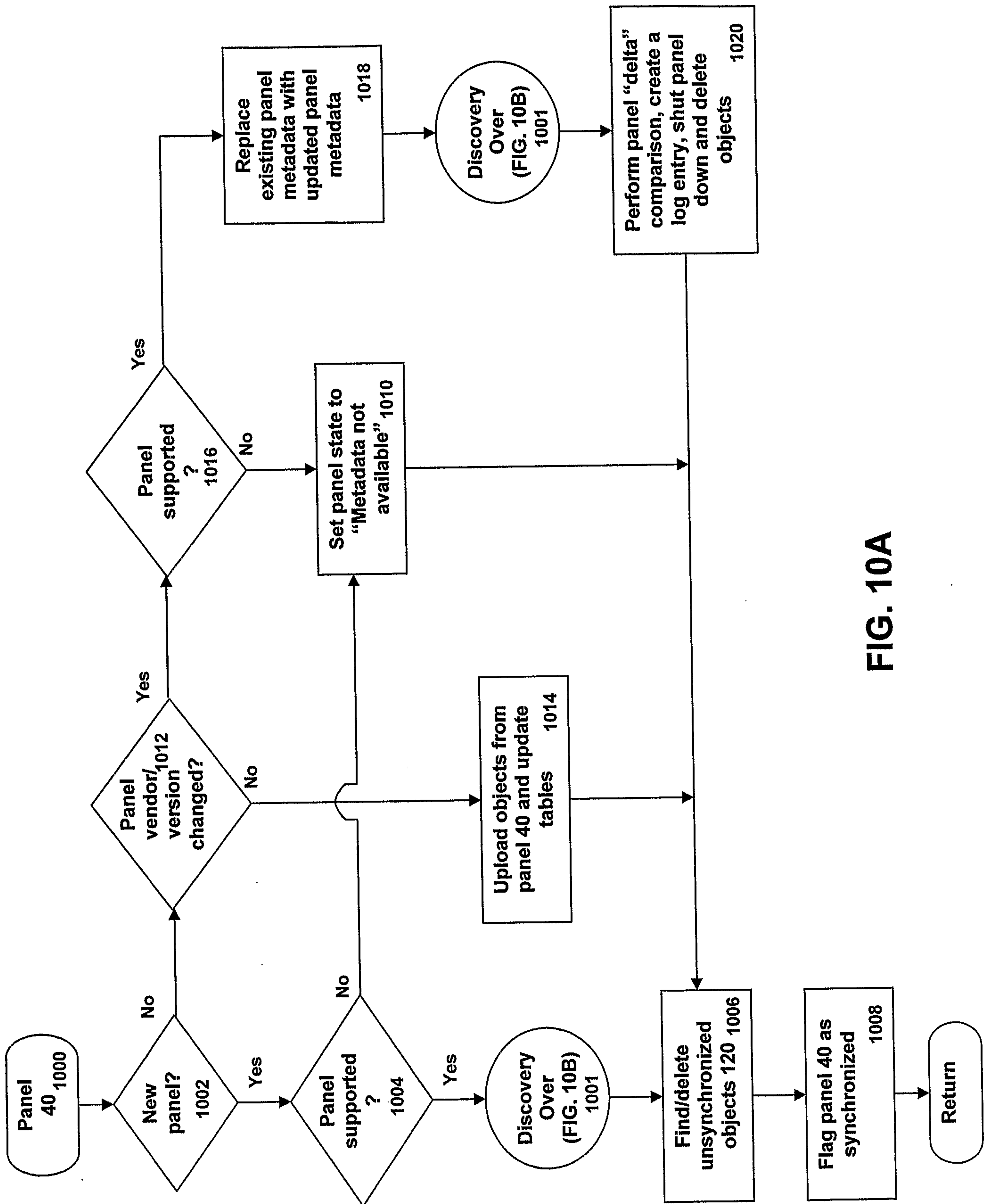


FIG. 10A

14/52

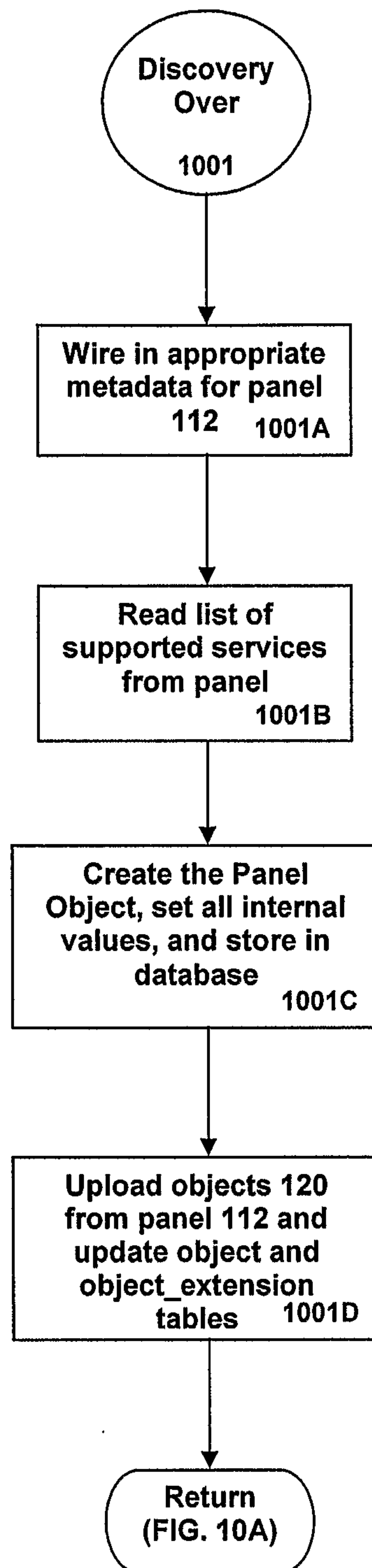


FIG. 10B

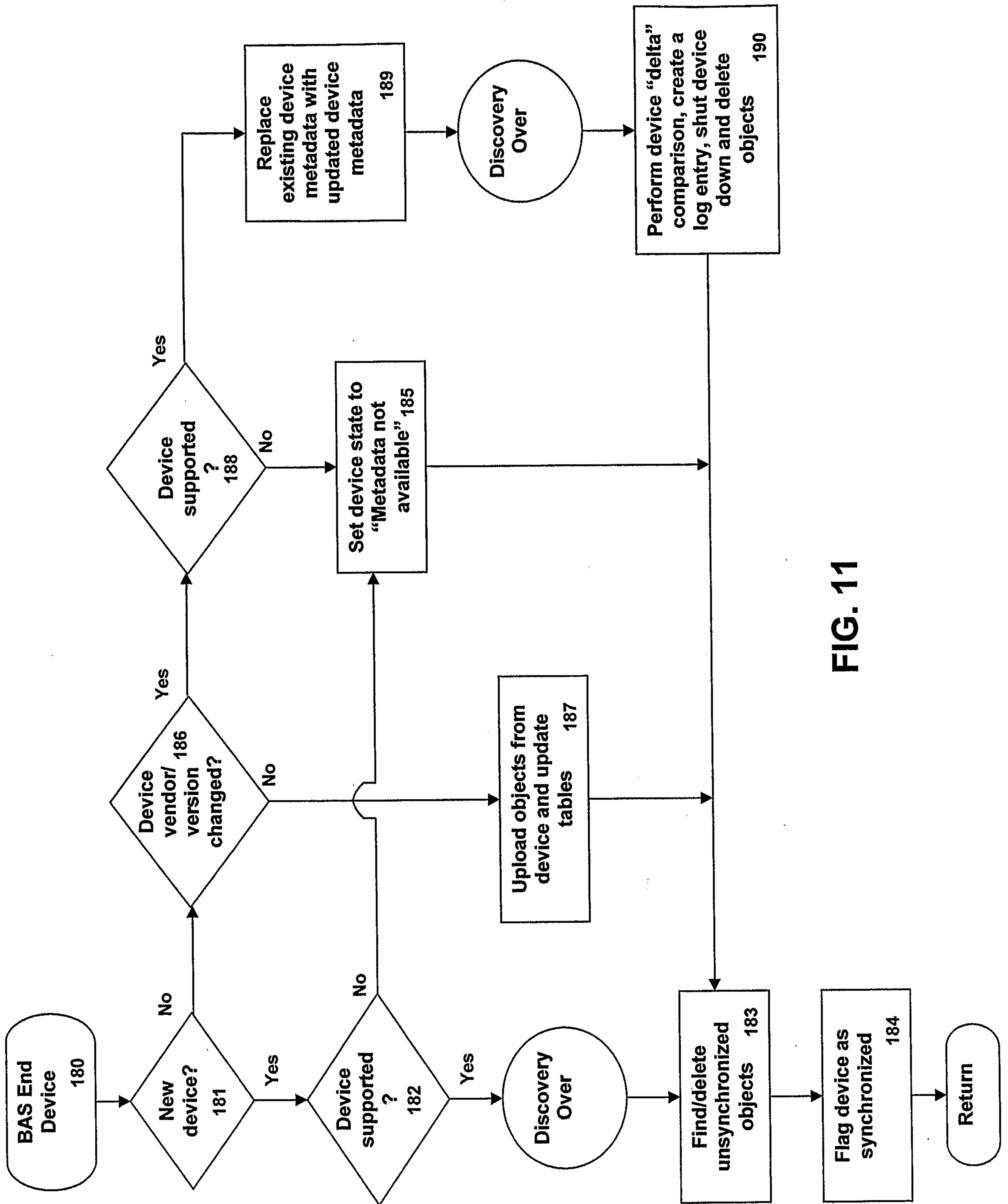


FIG. 11

16/52

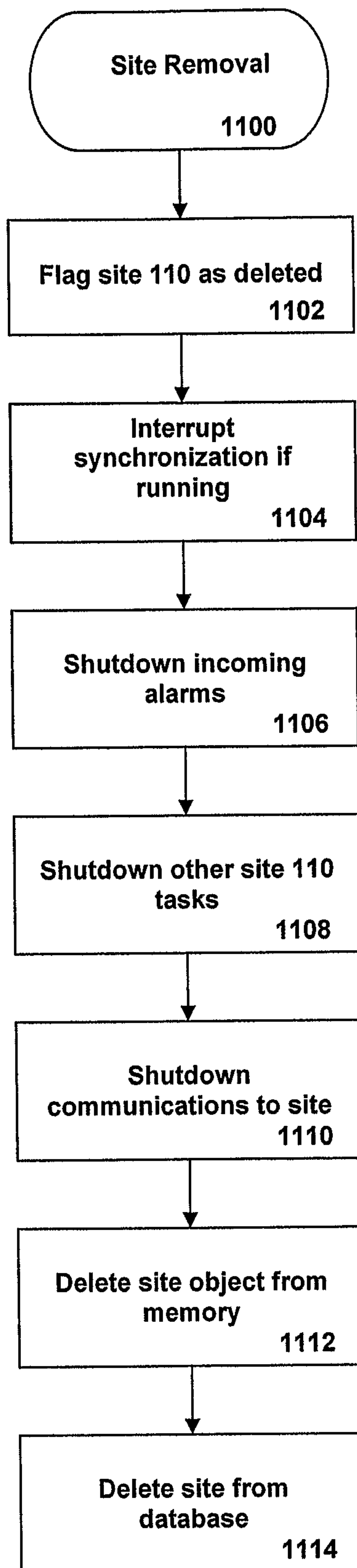
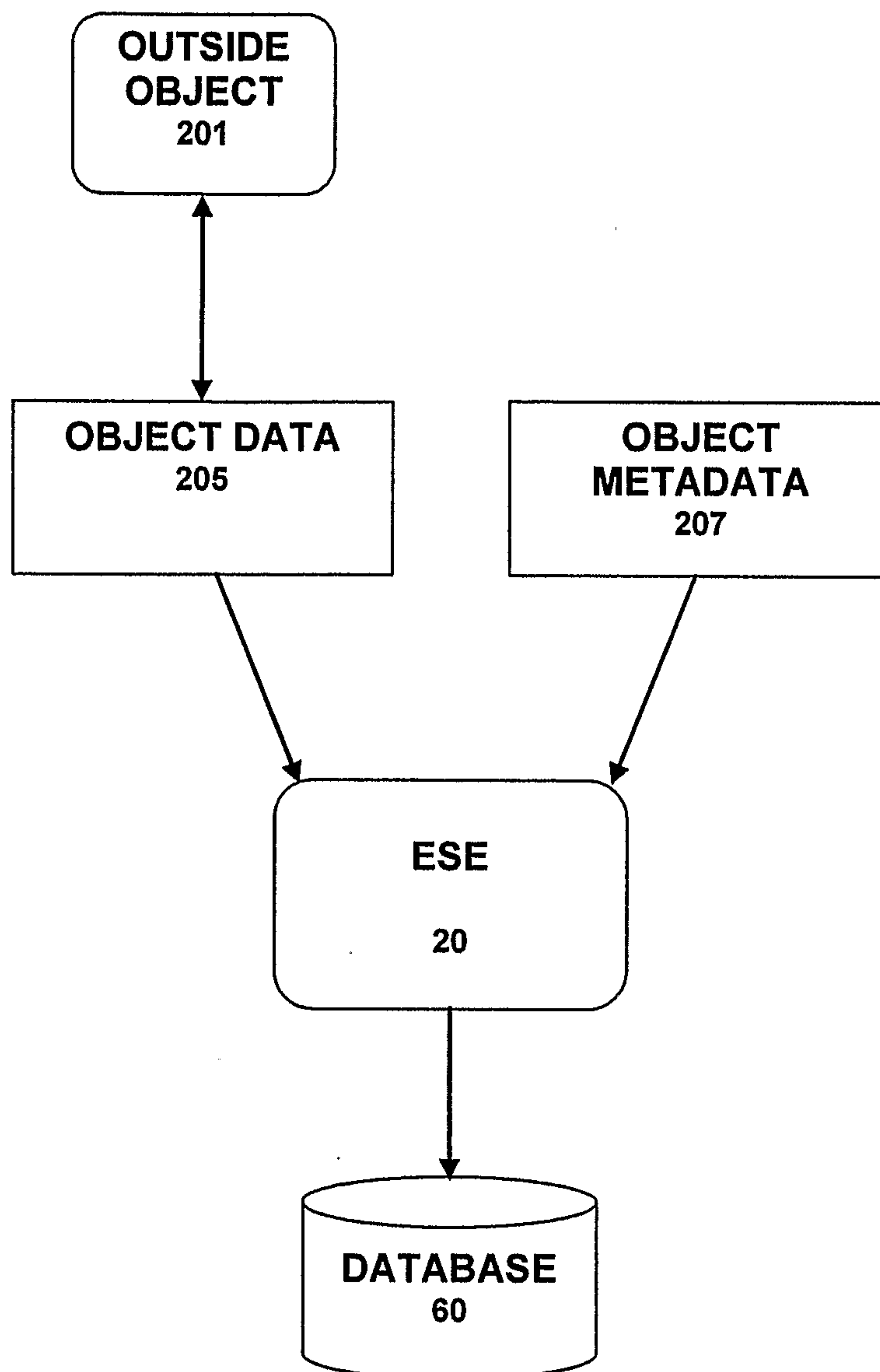
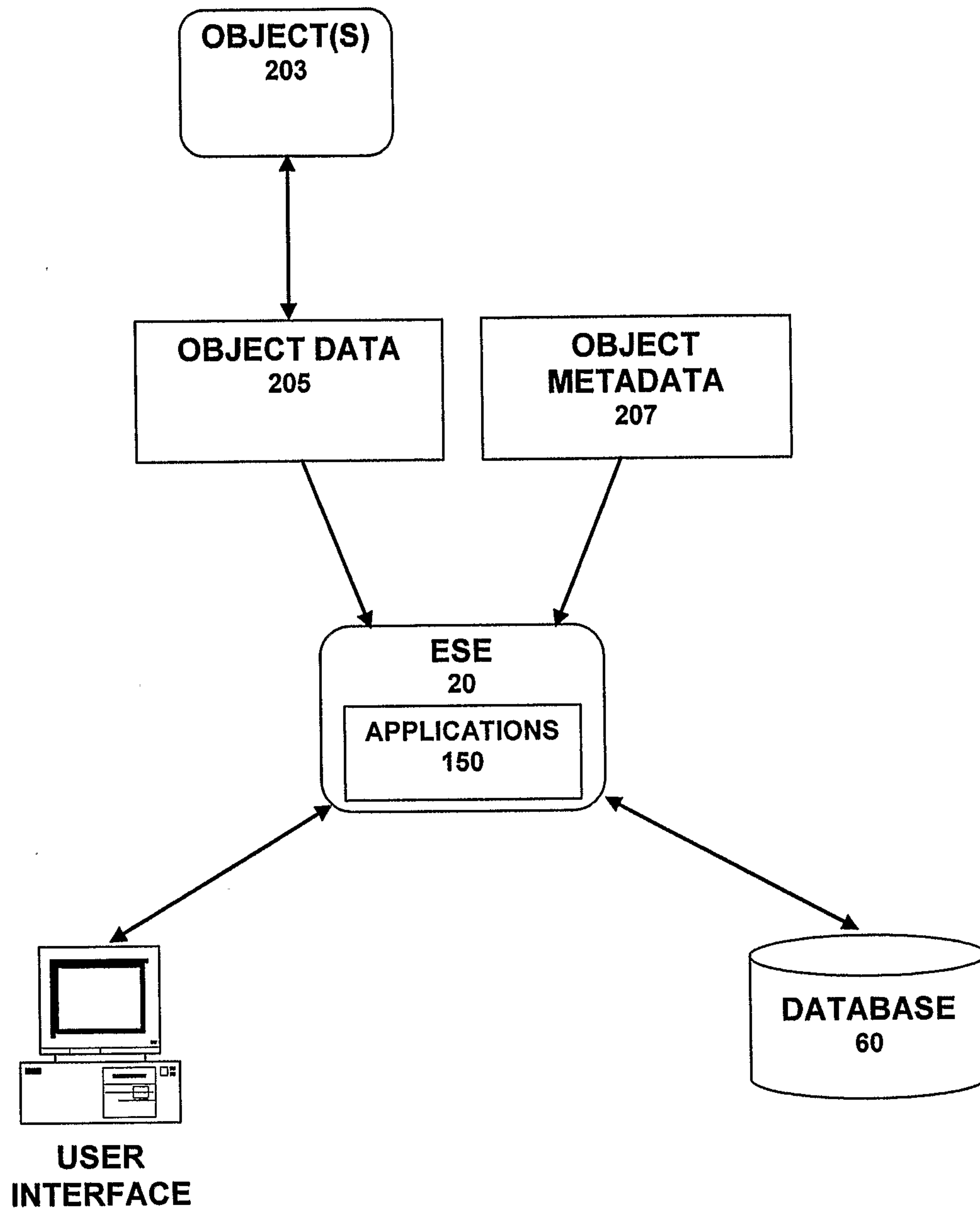


