



- (51) International Patent Classification:  
*G06Q 10/06* (2012.01)
- (21) International Application Number:  
PCT/US2014/016827
- (22) International Filing Date:  
18 February 2014 (18.02.2014)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
61/766,215 19 February 2013 (19.02.2013) US
- (71) Applicants: **SIEMENS AKTIENGESELLSCHAFT** [DE/DE]; Wittelsbacherplatz 2, 80333 München (DE). **SIEMENS PRODUCT LIFECYCLE MANAGEMENT SOFTWARE INC.** [US/US]; 5800 Granite Parkway, Suite 600, Plano, Texas 75024 (US).
- (72) Inventors: **NOETZELMANN, Oswin**; 712 Wood Mill Drive, East Windsor, New Jersey 08512 (US). **REUVENI, Rami**; 20 Meadowsweet Way, Irvine, California 92612 (US). **HAMBRIDGE, Victor Robert**; 17452 Santa Isabel,

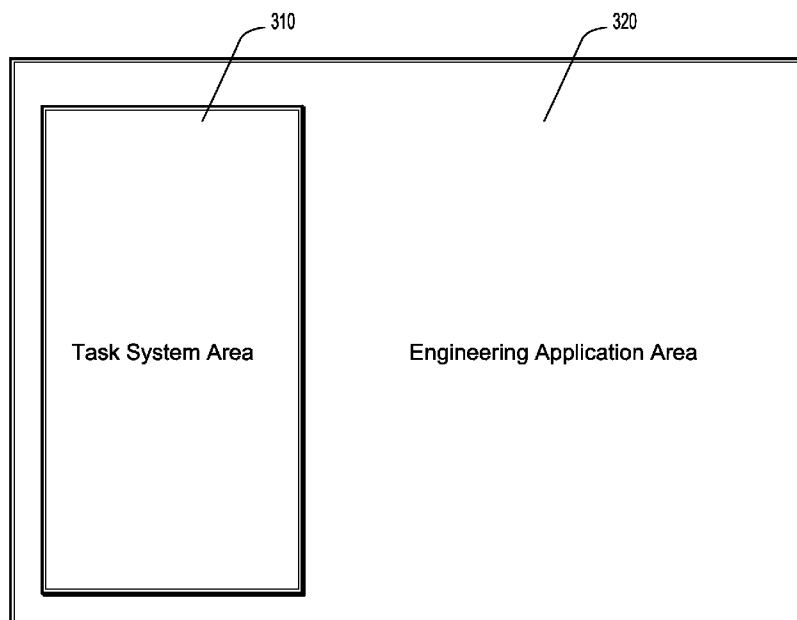
Fountain Valley, California 92708 (US). **DUREL, Marine**; 755 College Road East, Princeton, New Jersey 08540 (US). **OERTER, Tim**; 7701 Whitewater Drive, Huntington Beach, California 92648 (US). **PORTWAY, JR., Christopher Patrick**; 7890 E. Spring Street 7C, Long Beach, California 90815 (US). **VIELSAECKER, Dirk**; 8825 Canary Avenue, Fountain Valley, California 92708 (US). **RAGAVAN, Sarvananthan**; Kamillenweg 5, 50226 Frechen (DE). **NGUEFACK MBOGNING, Cyprien Pame**; Stresemannstr. 6, 53123 Bonn (DE). **STE-DEROTH, Daniela**; 2240 Park Newport, 409, Newport Beach, California 92660 (US).

(74) Agents: **PASCHBURG, Donald B.** et al.; Siemens Corporation- Intellectual Property Dept., 170 Wood Avenue South, Iselin, New Jersey 08830 (US).

(81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME,

[Continued on next page]

(54) Title: METHOD AND SYSTEM FOR VISUALIZING ENGINEERING TASKS IN A MULTIDISCIPLINARY ENGINEERING SYSTEM



(57) Abstract: A method for visualizing engineering tasks in a multidisciplinary engineering environment including: displaying an engineering application user-interface (UI) area (320); and displaying a task system UI area (310) in the engineering application UI area, wherein the task system UI area is movable over the engineering application UI area and a task is an association object-action that integrates with a multidisciplinary engineering system.

**FIG. 3**



MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

**(84) Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ,

**Published:**

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

METHOD AND SYSTEM FOR VISUALIZING ENGINEERING TASKS IN A  
MULTIDISCIPLINARY ENGINEERING SYSTEM

**CROSS-REFERENCE TO RELATED APPLICATIONS**

This application claims priority to U.S. provisional application no. 61/766,215, filed February 19, 2013, the disclosure of which is incorporated by reference herein in its entirety.

This application is related to PCT application entitled “METHOD AND SYSTEM FOR TASK ORIENTED ENGINEERING IN A MULTIDISCIPLINARY ENGINEERING SYSTEM”, attorney docket no. 2013P02900WO (8706-1426), and PCT application entitled “METHOD AND SYSTEM FOR OPTIMIZED PROJECTION IN A MULTIDISCIPLINARY ENGINEERING SYSTEM”, attorney docket no. 2013P02901WO (8706-1427), the disclosures of which are incorporated by reference herein in their entireties.

**BACKGROUND OF THE INVENTION**

**1. Technical Field**

The present invention relates to multidisciplinary engineering systems.

**2. Discussion of the Related Art**

A multi-disciplinary engineering system is a system that allows engineers from multiple disciplines to work on common or connected data. For example, a factory planner can work together with a mechanical engineer, an electrical engineer and an automation engineer to plan a new production line for a car door assembly. An example of a multi-disciplinary engineering system 100 is shown in FIG. 1.

Traditionally, the disciplines work separated from a data point of view and manual synchronization of the discipline specific data is very time consuming and error prone. For example, when the automation engineer introduces a new programmable logic controller (PLC) to the project, which he needs to automate the line, this information needs to be transported to the electrical engineer, so he can place it in the right electrical cabinet and plan its wiring. If this information is not transported or is distorted, it can have serious impact on the quality of the work of the electrical engineer and vice versa.

Data connections between the different disciplines can potentially be used to allow the system to support various functions such as notification/communication between disciplines (e.g., departments), change propagation (e.g., rule based), formalization of workflows (e.g., sign off procedures), multi-disciplinary report generation and usage of interdisciplinary common data structures.

An automation engineering application is a system which helps engineers to design and implement their factory plant automation. They can add their resources, like robots or conveyors, organize them in functional and physical areas, connect them to PLCs, etc. The goal of all this plant configuration and preparation is to generate data that can directly be used to setup and run the factory. For example, electric diagrams to connect all the resources together, PLC code to download in the PLCs in order to run the machines, and Human Machine Interface (HMI) screens to view and input data into the machines via touch screen panels in the factory.

The fact that this data is generated will save a considerable amount of time, since the engineers will not have to draw the electric diagrams themselves, or write the PLC or HMI code from scratch. In order to obtain the most accurate generated data, the

preparation work made in the automation engineering application is very important. The resources have to be connected the right way, all the network addresses have to be setup correctly for the PLC connections to work, etc. An example of generated data 200 from engineers input is shown in FIG. 2.

In FIG. 2, the first arrow with DATA is data input from the user, for example the type of robot used (e.g., specific model), the type of motor to run the conveyor, which resources are connected to which PLC, etc. This is not generated but is instead manually entered by the engineer and considered as the preparation work. The other arrows in FIG. 2, e.g., outputs from the Automation Application, is data which is this time generated. It is in this case PLC code, HMI screens and Electric diagrams. The Automation Application processes this user input to generate those outputs.

If this preparation work is not correctly done, for example a robot has not been associated to a PLC, the generated results would be wrong, which can have a very serious impact on the factory plant, for example, robots behaving in an unexpected way.

The previous systems do not present engineering tasks in a multi-disciplinary context to assist users in completing an engineering project. Traditionally the user had to learn all the user interface (UI) elements of discipline specific engineering applications – e.g., menus, dialogs, workspaces – to input and process the data required.

