

(19) World Intellectual Property  
Organization  
International Bureau



(43) International Publication Date  
10 February 2005 (10.02.2005)

PCT

(10) International Publication Number  
**WO 2005/013075 A2**

- (51) International Patent Classification<sup>7</sup>: **G06F**
- (21) International Application Number:  
PCT/US2004/024323
- (22) International Filing Date: 28 July 2004 (28.07.2004)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
10/632,328 1 August 2003 (01.08.2003) US
- (71) Applicant (for all designated States except US): **IDX INVESTMENT CORPORATION** [US/US]; 1400 Shelburne Road, South Burlington, VT 05403 (US).

- (72) Inventors; and
- (75) Inventors/Applicants (for US only): **FERGUSON, John, R.** [US/US]; 62 Nashoba Road, Concord, MA 01742

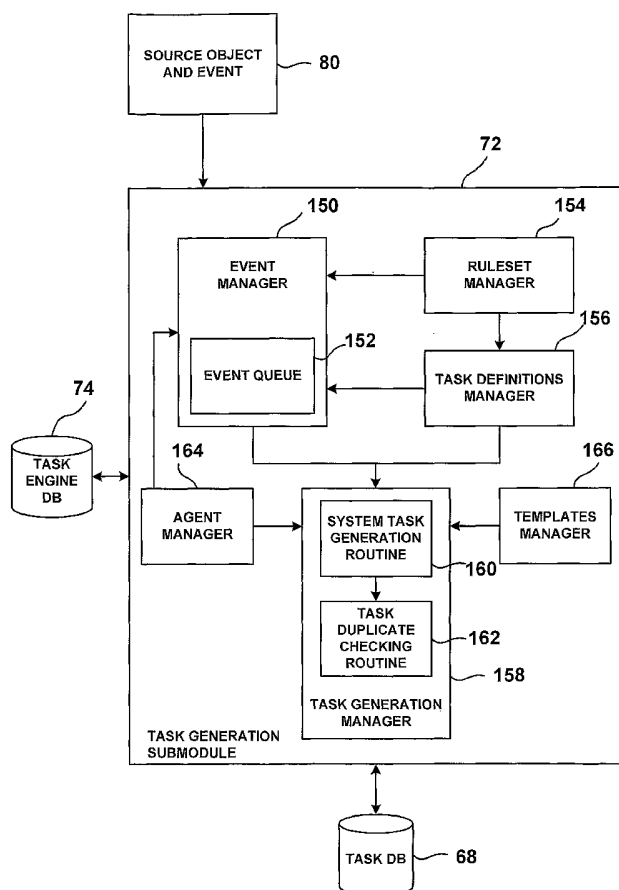
(US). **BRADLEY, Shannon** [US/US]; 224 Lake Street, Apt.1, Brighton, MA 02135 (US). **CANZANO, Stephen, A.** [US/US]; 9 Orchard Road, Shrewsbury, MA 01545 (US). **MCKENNA, Sean, T.** [US/US]; 45 Birchland Avenue, West Roxbury, MA 02132 (US). **TICE, James, R.** [US/US]; 35 Eldridge Road #206, Jamaica Plain, MA 02130 (US).

(74) Agent: **MEIER, Lawrence, H.**; Downs Rachlin Martin PLLC, 199 Main Street, P.O. Box 190, Burlington, VT 05402-0190 (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, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM,

[Continued on next page]

(54) Title: ENTERPRISE TASK MANAGER



(57) Abstract: An enterprise task management system (24) for managing data maintenance task across multiple applications. System (24) includes a task manager module (60) and a task engine module (62). Task manager module (60) includes two primary graphical user interfaces: a user interface (64) and a system administrator interface (66). User interface (64) includes an enterprise task list (22), a preview text box (96), and an instruction text box (100). Task engine module (62) includes the following sub-modules: ETM system setup (70) and task generation (72). Another aspect of the present invention is a system architecture for centrally managing the creation of tasks. Additional aspects of the present invention include a method of generating data maintenance tasks within an enterprise information system, a method of populating task data field using task templates, and a method of automatically generating and performing tasks within system (24).



TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**(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, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI,

**Published:**

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

*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.*

## ENTERPRISE TASK MANAGER

## FIELD OF THE INVENTION

[0001] The present invention relates generally to a task management system and more particularly relates to a task management system for centrally managing tasks across an entire enterprise.

## BACKGROUND

[0002] The rising cost of providing healthcare services has increased the consolidation of healthcare service providers and thus the size of integrated delivery networks (IDN). Management of large IDNs poses unique organizational problems. The result is often increased revenue cycles with respect to patient billing and poorer customer service. Existing task management structures are often not flexible enough to effectively handle the dynamic changes that occur on an on-going basis in large IDNs.

[0003] In order to manage the volume of data in IDNs, healthcare provider organizations have turned to sophisticated data management systems or applications. Typically, these systems include modules for maintaining or improving the quality of data. By improving the quality of data, i.e., correcting errors and or omissions in data, the revenue cycle for patient billing may be reduced and the overall service to customers may be improved. Generally, for each healthcare service provider in the IDN, a task list is developed and populated with tasks for improving data specific to that particular provider. A user such as a data clerk is often responsible for working the tasks that populate the task lists.

[0004] For large IDNs, the tasks that populate the task lists may include tasks input by users of the system, tasks input by third party systems, and tasks generated by the data management system itself. Prior art data management systems typically are not capable of integrating tasks from across an entire enterprise or IDN. Rather, prior art systems typically include a plurality of task lists and require users to “jump” from one task list to another to work tasks. In addition, prior art systems including multiple task lists do not allow a system administrator to easily manage all of the tasks in the system from a single interface. As a result, tasks are sometimes

duplicated, lost, incorrectly entered, and incorrectly worked. Generally, system resources are often not efficiently used.

#### SUMMARY OF THE INVENTION

**[0005]** One aspect of the present invention is an enterprise information system for managing data maintenance tasks across multiple applications within an enterprise. The system includes a task manager module having an enterprise task list, the enterprise task list including tasks generated outside the system at the multiple applications and tasks generated within the system, and a task engine module adapted to create tasks, the task engine module in communication with the task manager module.

**[0006]** Another aspect of the present invention is a method of generating data maintenance tasks within an enterprise information system, the method including the following steps: filing a source object and event to the system; determining whether the event is a custom event or a system event; adding all system events to an event queue; determining for each of the custom events whether a corresponding custom event ruleset is true; adding each of the custom events where the corresponding custom event ruleset is true to the event queue; determining for each of the custom or system events in the event queue whether a task definition corresponding to each of the custom or system events exists; determining for each task definition corresponding to each of the custom or system events whether a task definition ruleset is true; and generating a task for each of the custom or system events having a task definition ruleset that is true.

**[0007]** Still another aspect of the present invention is a graphical user interface for use in an enterprise information system for managing data maintenance tasks across multiple applications. The graphical user interface includes a first portion having an enterprise task list. The enterprise task list includes tasks generated outside the system at the multiple applications and tasks generated within the system.

**[0008]** Another aspect of the present invention is a system architecture for centrally managing the creation of tasks. The system architecture includes the following: a source object layer including one or more source objects, each of the one or more source objects having a

corresponding event; a target object layer including one or more target objects; and an enterprise task manager layer between said source object layer and said target object layer, wherein said enterprise task manager layer centrally manages relationships between each of the one or more source objects and each of the one or more target objects.

[0009] Still another aspect of the present invention is a method of populating task data fields using task templates. The method includes the following steps: populating each of the task data fields that are empty with data from corresponding data fields in a task definition template; populating each of the task data fields that are empty with data from corresponding data fields in a task name template; populating each of the task data fields that are empty with data from corresponding data fields in a source object template; and populating each of the task data fields that are empty with data from corresponding data fields in a system template.

[0010] Yet another aspect of the present invention is a method of automatically generating and performing tasks within an enterprise task management system. The method includes the following steps: providing an agent including a predetermined ruleset; comparing data against the predetermined rules of the agent; and causing the system to generate tasks according to the predetermined ruleset of the agent.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0011] For the purpose of illustrating the invention, the drawings show a form of the invention that is presently preferred. However, it should be understood that the present invention is not limited to the precise arrangements and instrumentalities shown in the drawings, wherein:

FIG. 1 is a simplified schematic diagram of a prior art configuration for accessing tasks created by a plurality of providers in an IDN;

FIG. 2 is a simplified schematic diagram of a configuration for accessing tasks created by a plurality of providers in an IDN according to one embodiment of the present invention;

FIG. 3 is a simplified schematic diagram of a computer system for accessing the ETM system according to one embodiment of the present invention;

FIG. 4 is a schematic diagram of the enterprise task manager system according to one embodiment of the present invention;

FIG. 5 is a schematic diagram of a graphical user interface according to one embodiment of the present invention;

FIG. 6 is a schematic diagram of a graphical user interface according to one embodiment of the present invention;

FIG. 7 is a flow chart of steps taken by a user when using a graphical user interface according to FIGS. 5 or 6;

FIG. 8 is a schematic diagram of a task generation sub-module according to one embodiment of the present invention;

FIG. 9 is a flow chart of a process for generating tasks by the system according to one embodiment of the present invention;

FIG. 10 is a flow chart of a process for creating tasks using agents according to one embodiment of the present invention;

FIG. 11 is a schematic of templates used to populate new task data tables according to one embodiment of the present invention;

FIG. 12 is a flow chart of a process for populating task data fields using templates according to one embodiment of the present invention;

FIG. 13 is a simplified schematic diagram of a prior art task management system architecture;

FIG. 14 is a block diagram of a prior art process for adding new objects to the architecture illustrated in FIG. 13;

FIG. 15 is a simplified schematic diagram of a task management system architecture according to one embodiment of the present invention; and

FIG. 16 is a block diagram of a process for adding new objects to the architecture illustrated in FIG. 3.

#### DETAILED DESCRIPTION OF THE INVENTION

**[0012]** The present invention is directed to an enterprise task management system for managing data maintenance tasks across multiple applications. The present invention includes a system having an enterprise task list, a graphical user interface, an architecture, and processes for creating and managing tasks across an enterprise such as an IDN. Examples of the disclosed invention are depicted in FIGS. 1-16, although it should be understood that the present invention is not limited to this (or any other) particular embodiment, but rather is intended to cover all systems, graphical user interfaces, architectures, and processes that fairly fall within the broad scope of the appended claims.

**[0013]** Turning to the drawings, wherein like reference numerals refer to like elements, FIG. 1 illustrates how a user or system administrator accesses data maintenance task lists in a prior art data management system. Typically, a user or system administrator 10 has to access a separate data management system for each healthcare service provider in the IDN, i.e., “HOSPITAL #1” 12, “LAB” 14, and “HOSPITAL #2” 16, to view its respective task list, i.e., task lists 13, 15, and 17.

**[0014]** FIG. 2 illustrates how a user or system administrator accesses data maintenance task lists in the enterprise task management (ETM) system of the present invention. In the ETM

system of the present invention, a user or system administrator 20 only has to access a single task list, i.e., an enterprise task list 22, in an ETM system 24, which are both described in greater detail below, to view all tasks in the IDN. Data maintenance tasks related to each of the healthcare service provider in the IDN, i.e., "HOSPITAL #1" 25, "LAB" 26, and "HOSPITAL #2" 27, are maintained on enterprise task list 22.

**[0015]** Referring to FIG. 3, ETM system 24 of the present invention is illustrated as being implemented in an appropriate computing environment. Although not required, the invention will be described generally in terms of computer-executable instructions, such as program modules, being executed by a conventional, general purpose, digital computer. Typically, program modules include routines, programs, objects, components, data structures, etc. that perform specific tasks. The invention may be practiced with a variety of computer system configurations, including networked client-server computing systems, hand-held devices, programmable consumer electronics, minicomputers, mainframe computers, and the like. The invention will typically, but not necessarily, be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network, e.g., a LAN, WAN or an Internet-based network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

**[0016]** It is contemplated that the system of the present invention will operate in a networked computing environment including a personal computer 30 and an Internet server 42. Personal computer 30 includes a computer central processing unit (CPU) 32, a computer memory 34, and input/output devices 36. Typically, ETM system computer programs 38 reside in a server memory 40 of an Internet server 42. Of course, ETM system computer programs 38 may reside in the memory of a local or wide area network server or in the memory of an individual desktop computer system. Typically, ETM system computer programs 38 may be accessed via a thin client access program 44 that resides in computer memory 34 of computer system 30. In such a configuration, ETM system computer programs 38 may be executed on either or both an Internet server CPU 46 and computer CPU 32. Input/output devices 36 typically include a storage

device 48, such as a hard disk drive, a keyboard 50, a pointing device 52, i.e., a mouse, and a display device 54, such as a monitor.