FIG. 12

**FIG. 13**

**FIG. 14**

19/52

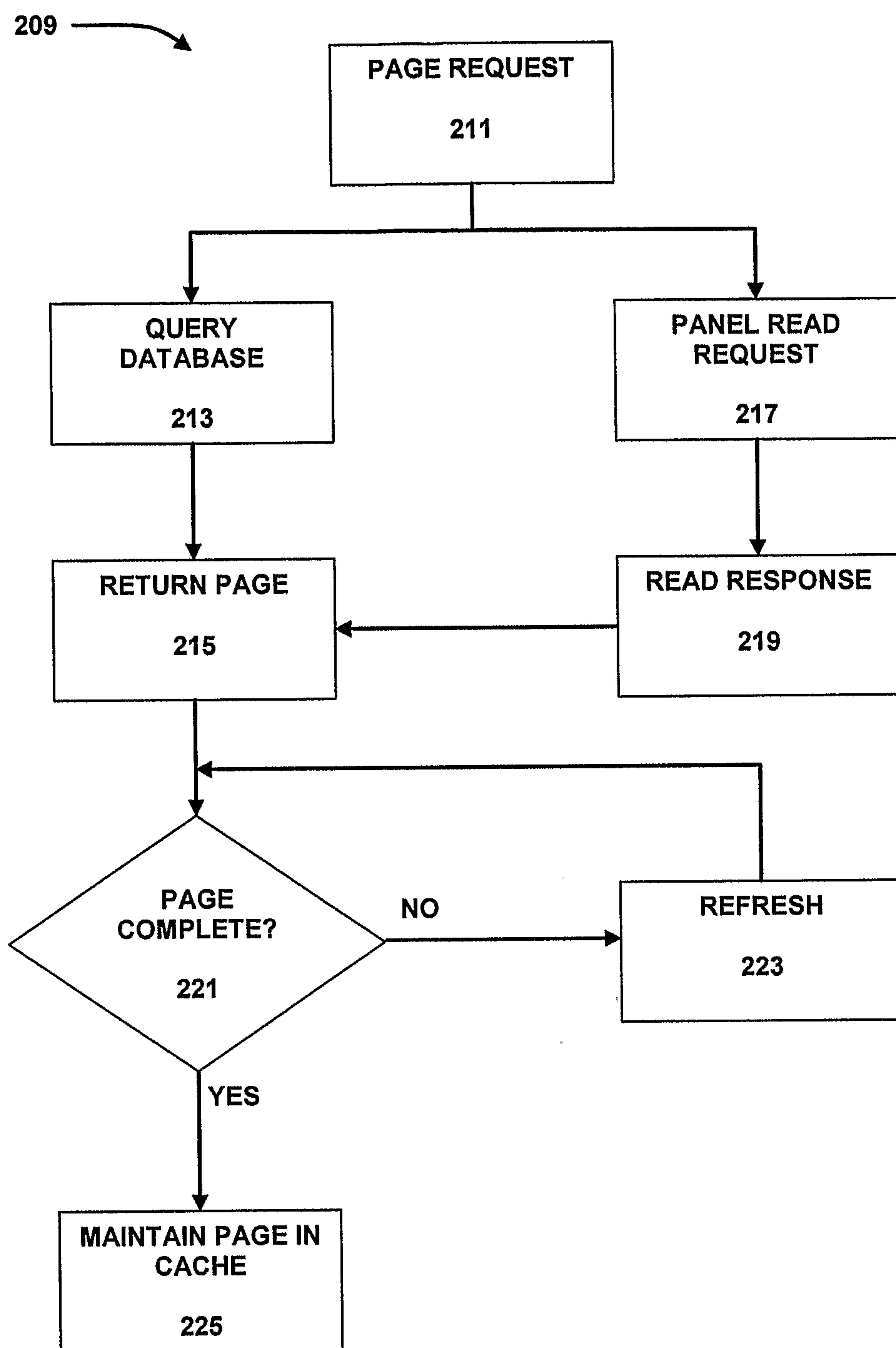


FIG.15

20/52

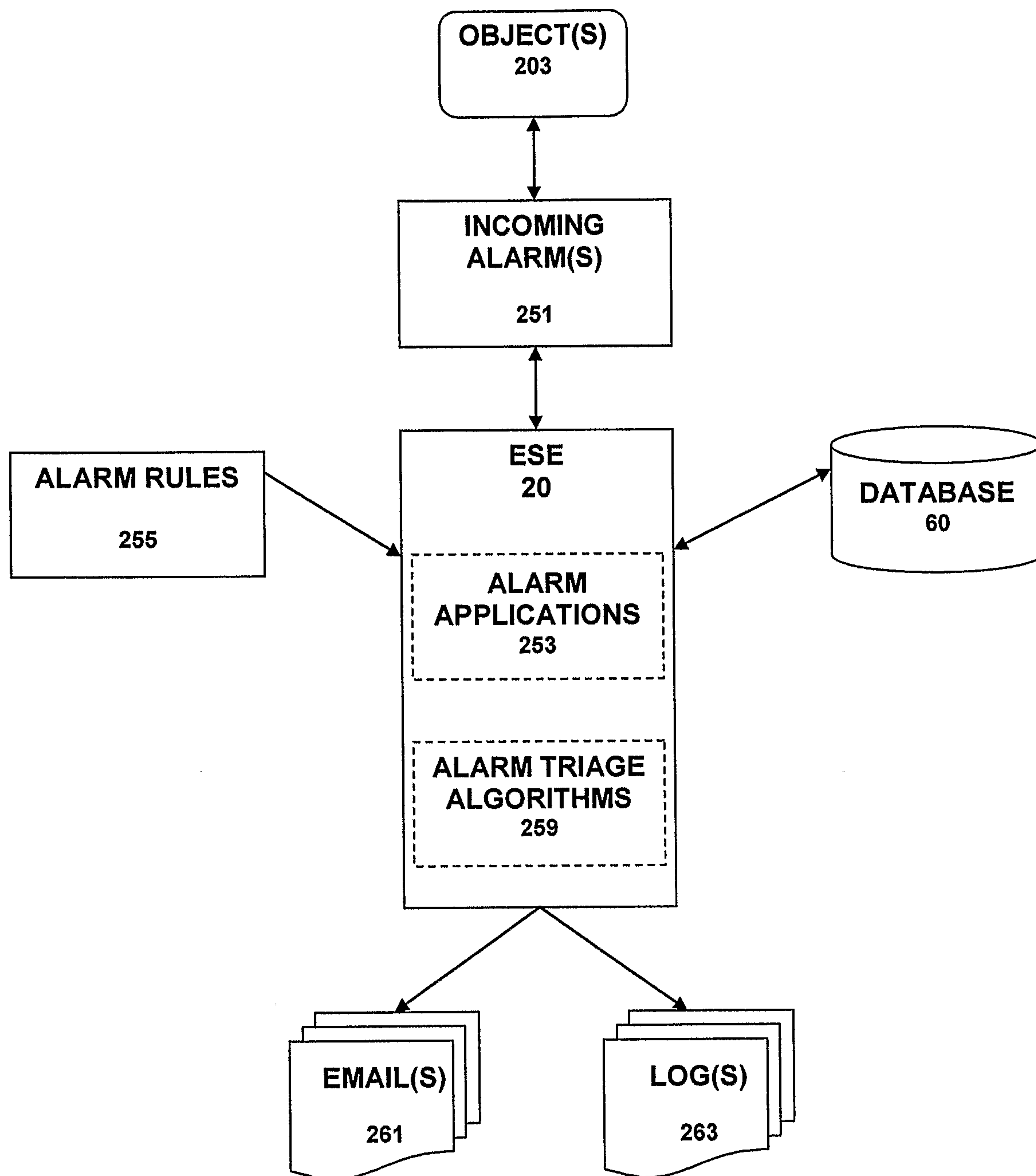


FIG. 16

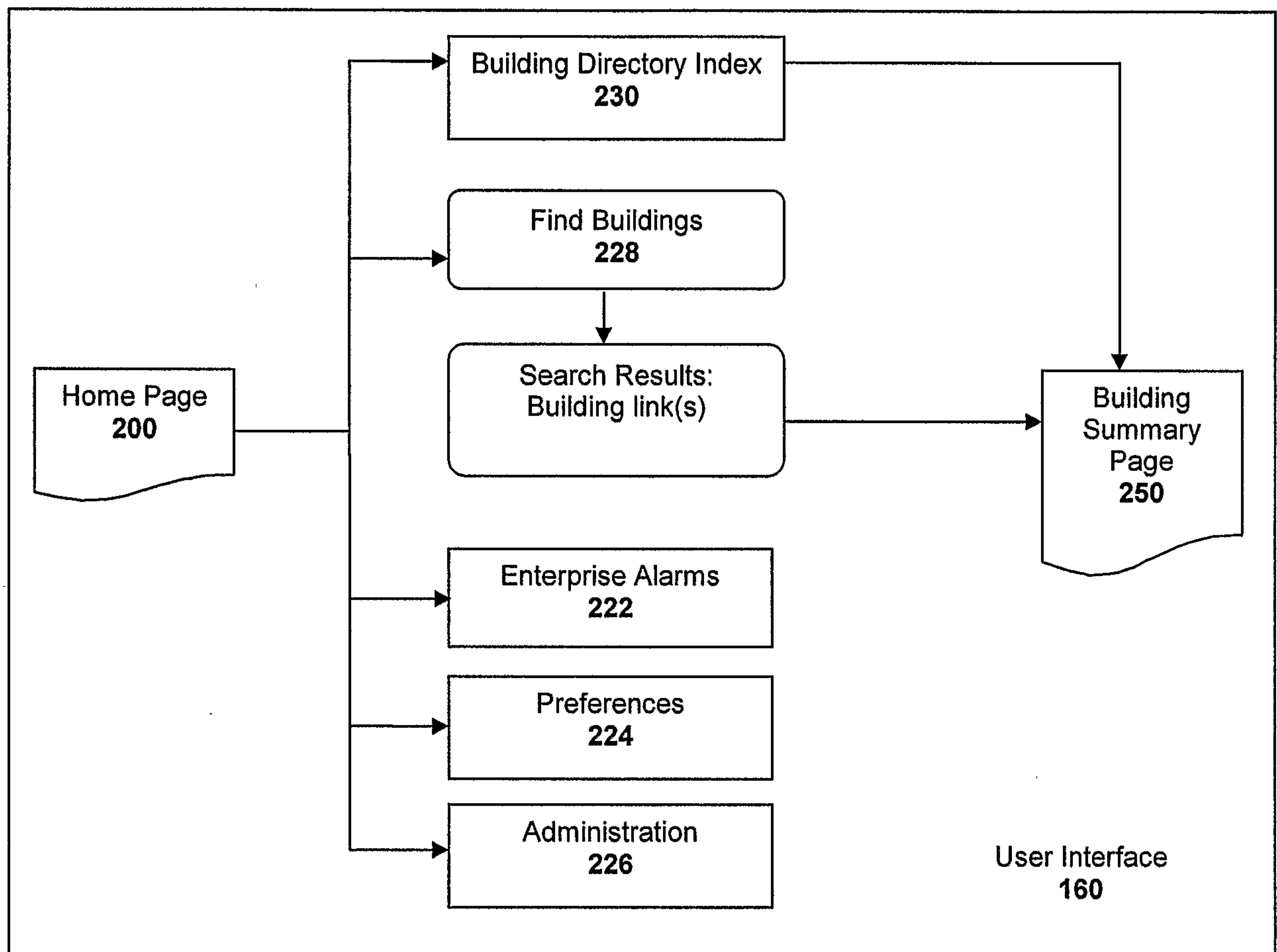


FIG. 17

202

200

228

230

238

242

244

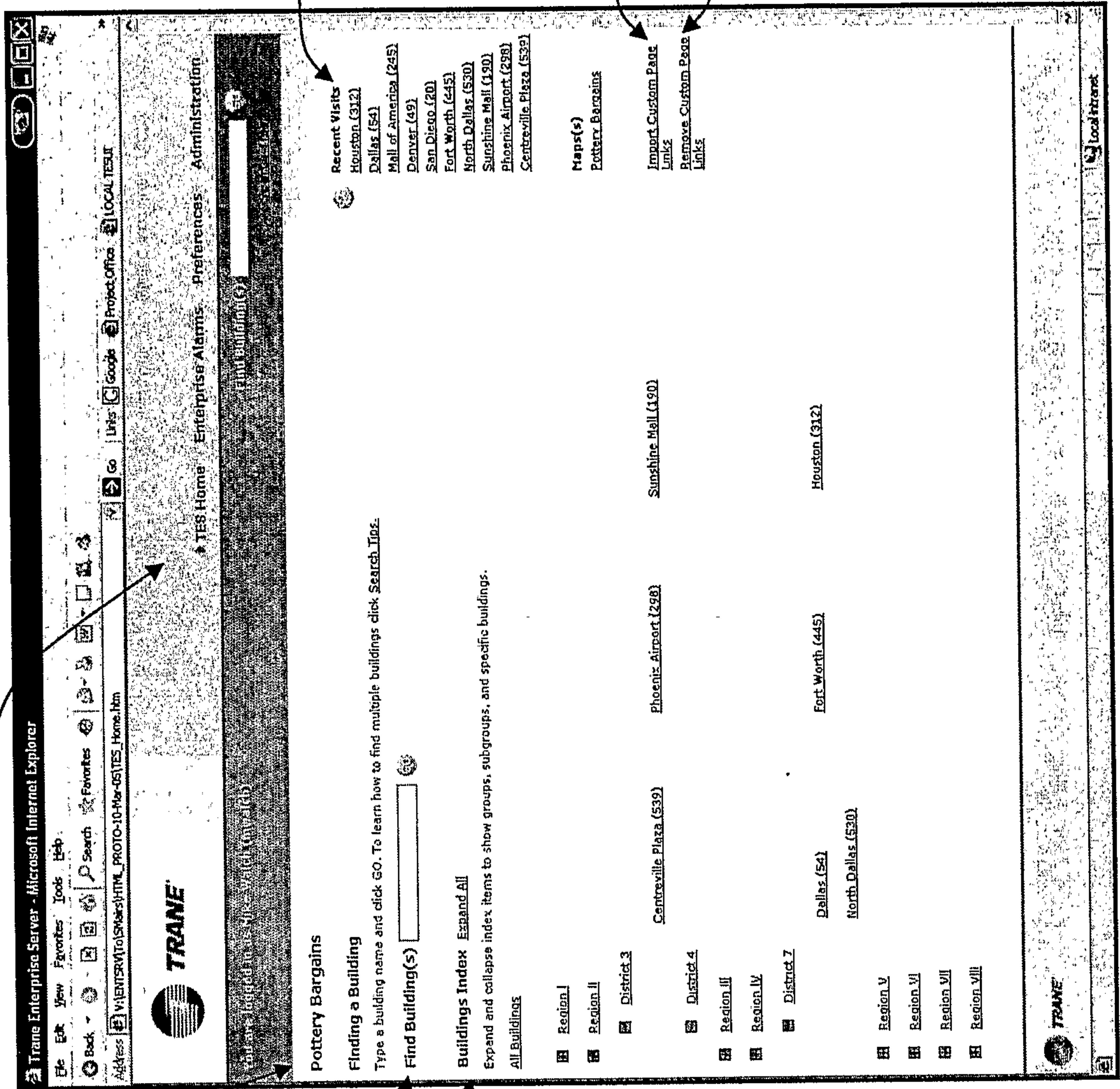


FIG. 18A