This is time consuming and provokes errors, since for every task, the user has to navigate through the user interfaces of all engineering applications involved. The more complex the engineering application is, the more menus and entries have to be navigated; therefore, the engineer will need more time to familiarize with the user interfaces and also to complete his project.

## SUMMARY OF THE INVENTION

According to an exemplary embodiment of the present invention, there is provided a method for visualizing engineering tasks in a multidisciplinary engineering environment, comprising: displaying an engineering application user-interface (UI) area; and displaying a task system UI area in the engineering application UI area, wherein the task system UI area is movable over the engineering application UI area and a task is an association object-action that integrates with a multidisciplinary engineering system.

Tasks are sequentially listed in the task system UI area. Tasks are visualized in graph form in the task system UI area. Tasks are visualized in the engineering application UI area. Tasks are sorted by action and objects are listed underneath corresponding actions. Tasks are sorted by objects and actions are listed underneath corresponding objects. Displayed tasks are filtered according to their domain, type, maturity or dependency level. Tasks are ordered and visualized depending on which action has to be completed first.

The method further comprises updating displayed tasks in real-time. Updating displayed tasks comprises: changing data in the multidisciplinary engineering system in response to an action performed by a user in the engineering application UI area; notifying a task system that the data has been changed in the multidisciplinary engineering system; checking, in the task system, whether the new data completes a task or if another task has to be created; and updating the task system UI area with a new task. A task is executed directly, indirectly or step-by-step by a user.

According to an exemplary embodiment of the present invention, there is provided a system for visualizing engineering tasks in a multidisciplinary engineering environment, comprising: a memory device for storing a program; a processor in communication with

the memory device, the processor operative with the program to: display an engineering application UI area; and display a task system UI area in the engineering application UI area, wherein the task system UI area is movable over the engineering application UI area and a task is an association object-action that integrates with a multidisciplinary engineering system.

Tasks are sequentially listed in the task system UI area. Tasks are visualized in graph form in the task system UI area. Tasks are visualized in the engineering application UI area. Tasks are sorted by action and objects are listed underneath corresponding actions, or tasks are sorted by objects and actions are listed underneath corresponding objects.

According to an exemplary embodiment of the present invention, there is provided a computer program product for visualizing engineering tasks in a multidisciplinary engineering environment, comprising: a non-transitory computer readable storage medium having computer readable program code embodied therewith, the computer readable program code comprising: computer readable program code configured to perform the steps of: displaying an engineering application UI area; and displaying a task system UI area in the engineering application UI area, wherein the task system UI area is movable over the engineering application UI area and a task is an association object-action that integrates with a multidisciplinary engineering system.

Tasks are sequentially listed in the task system UI area. Tasks are visualized in graph form in the task system UI area. Tasks are visualized in the engineering application UI area.

### **BRIEF DESCRIPTION OF THE DRAWINGS**

FIG. 1 is an example of a multidisciplinary engineering system;

FIG. 2 is an example of data generated from engineers input;

FIG. 3 illustrates a task system UI area according to an exemplary embodiment of the present invention;

FIG. 4A illustrates a task list view according to an exemplary embodiment of the present invention;

FIG. 4B illustrates a task graph view according to an exemplary embodiment of the present invention;

FIG. 5 illustrates task visualization within an engineering application workspace according to an exemplary embodiment of the present invention;

FIG. 6 illustrates task visualization sorted by an action according to an exemplary embodiment of the present invention;

FIG. 7 illustrates task visualization sorted by an object according to an exemplary embodiment of the present invention;

FIG. 8 illustrates a real-time UI update process according to an exemplary embodiment of the present invention;

FIG. 9A illustrates an implementation of a task visualization system according to an exemplary embodiment of the present invention;

FIG. 9B illustrates an implementation of a task visualization system according to an exemplary embodiment of the present invention; and

FIG. 10 illustrates a computer system in which an exemplary embodiment of the present invention may be implemented.

## DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

The principle of the visualization system according to an exemplary embodiment of the present invention is to display the set of tasks and thus the project status to the user in the clearest fashion.

A task is an association object-action that integrates with a multi-disciplinary engineering system. FIG. 3 illustrates how a task system UI area 310 can be integrated in an engineering application UI area 320.

The task system area 310 can be moved over the engineering application area 320.

The task visualization system offers multiple ways of viewing the tasks. One option is a list that can be sorted and organized by different criteria: e.g., sort the tasks by object or action, etc. (this will be discussed in more detail later). This allows the user chose between different approaches: execute all the actions of a specific object, or complete a specific action for all the objects. For any of those approaches, the task visualization system provides step-by-step guidance, which will allow the user to execute all his related tasks in one window in one sequence of actions. FIG. 4A illustrates an example of a task list view 410a in an engineering application area 420a.

Another way of visualizing the tasks is in a graph form. This can be helpful to visualize dependencies of tasks or tasks in the context of workflows and the various engineering disciplines. FIG. 4B illustrates an example of a task graph view 410b in an engineering application area 420b.

The task visualization can also be embedded within the engineering applications workspace. This has the benefit that the task can be accessed from all workspaces within the different engineering applications that visualize the objects that are part of a task. For

example, in a mechanical design tool, the tasks can appear as icons within a three dimensional (3D) model. Another example is the visualization of tasks within a two dimensional (2D) plant layout of an automation engineering application for a factory layout planner application. FIG. 5 illustrates the integration of tasks 530 into a 3D workspace 520. In this case, the tasks 530 are visualized in the 3D workspace 520, not a task system area 510.

The task visualization also provides filtering options, for the domain engineer to allow finding the right tasks quickly, even in very large engineering projects that can have thousands of open tasks. This will be discussed in more detail later.

A list with more examples of features that are part of the present invention can be found below. Note that this list is not meant to be complete and other features are possible to use with this invention.

The task visualization system allows the user to sort his set of tasks in different ways. One such example is by action.

The objects which have tasks associated with an action will be listed underneath the action. This corresponds to the user approach “I will complete this action for all the objects.” A concrete example in a project would be “I want to assign a function for my conveyor, robot and PLC.” An example of task visualization sorted by action 600 is shown in FIG. 6.

Tasks may be sorted by object. In this case, the actions which have a task associated with an object will be listed underneath the object. This corresponds to the user approach “I will complete all the actions of this object.” A concrete example would be “I

want to assign a function and assign a location for my conveyor.” An example of task visualization sorted by object 700 is shown in FIG. 7.

Tasks may be sorted by dependencies. Tasks can depend on each other, so one more option is to list tasks, based on their dependencies. One possibility is to visualize it as a tree-structure, another to show a graph.

Tasks may be sorted by engineering applications. The task system can provide tasks that involve multiple engineering disciplines and engineering systems. So another option to sort tasks is that tasks, which involve similar engineering applications, can be grouped together to reduce visual complexity.

Tasks may be sorted by workflow. Tasks can be integrated with formal workflows if supported by the multi-disciplinary system. For example, the Siemens Teamcenter repository platform supports a workflow model that could be integrated with the present invention. This means that tasks can be assigned and organized in workflows by a user, rather than just by what the engineering applications provide “out of the box.”

The action to change the way the tasks are sorted should be very simple, examples are either by clicking a button or right clicking on the task list and selecting the sorting method on the context menu.

The task visualization system provides the option to filter the displayed list of tasks by several criteria as follows. (1) Domain filter: allowing the user to choose to only display the tasks belonging to certain domains (e.g., mechanical, automation or electrical). (2) Type: display only tasks of certain task types. For example, display only tasks of type “Add Object to System.” (3) Maturity: tasks can be filtered by the maturity status of the objects involved, which is provided by the underlying multi-disciplinary engineering

model. (4) Dependency level: tasks can be interdependent and it is another feature to control how many levels of dependency should be visualized. These are examples for filtering options. Other options can be used with the present invention.

