# COMPUTER IMPLEMENTED METHOD AND SYSTEM FOR INTEGRATING MULTIPLE BUILDING SYSTEMS AND BUSINESS APPLICATIONS

## BACKGROUND OF THE INVENTION

**[0001]** This invention relates to supervisory control and data acquisition (SCADA) systems generally. More particularly, this invention relates to a method and system for user-customizable and process-driven integration of the control and response features of various independent and unrelated building systems and business applications.

## BRIEF SUMMARY OF THE INVENTION

**[0002]** Computerized systems for monitoring, diagnosing, operating and controlling the operations of a wide variety of facilities, including factories, power plants, warehouses, office buildings, apartment buildings, shopping centers and parking facilities are known. In such Supervisory Control and Data Acquisition (SCADA) systems, microprocessor devices convert device measurement and status inputs into computer data for logging and transmission to higher level processors. These supervisory processors make rules-based logical decisions for the operation of a subsystem or subsystem device and send out set points to dedicated controllers which make the changes to actuators. The SCADA network therefore connects to many controllers and field devices to gather information and make rules-based logical decisions. These SCADA systems typically integrate a large number of control devices through a number of diverse control networks. For example, there may be security, fire protection, heating, cooling, electrical and elevator control systems. U.S. Pat. No. 6,067,477 describes a computer integrated SCADA system for managing distributed field devices that control or monitor physical or logical entities and for providing users the ability to construct personalized SCADA applications. The computer controlled system provides real time continuous operational control by monitoring data received from one or more remote device sensors and actuating at least one of the plurality of control devices based on input from said remote sensors. Other building, site, entity management, and enterprise systems and applications exist including telephone, computer network, reservation/scheduling, appointment/calendar systems, public address, intercom, and document management systems, but these systems have not been included in or integrated with traditional SCADA systems. As-built integrated systems are known

in which, for example, fire control systems are linked to the security system so that fire doors are automatically unlocked when a fire alarm is sounded. Other systems have been similarly linked in relatively simple and direct fashion. However, all of these system linkages are completed manually and at the time of installation, and such linkages are not readily scalable or expandable to include other systems that were not part of the original installation. Moreover, the programmed responses of one system to data or input from the same or different system in prior art systems are purely rules-driven. In other words, all responses are hard programmed into the system, and consist primarily of "if x, then y" instructions. These instructions, or "rules," are set up by the system engineer when the system is installed, and the end user, i.e., building tenant, building supervisor or maintenance engineer, has no way to update, modify or delete the rules. Additionally, since prior art SCADA systems do not integrate enterprise systems, there has been no way for end-users to configure building systems to respond according to input from or data and information resident in enterprise systems. Specifically, traditional SCADA system information management schemes are not compatible with traditional enterprise application data and data formats.

**[0003]** The present invention provides a computer-implemented system and method by which an end user, for example, a building superintendent, maintenance technician, or even a tenant, can easily not only monitor and manage multiple building systems with a single interface, but in which the user can easily customize and revise the settings and automated responses for multiple building systems based on input and transactions from multiple different building systems and enterprise applications, even when the building systems, when installed, have had no connection or programmed interaction with other building systems or enterprise applications. Stated another way, under presently available systems, a building manager or tenant can program a first building system to react to a second building system or enterprise application status, information, transaction, or alarm, only if the first building system was designed and installed to work together with a second building system and/or enterprise application and with a specially designed and installed pre-integrated control system. However, if the building manager or tenant desires to program or configure a third building system to respond in a specific way based on the information, status, transaction, alarm or change in status in the first or second system building system, or in an enterprise application, he or she has no way of doing so. Indeed, the only way to integrate a

previously un-integrated building system with another building system, or with an enterprise system, is to remove both systems and re-install two pre-integrated systems. Alternatively, while it might be technically possible to build, in-situ, an integrating interface that specifically receives the reports, status, alarms, and transactions of two or more building systems and/or enterprise applications and allows a user to program the responses and actions of one system based on the input of another system, such in-situ retrofit of an integrated command interface between two or more pre-existing building systems and/or enterprise applications has not been considered economically feasible as compared to reinstallation of pre-integrated systems.