[0017] Referring now to FIG. 4, one embodiment of the enterprise task manager system (may be referred to herein as “ETM system”, “system”, or “ETM”) is illustrated. Dashed lines in FIG. 4 enclose ETM system 24 and its various components. ETM system 24 includes two primary modules, a task manager module 60 and a task engine module 62. Task manager module 60 includes two primary interfaces, a user interface 64 and a system administrator interface 66. Both user interface 64 and system administrator interface 66 are typically graphical user interfaces (GUI) that are displayed to a user or system administrator via a display device 54. Accordingly, user interface 64 and system administrator interface 66 may be referred to herein as GUI 64 and GUI 66, respectively. User interface 64 includes enterprise task list 22, which was described above with respect to FIG. 2. Task manager module 60 is supported by and in communication with a task database 68. Task engine module 62 primarily includes two sub-modules: ETM system setup 70 and task generation 72. Task engine module 62 is supported by and in communication with a task engine database 74.

[0018] As described above, enterprise task list 22 serves as a central repository for tasks (not shown) generated across an IDN or similar enterprise. The tasks included on enterprise task list 22 are stored in task database 68. Enterprise task list 22, which is part of user interface 64 within task manager module 60, may be populated with tasks 76 that are directly entered into task database 68 or tasks 78 that are indirectly entered into task database 68 via task generation sub-module 72 of ETM system 24.

[0019] Tasks 76 that are directly entered into task database 68 are typically entered by a user, system administrator, and or 3<sup>rd</sup> party data maintenance system. Tasks 76 may also be directly entered into task database 68 by ETM system 24 by an agent. As described in greater detail below, tasks 78 are developed by task generation sub-module 72 in response to interaction between ETM system 24 and any one or all of a user or system administrator, a 3<sup>rd</sup> party system, or by ETM system 24 itself through the use of agents or the like. Interaction with ETM system 24 causes a source object and associated event 80 to be filed to task generation sub-

module 72 of ETM system 24. Objects are components of a system that might require tasks. For example, in the context of a healthcare IDN, a patient, an invoice, and an appointment may all be objects. Typically, there are two types of objects: source and target. A source object is an object that files to the system and may trigger a task. For example, if a claim form edit object files to the system, it may trigger a task to update a patient's registration information. A target object is the object that needs attention for the task. For example, if a task instructs a user to update a patient's registration information, the target object is Patient. Events are the actions that occur when source objects file to the system. For example, when you edit a patient record and file, the source object is Patient and the event is Edit. Events are typically referred to as "statefull" in that they are surrounded by data from a source object. In contrast, a task is generally referred to as "stateless" because it begins as an empty shell that is populated with data based on the source object and event. Each time a source object files, either manually by a user or automatically by the system, task generation sub-module 72 evaluates the contents of the source object and associated event 80 and determines whether to generate and store a task in task database 68.

**[0020]** In use, a user primarily interacts with ETM system 24 via task manager module 60. Typically, a user accesses task manager module 60 via user interface 64 to view and work tasks that are included in enterprise task list 22. Task manager module 60 allows a user to create specific query-based views of enterprise task list 22, work tasks according to pre-determined work options, create/edit/view/remove tasks from enterprise task list 22, re-assign ownership of tasks to other users, change task status, etc. Depending on a user's skill and their level of access to the system, a user may access task engine module 62 to use ETM setup sub-module 70 for the creation of new tasks.

**[0021]** A system administrator typically interacts frequently with ETM system 24 via both task manager module 60 and task engine module 62. A system administrator accesses system manager module 60 via system administrator interface 66 and accesses task engine module 62 via ETM setup sub-module 70. A system administrator typically uses task manager module 60 to monitor productivity and task activity, create enterprise views to be used when viewing enterprise task list 22 as explained further below, and maintain overall system security. A system administrator typically accesses task engine module 62 to provide services to define

tasks, define rules that govern the generation, execution, escalation, and completion of tasks, and create agents which operate within the system to automatically create or perform tasks.

[0022] As described above, a user typically interacts with task manager module 60 of ETM system 24 to view and work tasks included on enterprise task list 22. As also described above, enterprise task list 22 is displayed to a user via GUI 64. Referring now to FIGS. 5-7, GUI 64 includes a first portion 90 and a second portion 92. First portion 90 includes enterprise task list 22. Second portion 92 includes a preview tab 94 and corresponding preview text box 96 and an instructions tab 98 and corresponding instructions text box 100 (shown in FIG. 6). Of course, in other embodiments, additional tabs may be included. In addition, any type of data, link, or instruction may be included in each tab included. A plurality of buttons facilitates user interaction with enterprise task list 22. Edit button 102 allows a user to edit a selected task 104. In FIGS. 5 and 6, only one task, selected task 104, is selected as denoted by the "X" in task selection column 106 to the left of "Task Name" column 107. However, a user may select a plurality of tasks for viewing and or working by marking the box in task selection column 106 for each task desired. New button 108 allows a user to manually add a new task to enterprise task list 22. Actions button 110 allows a user to select actions from a dropdown menu (not shown) having predefined actions for working common tasks. Examples of predefined actions include but are not limited to hiding preview or instructions tabs 94 and 98, viewing selected tasks, manually marking tasks as done, reassigning tasks, and adding notes to tasks.

[0023] Enterprise task list 22 generally includes a plurality of data columns 112 that include data related to a particular task. The particular data columns included in data columns 112 are determined according to a particularly selected view 114 that is selected from a view dropdown box 116. Particularly selected view 114 causes system 24 to run a query against task database 68. The tasks that match the view query are displayed on enterprise task list 22. In addition to data from task database 68, data from sources outside of task database 68, e.g., data from a target object, may be included in enterprise task list 22. As mentioned above, a system administrator, using task engine module 62, creates the views that populate view dropdown box 116. A user may typically access any views included in view dropdown box 116. After selecting a view, a user may alter selected view 114 by selecting a particular column header 118

thereby sorting enterprise task list 22 according to the data contained in the column beneath the selected header column 118.

**[0024]** Preview tab 94 and corresponding preview text box 96 allow a user to view additional data related to any of the tasks in enterprise task list 22. Typically, a user highlights a particular task using pointing device 52 to designate it as the presently selected task (not shown). Data related to the presently selected task populates preview text box 96. In this way, a significant amount of data may be displayed with respect to each task in enterprise task list 22.

**[0025]** To begin working a task, a user may select start task button 120 or select instructions tab 98. Selection of start task button 120 or instructions tab 98 causes the instructions tab to be selected thereby simultaneously displaying instructions text box 100 (as illustrated in FIG. 6) and hiding preview text box 96. Instructions text box 100 typically includes a list of options 122 for working the presently selected task. In FIG. 6, list of options 122 is numbered and arranged in a particular order. However, in other embodiments, list of options 122 may not be numbered or arranged in any particular order. In addition, in some instances, a task may only be informational and no instructions will be provided.

**[0026]** Referring now to FIG. 7, GUI 64 is arranged in such a manner as to facilitate efficient interaction by a user with enterprise task list 22. Typically, a user first selects a particular enterprise task list view from view dropdown box 116 at step 130. Next, at step 132, a user decides whether to sort particularly selected view 114 of enterprise task list 22 according to one of column headers 118. If the user decides to sort enterprise task list 22, the list is refreshed and sorted according to the column header 118 selected at step 134. If the user does not choose to sort enterprise task list 22, the process continues at step 136 where the user selects one or more tasks to view or work. Referring again to step 134, after enterprise task list 22 is sorted, the process continues at step 136 where the user selects one or more tasks to view or work. After selecting tasks to view or work, at step 138 the user determines whether to work one or more tasks or preview additional task information. If a user decides to work one or more tasks, at step 140 the user selects start task button 120 thereby causing instructions tab 98 and instructions text box 100 to be displayed. Alternatively, the user may select preview tab 94 at step 138 in

order to preview additional information about the one or more tasks selected. If preview tab 94 is selected, preview tab 94 and preview text box 96 are displayed. Although not illustrated in FIGS. 5-7, it is contemplated in the present invention that in addition to sorting tasks, the tasks in enterprise task list 22 may be filtered and or arranged in groups or categories.

**[0027]** A system administrator accesses task manager module 60 using GUI 66. GUI 66 generally includes all the same elements as GUI 64. However, GUI 66 will have added functionality that allows a system administrator to make enterprise-wide modifications to tasks, views, security levels, and other system parameters. Therefore, GUI 66 likely will differ from GUI 64 to include additional frames, buttons, and or other portions that facilitate a system administrator's increased functionality over a user.

**[0028]** As described above, task engine module 62 primarily includes two sub-modules: ETM setup 70 and task generation 72. A system administrator typically uses ETM setup sub-module 70 to configure the parameters of ETM system 24. A system administrator uses ETM setup sub-module 70 to define task names and their preview/instruction components, to define events, to define task definitions, to define task templates, to define agents, and to define views, roles, and users for a task.

**[0029]** Task generation sub-module 72 is utilized by ETM system 24 to generate tasks for inclusion in task engine database 68 and enterprise task list 22. Referring now to FIG. 8, task creation sub-module 72 includes a plurality of manager sub-modules that interact to generate tasks: an event manager 150 having an event queue 152; a ruleset manager 154; a task definitions manager 156; a task generation manager 158 having a system task generation routine 160 and a task duplicate checking routine 162; an agent manager 164; and a templates manager 166.

**[0030]** Ruleset manager 154 supplies ETM rulesets (not shown) to both event manager 150 and task definitions manager 156. Task definitions manager 156 supplies task definitions to event manager 150. Information is sent from event manager 150 and task definitions manager 156 to task generation manager 158 where system task generation routine 160 generates tasks using data supplied by templates manager 166. Templates are described in detail below with reference

to FIGS. 11 and 12. Alternatively, information may be sent from agent manager 164 to task generation manager 158 instead of from event manager 150 and task definitions manager 156. In addition, agent manager 164 may place events on event queue 152. Before sending tasks to task database 68, tasks generated by system task generation routine 160 are checked by task duplicate checking routine 162 to determine if identical tasks exist in the system. Acceptable tasks are stored in task database 68 for viewing in enterprise task list 22. The data that populates the managers in task generation sub-module 72 is provided by task engine database 74, which is generally populated based on setup input from a system administrator. Much of the data stored in task engine database 74 is grouped in dictionaries. For example, one dictionary may exist for task names and another for views.

**[0031]** Tasks generated by task generation sub-module 72 of ETM system 24 are devised through a specific process. Referring now to FIG. 9, the process in which system 24 generates tasks is illustrated. At step 170, a source object is filed to the system. As explained above, the source object is paired with a corresponding event. At step 172, it is determined whether the corresponding event is a system event or a custom event. System events are default events provided by system 24. In contrast, a user or system administrator defines custom events.

**[0032]** If the event is a system event, the system event is added to event queue 152 at step 174. If the event is a custom event, at step 176 it is determined whether the custom event ruleset, i.e., an ETM ruleset from ruleset manager 154, accompanying the custom event is true. ETM rulesets are rules that determine when to create custom events or when to create and review tasks. Creation rulesets define the conditions that create custom events and tasks. Review rulesets define the conditions under which the system reevaluates a task in enterprise task list 22 to determine if it still needs to be worked. If the custom event ruleset is not true, the process ends. If the custom event ruleset is true, the process continues at step 178 where the custom event is added to event queue 152.

**[0033]** Next, at step 180, it is determined for each event in event queue 152 whether a task definition exists for the particular event and related source object. A task definition links together the object, event, task name or task reference, and ETM ruleset. Task names in a task

definition may explicitly refer to a task name in a particular task name dictionary or provide a column reference to a particular task name dictionary. A system administrator typically defines task definitions in ETM setup sub-module 70 of task engine module 62. If no task definition exists, the process ends. If a task definition exists, the process continues at step 182 where it is determined whether the task definition ruleset, i.e., an ETM ruleset from ruleset manager 154, is true. If the task definition ruleset is not true, the process ends. If the task definition ruleset is true, the process continues at step 184 where a task is generated by task generation sub-module 72.

**[0034]** Next, at step 186, it is determined whether the task generated at step 184 is a duplicate, i.e., is already on file or already exists in task database 68. If the task generated at step 184 is found to already exist, the status of the previously generated task is reviewed at step 188. The status of the previously generated task is compared against a predetermined duplicate checking ruleset (not shown) at step 190. If the duplicate checking ruleset is not true, the process ends. If the duplicate checking ruleset is true, a new task may be generated to overwrite the original task (retain history of original task), the original task may be retained and no new task generated, the original task may be deleted and the new task added (lose history of original task), a duplicate task may be added, or the data fields of the new task may be merged with the data fields of the original task. At step 192, the new or modified task is added to task database 68. Referring to step 186, if it is found that the task generated in step 184 is not a duplicate, the newly generated task will be added to task database 68 of task manager module 60 at step 192.

**[0035]** As mentioned above, ETM system 24 may also create tasks via the use of an agent. Another embodiment of the present invention is a method of automatically generating tasks within an enterprise task management system. Referring to FIG. 10, at step 200, an agent having a predetermined ruleset is provided. Next, at step 202 and 203, a source object filed to system 24 and data in task database 68 and task engine database 74 are compared to the agent ruleset. Then, at step 204, it is determined whether the agent ruleset is true. If the agent ruleset is true, a task is generated at step 206 and added to task database 68. If the ruleset is not true, the process ends.