The task system, which works below the visualization system, orders the tasks by sequence, depending on which action has to be completed first, e.g., data or dependency. The user can choose to display his tasks according to that given sequence, to be sure to execute his tasks in the right order.

The list of tasks is updated in real-time, to allow the user to instantaneously know if he completed a task or if, on the contrary, a new task was created. This is possible through an event mechanism as shown in FIG. 8. Referring to FIG. 8, in step 1, when the user performs an action on the UI side of application 810, the data is changed in multi-disciplinary system 820. In step 2, task system 830 is notified that a change has been done and the data has been updated. In step 3, the task system 830 takes the new data into account and checks if it completes any task or if another task has to be created. In step 4, the task system 830 updates visualization system 840 with the new set of tasks.

A task can be completed with three different UI operations. One example is directly. In this case, if the user wants to directly complete a task, he can double click or right click and select the right context menu entry. The task visualization system will automatically open the form, within the application, containing the information needed for the task to be achieved.

The user can also indirectly complete a task, by opening himself a form in the application, and filling the data that will cause completion of a task.

If the user wants to execute all the related tasks at the same time (e.g., all the tasks related to a specific object or all the tasks related to a specific action), he can choose to open the step-by-step guidance tool, which will allow him to complete all the tasks in one sequence within the same window.

The following is a description of how the invention can be implemented, as an example. It possible to use other existing engineering systems.

FIG. 9A is an example implementation of a task system according to an exemplary embodiment of the present invention integrated into a multi disciplinary engineering system. It illustrates the same mechanism as FIG. 8 but with an existing application, as a possible implementation.

The lower part in FIG. 9A is a server 950, for example, Siemens Teamcenter. Its role is to hold and manage data and its different revisions, as it is a product lifecycle management (PLM) software.

Teamcenter, with the integrated multi disciplinary system, is a common platform that different engineering applications can connect to and use to communicate between each other. In this example implementation, the task system 930 would be embedded into the server 950.

Each engineering application 910 is connected to the Teamcenter server 950 (over a network 960), and exchanges information with the server 950. Each of the applications 910 displays tasks through a task visualization system 940, which, connected to the task system 930 located in the server 950, allows UI updates through the event mechanism. The task visualization system 940 overlaps a multi-disciplinary application part 920

An example workflow would be:

-a Line Designer engineer adds a robot to a plant layout. When the engineer saves, this data is sent to the shared platform Teamcenter 950 through the network 960, and stored. The storage could be in one of databases 970.

-the task system 930, embedded into the multi disciplinary system server 950 integrated into Teamcenter, processes the new information and decides that a new task has to be created for Automation Designer, which is to assign a Function to this new robot.

-a notification is sent to the Automation Designer application (leftmost 910), which updates its task visualization system 940 with the new task "Assign Function" for the new robot.

FIG. 9B is a screenshot of an example implementation of the task visualization system 940 that we would see in the Automation Designer application of FIG. 9A.

In FIG. 9B, the tasks are sorted by action, so we can see that the user needs to assign a function to seven objects, assign a location to five, select a template component for his fixture and assign a PLC to his PLC group 1.

In accordance with an exemplary embodiment of the present invention, a user can have a quick overview of the tasks he has to complete and therefore have a better time estimate for his project and quicker navigation through the engineering applications. Both result in time savings on an engineering project. A company may adopt task visualization according to an exemplary embodiment of the present invention, to the needs of specialized industries and engineering domains. The clear visualization of tasks also gives the users a better understanding of the project maturity status, even during the middle of milestone phases (which are a traditional way of synchronizing the work of different disciplines). These inventive features result in shortened time to prepare an automation

project, a shortened time to market for complex engineering projects and thus save investment and reduce risk for engineering companies.

As will be appreciated by one skilled in the art, aspects of the present invention may be embodied as a system, method or computer program product. Accordingly, aspects of the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a “circuit,” “module” or “system.” Furthermore, aspects of the present invention may take the form of a computer program product embodied in one or more computer readable medium(s) having computer readable program code embodied thereon.

Any combination of one or more computer readable medium(s) may be utilized. The computer readable medium may be a computer readable signal medium or a computer readable storage medium. A computer readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer readable storage medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can

contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

A computer readable signal medium may include a propagated data signal with computer readable program code embodied therein, for example, in baseband or as part of a carrier wave. Such a propagated signal may take any of a variety of forms, including, but not limited to, electro-magnetic, optical, or any suitable combination thereof. A computer readable signal medium may be any computer readable medium that is not a computer readable storage medium and that can communicate, propagate, or transport a program for use by or in connection with an instruction execution system, apparatus, or device.

Program code embodied on a computer readable medium may be transmitted using any appropriate medium, including but not limited to wireless, wireline, optical fiber cable, RF, etc., or any suitable combination of the foregoing.

Computer program code for carrying out operations for aspects of the present invention may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, Smalltalk, C++ or the like and conventional procedural programming languages, such as the “C” programming language or similar programming languages. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

Aspects of the present invention are described with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems) and computer program products according to embodiments of the invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

These computer program instructions may also be stored in a computer readable medium that can direct a computer, other programmable data processing apparatus, or other devices to function in a particular manner, such that the instructions stored in the computer readable medium produce an article or manufacture including instructions which implement the function/act specified in the flowchart and/or block diagram block or blocks.

The computer program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other devices to cause a series of operational steps to be performed on the computer, other programmable apparatus or other devices to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide processes for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

Referring now to FIG. 10, according to an exemplary embodiment of the present invention, a computer system 1001 can comprise, inter alia, a central processing unit (CPU) 1002, a memory 1003 and an input/output (I/O) interface 1004. The computer system 1001 is generally coupled through the I/O interface 1004 to a display 1005 and various input devices 1006 such as a mouse and keyboard. The support circuits can include circuits such as cache, power supplies, clock circuits, and a communications bus. The memory 1003 can include RAM, ROM, disk drive, tape drive, etc., or a combination thereof. Exemplary embodiments of present invention may be implemented as a routine 1007 stored in memory 1003 (e.g., a non-transitory computer-readable storage medium) and executed by the CPU 1002 to process the signal from a signal source 1008. As such, the computer system 1001 is a general-purpose computer system that becomes a specific purpose computer system when executing the routine 1007 of the present invention.

The computer system 1001 also includes an operating system and micro-instruction code. The various processes and functions described herein may either be part of the micro-instruction code or part of the application program (or a combination thereof) which is executed via the operating system. In addition, various other peripheral devices may be connected to the computer system 1001 such as an additional data storage device and a printing device.

The flowchart and block diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods and computer program products according to various embodiments of the present invention. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of code, which comprises one or more executable instructions for

implementing the specified logical function(s). It should also be noted that, in some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.

The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of the invention. As used herein, the singular forms “a,” “an” and “the” are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms “comprises” and/or “comprising,” when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

The corresponding structures, materials, acts, and equivalents of all means or step plus function elements in the claims below are intended to include any structure, material, or act for performing the function in combination with other claimed elements as specifically claimed. The description of the present invention has been presented for purposes of illustration and description, but is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to

those of ordinary skill in the art without departing from the scope and spirit of the invention. The embodiment was chosen and described to best explain the principles of the invention and the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.

## CLAIMS

What is claimed is:

1. A method for visualizing engineering tasks in a multidisciplinary engineering environment, comprising:  
  
displaying an engineering application user-interface (UI) area; and  
  
displaying a task system UI area in the engineering application UI area,  
  
wherein the task system UI area is movable over the engineering application UI area and a task is an association object-action that integrates with a multidisciplinary engineering system.
2. The method of claim 1, wherein tasks are sequentially listed in the task system UI area.
3. The method of claim 1, wherein tasks are visualized in graph form in the task system UI area.
4. The method of claim 1, wherein tasks are visualized in the engineering application UI area.
5. The method of claim 1, wherein tasks are sorted by action and objects are listed underneath corresponding actions.