**[0004]** Notably, while sophisticated prior art SCADA systems can receive the data, input, alarms, etc., from multiple building systems and allow users to configure each of those multiple building systems, no prior art SCADA system has had the capability to permit an end user to use data from one building system as an input for the control of another building system, unless as mentioned above, the two building systems were installed together with an integrated control system. Moreover, no prior art SCADA system has had the capability to permit an end user to use data from enterprise applications as input for the control of a building system based on business requirements, and to easily revise the response features of building systems as business requirements change.

**[0005]** Accordingly, the present invention provides a drop-in command and interface software module that allows a user to integrate control, command and response functions between previously installed and un-integrated building systems without the need for expensive and time-consuming retrofits of actual physical equipment. Thus, the present invention allows the user to configure the activation or deactivation of one system function based on the data, alarm, event, or transaction from one or more other systems. Moreover, as additional or replacement systems are added to a building's systems, the present invention can easily accommodate the new additions without reconfiguring the entire control system. In short, the present invention takes the data and control functions from any and every control system and allows a user to build his or her own intelligent building, and to edit the control processes on their own at any time. The building systems and enterprise applications that can be integrated with this invention include, but are not limited to, HVAC, fire control, access control, video monitoring, public address, exterior lighting and signage,

interior public and private space lighting, escalator and elevator systems, telephone systems, room scheduling/reservation systems, resource management systems, data and information technology systems, document management systems, e-mail servers, fax servers, SMS servers, building tenant/visitor database systems, and customer relationship/management systems.

**[0006]** According to another aspect of the invention, the responses of building systems are process driven, rather than rules based, and the response process is fully editable by the end user. For example, when the SCADA system receives data from a remote sensor indicating that smoke has been detected, the system of the invention launches a process, which process has been designed by the user, rather than simply and automatically triggering an audible alarm and sending a notification to the fire department. For example, according to the invention, an end user can design a smoke detection response process according to which the system queries various systems to determine whether any persons are currently occupying the space where smoke has been detected. The system can also query the HVAC system to determine whether there has been a rise in temperature in the area where smoke has been detected. The system can be programmed to make various other queries to verify the signal, determine its scope and severity, and determine what personnel and resources are at risk. Based on the data received based on any one or more of these inquiries, the system can be programmed to command various systems in various ways – for example, the security system can be commanded to prevent entry into the space; the HVAC system can be commanded to close or divert air flow into or out of the affected area, and various persons can be alerted. The system can be programmed to wait for end user intervention before an alarm is sounded, for example, if based on the queries, the threat level doesn't meet a particular threshold. And all of these queries and responses are full customizable and editable by the end user. At any time, the end user can add or delete steps to the response process using the systems process design interface.

**[0007]** According to another aspect of the invention, the system allows for multiple processes for a given input, depending, for example, on the location of the sensor providing the input. That is, the process that is launched when one smoke detector (for example a smoke detector located in a conference room) detects smoke can be completely different from the process that is launched when another smoke detector (for example, a smoke detector located in the parking garage) detects

smoke. This goes for each sensor in the system. A completely different process may be programmed for launch for every individual sensor. For example, one tenant may want a very simple response process when smoke is detected in its tenant space. Another tenant in the same building may want a more complex, or in any event, a different process, launched when smoke is detected in its tenant space. According to the system of the invention, each tenant in a building may program the building system response processes according to its own business needs and management policies.