[0036] Some agents may merely cause an event to be placed in event queue 152. Other agents will by-pass event queue 152 and cause tasks to be generated and added to task database 68. Other agents may also include a predetermined agent workflow that allows the system to automatically work the task. For example, an agent may be created that automatically updates certain fields of data within task database 68. The user may initiate application of agent rulesets via queries. Alternatively, agents may be scheduled to operate on a predetermined basis or because of certain events within the system.

[0037] Another embodiment of the present invention includes a method of populating task data fields using templates. Templates are placeholders for pointers to default data that system 24 requires to create a task. For example, when system 24 creates a task, it needs to know the date on which the task should be started or reviewed, and the task's priority. System 24 looks to the various task templates for this and other required data. Templates exist at the system, object, task name, and task definition level. System 24 typically uses templates in a set order.

[0038] Referring to FIG. 11, when system 24 generates a task, a new task data table having task data fields (also referred to as task attributes) including pertinent data regarding the task must be populated before the task can be added to enterprise task list 22. FIG. 11 schematically illustrates the typical templates used to populate task data fields A1, A2, A3, and A4 of new task data table 210. In FIG. 11, the pertinent data for task data fields A1, A2, A3, and A4 of new task data table 210 are "1", "3", "6", and "10", respectively. The typical templates include task definition template 212, task name template 214, target object template 216, and system template 218. In FIG. 11, all of the templates include the same task data fields, A1, A2, A3, and A4. Of course, in other embodiments, each template may contain different task data fields. In addition, in some cases only a few or none of the task data fields in a particular template will overlap with the task data fields in other templates.

[0039] As mentioned above, system 24 typically uses templates in a set order: first using values from task definition template 212; then task name template 214; then target object template 216; and lastly system template 218. In FIG. 11, task data field A1 of new task data table 210 is populated with data from task data field A1 of task definition template 212, task data

field A2 of new task data table 210 is populated with data from task data field A2 of task name template 214, task data field A3 of new task data table 210 is populated with data from task data field A3 of target object template 216, and task data field A4 of new task data table 210 is populated with data from task data field A4 of system template 218.

**[0040]** FIG. 12 illustrates the steps for populating a new task data table using templates. First, in step 300, system 24 goes to the task definition template. Next, at step 302, for each of the task data fields that do not include data, the empty task data field(s) are populated with data from the corresponding data field, if any, in the task definition template. Then, at step 304, it is determined whether any of the task data fields remain empty. If none of the task data fields remain empty, the process ends. If any task data fields remain empty, system 24 goes to the task name template at step 306. Then at step 308, for each empty task data field, the empty task data field is populated with data from the corresponding data field, if any, in the task name template. Next, at step 310, it is determined whether any empty task data fields remain. If no empty task data fields remain, the process ends. If any task data fields remain, the process continues at step 312. At step 312, system 24 goes to the source object template. Then, at step 314, for each remaining empty task data field, the empty task data field is populated with data from the corresponding data field, if any, in the source object template. At step 316, it is then determined whether the task includes any remaining empty data fields. If there are no remaining empty task data fields, the process ends. If any task data fields remain empty, the process continues at step 318. At step 318, system 24 goes to the system template. Then, at step 320, any empty data fields are populated with data from corresponding data fields, if any, in the system template. After completion of step 320, the process ends. Of course, in other embodiments, more or fewer templates may be included.

**[0041]** Another aspect of the present invention is the overall architecture of ETM system 24. FIGS. 13 and 14 illustrate the architecture and processes often found in prior art systems. Prior art system architecture generally includes a plurality of source objects and associated events 400, i.e., "PATIENT", "APPOINTMENT", "INVOICE", and "INSURANCE ELIGIBILITY". Each of source objects and associated events 400 may cause a task 402 to be performed on one or more target objects 404 depending on predetermined relationships between the objects.

[0042] As illustrated with respect to “INSURANCE ELIGIBILITY” in FIG. 1 and the process in FIG. 2, when a new object 406 is added, relationships 408 must be established between new object 406 and every related existing object, i.e., “PATIENT”, “APPOINTMENT”, and “INVOICE” in FIG. 1 and existing objects 1 thru n in FIG. 14.

[0043] In a typical architecture, when a new object is added to the system, it is generally designed with transactions (methods or procedures) that it supports. Existing objects wishing to integrate with new ones must define an interface for each of the new object's transactions that it wishes to support. Conversely, new object's wishing to integrate with existing ones must define the interface for each transaction it needs to support. In such an architecture, the interfaces expand geometrically because each existing object needs to interact with some or all of the transactions supported by the new object while at the same time each new object must interact with some or all of the transactions of the existing objects. Typically, these kinds of interfaces are statefull with the need to maintain the state after the transaction has been completed greatly adding to the complexity of the transaction itself.

[0044] FIGS. 15 and 16 illustrate the “hub and spoke” architecture used in ETM system 24 of the present invention. Referring to FIG. 15, a first layer 500 includes a plurality of source objects 502 and corresponding events 504. A second layer 506 includes a plurality of tasks 508 to be performed on a plurality of target objects 510. A third layer 512 is interposed between first layer 500 and second layer 506. Third layer 512 includes ETM system 24 and its various components.

[0045] Referring now to FIG. 16, the architecture illustrated in FIG. 15 simplifies the addition of new source objects within the system. A new source object is added at step 550 and ETM system 24 automatically establishes and manages relationships between the new source object and all target objects, as indicated at step 552. In contrast to the prior art architecture illustrated in FIGS. 13 and 14, the architecture of the present invention helps simplify the addition of new source objects within the system and reduces the amount of system administrator level input

required to add new source objects. In addition, the architecture of the present invention is effective whether the source objects and target objects are the same or different.

**[0046]** In the architecture of the present invention, as an object is added to the system, its events and methods (transactions) are declared to the ETM setup sub-module 70. Direct interfaces between new and existing objects can be avoided (although there may still be the need for some object interaction). Through task definitions, users determine the interaction between events occurring with source objects to tasks (actions) to be taken on target objects. Many such tasks can be completed automatically, by agents, eliminating any user interaction. The creation of such tasks and the actions taken (whether automatic or manual) are always stateless, which greatly simplifies surrounding code. The surrounding code no longer needs to concern itself with properly maintaining and/or updating the state when the transaction (method) occurs.

**[0047]** Another aspect of the present invention includes a task link table. A task link table is maintained that keeps track of all the relationships between source objects, tasks, and target objects. This information is useful to track the number of target objects that are corrected by completion of a single task. For example, when printing claims, a claim will not print if a physician's identification number is incorrect. If 50 out of 1000 claims are directed to a single physician with a faulty identification number, it is desired that only a single task be generated by the system to correct the faulty identification number. However, the task link table will let a user or system administrator know that completion of the single task actually fixes 50 objects or 50 claims.

**[0048]** Another aspect of the present invention is a diagnostic tool for monitoring the status of tasks and recording the history of work actions performed on tasks. A task-tracking attribute allows a user or system administrator to monitor the real-time status of any given task in the system. In addition, a historical register that records the actions worked on each task is a valuable learning tool that can be used to efficiently improve workflows. In connection with a historical register, a user may provide notations that explain actions taken or other pertinent information that may be helpful to a subsequent user.

[0049] The architecture of the present invention provides a flexible system for managing tasks across an enterprise. Centrally managing the relationships between source objects and target objects allows both new objects and new rulesets to be introduced to the system without requiring significant hard coding.

[0050] Enterprise task manager of the present invention includes a graphical user interfaces that provide both an end-user and a system administrator with a significant amount of information regarding tasks without requiring the user or system administrator to navigate through a complicated web of windows and menus. In addition, the preview and instruction text boxes allow a user to efficiently and accurately work tasks.

[0051] The overall structure of the enterprise task manager significantly improves the efficiency of users by creating a centrally located enterprise-wide task list that includes tasks that are manually input by users, created by the system, and created by third party applications. In this way, both the user and system administrator has access in a single location to all the tasks in the system.

[0052] For system created tasks, the method of creating tasks using templates insures that the task data fields are accurately and efficiently populated. By arranging the templates in a hierarchical order, the task data fields are populated with the most up-to-date data. The method of creating tasks using agents is an effective way to mine data stored in the enterprise task management system for any errors and to create and perform tasks to correct the data errors automatically. The use of agents is an effective way to automatically improve the quality of data stored within the enterprise task management system.

[0053] One embodiment of the present invention has been disclosed and discussed herein, although it should be understood that the present invention is not limited to this (or any other) particular embodiment. On the contrary, the present invention is intended to cover all alternatives, modifications and equivalents that may be included within the spirit and scope of the appended claims.

What is claimed is:

1. An enterprise information system for managing data maintenance tasks across multiple applications within an enterprise, the system comprising:
  - a task manager module having an enterprise task list, said enterprise task list including tasks generated outside the system at the multiple applications and tasks generated within the system; and
  - a task engine module adapted to create tasks, said task engine module in communication with said task manager module.
2. An enterprise information system according to claim 1, wherein said task manager module further comprises predetermined task instructions for working each of the tasks.
3. An enterprise information system according to claim 1, wherein said enterprise task list includes a plurality of data fields and said task manager module is adapted to sort said enterprise task list according to any one or more of said data fields.
4. An enterprise information system according to claim 1, wherein tasks created within said task engine module are communicated to said task manager module for inclusion in said enterprise task list.
5. An enterprise information system according to claim 4, wherein tasks created by said task engine module are automatically worked by the system.
6. An enterprise information system according to claim 4, wherein tasks created by said task engine module are presented to be worked by a user.
7. An enterprise information system according to claim 1, wherein said task engine module automatically creates tasks according to defined rules.
8. An enterprise information system according to claim 7, wherein said task engine module mines data across the enterprise information system according to said defined rules to identify

missing or incorrect data and automatically creates tasks for correcting the missing or incorrect data.

9. An enterprise information system according to claim 2, wherein each of said predetermined task instructions includes one or more links to sites outside of the system.

10. A method of generating data maintenance tasks within an enterprise information system, comprising the steps of:

- filing a source object and event to the system;
- determining whether said event is a custom event or a system event;
- adding all system events to an event queue;
- determining for each of said custom events whether a corresponding custom event ruleset is true;
- adding each of said custom events where said corresponding custom event ruleset is true to said event queue;
- determining for each of said custom or system events in said event queue whether a task definition corresponding to each of said custom or system events exists;
- determining for each task definition corresponding to each of said custom or system events whether a task definition ruleset is true; and
- generating a task for each of said custom or system events having a task definition ruleset that is true.

11. A method according to claim 10, further comprising the step of adding said task to an enterprise task list.

12. A method according to claim 11, further comprising the step of determining whether said task is a duplicate task.

13. A method according to claim 10, wherein at least one of said tasks generated is worked by the system.

14. A method according to claim 10, wherein at least one of said tasks generated is presented to be worked by a user.
15. A method according to claim 10, wherein each of said tasks generated includes a predetermined set of instructions for working each of said tasks.
16. A graphical user interface for use in an enterprise information system for managing data maintenance tasks across multiple applications, the graphical user interface comprising:
  - a first portion including an enterprise task list, said enterprise task list including tasks generated outside the system at the multiple applications and tasks generated within the system.
17. A graphical user interface according to claim 16, further comprising:
  - a second portion including information regarding each of the tasks; and
  - a third portion including predetermined instructions for working each of the tasks.
18. A graphical user interface according to claim 17, wherein said predetermined instructions include one or more links to websites outside of the system.
19. A graphical user interface according to claim 17, wherein said predetermined instructions for working each of the tasks is sorted according to a predetermined workflow.
20. A graphical user interface according to claim 17, wherein said first, second, and third portions are included in first, second, and third frames, respectively.
21. A graphical user interface according to claim 17, wherein said second and third portions are included in layered, tabbed frames.
22. A system architecture for centrally managing the creation of tasks comprising:
  - a source object layer including one or more source objects, each of said one or more source objects having a corresponding event;
  - a target object layer including one or more target objects; and

an enterprise task manager layer between said source object layer and said target object layer, wherein said enterprise task manager layer centrally manages relationships between each of said one or more source objects and each of said one or more target objects.

23. A system architecture according to claim 22, wherein said enterprise task manager further comprises a ruleset manager module including an event queue.

24. A system architecture according to claim 22, wherein said ruleset manager receives each of said one or more source objects and corresponding event, determines whether one or more task definitions exist for each of said one or more source object and corresponding event, compares each of said one or more source object and corresponding event to predetermined task definition rulesets, and generates tasks to be performed on said one or more target objects based on said predetermined task definition rulesets.

25. A system architecture according to claim 22, wherein each of said corresponding events may be one of a system event and a custom event.

26. A system architecture according to claim 25, wherein each of said system events is automatically added to said event queue.

27. A system architecture according to claim 25, wherein each of said custom events is compared to a custom event ruleset to determine whether to add said custom event to said event queue.