6. The method of claim 1, wherein tasks are sorted by objects and actions are listed underneath corresponding objects.

7. The method of claim 1, wherein displayed tasks are filtered according to their domain, type, maturity or dependency level.

8. The method of claim 1, wherein tasks are ordered and visualized depending on which action has to be completed first.

9. The method of claim 1, further comprising updating displayed tasks in real-time.

10. The method of claim 9, wherein updating displayed tasks comprises:  
changing data in the multidisciplinary engineering system in response to an action performed by a user in the engineering application UI area;  
notifying a task system that the data has been changed in the multidisciplinary engineering system;  
checking, in the task system, whether the new data completes a task or if another task has to be created; and  
updating the task system UI area with a new task.

11. The method of claim 1, wherein a task is executed directly, indirectly or step-by-step by a user.

12. A system for visualizing engineering tasks in a multidisciplinary engineering environment, comprising:

- a memory device for storing a program;
- a processor in communication with the memory device, the processor operative with the program to:
  - display an engineering application user-interface (UI) area; and
  - display a task system UI area in the engineering application UI area,

wherein the task system UI area is movable over the engineering application UI area and a task is an association object-action that integrates with a multidisciplinary engineering system.

13. The system of claim 12, wherein tasks are sequentially listed in the task system UI area.

14. The system of claim 12, wherein tasks are visualized in graph form in the task system UI area.

15. The system of claim 12, wherein tasks are visualized in the engineering application UI area.

16. The system of claim 12, wherein tasks are sorted by action and objects are listed underneath corresponding actions, or tasks are sorted by objects and actions are listed underneath corresponding objects.

17. A computer program product for visualizing engineering tasks in a multidisciplinary engineering environment, comprising:

a non-transitory computer readable storage medium having computer readable program code embodied therewith, the computer readable program code comprising:

computer readable program code configured to perform the steps of:

displaying an engineering application user-interface (UI) area; and

displaying a task system UI area in the engineering application UI area,

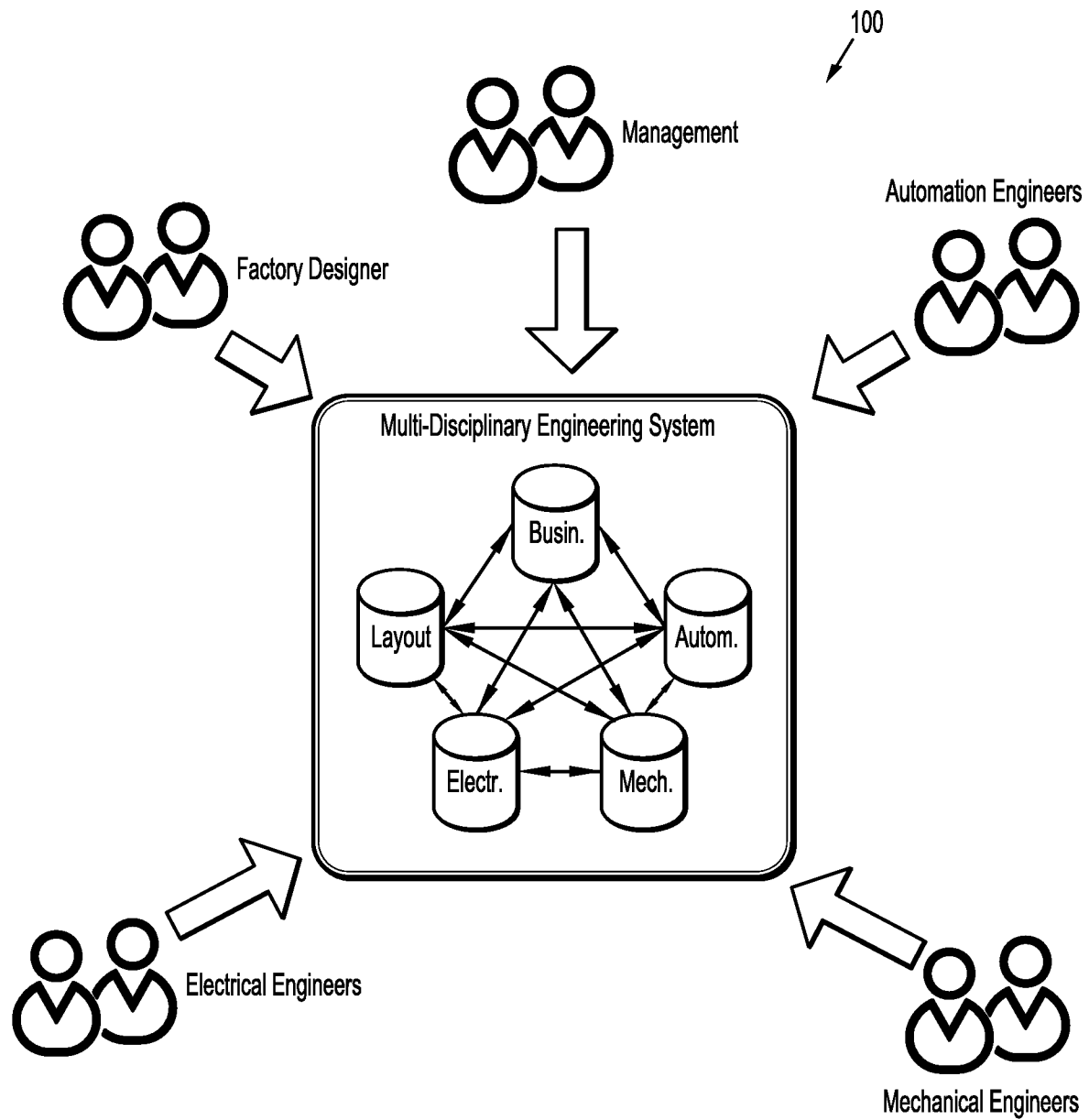
wherein the task system UI area is movable over the engineering application UI area and a task is an association object-action that integrates with a multidisciplinary engineering system.