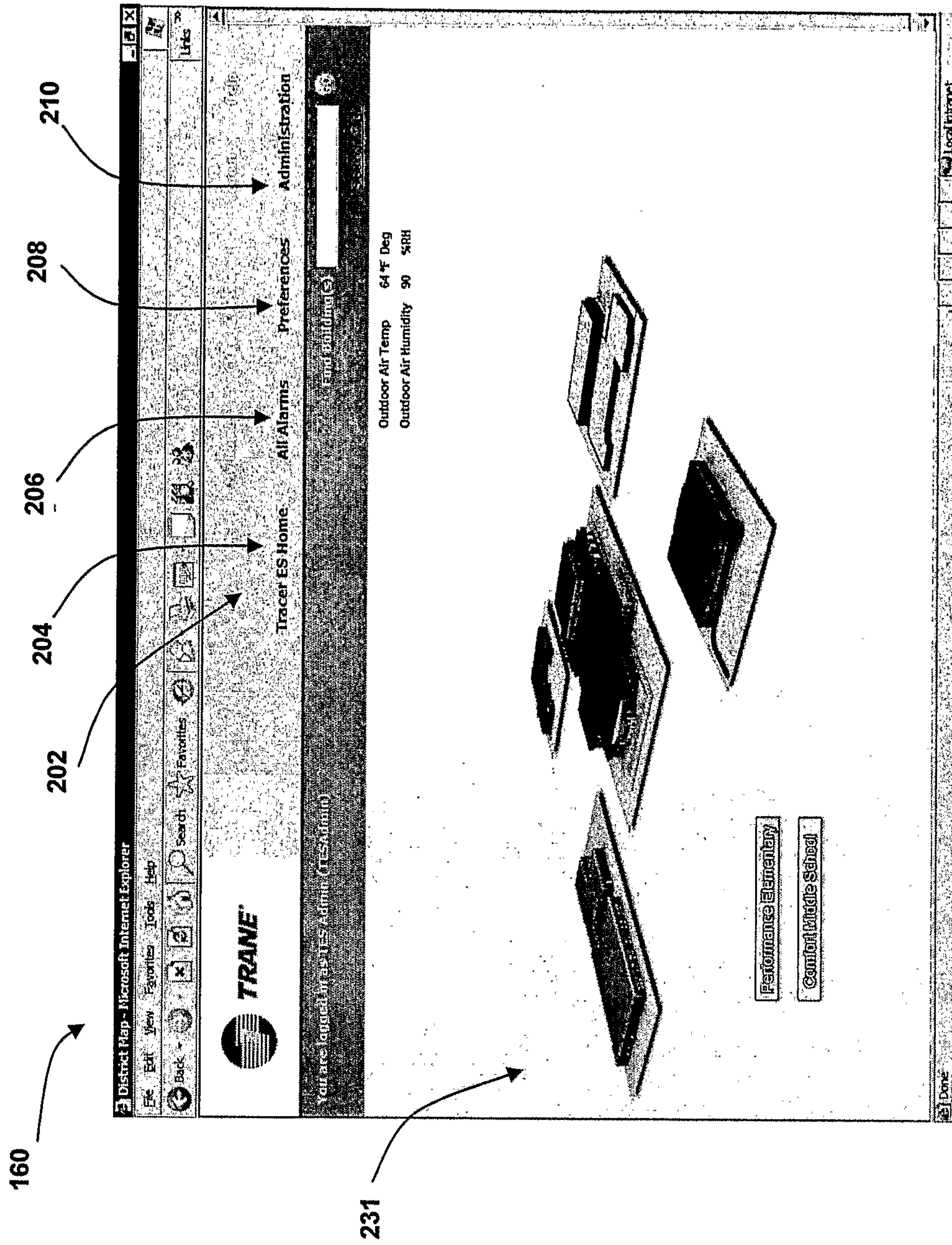


FIG. 18B

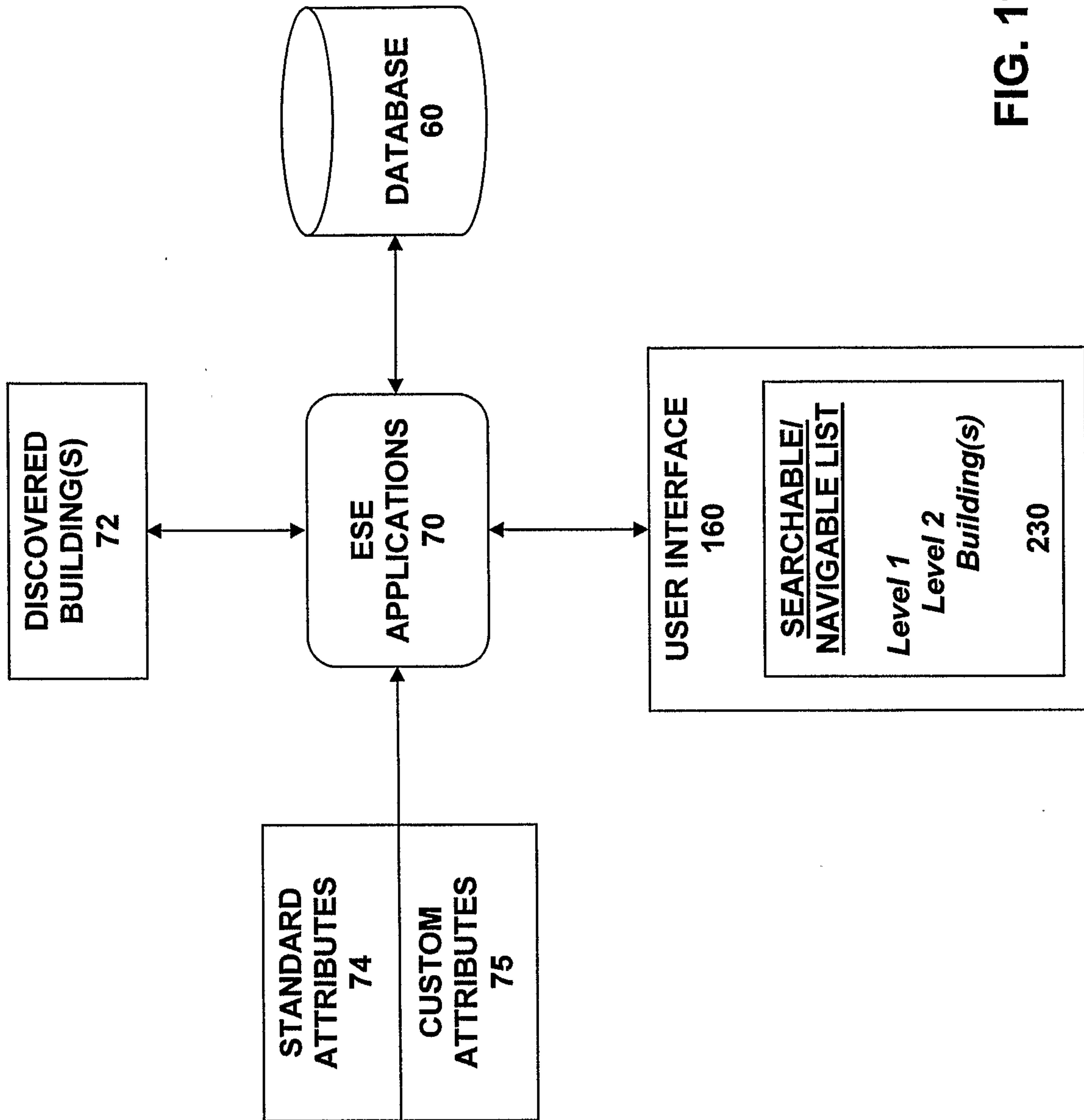


FIG. 19

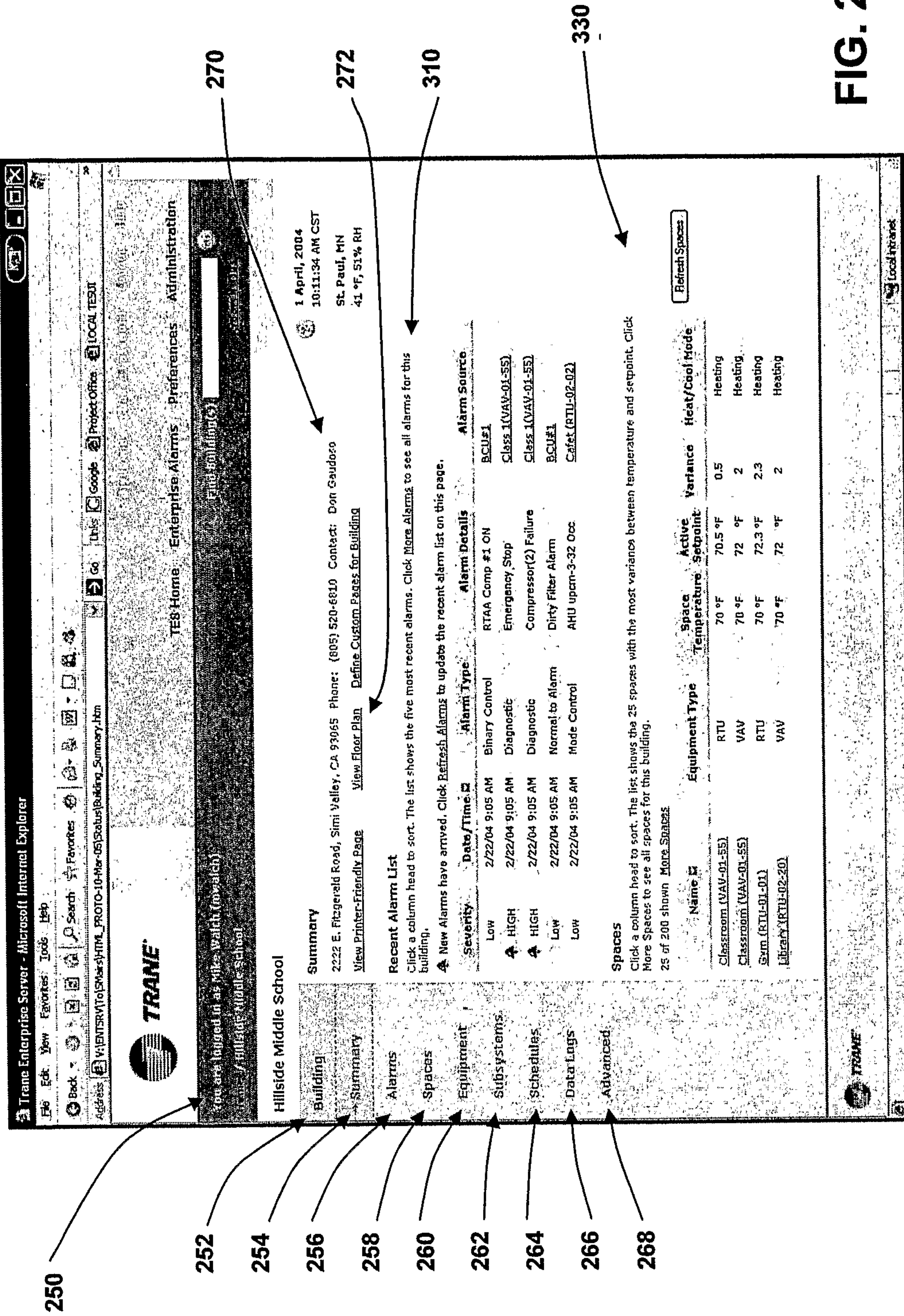


FIG. 20A

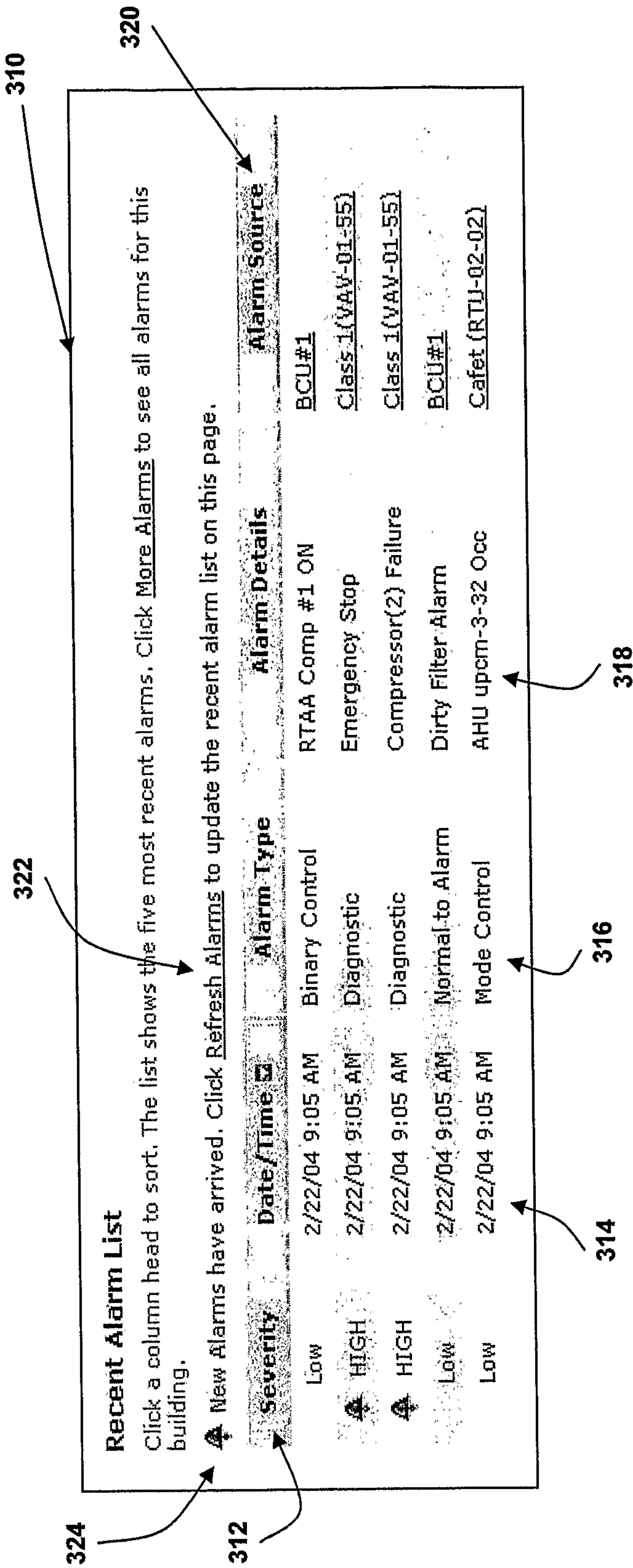


FIG. 20B

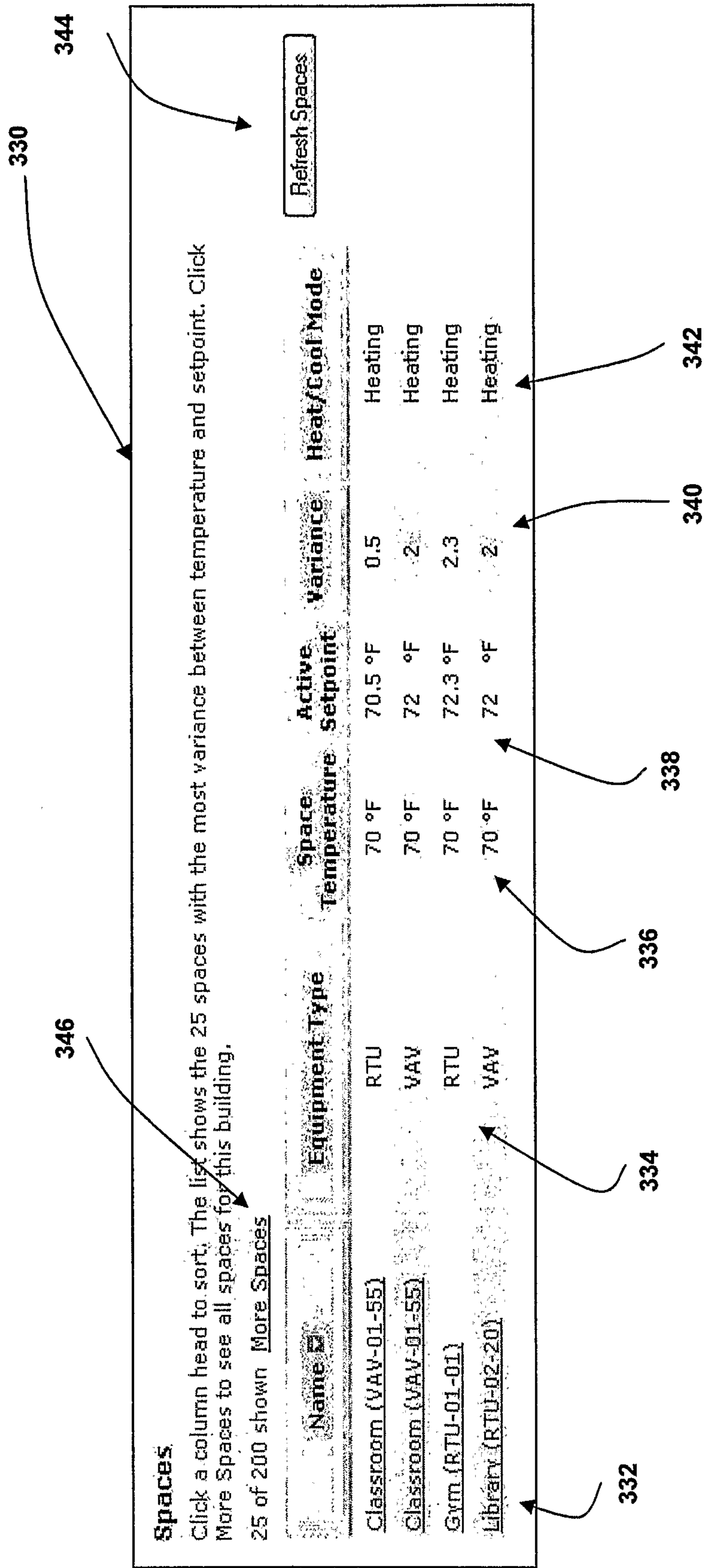
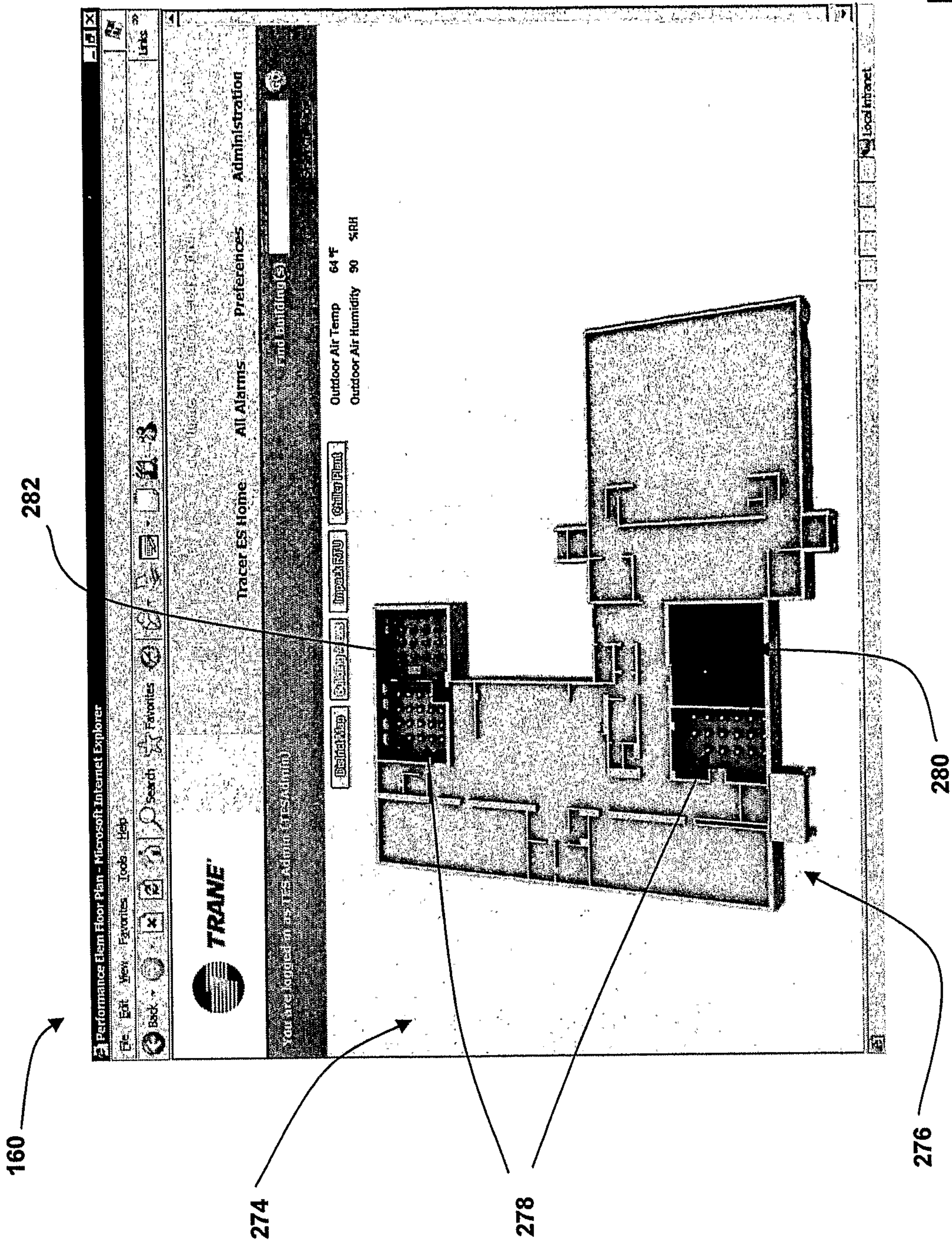