28. A method of populating task data fields using task templates, the method comprising the steps of:

populating each of the task data fields that are empty with data from corresponding data fields in a task definition template;

populating each of the task data fields that are empty with data from corresponding data fields in a task name template;

populating each of the task data fields that are empty with data from corresponding data fields in a source object template; and

populating each of the task data fields that are empty with data from corresponding data fields in a system template.

29. A method according to claim 28, wherein the steps are performed in the following order:

(1) populating each of the task data fields that are empty with data from corresponding data fields in a task definition template;

(2) populating each of the task data fields that are empty with data from corresponding data fields in a task name template;

(3) populating each of the task data fields that are empty with data from corresponding data fields in a source object template; and

(4) populating each of the task data fields that are empty with data from corresponding data fields in a system template.

30. A method of automatically generating and performing tasks within an enterprise task management system, the method comprising the steps of:

providing an agent including a predetermined ruleset;

comparing data against said predetermined rules of said agent; and

causing the system to generate tasks according to said predetermined ruleset of said agent.

31. A method according to claim 30, wherein said comparing step is performed at predetermined time intervals.

32. A method according to claim 30, wherein a query including said comparing step may be performed at any time.

33. A method according to claim 30, wherein said data is one or more source object and event pairs filed to the system.

34. A method according to claim 30, wherein said data is task data.
35. A method according to claim 30, further comprising the step of:  
providing an agent including a predetermined workflow for working a task; and  
causing the system to automatically work the task according to said predetermined workflow of said agent.
36. An enterprise information system for managing data maintenance tasks across multiple applications, the system comprising:  
an enterprise task list, said enterprise task list including tasks generated outside the system at the multiple applications and tasks generated within the system;  
a means for the system to create and add tasks to said enterprise task list; and  
a means for adding tasks created outside the system to said enterprise task list.
37. An enterprise information system according to claim 36, wherein said enterprise task list includes a plurality of data fields and a means for sorting said enterprise task list according to any one or more of said data fields.
38. A system architecture for centrally managing the creation of tasks comprising:  
a source object layer including one or more source objects, each of said one or more source objects having a corresponding event;  
a target object layer including one or more target objects; and  
an enterprise task manager layer between said source object layer and said target object layer, said enterprise task manager layer having a means for centrally managing relationships between each of said one or more source objects and each of said one or more target objects.
39. A computer readable signal in which is contained a program for generating data maintenance tasks within an enterprise information system, the program comprising instructions that, when executed by a computer, perform the steps of:  
filing a source object and event to the system;  
determining whether said event is a custom event or a system event;

adding all system events to an event queue;  
determining for each of said custom events whether a corresponding custom event ruleset is true;  
adding each of said custom events where said corresponding custom event ruleset is true to said event queue;  
determining for each of said custom or system events in said event queue whether a task definition corresponding to each of said custom or system events exists;  
determining for each task definition corresponding to each of said custom or system events whether a task definition ruleset is true; and  
generating a task for each of said custom or system events having a task definition ruleset that is true.

40. A computer readable signal according to claim 39, further comprising the step of adding said task to an enterprise task list.

41. A computer readable signal according to claim 40, further comprising the step of determining whether said task is a duplicate task.

42. A computer readable signal according to claim 40, wherein at least one of said tasks generated is worked by the system.

43. A computer readable signal according to claim 40, wherein at least one of said tasks generated is worked by a user.

44. A computer readable signal according to claim 40, wherein each of said tasks generated includes a predetermined set of instructions for working each of said tasks.

45. A computer readable signal in which is contained a program for populating task data fields using task templates, the program comprising instructions that, when executed by a computer, perform the steps of:

populating each of the task data fields that are empty with data from corresponding data fields in a task definition template;

populating each of the task data fields that are empty with data from corresponding data fields in a task name template;

populating each of the task data fields that are empty with data from corresponding data fields in a source object template; and

populating each of the task data fields that are empty with data from corresponding data fields in a system template.

46. A computer readable signal according to claim 45, wherein the steps are performed in the following order:

(1) populating each of the task data fields that are empty with data from corresponding data fields in a task definition template;

(2) populating each of the task data fields that are empty with data from corresponding data fields in a task name template;

(3) populating each of the task data fields that are empty with data from corresponding data fields in a source object template; and

(4) populating each of the task data fields that are empty with data from corresponding data fields in a system template.

47. A computer readable signal in which is contained a program for automatically generating and performing tasks within an enterprise task management system, the program comprising instructions that, when executed by a computer, perform the steps of:

providing an agent including a predetermined ruleset and a predetermined workflow for working a task;

comparing source object and event pairs filed to the system against said predetermined rules of said agent;

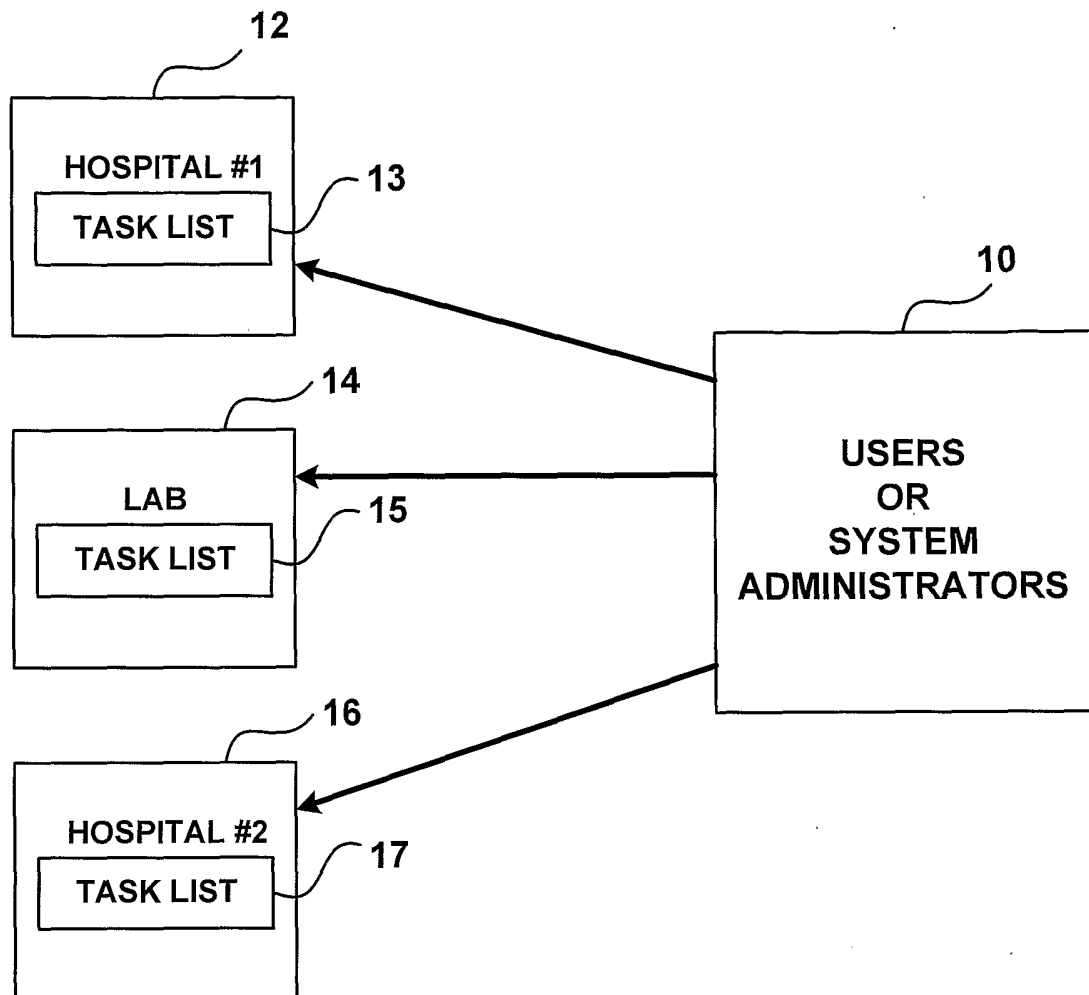
causing the system to generate tasks according to said predetermined ruleset of said agent; and

causing the system to automatically work the tasks according to said predetermined workflow of said agent.

48. A computer readable signal according to claim 47, wherein said comparing step is performed at predetermined time intervals.

49. A computer readable signal according to claim 47, wherein said comparing step may be performed at any time.

1/14



(PRIOR ART)