18. The computer program product of claim 17, wherein tasks are sequentially listed in the task system UI area.

19. The computer program product of claim 17, wherein tasks are visualized in graph form in the task system UI area.

20. The computer program product of claim 17, wherein tasks are visualized in the engineering application UI area.

1/11

**FIG. 1**

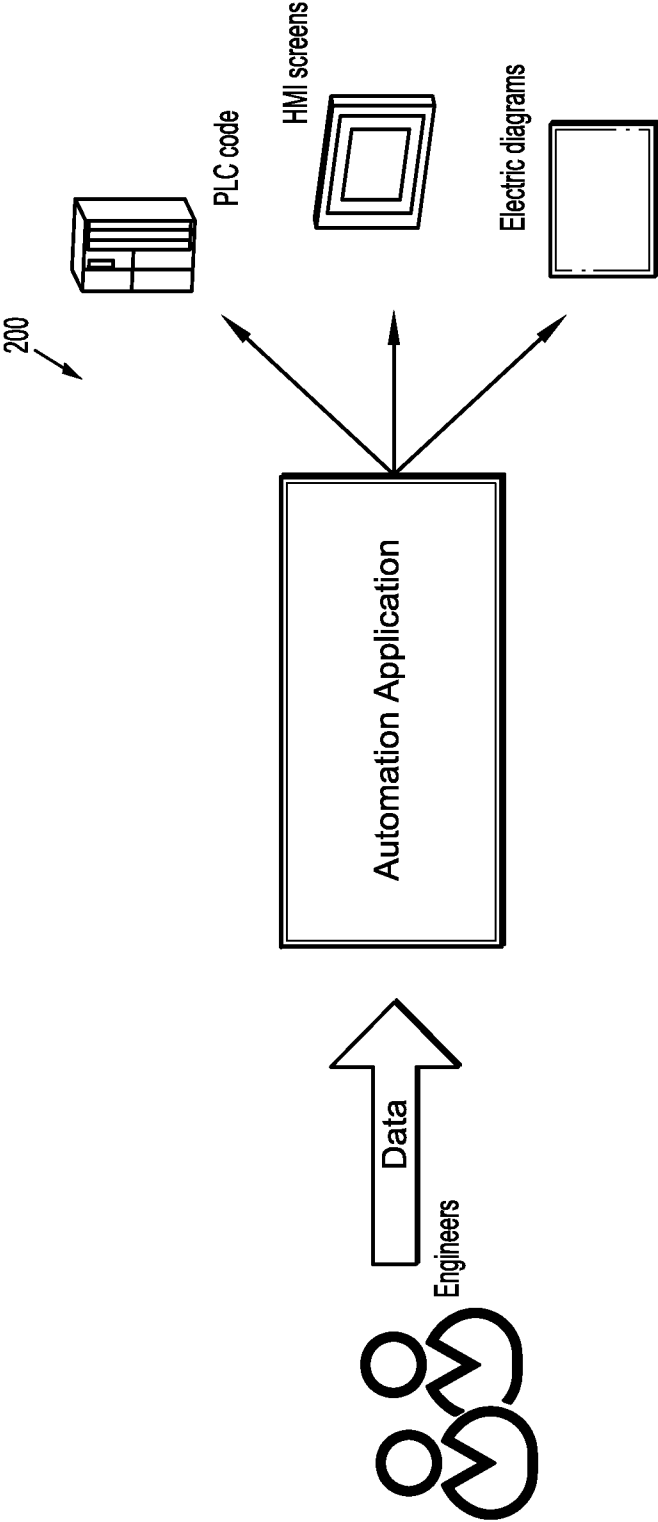


FIG. 2

3/11

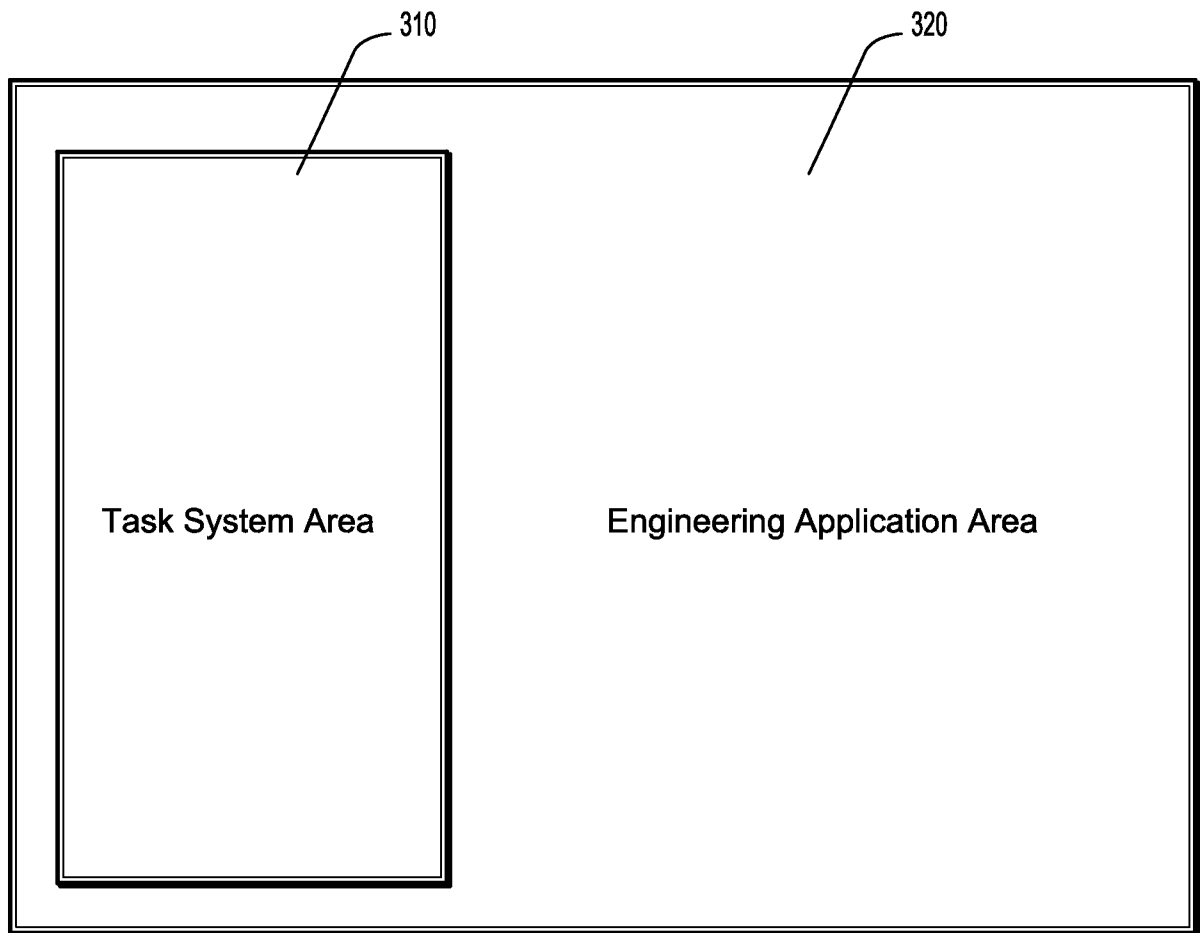


**FIG. 3**

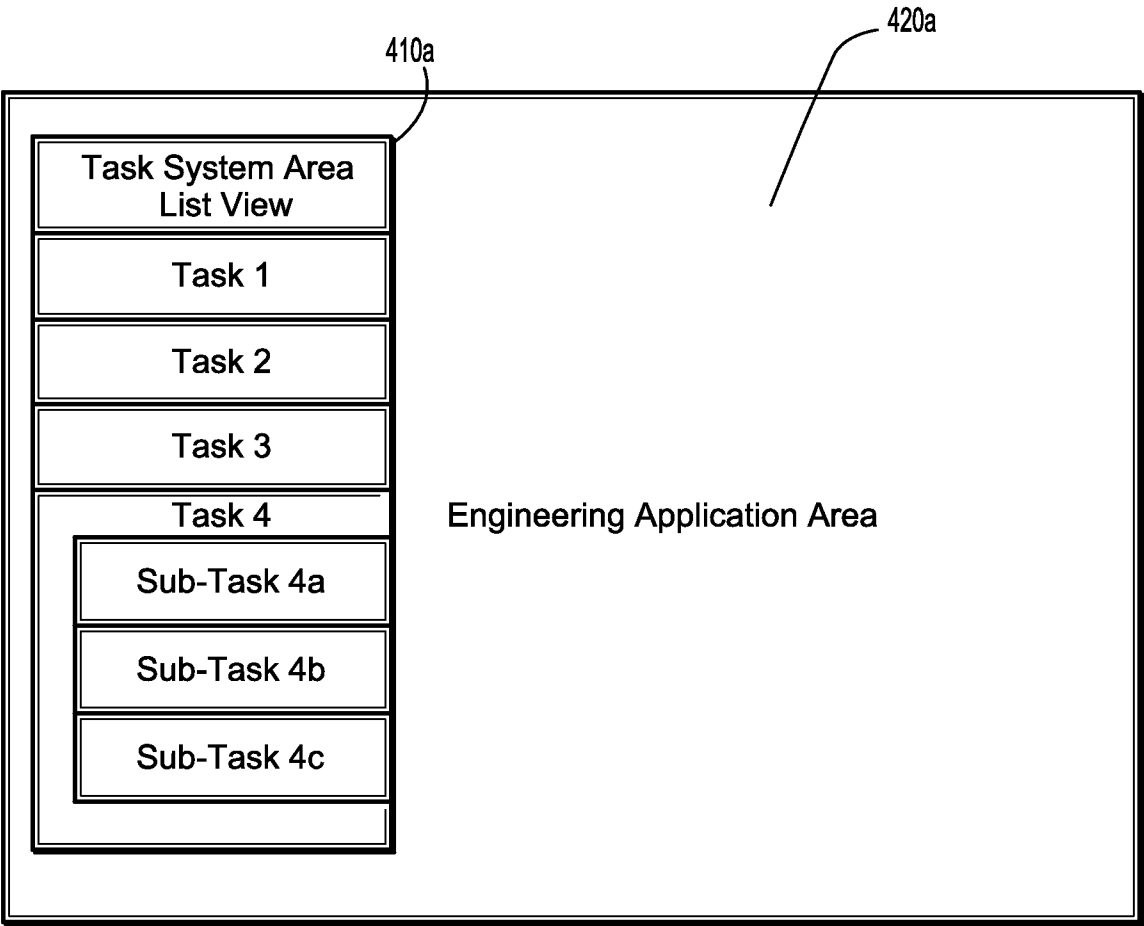
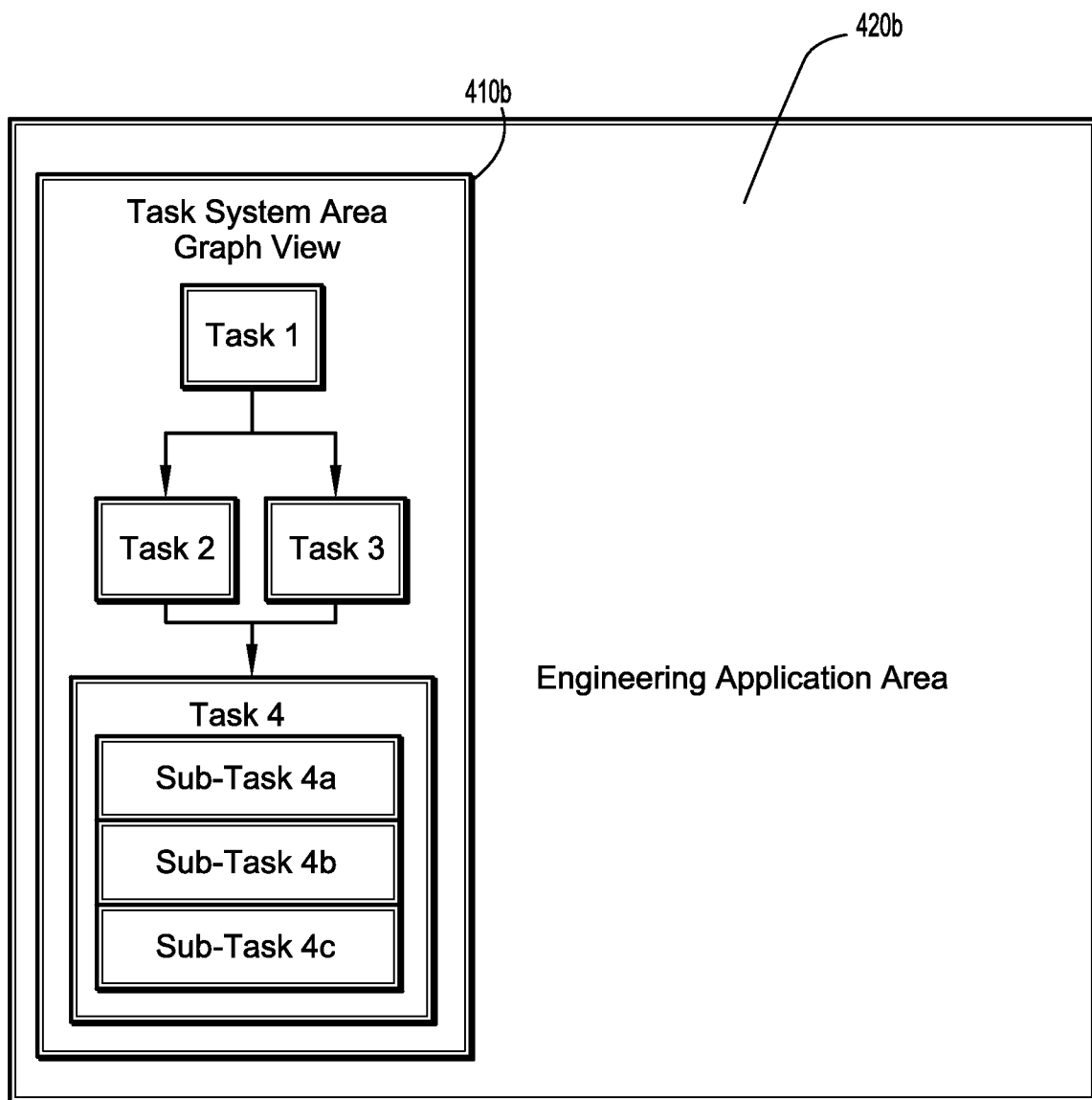
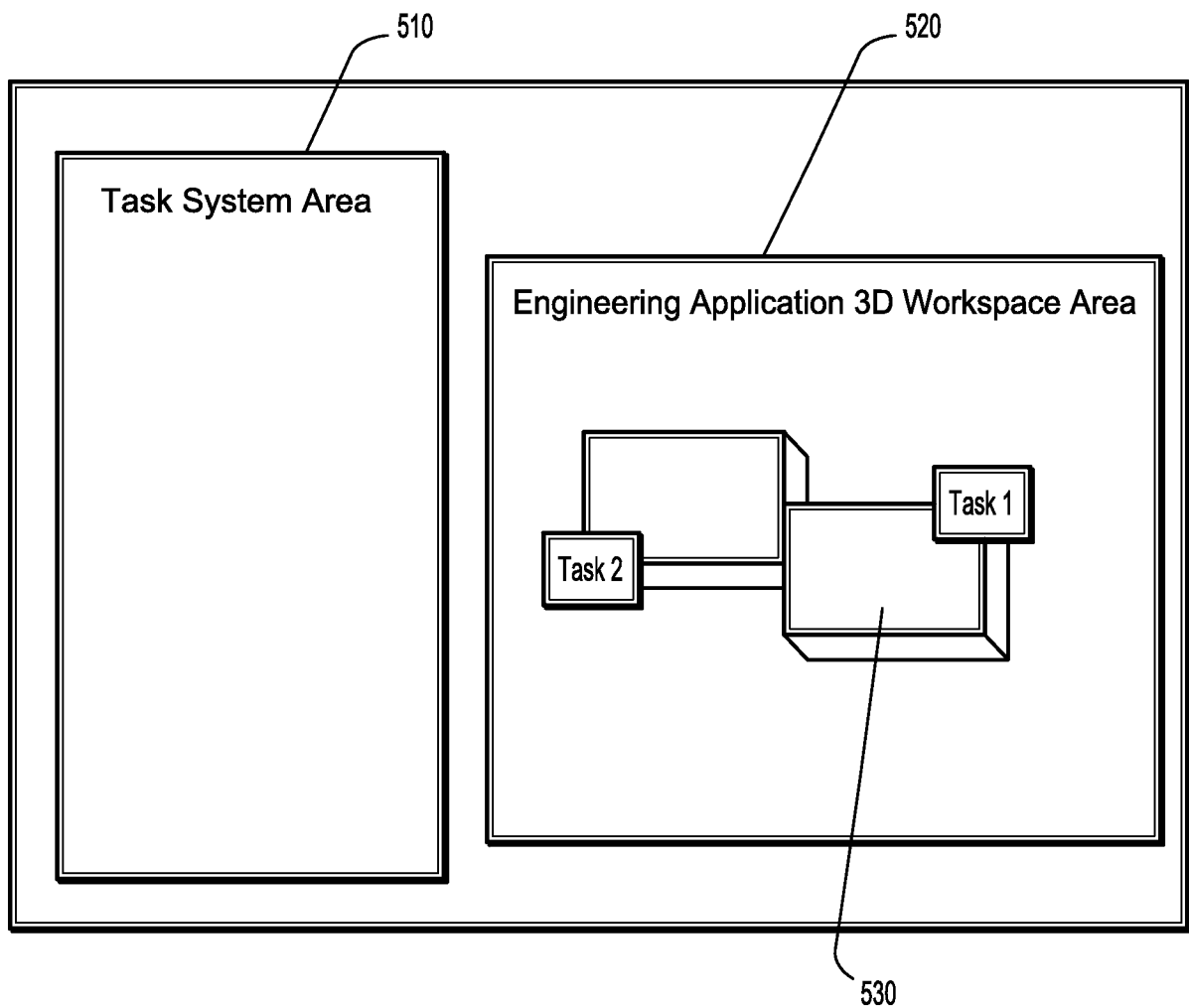