FIG. 20C



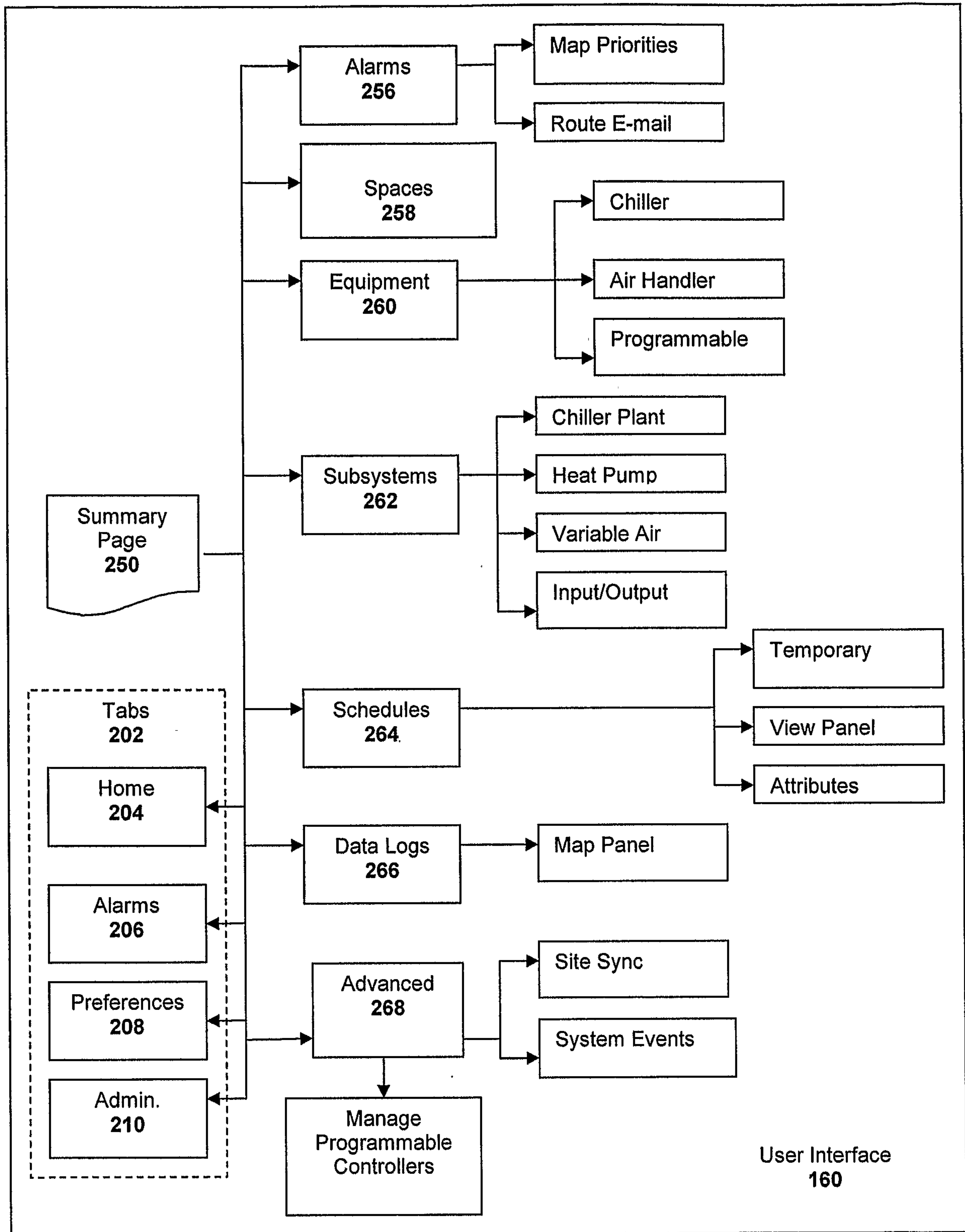


FIG. 21

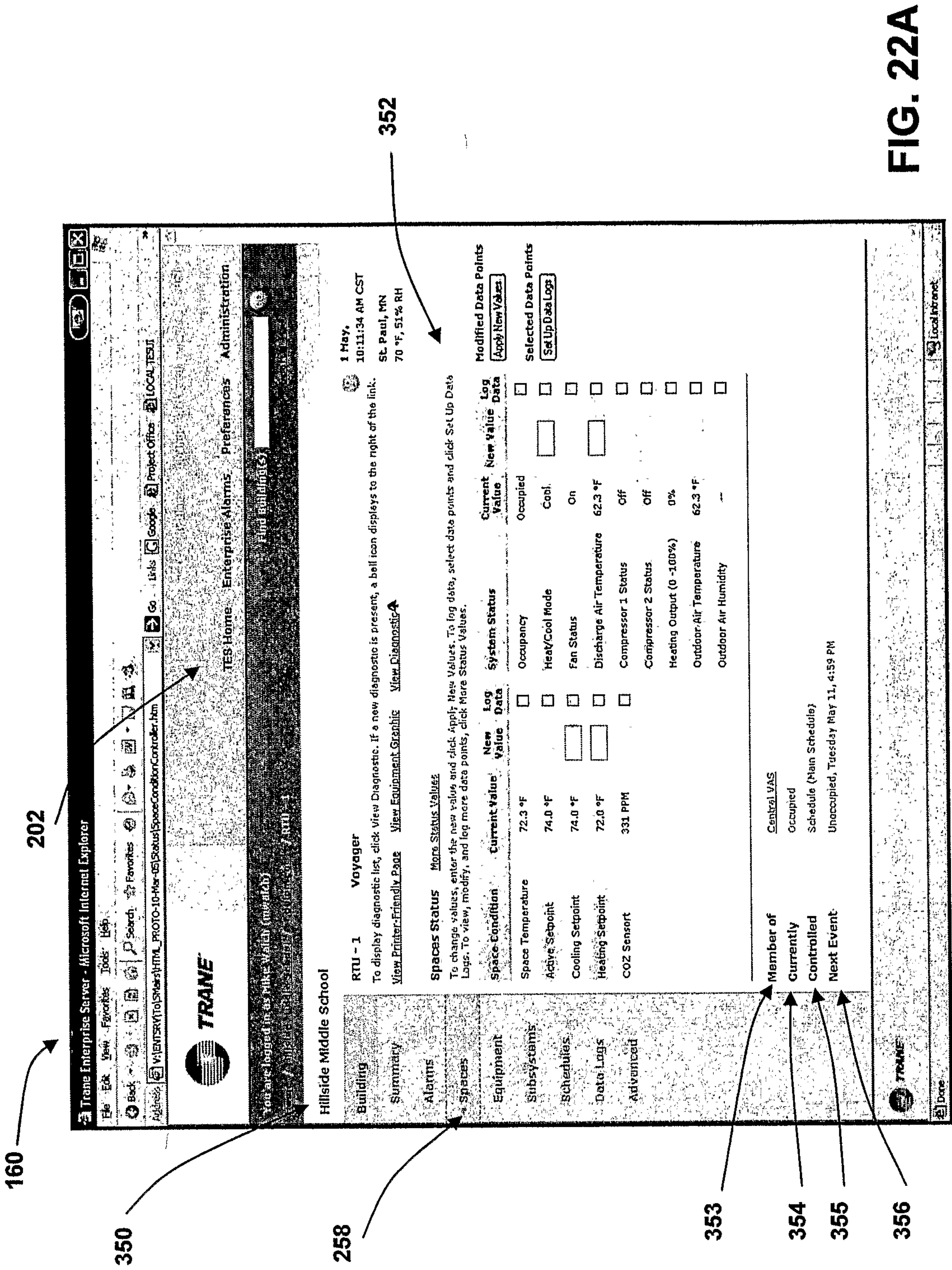


FIG. 22A

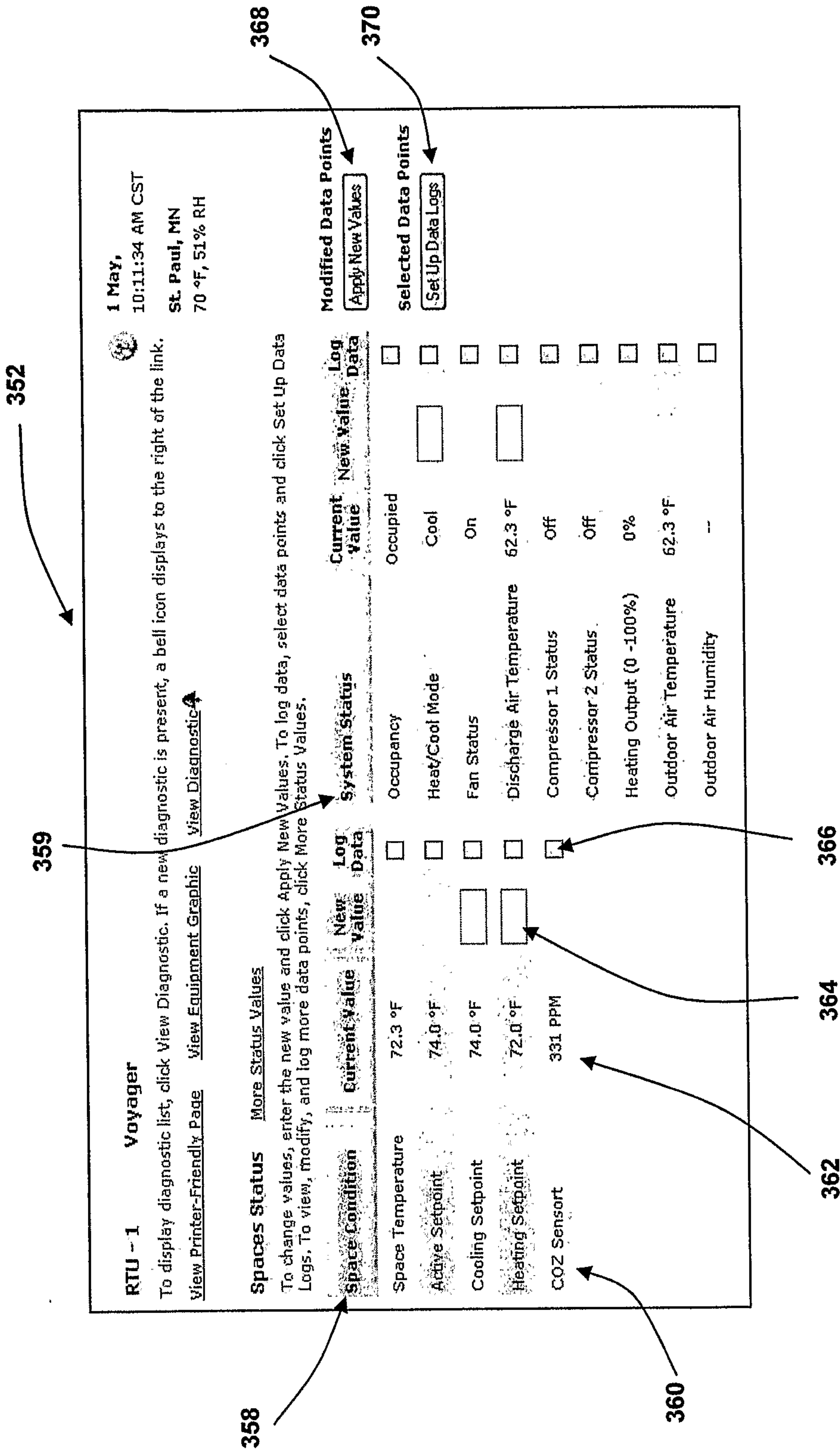


FIG. 22B

32/52