**[0008]** According to a further aspect of the invention, the present invention accommodates input from both value-oriented systems (also referred to as point-oriented systems) and transaction-oriented systems (also referred to as event-oriented systems). Value or status-oriented systems monitor and/or control the state of something, typically based on a value or threshold. Value or status-oriented systems include fire control systems, HVAC systems, etc. In these systems, values or statuses are set and/or monitored, and action is only taken when a status or value changes. Transaction or event based systems are systems that monitor and allow (or prevent) transactions or events. More specifically, transaction or event-based systems have a much richer and more complex data set. Transaction or event-based systems maintain a data record of system transactions or events, which data record includes, for example, the date of the event or transaction, the time of day of the transaction, the location of the event or transaction, and, in the instance of entry control, the identity of the access card that triggered the event or transaction, and the person to whom the access card is assigned. Prior Art SCADA systems allowed for the integration of multiple value or point-oriented systems using the common object model (although not for end-user configuration of response settings), but they did not allow for the integration of value-oriented systems with transaction based systems because the address space (the data record) for a transaction-based system is constantly growing, whereas the address-space for a point-oriented is fixed. That is, a point-oriented system can be easily programmed to communicate with another point oriented system because the addresses for each system are fixed. Not so for a transaction-based system. Accordingly, the inventors saw a need to develop a way to allow for the integration of point-oriented building systems and transaction-oriented building systems and enterprise applications. The present invention therefore provides a system and method for integrating both point-based

building systems and transaction-based building systems and enterprise applications and the end-user building, configuration, and customization of response processes using input from both value- and transaction-oriented systems.

**[0009]** A more detailed description of features of the invention is set forth below in the detailed description of the invention, with reference to the figures. While this invention is described with respect to building systems (e.g., HVAC, fire control, etc.) and enterprise applications (e.g., calendar applications, reservation applications, e-mail application, document management applications, database management applications), this invention is intended to also apply to the integration, and customizable control of response processes for other infrastructure and/or municipal management systems, such as traffic lights, water fountains, police and fire communication/dispatch systems, as well as to administrative applications which are building-specific, for example, tenant management applications, etc. The following detailed description of the invention is not intended to limit or exclude these embodiments from the invention.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0010]** FIG. 1 is a representation of prior art SCADA systems, having a portal layer, a data layer and a device and application layer.

**[0011]** FIG. 2 is a representation of a SCADA system according to the invention, in which a process management layer resides between the portal layer and the data layer.

**[0012]** FIG. 3 is a representation of an early step in a user interface for a process editor according to which an end user might build or edit a response process according to an embodiment of the invention.

**[0013]** FIG. 4 is a representation of a first subsequent step in a user interface for a process editor according to which an end user might build or edit a response process according to an embodiment of the invention.

**[0014]** FIG. 5 is a representation of a second subsequent step in a user interface for a process editor according to which an end user might build or edit a response process according to an embodiment of the invention.

**[0015]** FIG. 6 is a representation of a third subsequent step in a user interface for a process editor according to which an end user might build or edit a response process according to an embodiment of the invention.

**[0016]** FIG. 7 is a representation of a fourth subsequent step in a user interface for a process editor according to which an end user might build or edit a response process according to an embodiment of the invention.

**[0017]** FIG. 8 is a representation of a fifth subsequent step in a user interface for a process editor according to which an end user might build or edit a response process according to an embodiment of the invention.

**[0018]** FIG. 9 is a representation of a sixth subsequent step in a user interface for a process editor according to which an end user might build or edit a response process according to an embodiment of the invention.

**[0019]** FIG. 10 is a representation of a building hierarchy and two different configured processes for the event "security card access denied" depending on the location of the event in the building hierarchy.

**[0020]** FIG. 11 is a representation of a building hierarchy and a status list of configured processes that might be under way at a given time for various triggering events.

**[0021]** FIG. 12 is a representation of a set of process steps including a "wait for door lock status" flow control step.

**[0022]** FIG. 13 is a representation of a set of process steps including a process loop set to wait for a condition to be satisfied before the next step in the process is executed.