**FIG. 4A**

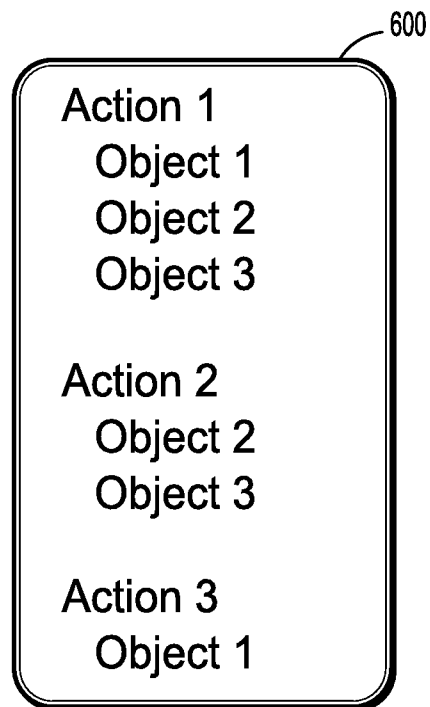
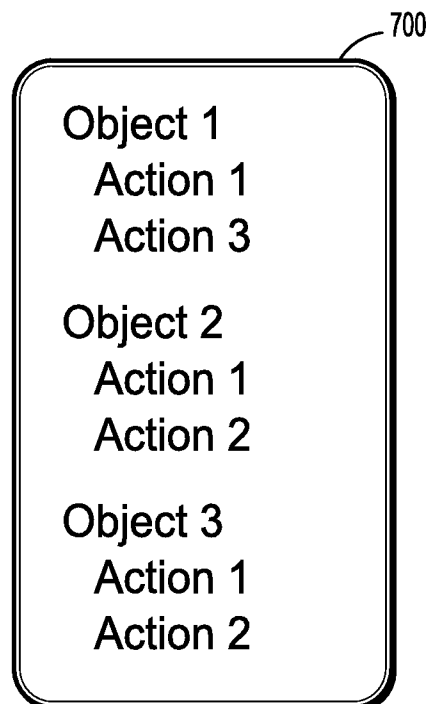
5/11