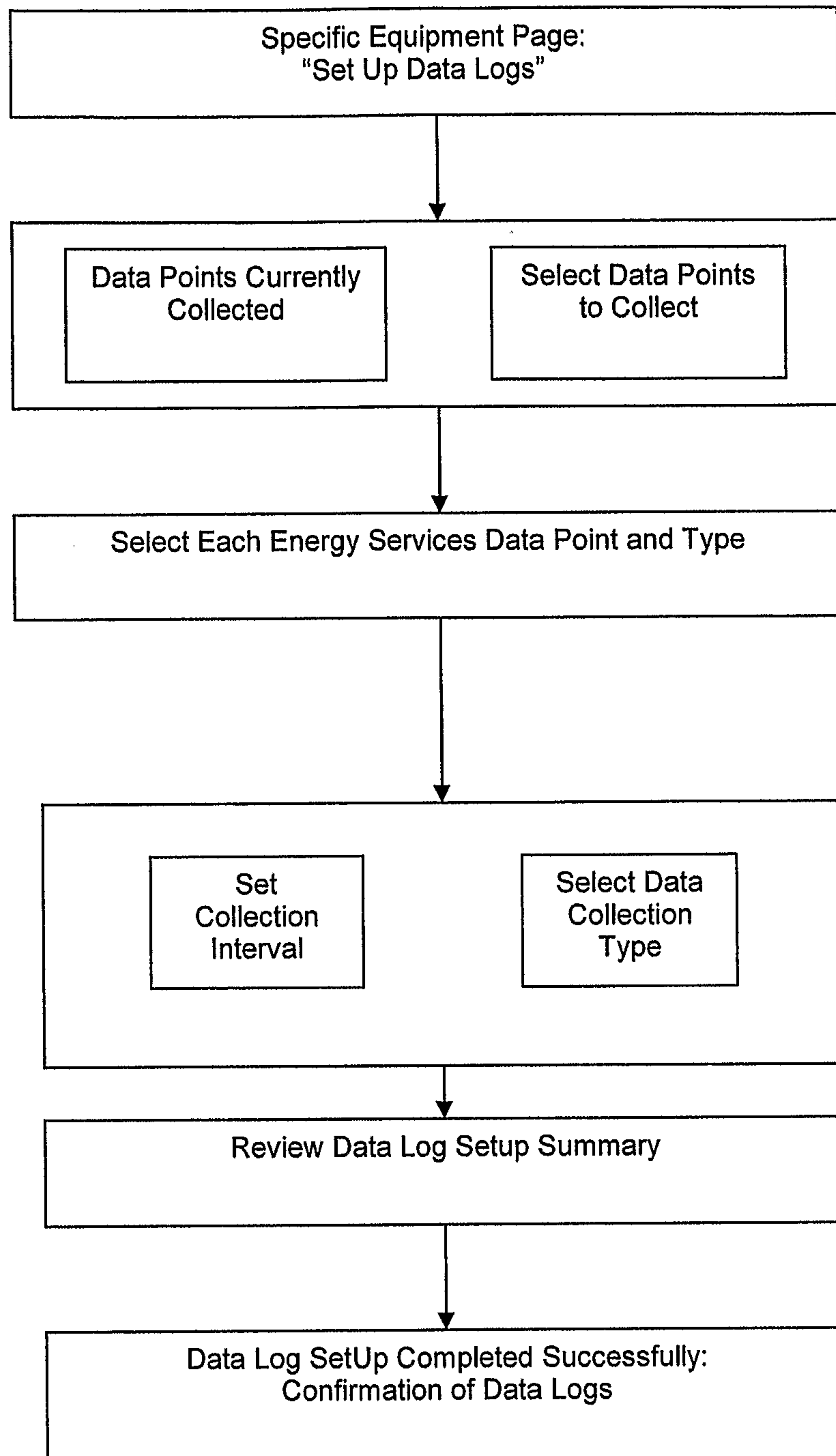
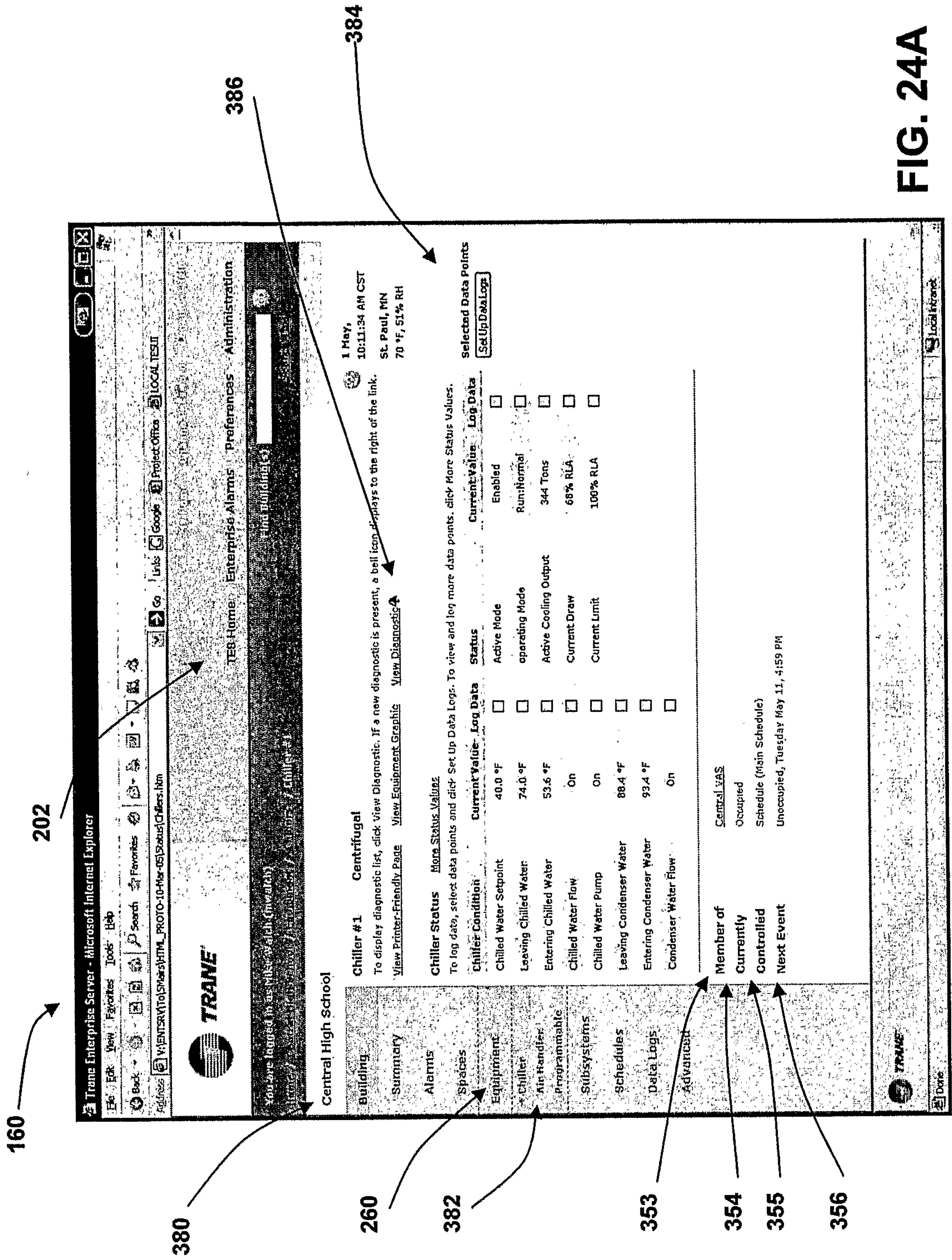


FIG. 23



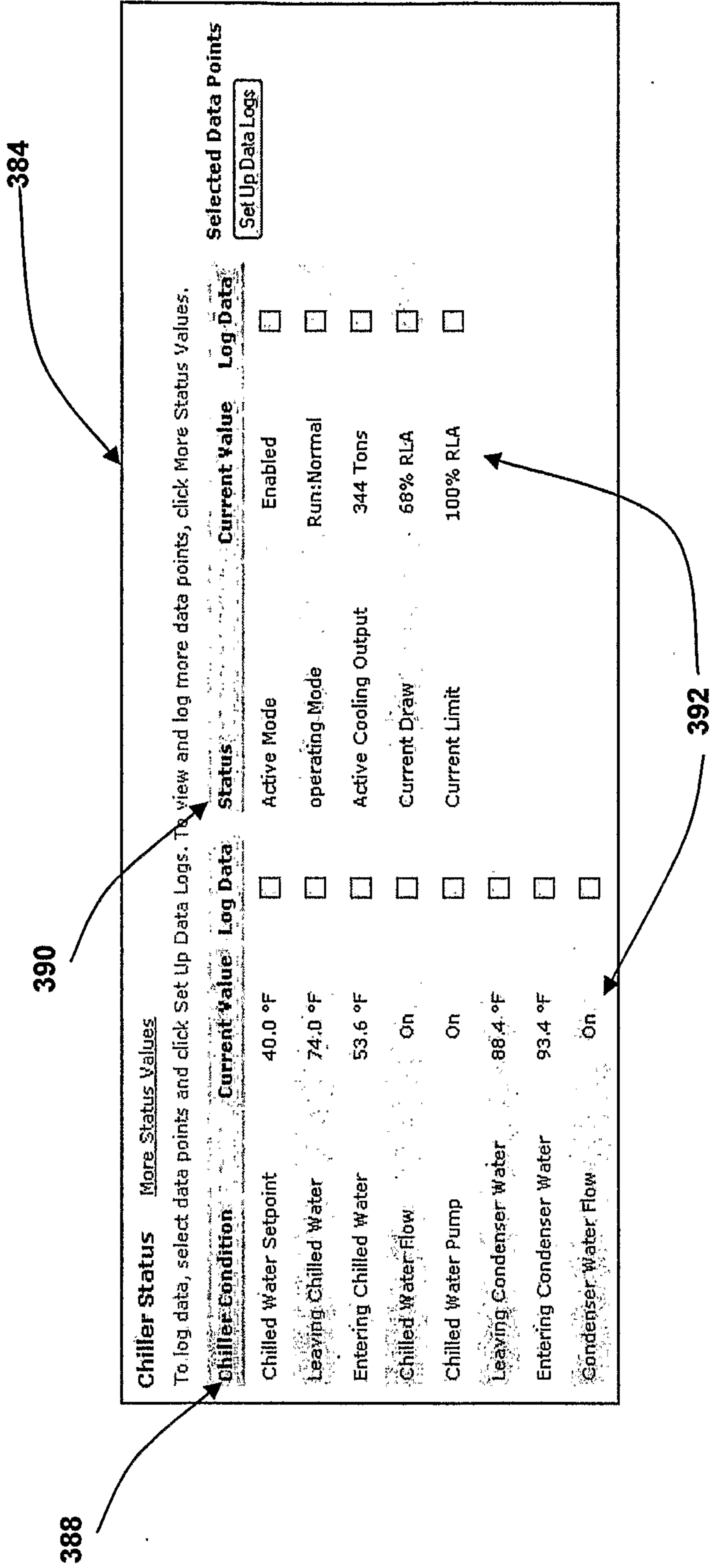
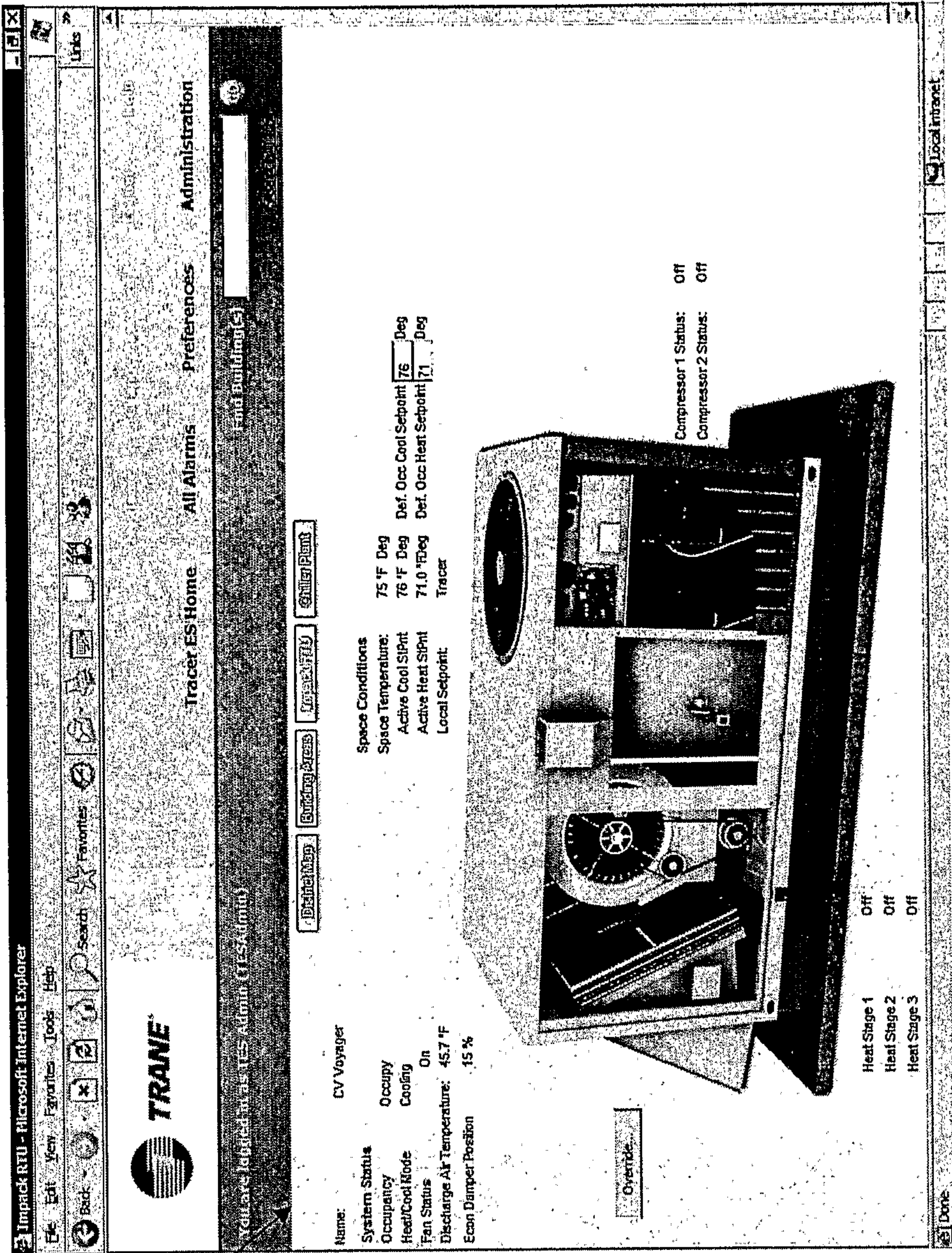