**[0023]** FIG. 14a is a representation of an active process list that might be presented to a user upon request for active processes.

**[0024]** FIG. 14b is a representation of a display that might be presented to a user upon a request for more detail as to the status of a particular active process.

**[0025]** FIG. 15 is a representation of a list of process milestones that might be presented to a user upon request for status of a particular process.

**[0026]** FIG. 16 is a representation of a graphical display that might be presented to a user to illustrate statistics relating to various processes.

**[0027]** FIG. 17 is a representation of a different graphical display that might be presented to a user to illustrate statistics relating to various processes.

## DETAILED DESCRIPTION OF THE INVENTION

**[0028]** Figure 2 shows a SCADA system according to the invention in which the process management system of the invention resides between the portal layer and the meta data layer. Field devices, control applications and enterprise applications constitute the device and application layer. Data and information from various and diverse field devices and control applications are represented and stored in the SCADA system at the meta data layer according to known methods and systems. According to the invention, data and information from various enterprise applications are also represented and stored in the SCADA system at the meta data layer. The portal layer includes applications and interfaces for logging onto and accessing the SCADA system, once again according to known methods and systems. However, where, according to the prior art, a user would use the portal layer to monitor, query, and command each device and application separately, the present invention provides a process management layer which comprises an interface module which communicates with the portal layer, a process configuration module which an end user accesses via the interface module and which guides the user through device selection and response process configuration, and (what) which stores the configured process and which executes process launch and process monitoring when there is a particular status change, event or transaction at a field

device or control or enterprise application at the device and application layer. System access and process configuration by an end user, as well as process launch and execution upon a status or event trigger can be explained by way of example.

**[0029]** Generally, setting up an integration logic using a process involves the following steps: (1) identify the object, the activation of which is intended to cause a process to launch; (2) define the steps in the process using a process editor module; (3) define any filtering logic to qualify events as a process-triggering event; and (4) arm the process to be launched upon a given event for a given location.

**[0030]** Example 1: System Access and Process Configuration

**[0031]** The present example describes how the user selects a device, and how a user goes about configuring a response process based on a particular status change/event/transaction. In particular, this example illustrates the response process configuration steps in the instance of an "card access denied event" in a card access entry/exit security system.

**[0032]** Defining the Process

**[0033]** To begin the configuration, the user opens a process editor module and selects the "object type" of the device, actor, or event for which the user wishes to configure a response process. Object types for which an end user can configure a response process include users, equipment, work orders, and tenants. In the case where the user wishes to configure a response process upon the trigger of a card access denied event, the user selects "equipment" from among the "object type" categories and then follows a series of menu prompts to select a category of devices, a subcategory of devices, or a single device, and a trigger event, for which to configure a response. In this example, the user selects a category of devices: "all card access devices", and selects "card access denied" as the event.

**[0034]** Once the object/device and event trigger are selected, the user begins the configuration of the response process. The system presents the user a menu of various work-performing steps, including "notifications," "actions," and flow-controlling steps such as "conditional branching",

"delay", "waiting for a criteria to be set", "looping", "conditional termination", "capturing process variables" and "logging milestones."

**[0035]** According to this example, the user selects as a first step in the process the notification of the card owner whose assigned card triggered the "access denied" event. The user may select "notification" as the action, specify the recipient of the notification, the mode of the notification, and even the text of the notification. See Fig. 4. The identity of the card owner would be part of the data that would be part of the triggering event. The user may also configure the system to make multiple notifications, as well. For example, in addition to notifying the card owner, the user may configure the system to notify the security station nearest the location where the card access denied event occurred. The notification to security might instruct them to view the door and the person trying to pass through the door and make a decision as to whether to open the door manually. In order to facilitate security's inspection of the door, the user might configure the process to access lighting control to turn on lights in the area surrounding the location where the card access denied event occurred. To accomplish this, the user would use the process editor to access lighting equipment control as shown in Figure 5. According to this example, the user would configure an action to send "value 1" to "ON/OFF control" point of the "light" (equipment) in the "installed location" of the card reader. The user might also configure the system to send a notification to "security manager" of that location. As with the notification to the card owner and the security station, the notification to the security manager may be customized. For example, the user may configure the system to access the phone number of the card owner from enterprise systems and provide the card owner's phone number to the security manger so that the security manager can call the affected user. See Fig. 6.