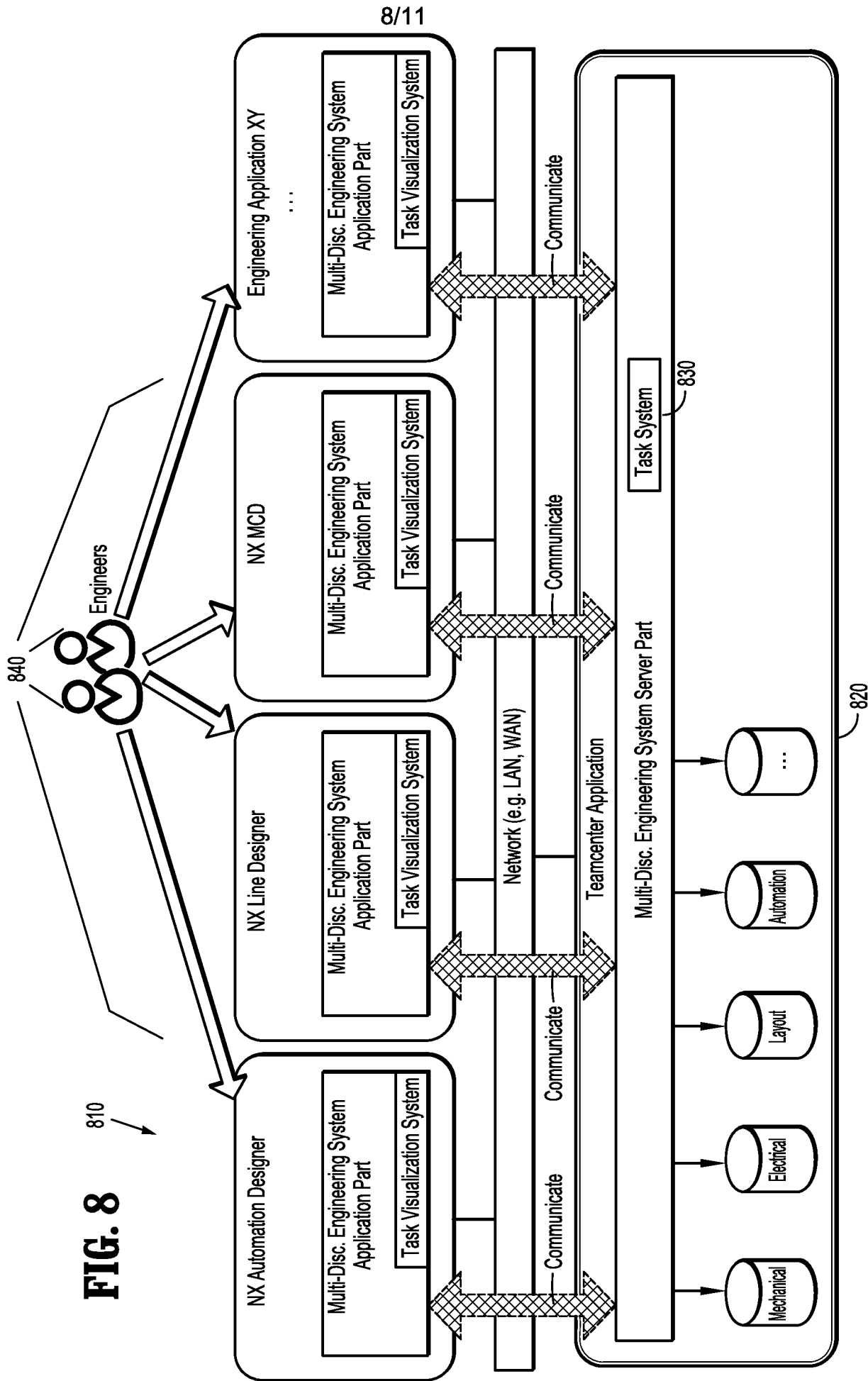
**FIG. 4B**

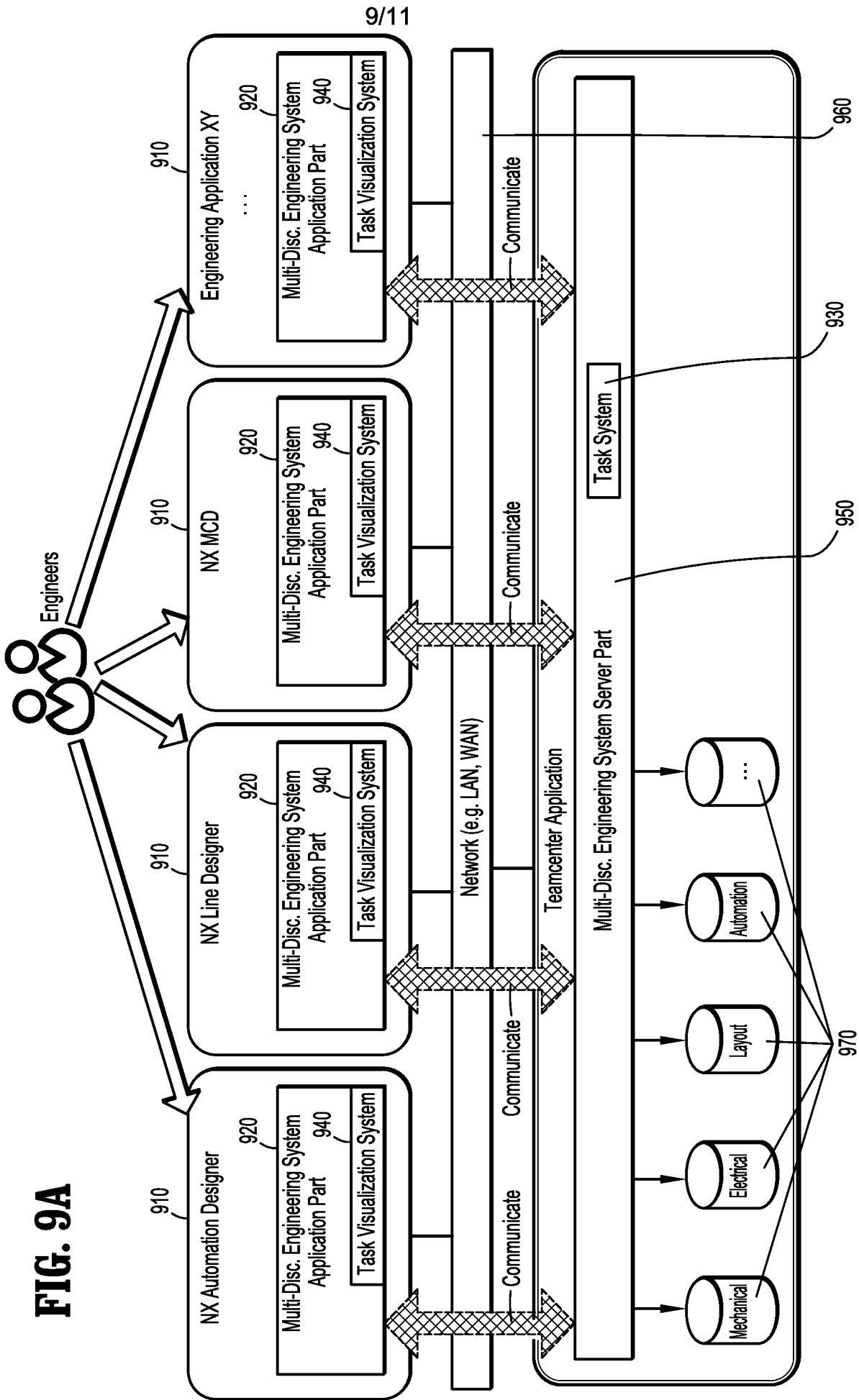
6/11

**FIG. 5**

7/11

**FIG. 6****FIG. 7**





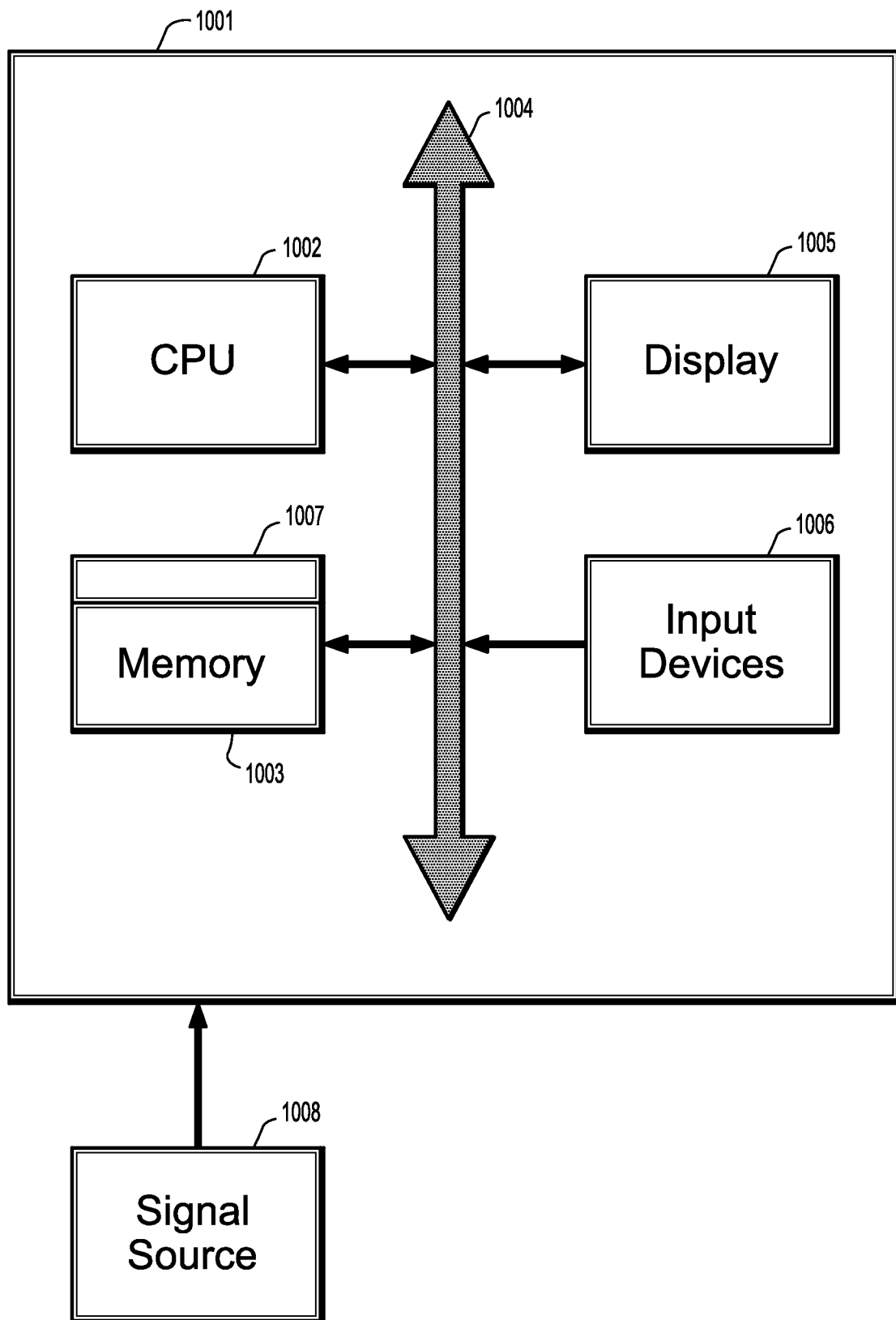
10/11

940

| Task Navigator                 |                |              |
|--------------------------------|----------------|--------------|
| Name                           | Functional ... | Location Tag |
| [-] Assign Function            |                |              |
| [-] = RB : Robot               |                |              |
| [-] = RB : Robot               |                |              |
| [-] = CV : Conveyor            |                |              |
| [-] A : SIMATICS7 300          |                |              |
| [-] B : Sensor                 |                |              |
| [-] S : Safety Switch          |                | +L1          |
| [-] = FX : Fixture             |                |              |
| [-] Assign Location            |                |              |
| [-] = CV1 : Conveyor           | = =RC1=CV1     |              |
| [-] = CV : Conveyor            |                |              |
| [-] A : SIMATICS7 300          |                |              |
| [-] B : Sensor                 |                |              |
| [-] = FX : Fixture             |                |              |
| [-] Select Template Components |                |              |
| [-] = FX : Fixture             |                |              |
| [-] Assign PLC to PLC Group    |                |              |
| G1 : PLCGroup 1                |                |              |
|                                |                |              |

**FIG. 9B**

11/11

**FIG. 10**