FIG. 24B

FIG. 24C

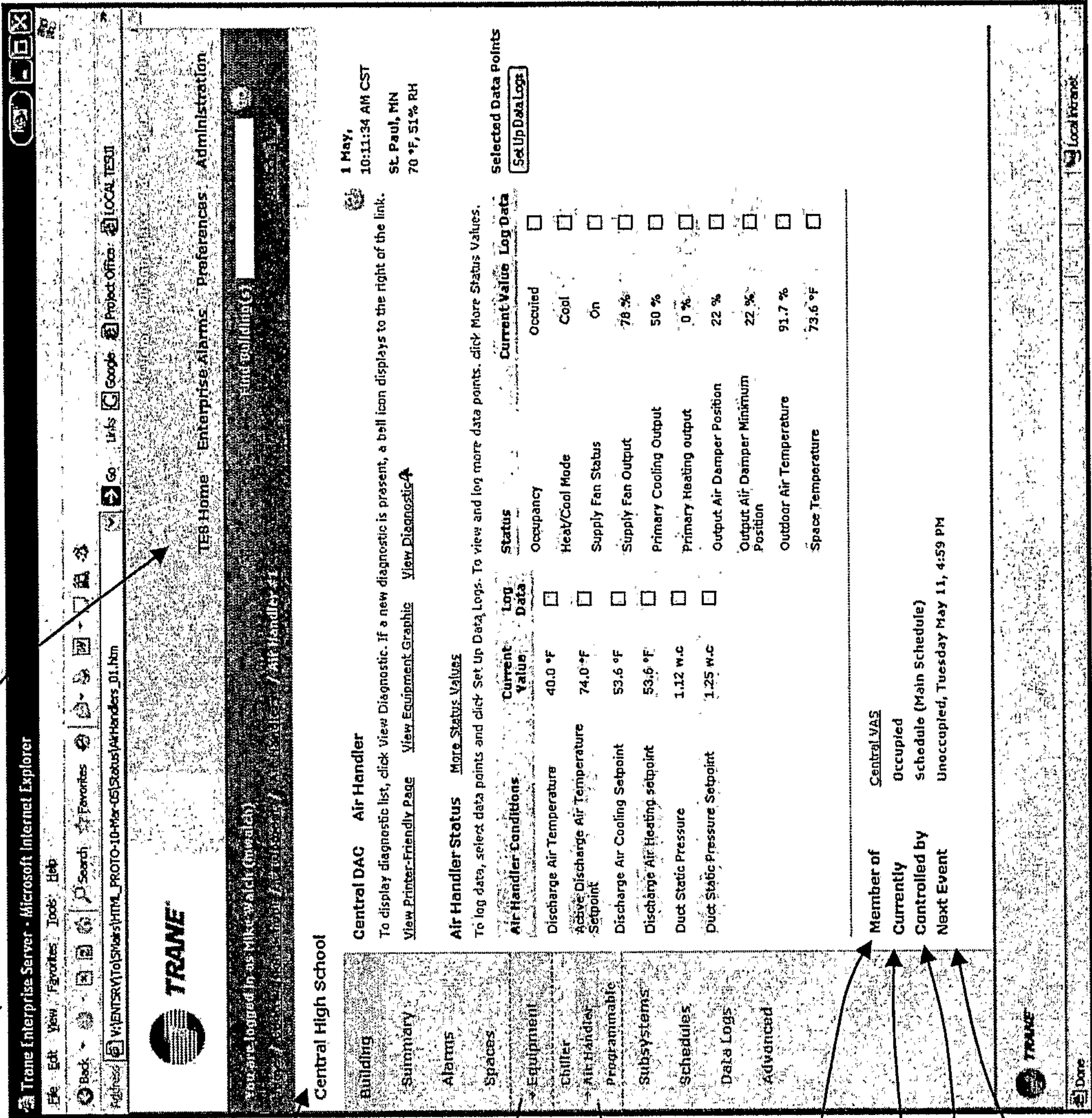


160

381

160

202



394

260

382

353

354

355

356

FIG. 25

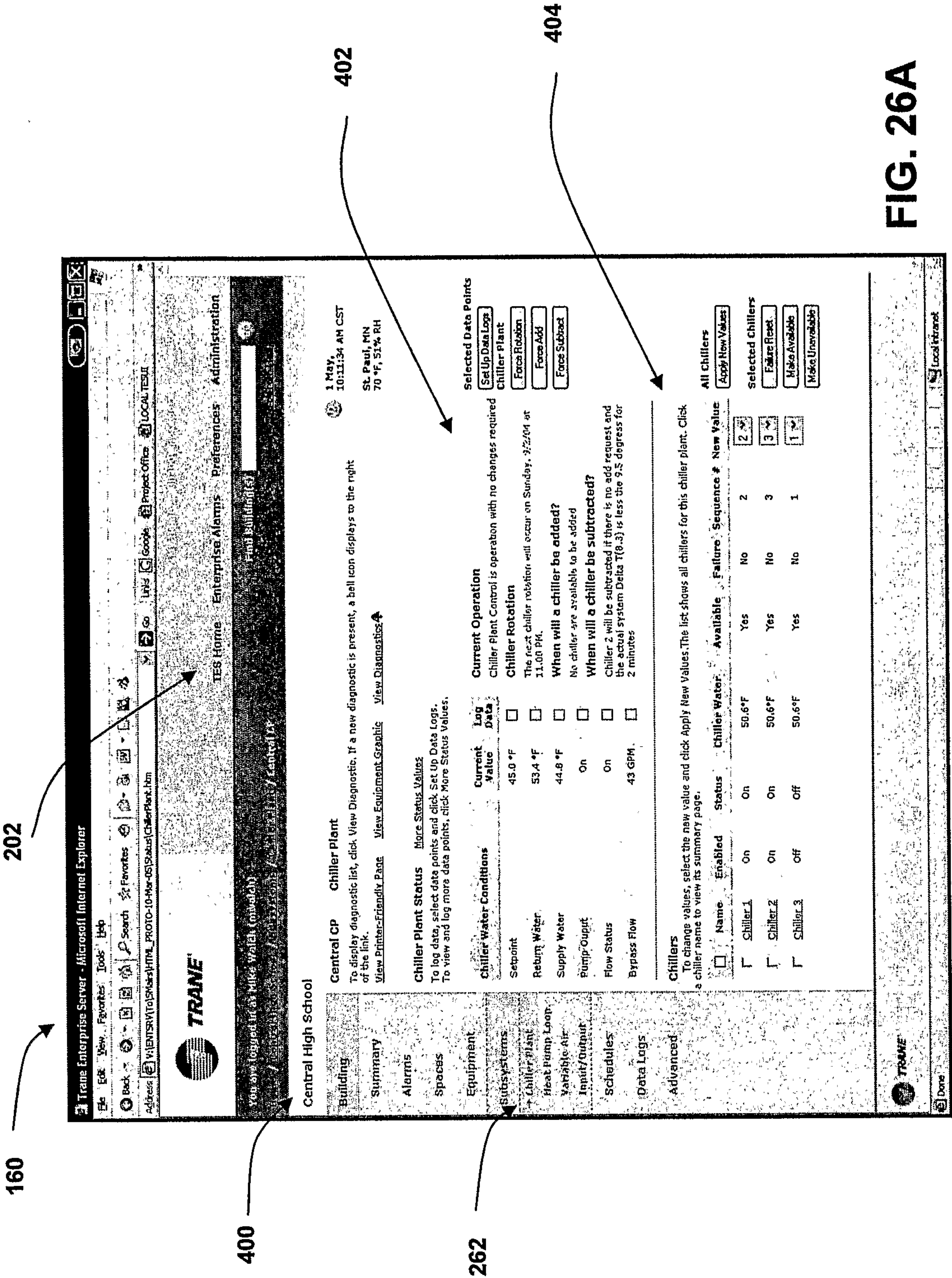


FIG. 26A

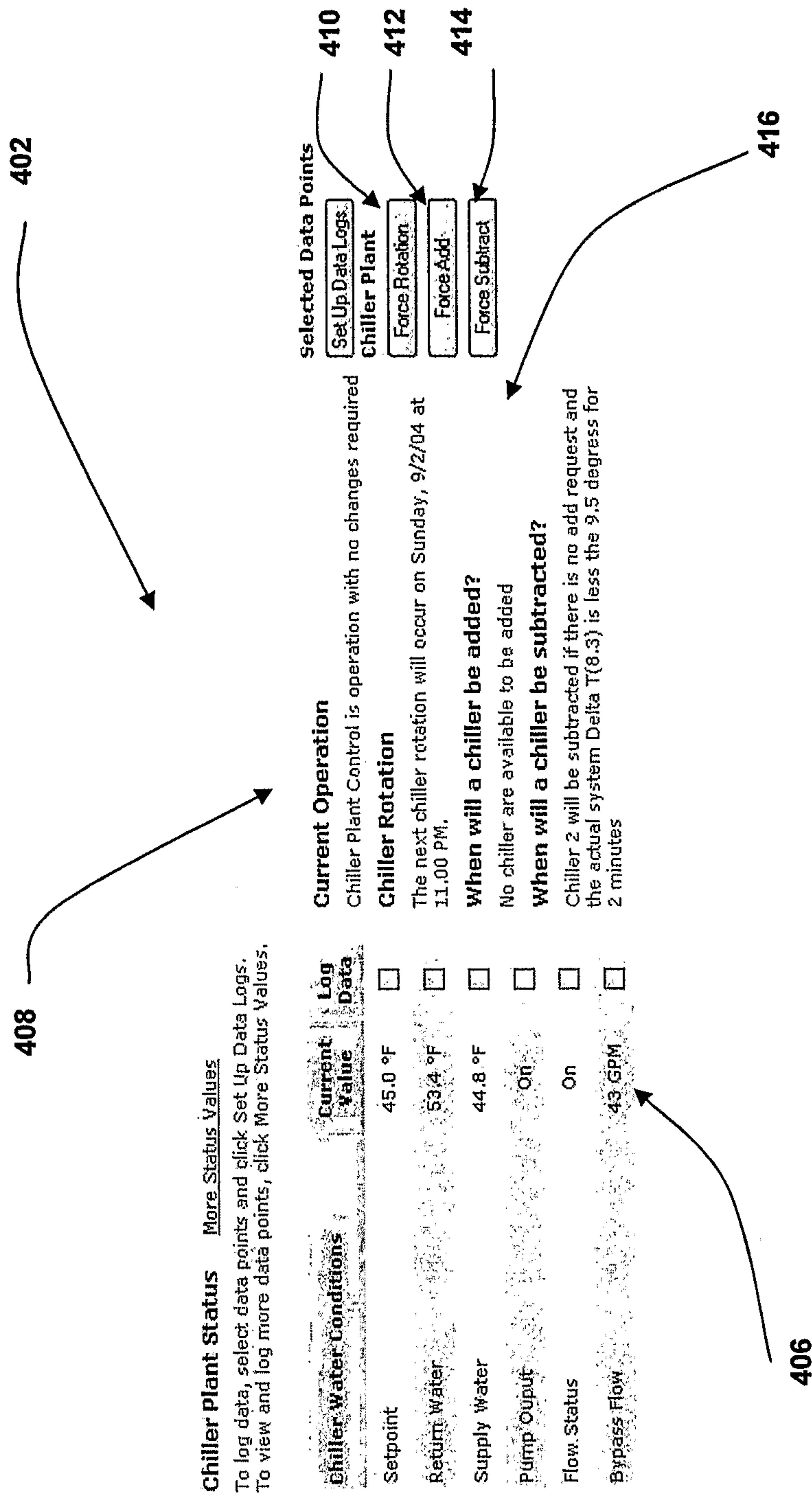


FIG. 26B

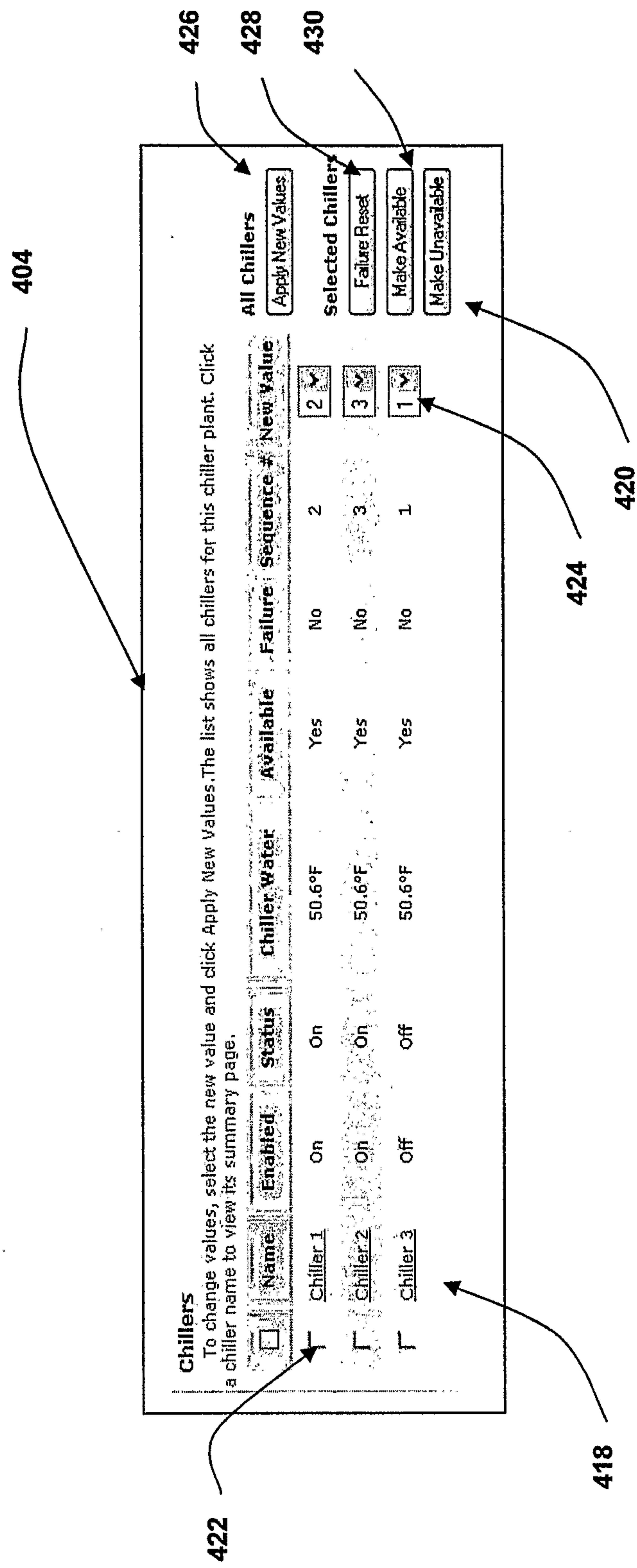
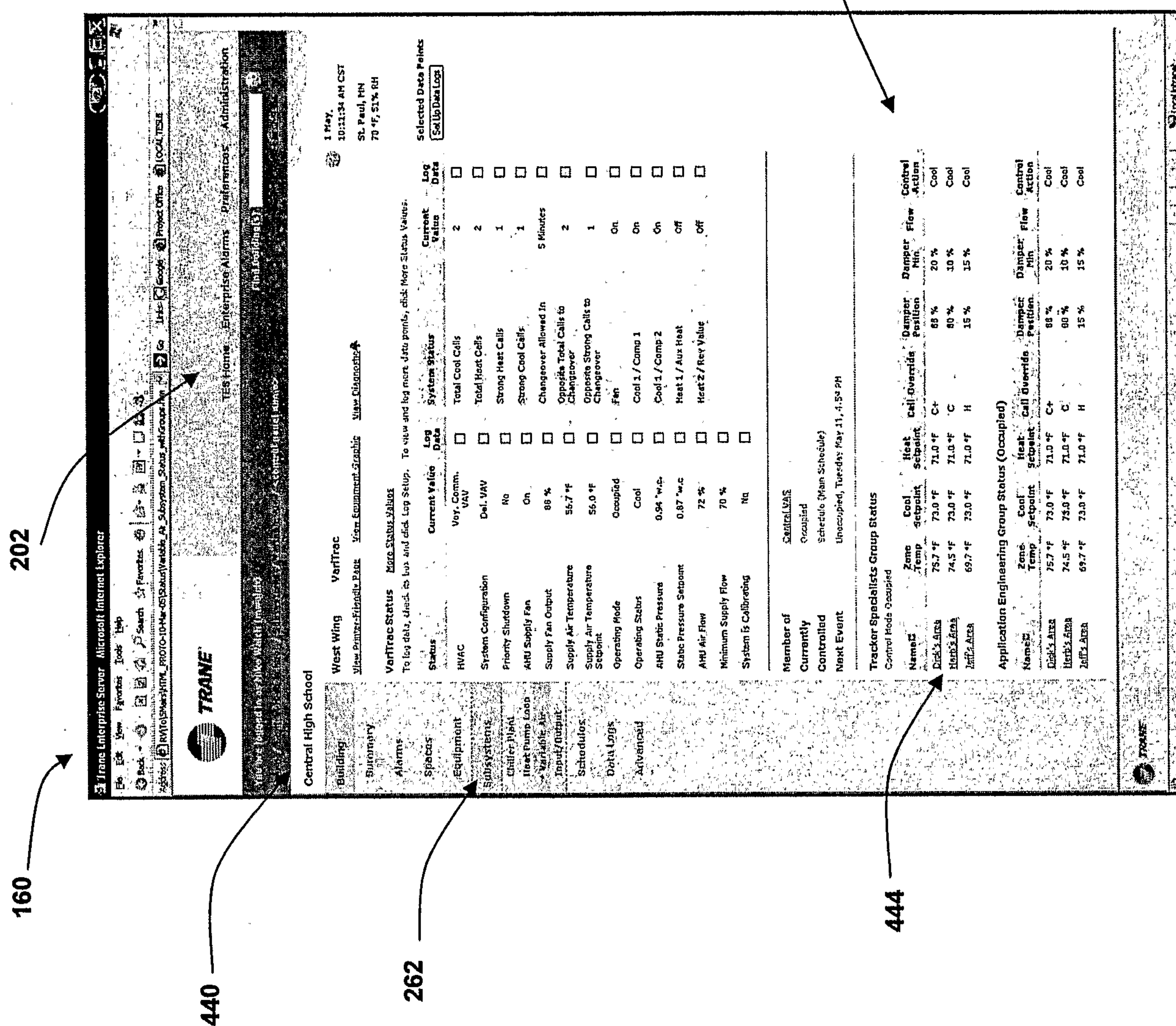