FIG. 1

2/14

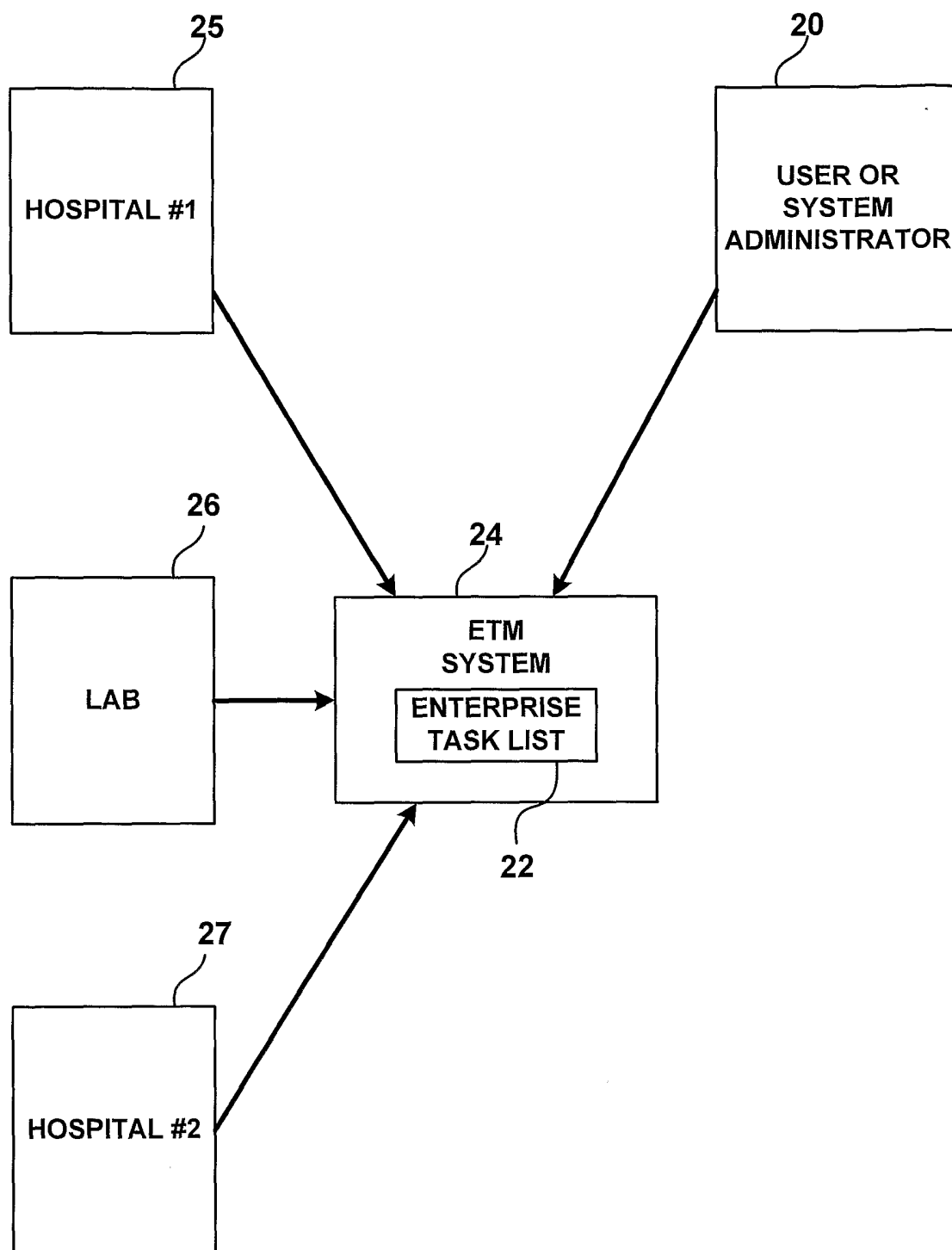


FIG. 2

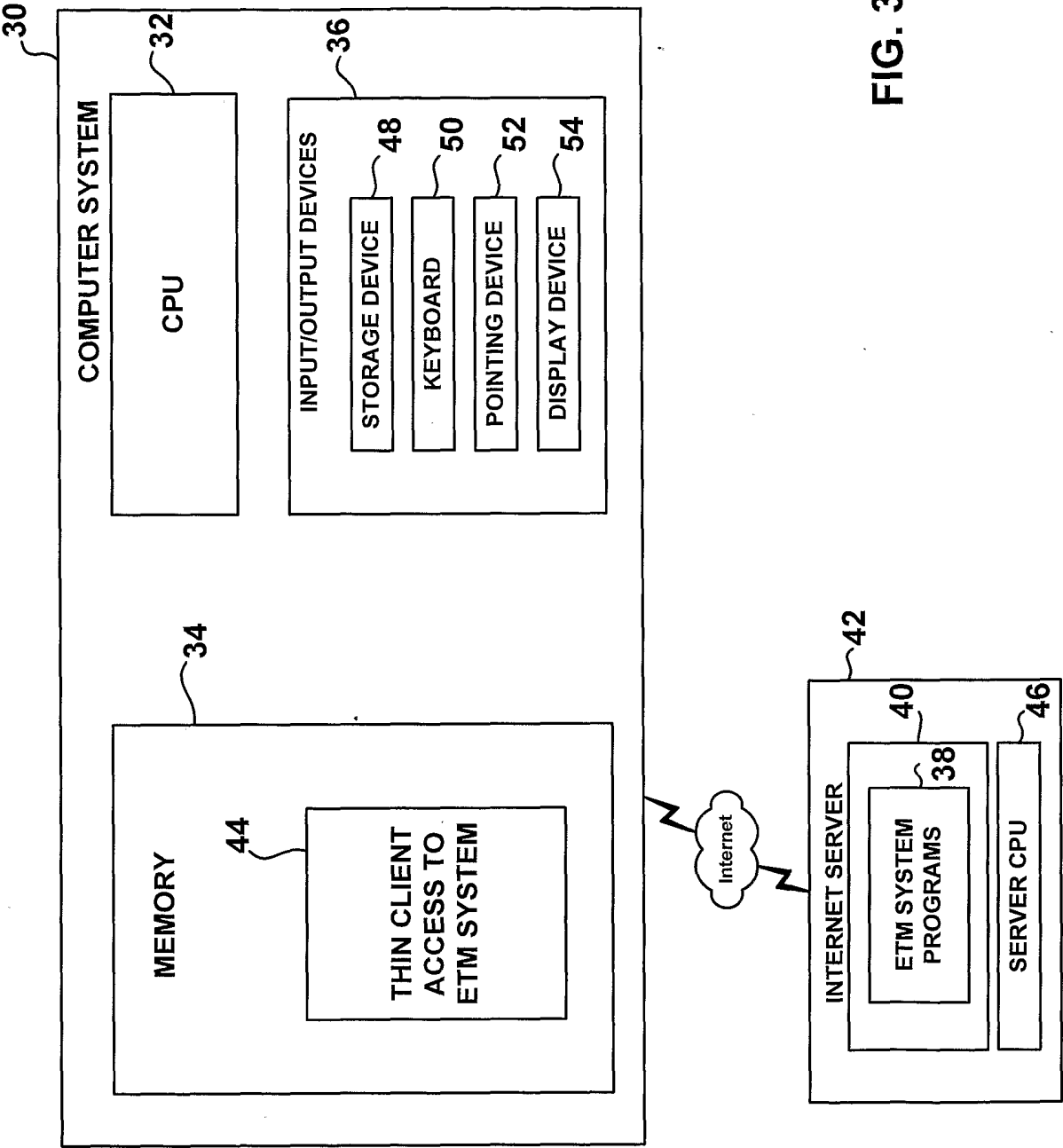


FIG. 3

4/14

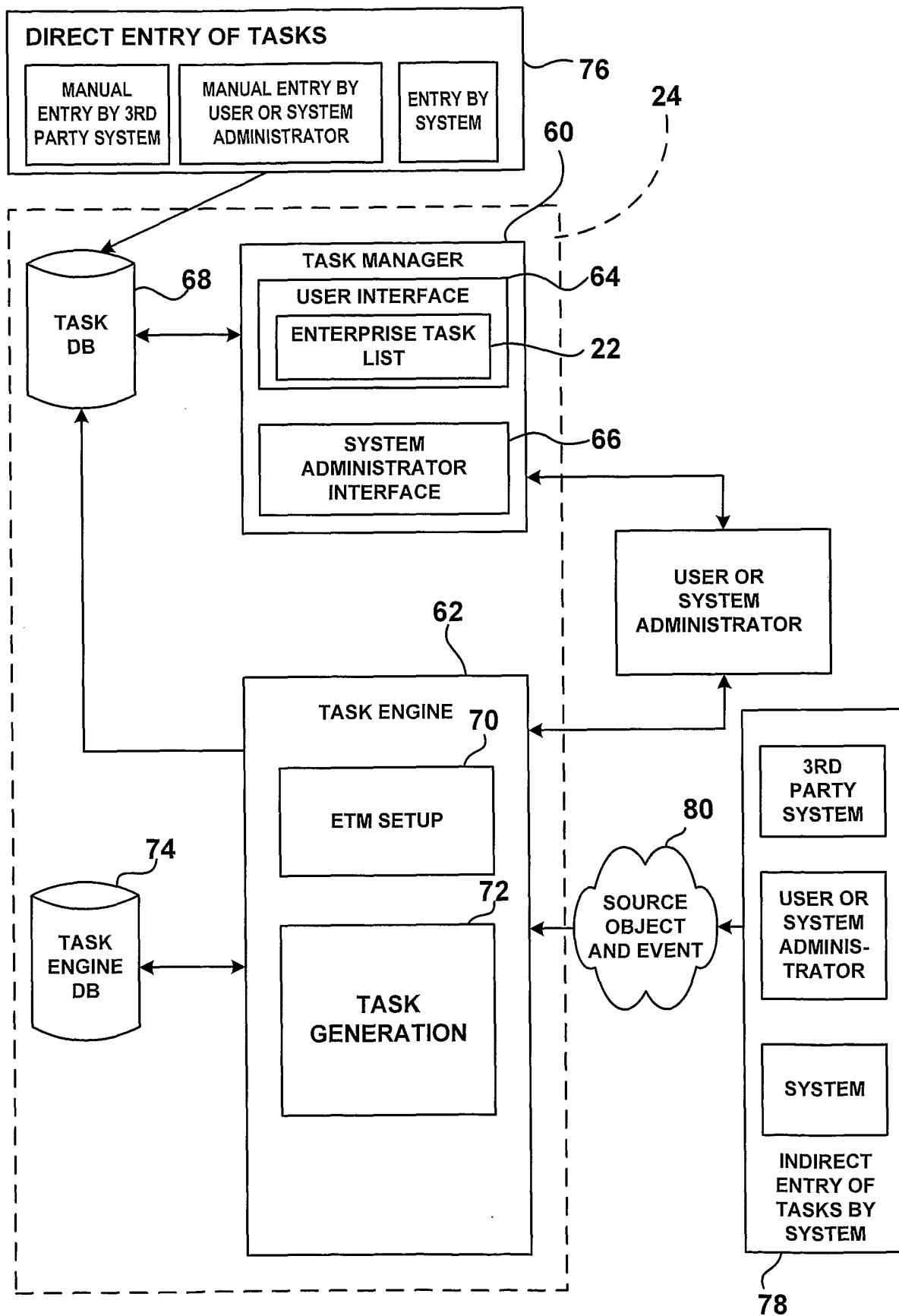


FIG. 4

5/14 92

**TASK LIST**

| 106                                 | Task Name              | Edit Message                  | Patient Name      | Inv #   | Inv Bal | Filing Date |
|-------------------------------------|------------------------|-------------------------------|-------------------|---------|---------|-------------|
| <input checked="" type="checkbox"/> | Fatal Front End edit   | F - Missing Ref Phy/Con Chg   | CONTRACT, TEST    | 8002041 | 283.00  | 09/02/2002  |
| <input type="checkbox"/>            | Fatal Front End edit   | F - Missing Dx 1 Code         | CONTRACT, TEST    | 8002041 | 283.00  | 09/02/2002  |
| <input type="checkbox"/>            | Fatal Front End edit   | F - Missing Hosp Ip/Op/Lab    | CONTRACT, TEST    | 8002041 | 283.00  | 09/02/2002  |
| <input type="checkbox"/>            | Fatal Front End edit   | F - Missing Dx 1 Code         | CONTRACT, TEST    | 8002162 | 1000.00 | 09/03/2002  |
| <input type="checkbox"/>            | FatalRegistration Edit | R - Pt Age Require Empl St... | DAVIS, LAWANDA S  | 8001372 | 40.00   | 06/15/2002  |
| <input type="checkbox"/>            | FatalRegistration Edit | R - Pt Age Require Empl St... | DAVIS, LAWANDA S  | 8001373 | 40.00   | 06/15/2002  |
| <input type="checkbox"/>            | FatalRegistration Edit | R - Pt Age Require Empl St... | RAMSEY, PAYTON M. | 8001537 | 40.00   | 06/15/2002  |
| <input type="checkbox"/>            | FatalRegistration Edit | R - Pt Age Require Empl St... | RAMSEY, PAYTON M. | 8001538 | 40.00   | 06/15/2002  |
| <input type="checkbox"/>            | FatalRegistration Edit | R - Pt Age Require Empl St... | SEXTON, TAMMY L.  | 8000943 | 29.00   | 06/12/2002  |
| <input type="checkbox"/>            | FatalRegistration Edit | R - Pt Age Require Empl St... | SIMS, SATIN       | 8001332 | 254.00  | 06/15/2002  |
| <input type="checkbox"/>            | FatalRegistration Edit | R - Pt Age Require Empl St... | SUTHERLIN, MONICA | 8000475 | 254.00  | 06/12/2002  |
| <input type="checkbox"/>            | Fatal Front End edit   | F - Missing Dx 1 Code         | TEST, HEZEKIAH    | 8001999 | 75.00   | 09/02/2002  |
| <input type="checkbox"/>            | Fatal Front End edit   | F - Missing Hosp Ip/Op/Lab    | TEST, HEZEKIAH    | 8001999 | 75.00   | 09/02/2002  |
| <input type="checkbox"/>            | FatalRegistration Edit | R - Missing Cert #            | CONTRACT, TEST    | 8002169 | 200.00  | 09/07/2002  |
| <input type="checkbox"/>            | FatalRegistration Edit | R - Missing Cert #            | FAGALA, CHARLES   | 8002096 | 0.00    | 08/27/2002  |

**View:**

**Claim Edits**

**94**

**Preview**

Fatal Front End edit  
F - Missing Ref Phy/Con Chg

CONTRACT, TEST  
SSN: 123-45-6789  
MRN: 08004744  
Sex: M DOB: 02/25/1978  
111 Mockingbird Ln

Allston, MA 02134  
Pat Phone:  
Alt Phone:

FSC: BLUE SHIELD ALLIANCE  
CHOICE  
InsCo: BLUE SHIELD MISSOURI  
Cert: 12312312  
EffDt:

Inv #: 8002041  
DOS: 03/05/2003  
Prov: WRIGHT MD, JAMES  
B/A: IMMEDIATE EDUCATION  
Loc: WOHL CLINIC OUTPATIENT

Ref Phy:  
Bal: 283.00  
FSC: BLUE SHIELD ALLIANCE  
CHOICE

**98**

**Instructions**

**96**

**Start Task**

**112**

**114**

**116**

**102**

**Edit**

**108**

**New**

**110**

**Actions**

**112**

**114**

FIG. 5

64

90 112 118 112 114 116

106 TASK LIST 107

104

102 108 110

120

FIG. 6

22

92

6/14

94

98

100

122

124

126

128

130

132

134

136

138

140

142

144

146

148

150

152

154

156

158

160

162

164

166

168

170

172

174

176

178

180

182

184

186

188

190

192

194

196

198

200

202

204

206

208

210

212

214

216

218

220

222

224

226

228

230

232

234

236

238

240

242

244

246

248

250

252

254

256

258

260

262

264

266

268

270

272

274

276

278

280

282

284

286

288

290

292

294

296

298

300

302

304

306

308

310

312

314

316

318

320

322

324

326

328

330

332

334

336

338

340

342

344

346

348

350

352

354

356

358

360

362

364

366

368

370

372

374

376

378

380

382

384

386

388

390

392

394

396

398

400

402

404

406

408

410

412

414

416

418

420

422

424

426

428

430

432

434

436

438

440

442

444

446

448

450

452

454

456

458

460

462

464

466

468

470

472

474

476

478

480

482

484

486

488

490

492

494

496

498

500

502

504

506

508

510

512

514

516

518

520

522

524

526

528

530

532

534

536

538

540

542

544

546

548

550

552

554

556

558

560

562

564

566

568

570

572

574

576

578

580

582

584

586

588

590

592

594

596

598

600

602

604

606

608

610

612

614

616

618

620

622

624

626

628

630

632

634

636

638

640

642

644

646

648

650

652

654

656

658

660

662

664

666

668

670

672

674

676

678

680

682

684

686

688

690

692

694

696

698

700

702

704

706

708

710

712

714

716

718

720

722

724

726

728

730

732

734

736

738

740

742

744

746

748

750

752

754

756

758

760

762

764

766

768

770

772

774

776

778

780

782

784

786

788

790

792

794

796

798

800

802

804

806

808

810

812

814

816

818

820

822

824

826

828

830

832

834

836

838

840

842

844

846

848

850

852

854

856

858

860

862

864

866

868

870

872

874

876

878

880

882

884

886

888

890

892

894

896

898

900

902

904

906

908

910

912

914

916

918

920

922

924

926

928

930

932

934

936

938

940

942

944

946

948

950

952

954

956

958

960

962

964

966

968

970

972

974

976

978

980

982

984

986

988

990

992

994

996

998

1000

1002

1004

1006

1008

1010

1012

1014

1016

