Abstract: A System, method, device, computer program, and/or interface for providing a user interface (UI) to a user. The method may include the act of accessing user profile information including stored task information for one or more users of a plurality of users. The method may also include the act of assigning a task of a plurality of tasks to a user of the plurality of users. The method may further include the act of determining whether the assigned task corresponds with the stored task information for the user. The method may also include the act of setting a working environment based upon the determination. Further, the method may include the act of rendering the set working environment using the UI. The task may be selected from a plurality of tasks. The method may also determine the tasks from a workflow such as a business process workflow.
— before the expiration of the time limit for amending the
claims and to be republished in the event of receipt of
amendments (Rule 48.2(h))
SYSTEM FOR DYNAMICALLY CONFIGURING TASK-BASED ENVIRONMENT
AND METHOD OF OPERATION THEREOF

FIELD OF THE PRESENT SYSTEM:
The present system relates to at least one of a system, method, user interface, and apparatus for configuring a working environment and setting a dynamic task-oriented user interface based on user-specific information.

BACKGROUND OF THE PRESENT SYSTEM:
Typically, in an enterprise environment, a business process or project may be described using a sequence of steps known as tasks which may define a workflow. The tasks may include essential tasks which must be completed before the business process or project may be considered to be complete. In today's technologically-complex environment, in an attempt to complete business processes or projects or other goals in a timely and efficient manner, more and more enterprises are increasingly relying upon using information technology (IT) systems, such as business process management systems (BPMs) or workflow systems (WSs), to manage the execution of their business processes or projects by assigning tasks to individual workers and tracking and/or monitoring their progress based on a pre-defined process model.

In a typical enterprise environment, many enterprise workers contribute to the enterprise by performing daily operations such as completing tasks that are assigned to the workers by information technology systems. These IT systems may use BPM and/or workflow applications, to create tasks and/or assign these tasks to workers. When tasks are assigned, the IT systems typically notify workers of these tasks. For example, the workers will typically receive notification of a task at their task inbox. When workers open their task, they will usually see some description of the task and a form to submit information back when they complete the task. The IT systems can typically track when, and to whom, a task was assigned. When individual workers report that their tasks are completed, the IT systems can create or select new tasks to assign to the workers according to the design of the business process model.
However, conventional IT systems, such as BPM and workflow systems, do not address how workers work on their tasks. Accordingly, the workers must select tools and other aids of their choice to perform their tasks to achieve the necessary results. This can take time and lead to inefficiency. For example, when tasks are assigned to workers, they must typically set up their "tools" in a working environment to work on the assigned tasks every time they work on the tasks. Such tools can include work-in-progress files, productivity tools, web sites, applications, etc. Unfortunately, many tasks are quite complicated and may require days, weeks, or even years to complete. Workers generally need to use multiple tools to work on their tasks, and they normally cannot keep all of the tools running from the time that they start a task until they finish the task. Further, in today's multi-tasking enterprise environment, workers may need to switch to another task (i.e., between tasks) or they may need to shut down their computer at the end of a workday. Therefore, it is quite common for productivity to suffer because workers may have to set up a different work environment to perform another task. This problem may be exacerbated as workers are burdened with larger workloads, and, thus, must frequently switch between projects. Further, each time a worker shuts off his/her computer, the worker may have to re-launch all the tools when the computer is started up again. This typically happens when a worker goes home each day or switches tasks.

Unfortunately, conventional BPM and workflow systems do not build task specific working environments for workers to easily and efficiently accomplish their tasks.

Thus, conventional BPM workflow systems generally assign tasks that although may include consideration of a worker's skillset and abilities, however it does not address the need and requirements of individuals when assigning tasks. For example, conventional BPM workflow systems typically assign tasks to users, provide users with the information needed to execute their tasks, and provide a mechanism for users to submit the results back to the BPM or workflow systems. Accordingly, conventional BPM workflow systems may determine when a task has been completed and initialize a next task and assign this task to a worker. However,
other than in this way, conventional BPM workflow systems do little to increase productivity of the worker.