FIG. 26C

FIG. 27



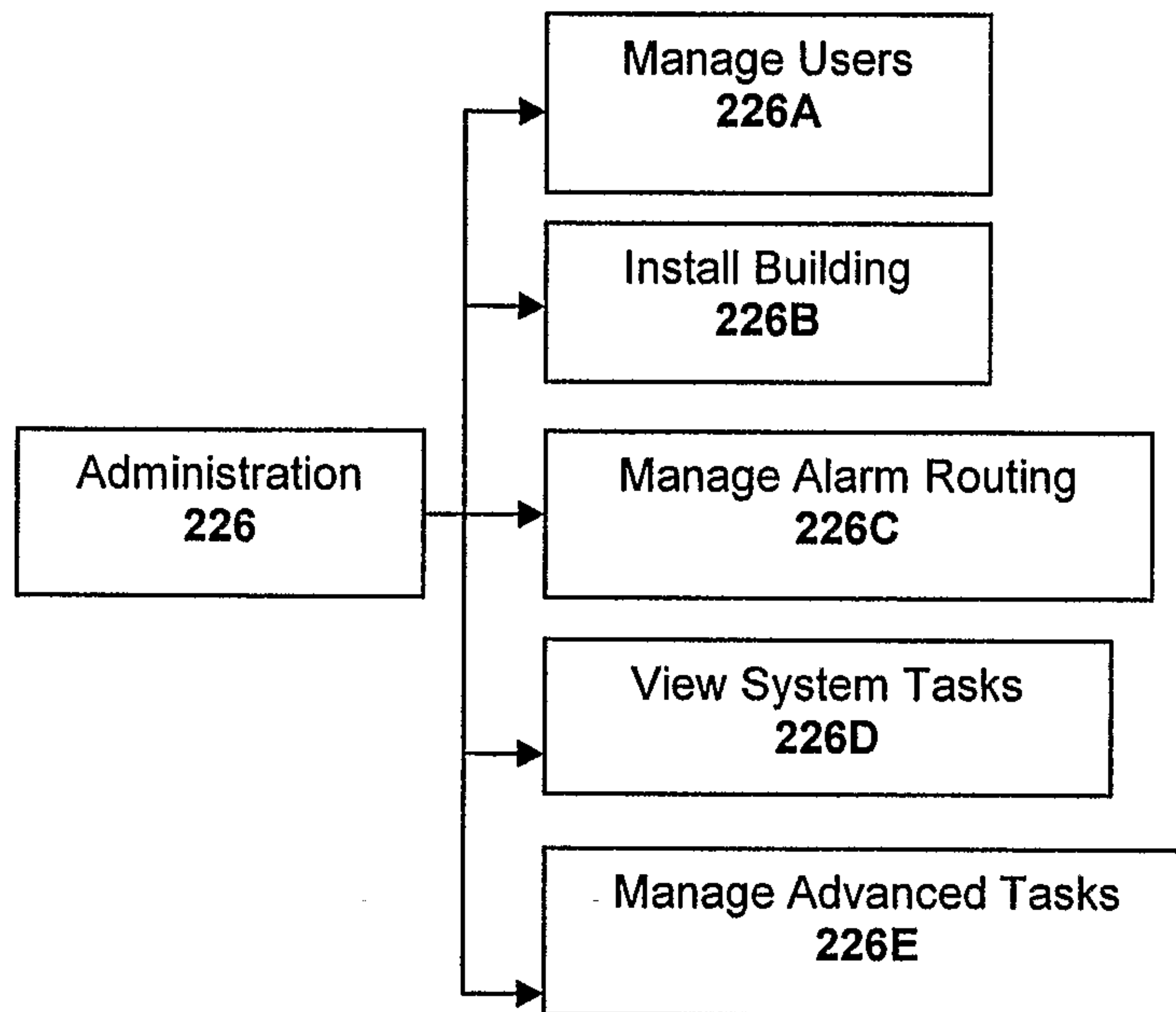
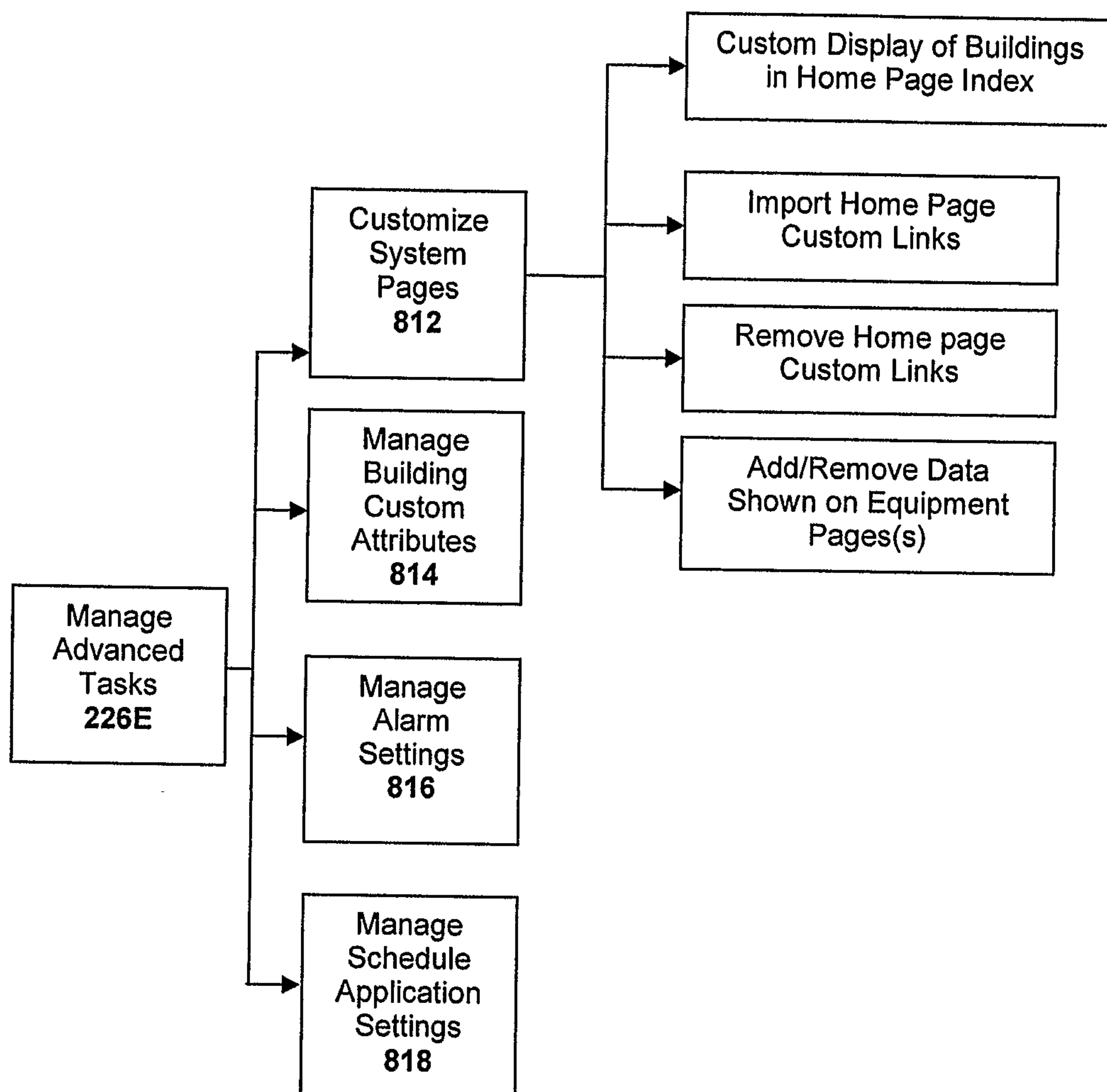
**FIG. 28**



FIG. 29

**FIG. 30**

160

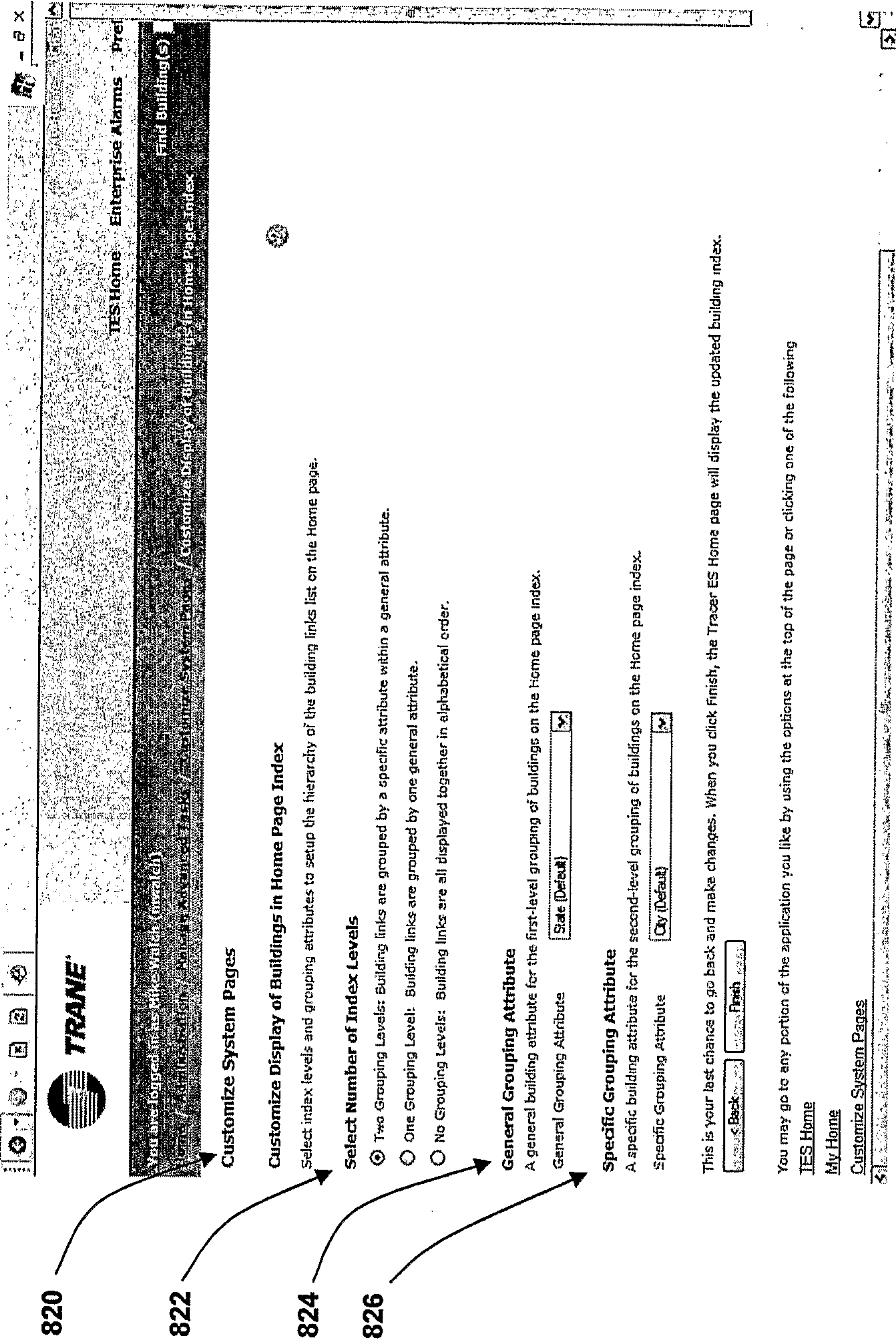


FIG. 31

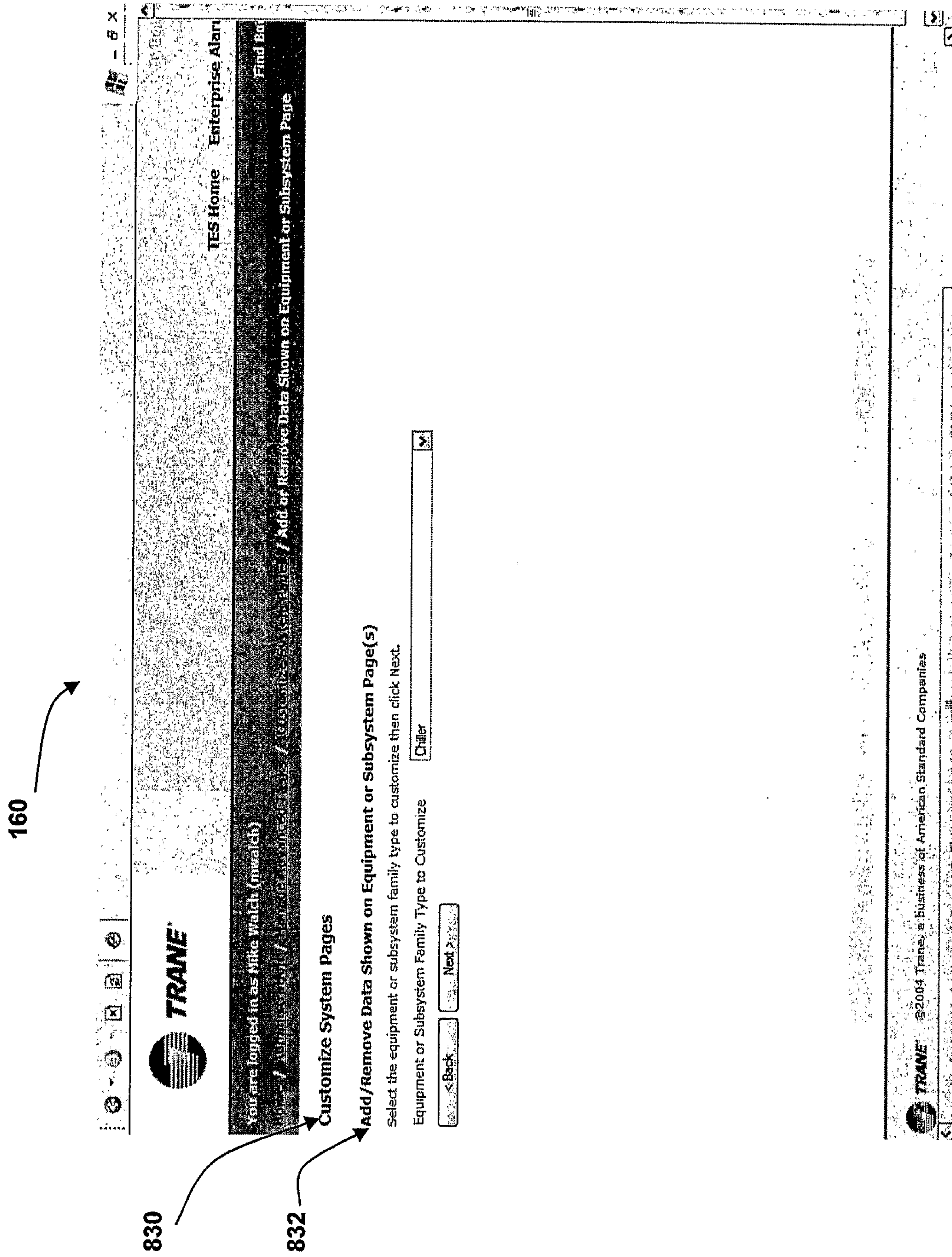


FIG. 32

160

834

836

ES

MS - B X

Tracer ES Home

Enterprise Alarms

Preferences

Administrative

You are logged in as Mike Walsh (mwalsh)

Find Building(s)

View Printer-Friendly Page

Data on Equipment Page

Customize System Pages

Add/Remove Chiller Equipment Summary Data

Select to show data points on the summary page and clear to delete data points from the summary page.

View Printer-Friendly Page

Currently Displayed	System Default	Chiller Conditions Table	Currently Displayed	System Default	Status Table
☒	Yes	Chilled Water Setpoint	☒	Yes	Active Mode
☒	Yes	Leaving Chilled Water	☒	Yes	Operating Mode
☒	Yes	Entering Chilled Water	☒	Yes	Active Cooling Output
☒	Yes	Chilled Water Flow	☒	Yes	Current Draw
☒	Yes	Chilled Water Pump	☒	Yes	Current Limit
☐	No	Leaving Condenser Water	☐	No	Alt Status 1
☐	No	Entering Condenser Water	☐	No	Alt Status 2
☐	No	Alt Temp 1	☐	No	Alt Status 3
☐	No	Alt Temp 2	☐	No	Alt Status 4
			☐	No	Alt Status 5
			☐	No	Alt Status 6

This is your last chance to go back and make changes. When you click Finish, the equipment summary pages display data selected in the list above.

Back

Finish

You may go to any portion of the application you like by using the options at the top of the page or clicking one of the following

Tracer ES Home

My Home

Customize System Pages

FIG. 33

160

840

842

844

846

848

850

160

TRANE

TES Home Enterprise Alarms Preferences

You are logged in as Mike Waleit (mwalch)

Find Building (-)

Add New Custom Building Attribute

Add New Custom Building Attribute

Complete the steps in this wizard to create a new custom attribute that can be applied to all buildings in the system.