1018

1020

1022

1024

1026

1028

1030

1032

1034

1036

1038

1040

1042

1044

1046

1048

1050

1052

1054

1056

1058

1060

1062

1064

1066

1068

1070

1072

1074

1076

1078

1080

1082

1084

1086

1088

1090

1092

1094

1096

1098

1100

1102

1104

1106

1108

1110

1112

1114

1116

1118

1120

1122

1124

1126

1128

1130

1132

1134

1136

1138

1140

1142

1144

1146

1148

1150

1152

1154

1156

1158

1160

1162

1164

1166

1168

1170

1172

1174

1176

1178

1180

1182

1184

1186

1188

1190

1192

1194

1196

1198

1200

1202

1204

1206

1208

1210

1212

1214

1216

1218

1220

1222

1224

1226

1228

1230

1232

1234

1236

1238

1240

1242

1244

1246

1248

1250

1252

1254

1256

1258

1260

1262

1264

1266

1268

1270

1272

1274

1276

1278

1280

1282

1284

1286

1288

1290

1292

1294

1296

1298

1300

1302

1304

1306

1308

1310

1312

1314

1316

1318

1320

1322

1324

1326

1328

1330

1332

1334

1336

1338

1340

1342

1344

1346

1348

1350

1352

1354

1356

1358

1360

1362

1364

1366

1368

1370

1372

1374

1376

1378

1380

1382

1384

1386

1388

1390

1392

1394

1396

1398

1400

1402

1404

1406

1408

1410

1412

1414

1416

1418

1420

1422

1424

1426

1428

1430

1432

1434

1436

1438

1440

1442

1444

1446

1448

1450

1452

1454

1456

1458

1460

1462

1464

1466

1468

1470

1472

1474

1476

1478

1480

1482

1484

1486

1488

1490

1492

1494

1496

1498

1500

1502

1504

1506

1508

1510

1512

1514

1516

1518

1520

1522

1524

1526

1528

1530

1532

1534

1536

1538

1540

1542

1544

1546

1548

1550

1552

1554

1556

1558

1560

1562

1564

1566

1568

1570

1572

1574

1576

1578

1580

1582

1584

1586

1588

1590

1592

1594

1596

1598

1600

1602

1604

1606

1608

1610

1612

1614

1616

1618

1620

1622

1624

1626

1628

1630

1632

1634

1636

1638

1640

1642

1644

1646

1648

1650

1652

1654

1656

1658

1660

1662

1664

1666

1668

1670

1672

1674

1676

1678

1680

1682

1684

1686

1688

1690

1692

1694

1696

1698

1700

1702

1704

1706

1708

1710

1712

1714

1716

1718

1720

1722

1724

1726

1728

1730

1732

1734

1736

1738

1740

1742

1744

1746

1748

1750

1752

1754

1756

1758

1760

1762

1764

1766

1768

1770

1772

1774

1776

1778

1780

1782

1784

1786

1788

1790

1792

1794

1796

1798

1800

1802

1804

1806

1808

1810

1812

1814

1816

1818

1820

1822

1824

1826

1828

1830

1832

1834

1836

1838

1840

1842

1844

1846

1848

1850

1852

1854

1856

1858

1860

1862

1864

1866

1868

1870

1872

1874

1876

1878

1880

1882

1884

1886

1888

1890

1892

1894

1896

1898

1900

1902

1904

1906

1908

1910

1912