**[0036]** The user might then configure the process to determine whether the card access denied event is due to a malfunction, i.e., a maintenance problem. According to this step, the user might configure the process "wait for" (continually monitor) the "door lock status" for next 3 minutes. If the door is opened manually within that time, the system can be configured to conclude that the card reader is not functioning properly, and send a notification to a maintenance manager of the problem and the location. See Fig. 7. The user can custom configure the message to the maintenance

manager to reference potentially faulty equipment, the card reader in this case, as well as card owner whose card triggered the card access denied event. See Fig. 7.

**[0037]** Once the user has added all the notifications, actions and (other steps) that are desired to be included in the response process, the user will identify the process with a "process name" and upload it to the system. In the case of Figure 8, the configured process has been given the name "Card Access Denied." See Figure 8.

**[0038]** Arming the process for a specific location

**[0039]** Once the user has configured and named a process, the user may then select the event for which the process is to be launched, and the location(s) where the event must occur in order for the process to be launched. In the case of Figure 9, the user has configured the process "Card Access Denied" to be launched whenever the event "Security . CardAccessDenied" occurs anywhere in Building 3.

**[0040]** Example 2: Status or Event Trigger and Process Launch

This example describes how the system of the invention operates according to a user-configured process when a particular status/event trigger occurs. Generally, when a triggering event occurs, the appropriate process(es) will be launched and a "process instance" will be created in the system for each process launched. The user can view the list of process instances running in a given location at any given time, and open any of the process instances to monitor its current status. The system will show current step of the process, as shown for example in Figs. 14a and 14b.

**[0041] Execution Steps: Determining Event Location**

**[0042]** When an event is received by the system, the system first determines the "event location" through object relationships. For example, when a "card access denied" event is received, the event location is considered to be the "installed location of the card reader" that generated the event. However, there are situations where "event location" is not the physical location of the activity that triggered the event. For example, an "email received" event would consider the "location of the office" of the staff member who received the email.

**[0043] Execution Steps: Locating the Process Definition**

**[0044]** The system then uses the "event location" to locate the proper process definition to execute. There can be many process definitions that are qualified to act upon a "card access denied" event. Therefore, the system must locate the appropriate process definition for that particular event location.

**[0045]** First, the system checks if there is a Process Definition armed at the Event Location for this event. If so, it would be the correct process definition. If not, system then searches the "parent location" of the event location. System continues to search upwards the location hierarchy until a process definition is found.

**[0046]** In effect, this allows a user to arm a process definition at the Building level to act upon all card access denied events happening anywhere in the building (without having to configure it at each floor). However, if a special treatment is necessary for the top floor where company directors are sitting, a special process could be armed only for the top-floor as an overriding process. So, if the event location is top-floor, special process will be executed whereas any other event would cause building-level process to execute. Fig. 10 shows an example of location hierarchy and one process definition at the building level and a separate process definition at the third floor level. In the event of a card access denied event on the third floor, the process named "Immediate Escalation" would be launched. Anywhere else in the building, the process "card access denied" would be launched.

**[0047] Execution Steps: Creating Process Instance**

**[0048]** Once the system has identified the appropriate process definition, the creates a "process instance" for that particular event. Each process instance is a "living entity" that is automatically programmed to follow the steps defined in the process definition. Each process instance will remember its last executed step, and keep executing the steps until it reaches the "end" step. Figure 11 shows a list of active process instances for Building A, together with their current status.