Attribute Name

Select Type of Attribute

☐ **Two Choices** - You define two usually opposite values that users choose between
Example - Yes or No, True or False, On or Off

☐ **Fixed Choice List** - You define list and users can pick only from the list or add new items
Example 0 Class 1 or Class 3

☐ **Growing Choice List** - You define initial list and users can select from list or add new items
Example: red or Green or Blue or Yellow or <User Color 1> or <User Color 2>

☐ **Free Form Value** - Users define single content item appropriate and unique for each building
Example - Joe Smith 651-555-1212; this is a large building; Serviced by Nashville Office

TRANE

FIG. 34

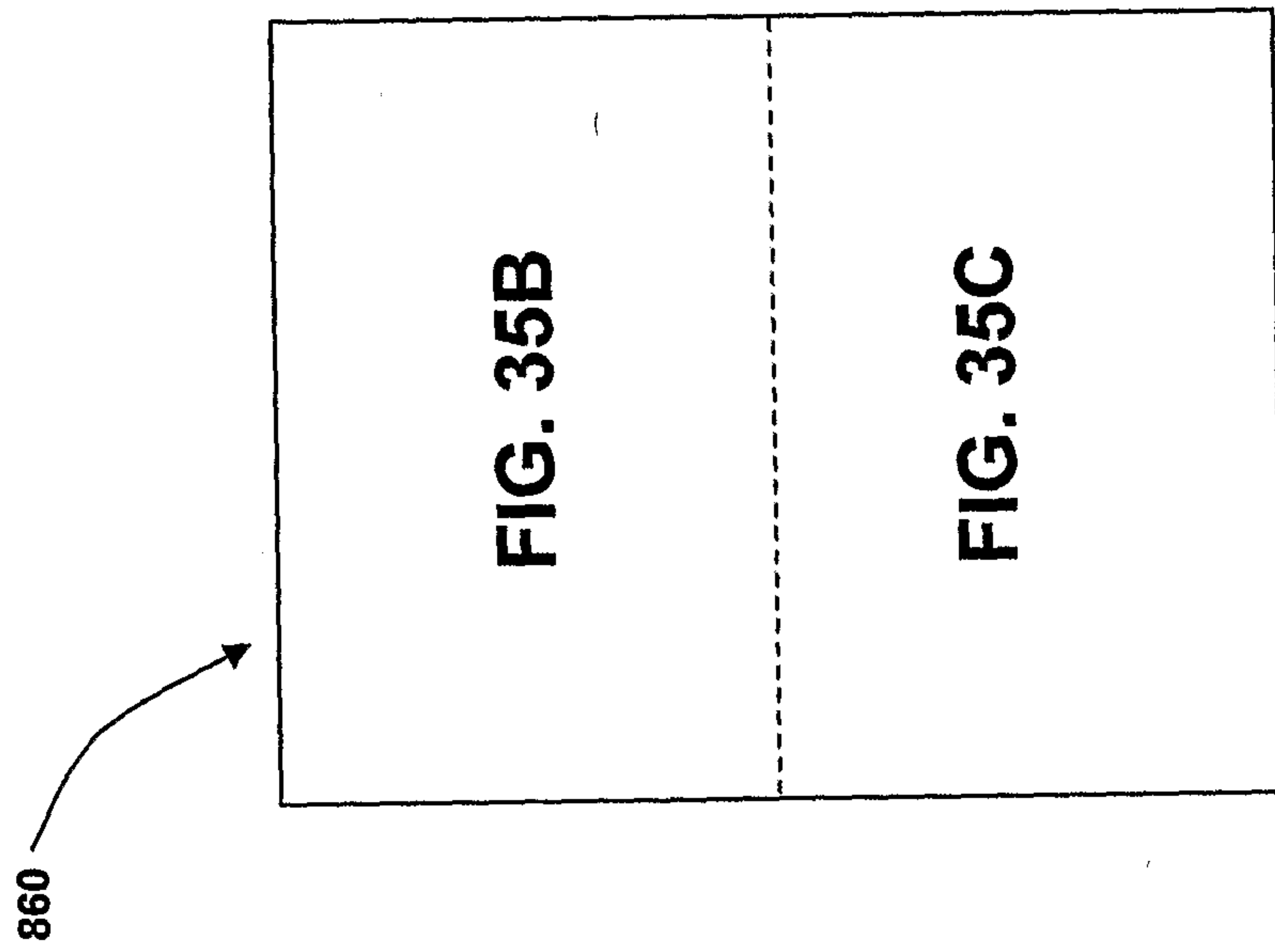


FIG. 35A

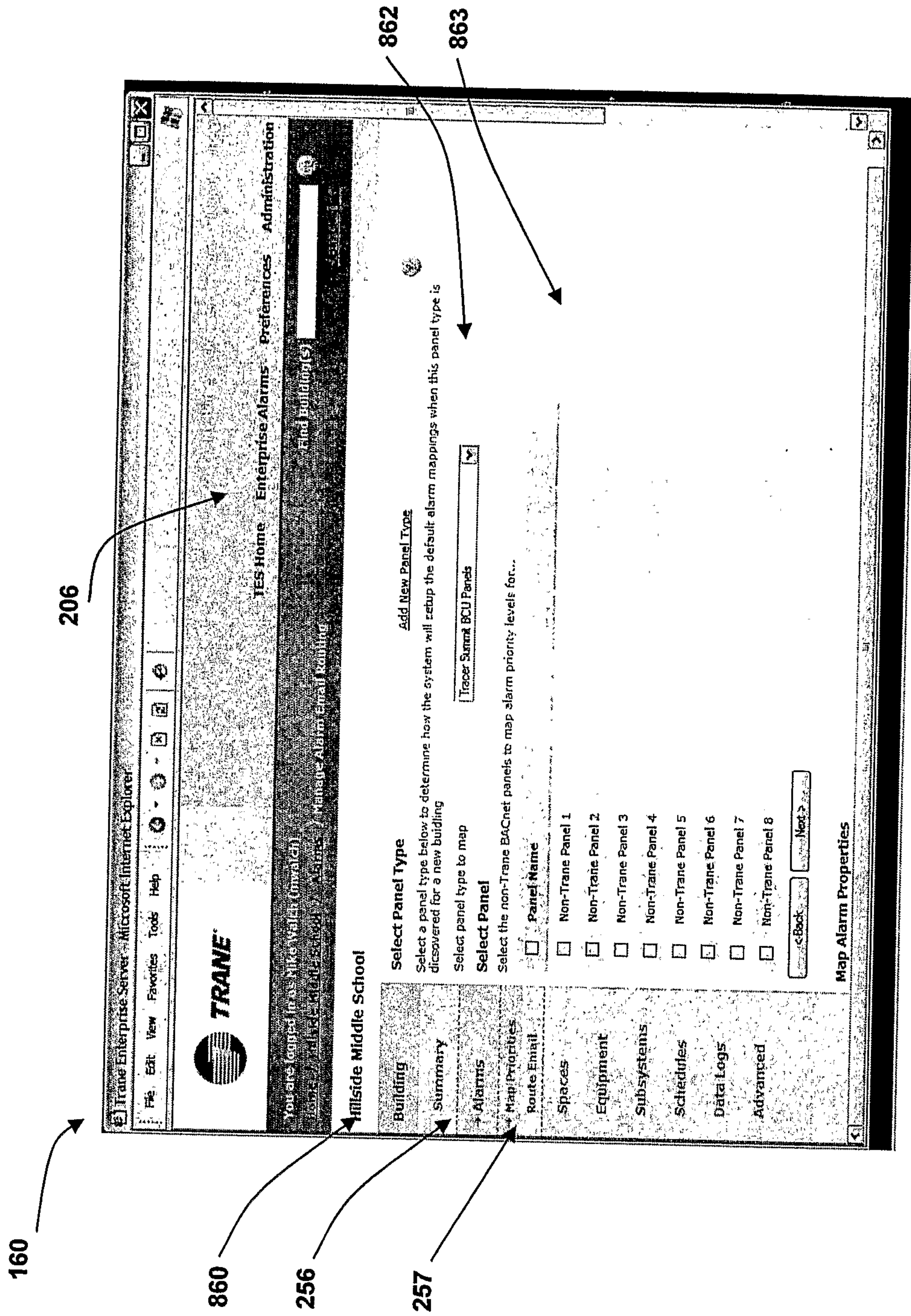


FIG. 35B

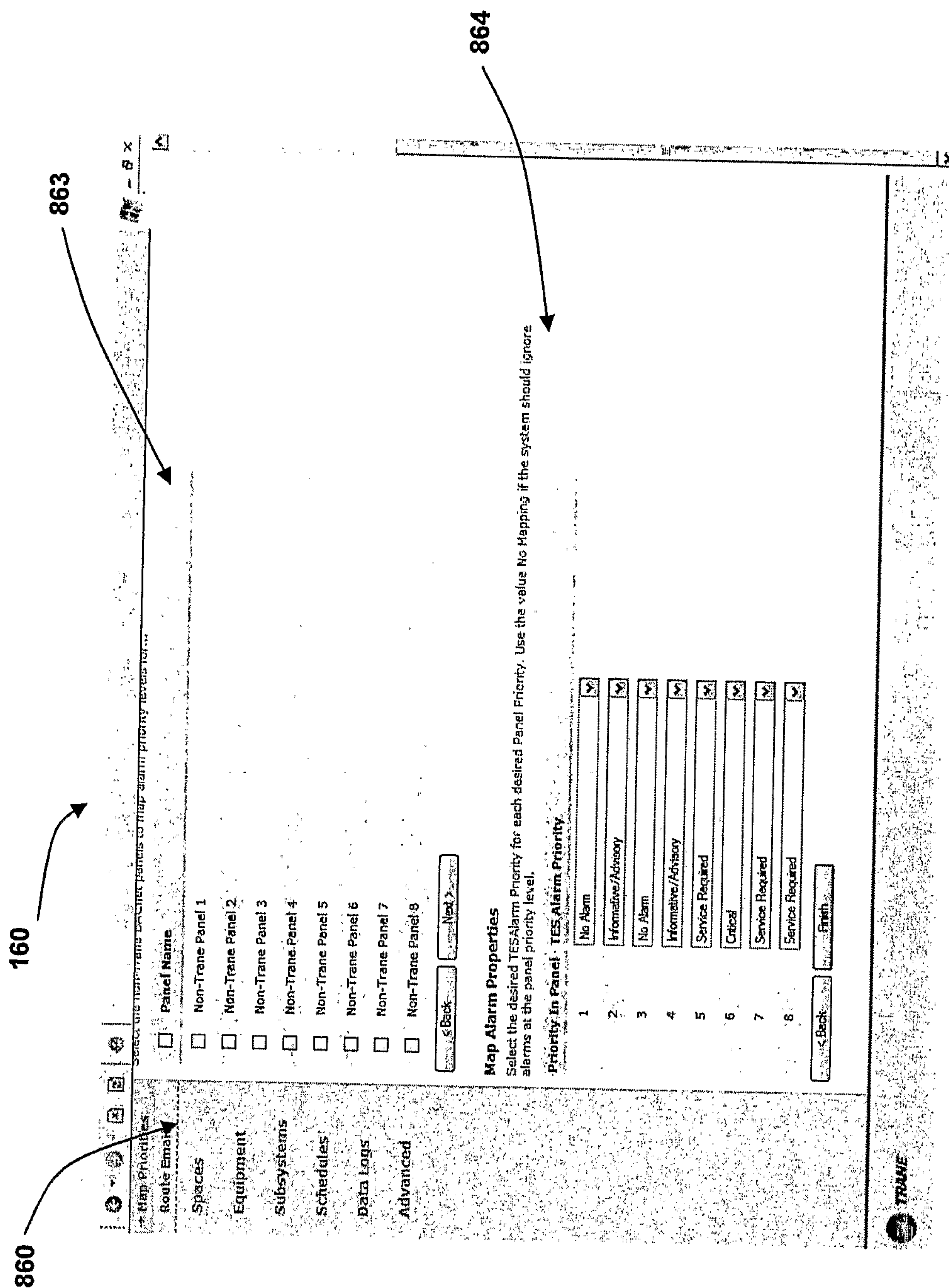
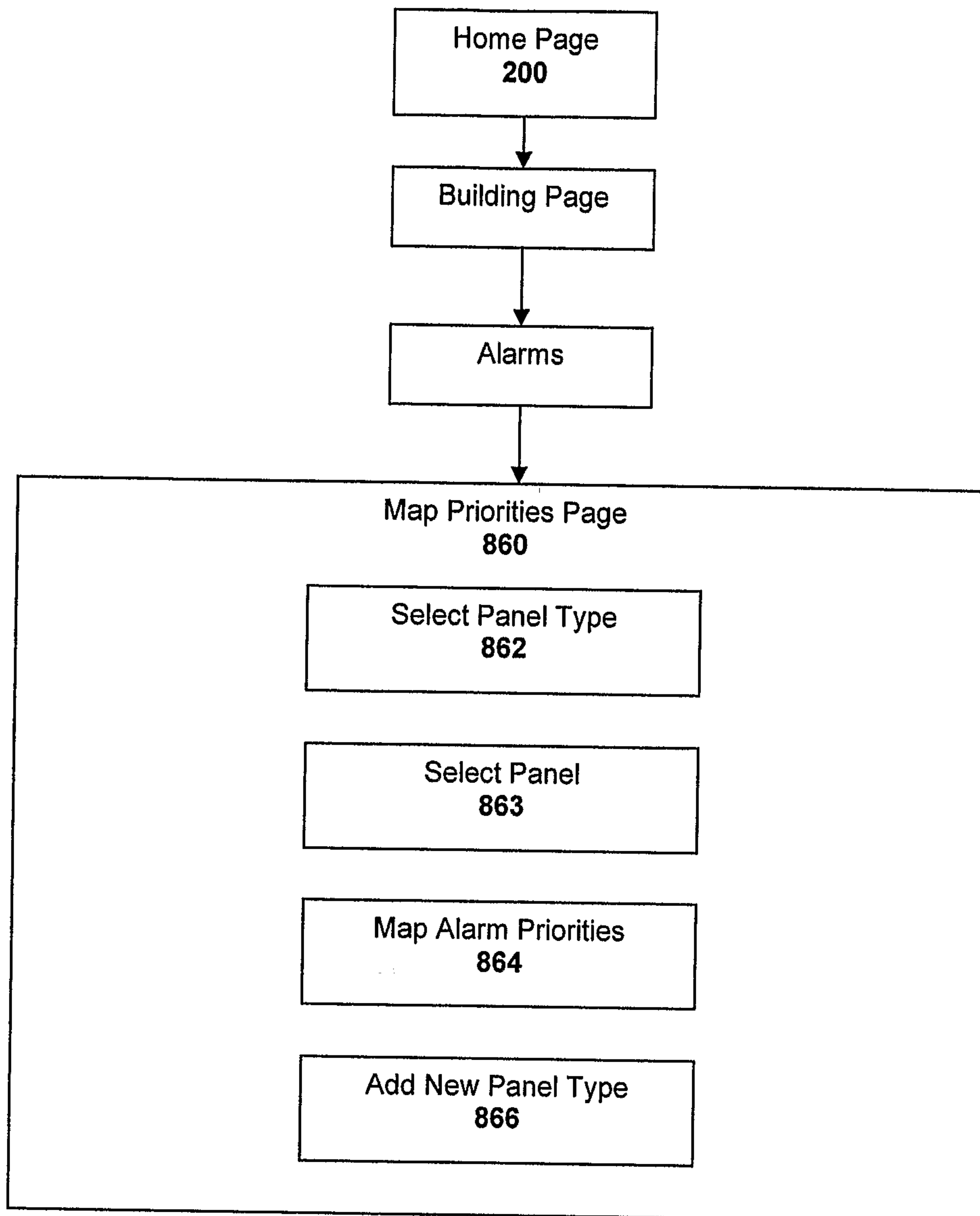


FIG. 35C

51/52**FIG. 36**

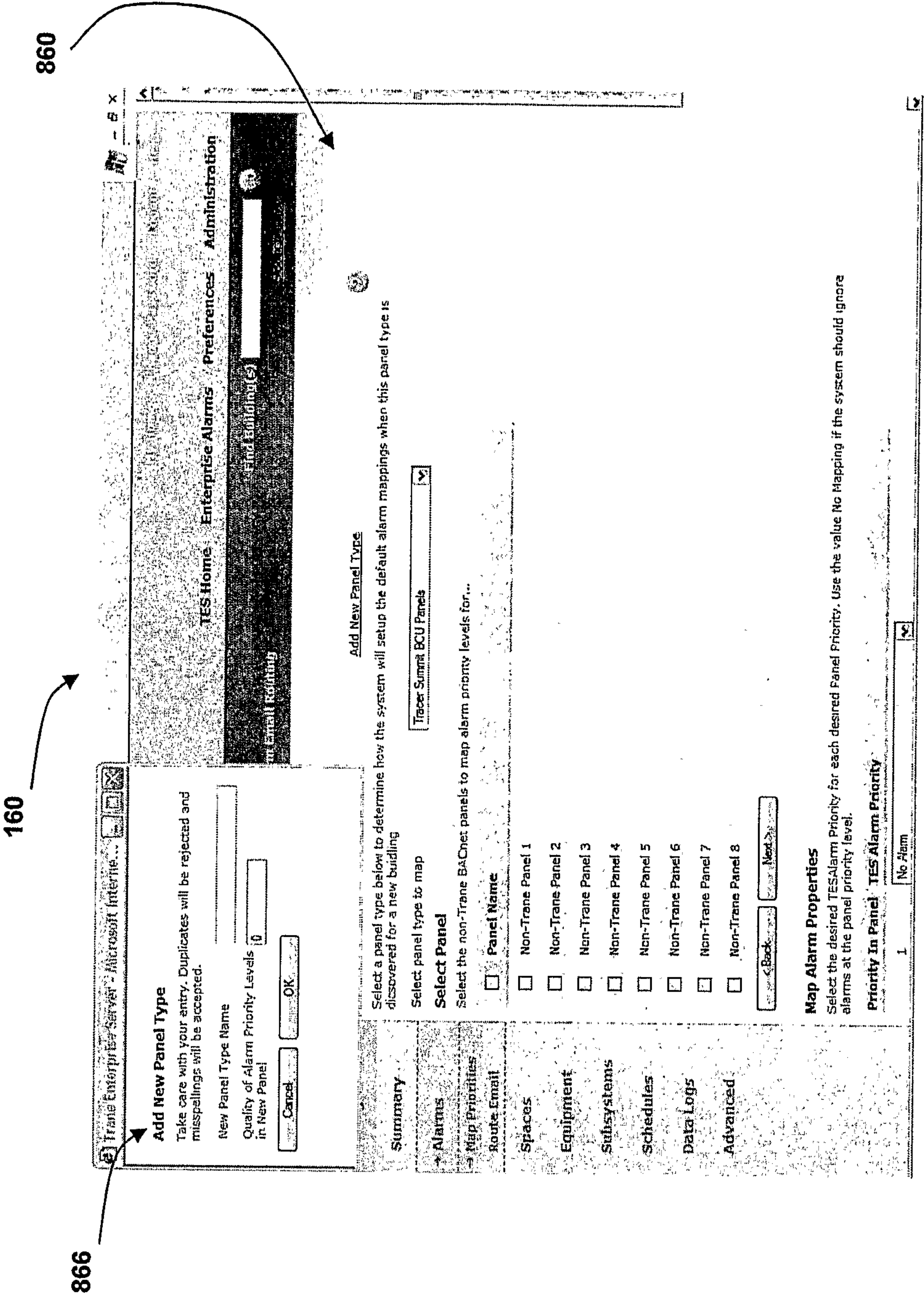


FIG. 37