1914

1916

1918

1920

1922

1924

1926

1928

1930

1932

1934

1936

1938

1940

1942

1944

1946

1948

1950

1952

1954

1956

1958

1960

1962

1964

1966

1968

1970

1972

1974

1976

1978

1980

1982

1984

1986

1988

1990

1992

1994

1996

1998

2000

2002

2004

2006

2008

2010

2012

2014

2016

2018

2020

2022

2024

2026

2028

2030

2032

2034

2036

2038

2040

2042

2044

2046

2048

2050

2052

2054

2056

2058

2060

2062

2064

2066

2068

2070

2072

2074

2076

2078

2080

2082

2084

2086

2088

2090

2092

2094

2096

2098

2100

2102

2104

2106

2108

2110

2112

2114

2116

2118

2120

2122

2124

2126

2128

2130

2132

2134

2136

2138

2140

2142

2144

2146

2148

2150

2152

2154

2156

2158

2160

2162

2164

2166

2168

2170

2172

2174

2176

2178

2180

2182

2184

2186

2188

2190

2192

2194

2196

2198

2200

2202

2204

2206

2208

2210

2212

2214

2216

2218

2220

2222

2224

2226

2228

2230

2232

2234

2236

2238

2240

2242

2244

2246

2248

2250

2252

2254

2256

2258

2260

2262

2264

2266

2268

2270

2272

2274

2276

2278

2280

2282

2284

2286

2288

2290

2292

2294

2296

2298

2300

2302

2304

2306

2308

2310

2312

2314

2316

2318

2320

2322

2324

2326

2328

2330

2332

2334

2336

2338

2340

2342

2344

2346

2348

2350

2352

2354

2356

2358

2360

2362

2364

2366

2368

2370

2372

2374

2376

2378

2380

2382

2384

2386

2388

2390

2392

2394

2396

2398

2400

2402

2404

2406

2408

2410

2412

2414

2416

2418

2420

2422

2424

2426

2428

2430

2432

2434

2436

2438

2440

2442

2444

2446

2448

2450

2452

2454

2456

2458

2460

2462

2464

2466

2468

2470

2472

2474

2476

2478

2480

2482

2484

2486

2488

2490

2492

2494

2496

2498

2500

2502

2504

2506

2508

2510

2512

2514

2516

2518

2520

2522

2524

2526

2528

2530

2532

2534

2536

2538

2540

2542

2544

2546

2548

2550

2552

2554

2556

2558

2560

2562

2564

2566

2568

2570

2572

2574

2576

2578

2580

2582

2584

2586

2588

2590

2592

2594

2596

2598

2600

2602

2604

2606

2608

2610

2612

2614

2616

2618

2620

2622

2624

2626

2628

2630

2632

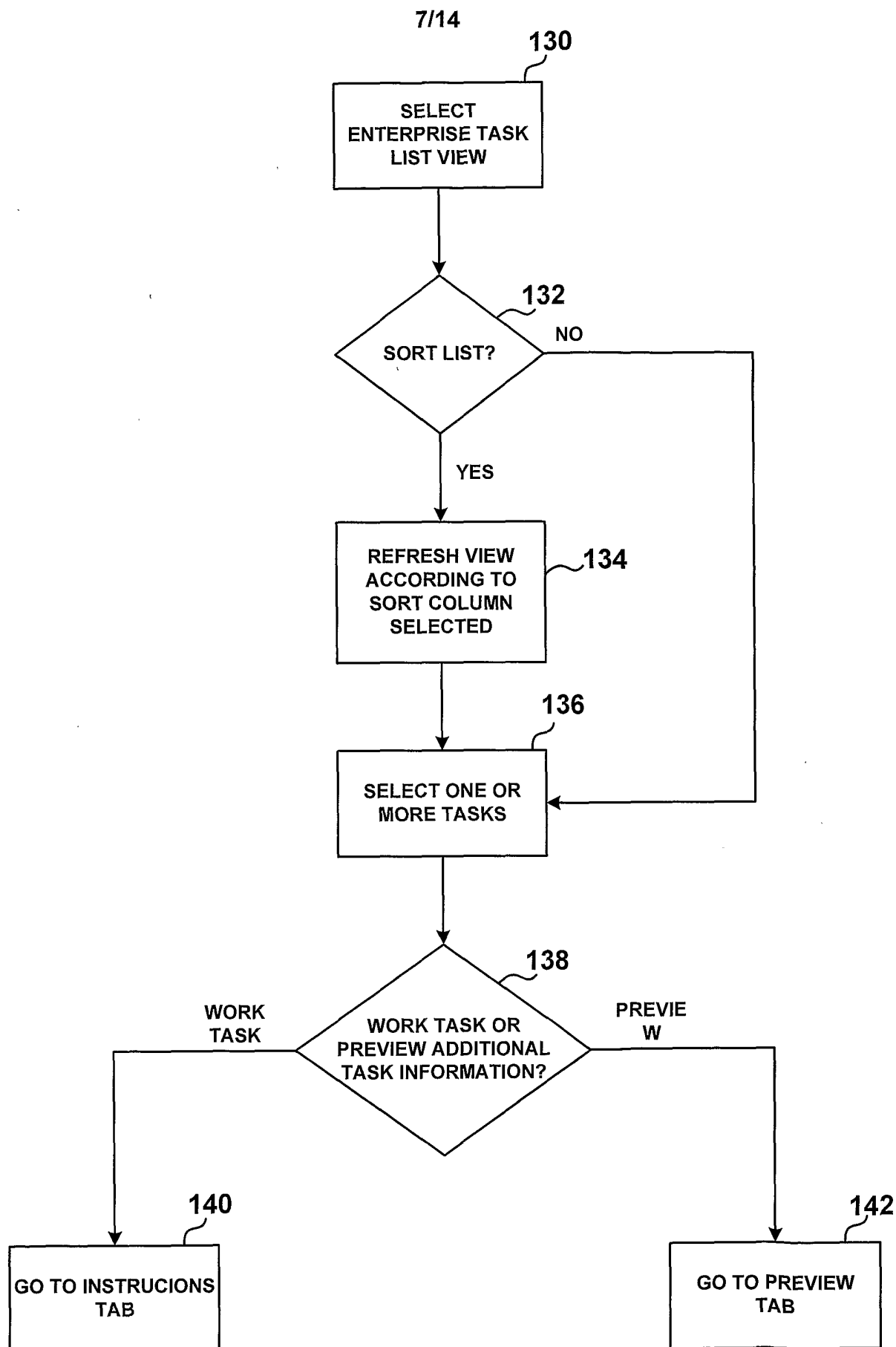


FIG. 7

8/14

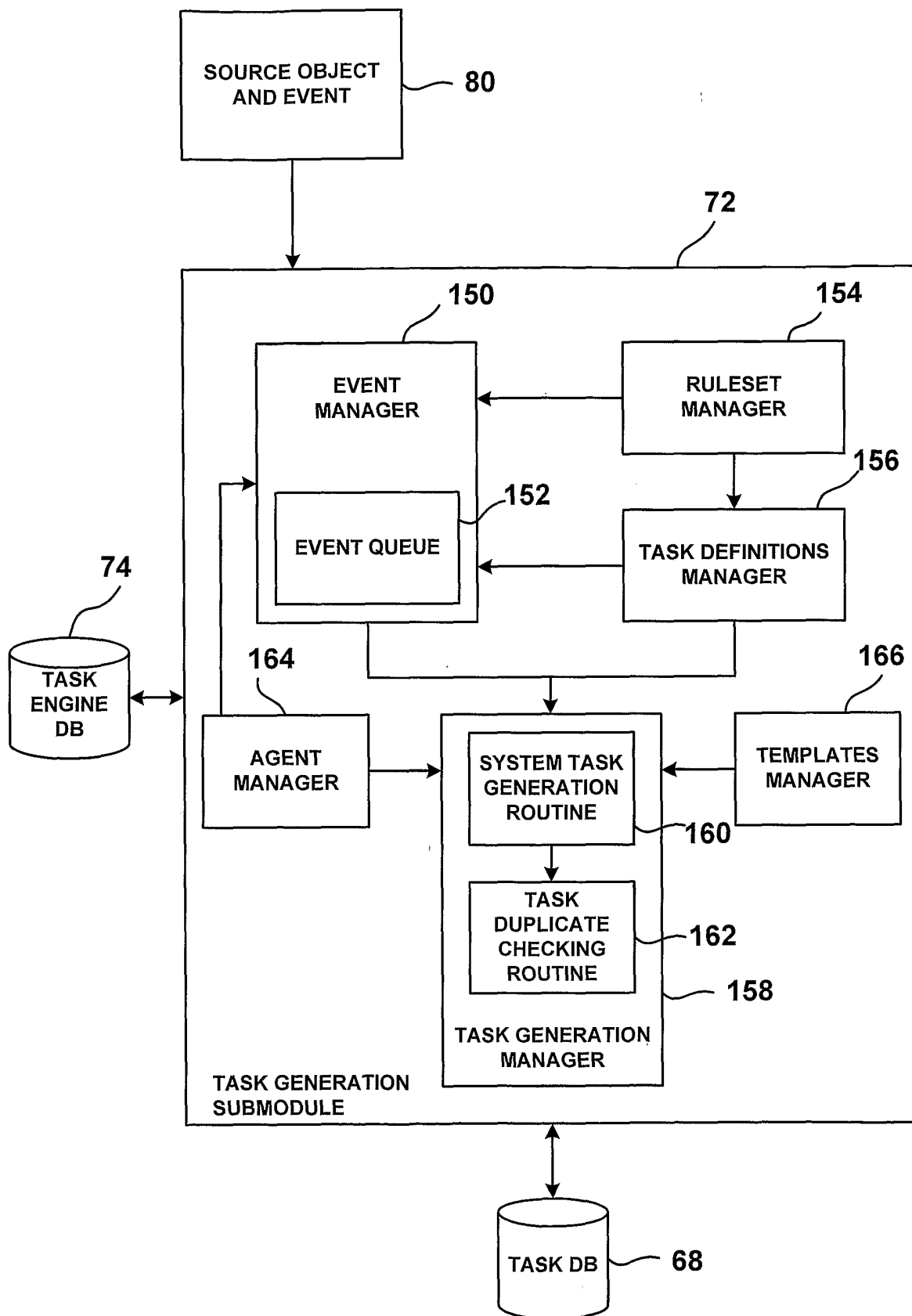


FIG. 8

9/14

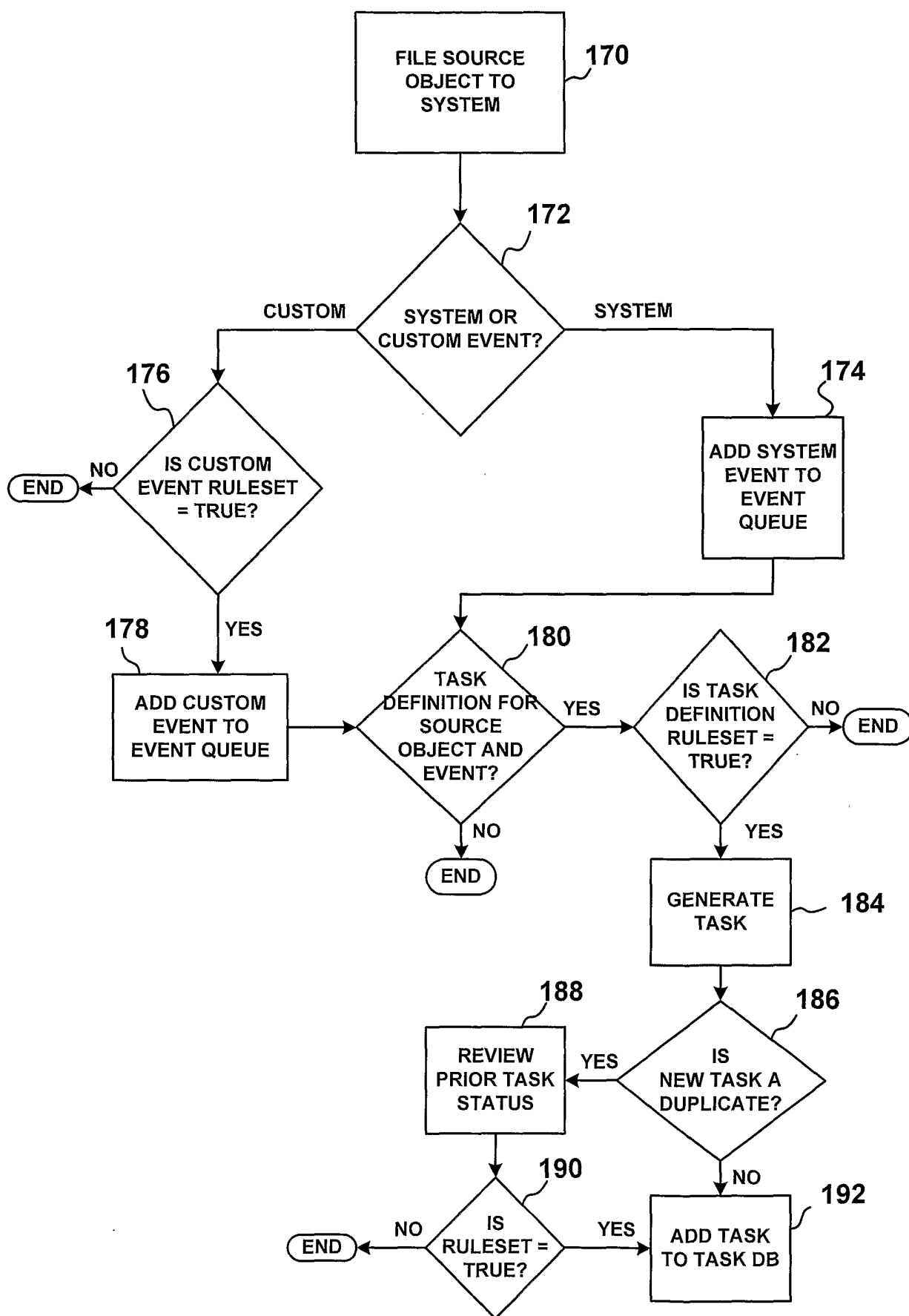


FIG. 9

10/14

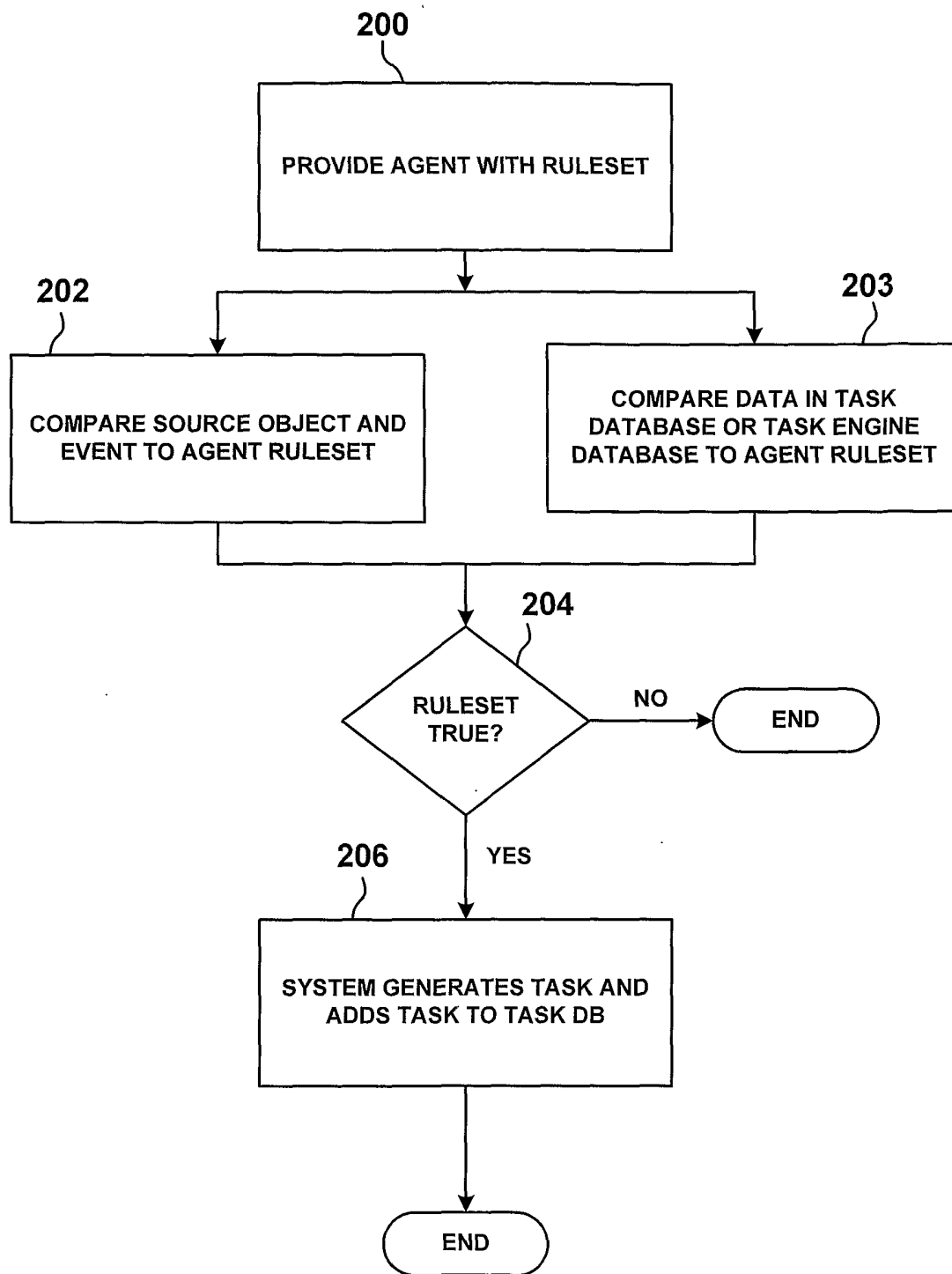


FIG. 10

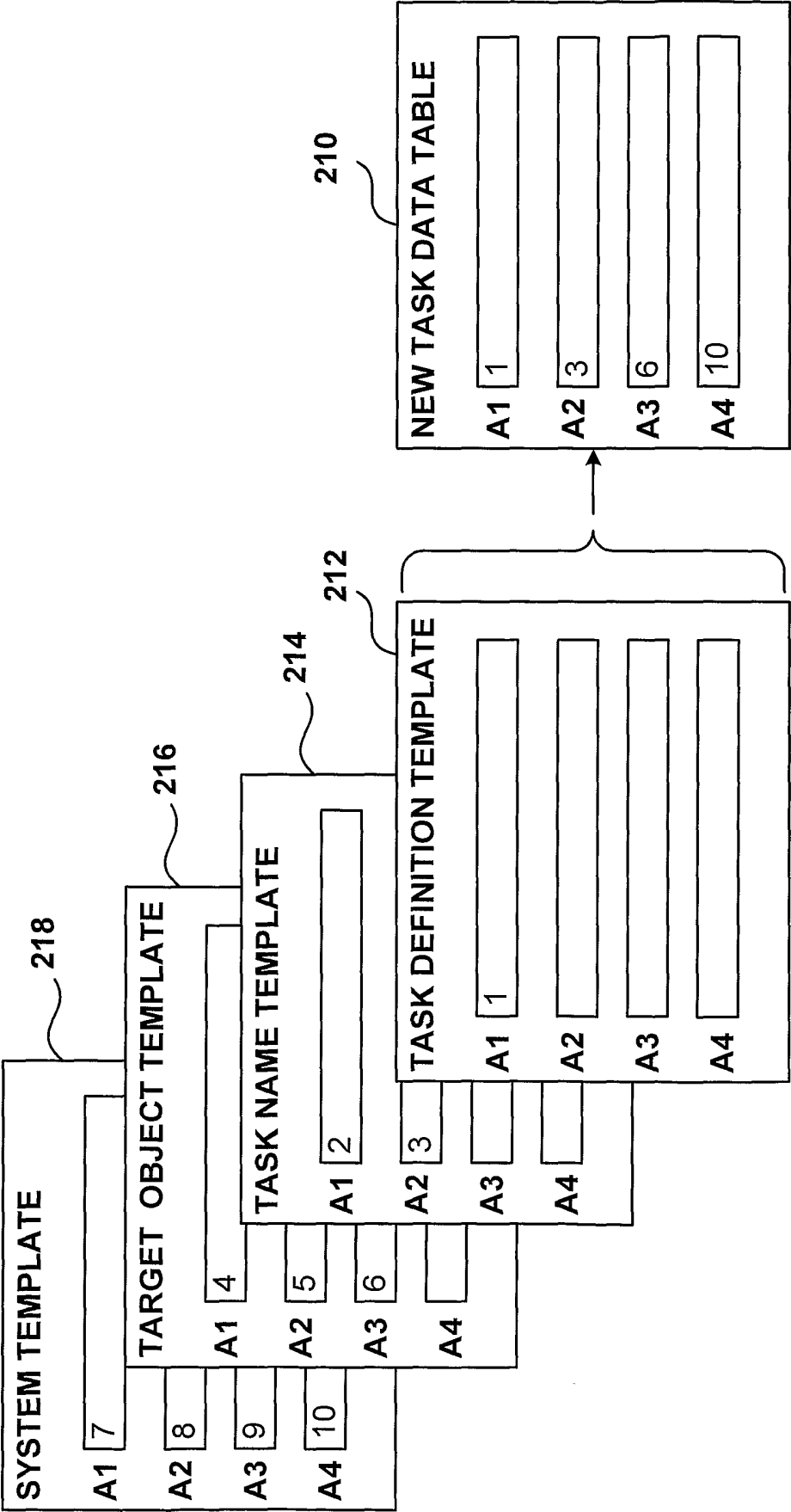


FIG. 11

12/14

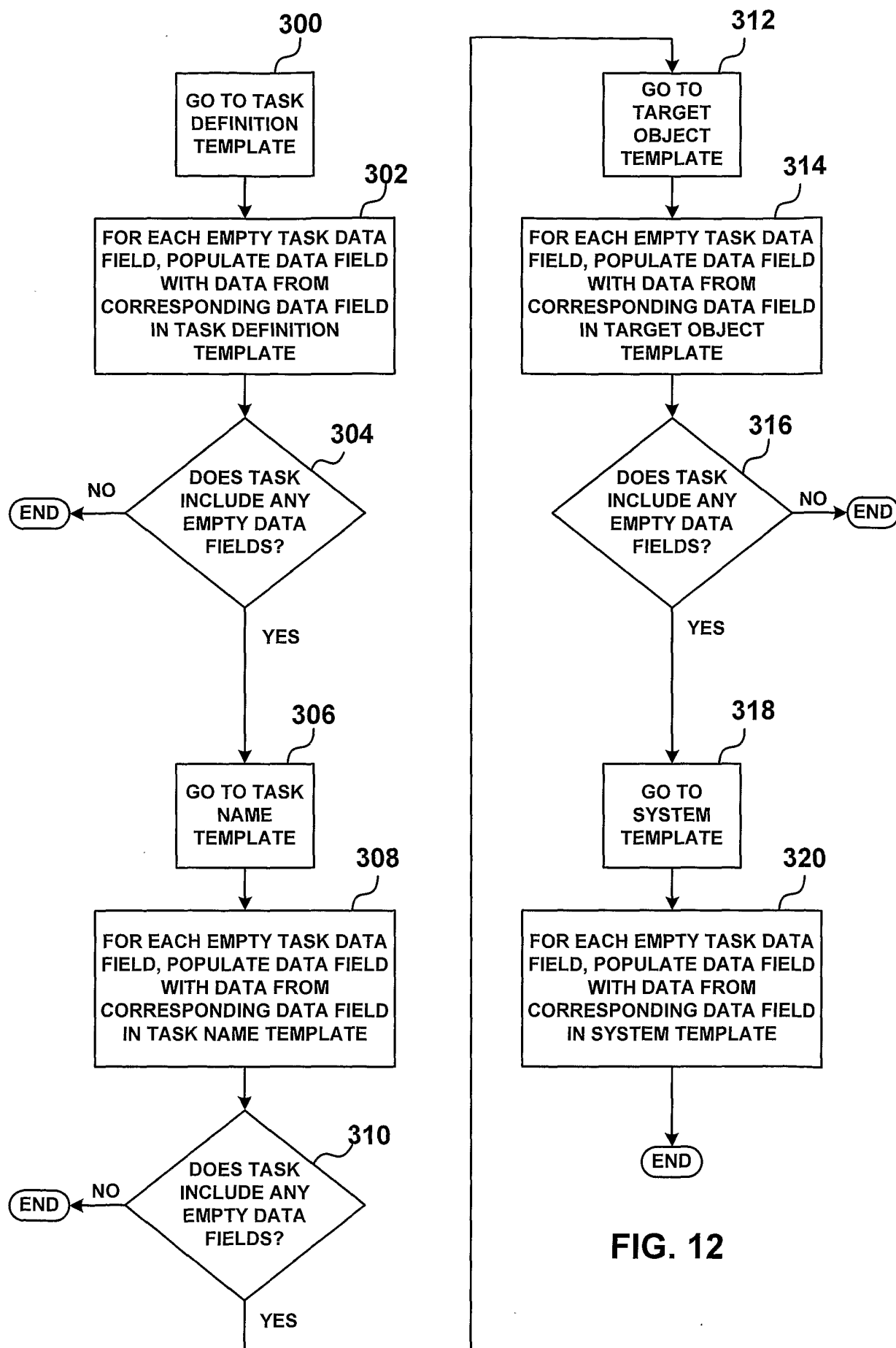
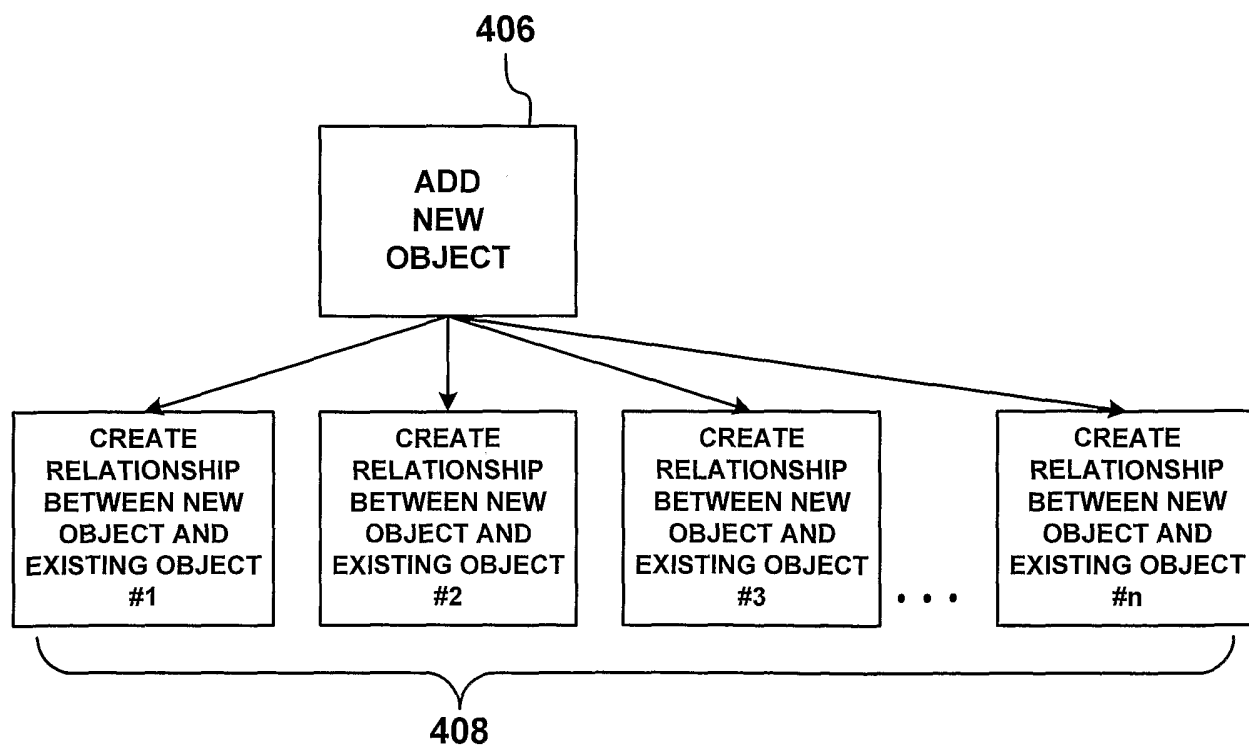
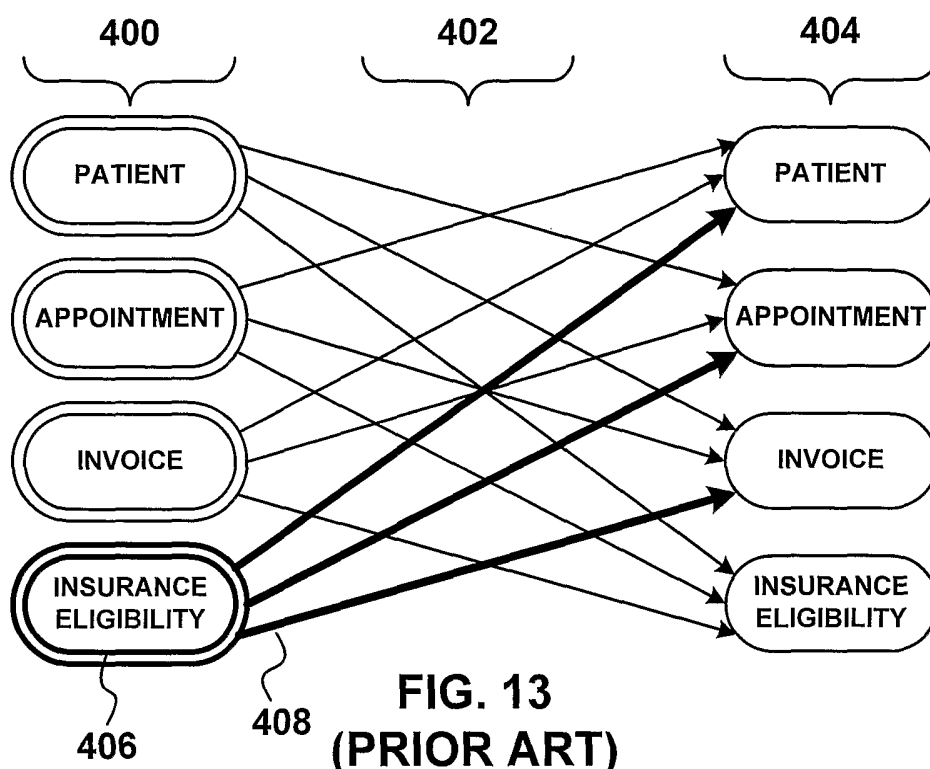


FIG. 12

13/14



14/14

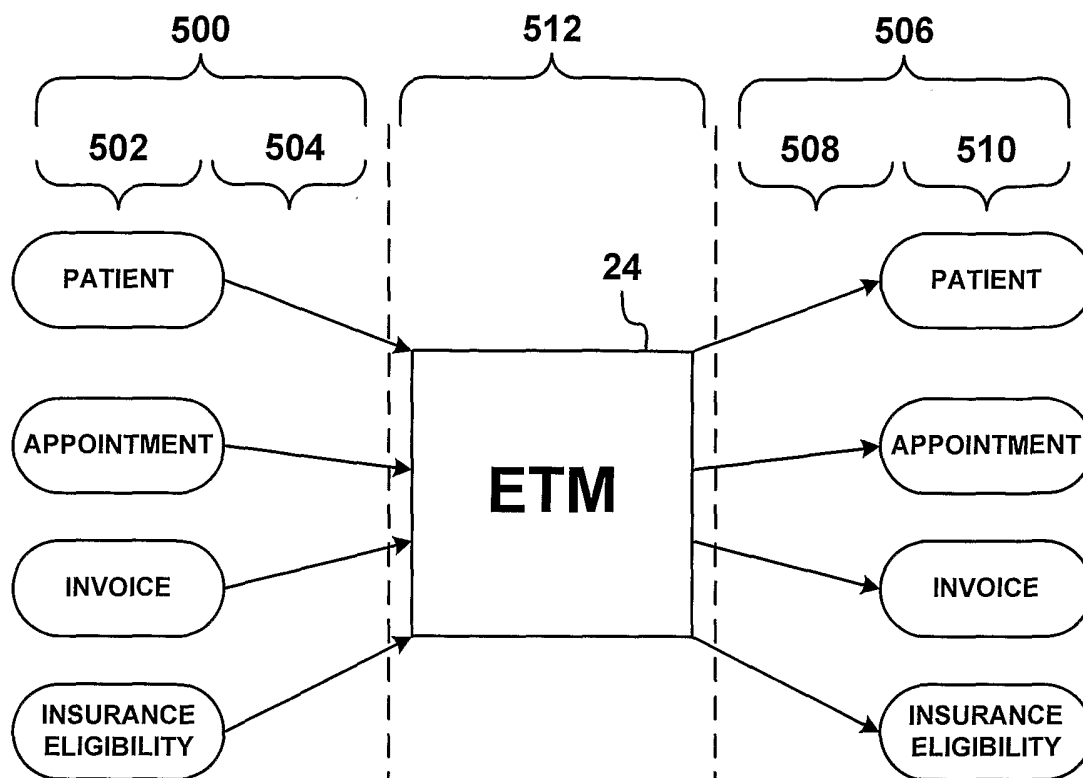


FIG. 15

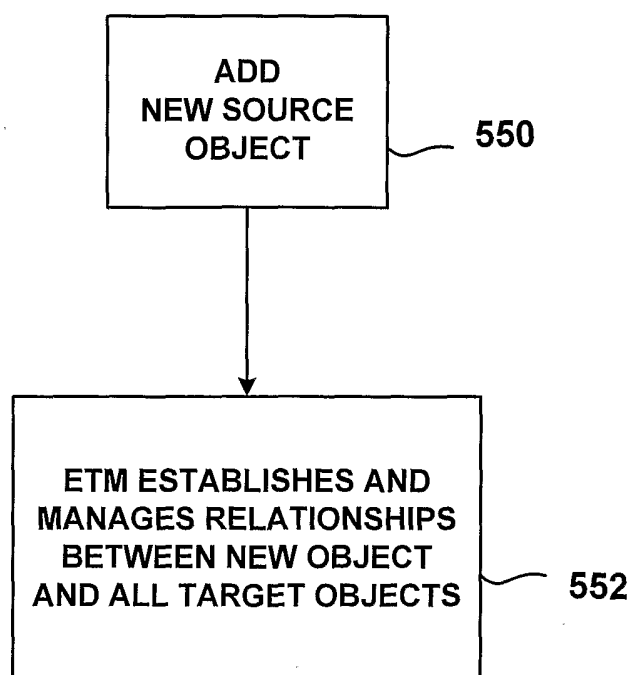


FIG. 16