**[0049] Execution Steps: Executing Process Steps**

**[0050]** Each process instance includes a record of its corresponding process definition along with additional data specific to that instance – such as last executed step, and the data came along with the event. The steps of the process instance are executed by the "process engine," an executable that periodically looks at process instances that require execution.

**[0051]** There are generally two kinds of process steps: (1) Queries and Actions, and (2) Flow Control. A "Query" means sending a request for some information (described in the query-data that goes along with the query). An "Action" means sending a request to perform a step such as modifying some data in the database or asking lighting control system to turn on lights.

**[0052]** Flow control steps set up the logical sequence of the process steps. These steps include condition based branching, waiting for a condition to occur, looping until a particular condition is satisfied, jumping out of a sequence or initiating another process instance. FIG. 12 shows a set of process steps that include waiting type flow control steps. FIG. 13 shows a set of process steps that include looping-type flow control steps.

**[0053] Monitoring Process Instance, Recording Milestones, Generating and Presenting Process Statistics**

**[0054]** Process instances can be monitored visually with respect to the process definition. Figures 14a and 14b show an example of a list of active process (FIG 14a), and a graphic showing the current or last executed step of a selected active process (FIG. 14b). As shown in FIG. 15, the system can record the details of the process instance ("milestones") for future analysis. Furthermore, after many process instances have been executed over time, the data from recorded milestones may be presented as "process statistics" for the user's analysis. Examples of such statistical presentations are shown in Figures 16 and 17.

## CLAIMS

1. A computer-implemented method for allowing a user to configure automated device or application responses in a first building system and at least one enterprise application, comprising providing a process management software module and configuring it to communicate with said first building system and said enterprise application, configuring said process management software module to be accessible over the internet by end users, configuring said process management software module to allow users to configure automated building system and enterprise application response processes using input from said first building system and said enterprise application.

2. A computer-implemented method for allowing a user to configure automated device or application responses in a first building system and at least one enterprise application, comprising providing a process management software module and configuring it to communicate with said first building system and said enterprise application, configuring said process management software module to be accessible over the internet by end users, configuring said process management software module to allow users to configure automated building system and enterprise application response processes using input from said first building system and said enterprise application,wherein said first building system and said enterprise application were not installed together with an integrated control system.

3. A computer-implemented method for allowing a user to configure automated device or application responses in a first building system and at least one enterprise application, comprising providing a process management software module and configuring it to communicate with said first building system and said enterprise application, configuring said process management software module to be accessible over the internet by end users, configuring said process management software module to allow users to configure automated building system and enterprise application response processes using input from said first building system and said enterprise application, wherein said method allows a user to use data from enterprise applications as input for the control of a building system based on business requirements, and to revise the response features of building systems as business requirements change.

4. A computer readable medium having code configured to allow a user to integrate control, command and response functions between previously installed and un-integrated building systems and enterprise applications wherein said code allows the user to configure the activation or deactivation of one system function based on the data, alarm, event, or transaction from one or more other systems.

5. A computer-implemented method for allowing a user to configure automated device or application responses in a first building system and at least one enterprise application, comprising providing a process management software module and configuring it to communicate with said first building system and said enterprise application, configuring said process management software module to be accessible over the internet by end users, configuring said process management software module to allow users to configure automated building system and enterprise application response processes using input from said first building system and said enterprise application, wherein the process management software module can be configured to communicate with additional and replacement building systems and enterprise applications that are added to the building after the process management software module is initially configured, without reconfiguring the entire control system.

6. A computer-implemented method for allowing a user to configure automated device or application responses in a first building system and at least one enterprise application, comprising providing a process management software module and configuring it to communicate with said first building system and said enterprise application, configuring said process management software module to be accessible over the internet by end users, configuring said process management software module to allow users to configure automated building system and enterprise application response processes using input from said first building system and said enterprise application, wherein the process management software module takes the data and control functions from a plurality of building systems and enterprise applications and allows a user to build his or her own intelligent building, and to edit the control processes on their own at any time.

7. A computer-implemented method for allowing a user to configure automated device or application responses in a first building system and at least one enterprise application, comprising providing a process management software module and configuring it to communicate with said first building system and said enterprise application, configuring said process management software module to be accessible over the internet by end users, configuring said process management software module to allow users to configure automated building system and enterprise application response processes using input from said first building system and said enterprise application, wherein the building systems and enterprise applications are selected from the group consisting of HVAC, fire control, access control, video monitoring, public address, exterior lighting and signage, interior public and private space lighting, escalator and elevator systems, telephone systems, room scheduling/reservation systems, resource management systems, data and information technology systems, document management systems, e-mail servers, fax servers, SMS servers, building tenant/visitor database systems, and customer relationship/management systems.

8. A computer-implemented method for allowing a user to configure automated device or application responses in a first building system and at least one enterprise application, comprising providing a process management software module and configuring it to communicate with said first building system and said enterprise application, configuring said process management software module to be accessible over the internet by end users, configuring said process management software module to allow users to configure automated building system and enterprise application response processes using input from said first building system and said enterprise application, wherein the responses of building systems and enterprise applications are process driven and the response process is editable by the user.

9. A method according to claim 8 wherein the user can add or delete steps to the response process using a process design interface.

10. A computer-implemented method for allowing a user to configure automated device or application responses in a first building system and at least one enterprise application, comprising providing a process management software module and configuring it to communicate with said first building system and said enterprise application, configuring said process management software module to be accessible over the internet by end users, configuring said process management software module to allow users to configure automated building system and enterprise application response processes using input from said first building system and said enterprise application, wherein a different process may be configured for every device in a building system.

11. A computer-implemented method for allowing a user to configure automated device or application responses in a first building system and at least one enterprise application, comprising providing a process management software module and configuring it to communicate with said first building system and said enterprise application, configuring said process management software module to be accessible over the internet by end users, configuring said process management software module to allow users to configure automated building system and enterprise application response processes using input from said first building system and said enterprise application, wherein a first user may configure a first building system to respond in a first user's tenant space to a first event according to a first process and a second user may configure said first building system to respond in a second user's tenant space to said first event according to a second process.

12. A computer-implemented method for allowing a user to configure automated device or application responses in a first building system and at least one enterprise application, comprising providing a process management software module and configuring it to communicate with said first building system and said enterprise application, configuring said process management software module to be accessible over the internet by end users, configuring said process management software module to allow users to configure automated building system and enterprise application response processes using input from said first building system and said enterprise application, wherein the process management software module receives input from both value-oriented systems and transaction-oriented systems.

13. A computer-implemented method for allowing a user to configure automated device or application responses in a first building system and at least one enterprise application, comprising providing a process management software module and configuring it to communicate with said first building system and said enterprise application, configuring said process management software module to be accessible over the internet by end users, configuring said process management software module to allow users to configure automated building system and enterprise application response processes using input from said first building system and said enterprise application, wherein an end user may configure and customize building system response systems using input from both value-oriented and transaction-oriented systems.

14. A computer-implemented method for allowing a user to configure automated device or application responses in a first building system and at least one enterprise application, comprising providing a process management software module and configuring it to communicate with said first building system and said enterprise application, configuring said process management software module to be accessible over the internet by end users, configuring said process management software module to allow users to configure automated building system and enterprise application response processes using input from said first building system and said enterprise application, wherein the process management software module receives input from both building systems and enterprise applications.

15. A computer-implemented method for allowing a user to configure automated device or application responses in a first building system and at least one enterprise application, comprising providing a process management software module and configuring it to communicate with said first building system and said enterprise application, configuring said process management software module to be accessible over the internet by end users, configuring said process management software module to allow users to configure automated building system and enterprise application response processes using input from said first building system and said enterprise application, wherein the process management software module issues commands to building systems and enterprise applications.