Further, as complex user interfaces (UIs) become more common, conventional BPM or workflow systems have retained proprietary formats typical of computer systems which were common during the previous decade and do not provide up-to-date enhanced UI functionality. Further, although there have been attempts to provide UIs with enhanced UI functionality, these attempts fail to provide a dynamic task-based BPM or workflow system. For example, iGoogle™ and/or Netvibes™ applications provide users with the capability to drag and drop different widgets into a browser in order to syndicate information distributed by different sources. These widget systems allow users to preserve different widget panels so that users can launch the same set of widgets each time the user uses the application. Further, an even more advanced UI application known as DreamForce™ allows users to define a relationship between widgets. Accordingly, users can link the output of one widget to the input of another widget. When these links are set, the widget system can automatically exchange data among widgets so that when one widget is updated, all the related widgets can be updated accordingly. However, these systems and applications do not preserve the state of the widget, so even though a user can use the widget to work on his tasks, he will always have to start with a default state of any widget in the widget system. Further, these systems do not provide a dynamic task-oriented user interface based on user-specific information for efficiently accomplishing a workflow task.

Moreover, although some widget applications like Adobe Genesis™ have the ability to preserve state information of individual "tiles" within a working environment, Genesis™ doesn't provide a user configurable task and task instance association with a widget and working panel. Accordingly, the Genesis™ application does not associate a working environment panel with a task type or template. In other words, users cannot define a working environment for a specific task type so that the same environment can be launched automatically with predefined task instances. Accordingly, Genesis™ does not provide a dynamic task-oriented user interface based on user-specific information.
Moreover, none of the prior art task-oriented user interfaces provide a system, method, user interface, and device to configure a working environment and set a task-oriented user interface base upon contextual information.

**SUMMARY OF THE PRESENT SYSTEM:**

It is an object of the present system to overcome disadvantages and/or make improvements in the prior art.

The present system includes a system, method, device, and interface for rendering a user interface for completing a task of a plurality of tasks of a workflow. The workflow may correspond with, for example, a business workflow such as a BPM workflow.

In accordance with the present system, task instances may be set in accordance with a context-sensitive query that may include one or more of user profile information, user task history, user preferences, and task type.

According to an aspect of the present system, there is provided a method of providing a UI including the acts of: accessing user profile information including stored task information for one or more users of a plurality of users; assigning a task of a plurality of tasks to a user of the plurality of users; determining whether the assigned task corresponds with the stored task information for the user; setting a working environment including a plurality of applications based upon the determination; and rendering the set working environment using the UI. The method may further include an act of informing the user of the task before a user can launch the task.

The method may also include an act of storing information related to the task in accordance with a selected save option of a plurality of save options. Further, the act of setting may include: setting a first working environment when it is determined that the assigned task corresponds with the stored task information for the user; and setting a second working environment different from the first working environment when it is determined that the assigned task does not correspond with the stored task information for the user. Moreover, the first working environment may include an application configuration selected by the user. Further, the second working environment may include a default application configuration.
Moreover, the method may include acts of: launching the assigned task; inputting, by the user, instance values; and/or storing user instance values in accordance with a save option when the task is shut down.

According to yet another aspect of the present system, there is provided a system which may provide a UI. The system may include a controller which accesses user profile information that includes stored task information for one or more users of a plurality of users, assigns a task of a plurality of tasks to a user of the plurality of users, determines whether the assigned task corresponds with the stored task information for the user, and/or sets a working environment based upon the determination. The system may also include a rendering device which renders the set working environment using the UI. According to the system, the controller may render information to inform the user of the assigned task on the rendering device.

The system may also include a storage device which may store information related to the task in accordance with a selected save option of a plurality of save options. Further, the controller may: set a first working environment when it is determined that the assigned task corresponds with the stored task information for the user; and/or may set a second working environment different from the first working environment when it is determined that the assigned task does not correspond with the stored task information for the user. According to the system, the first working environment may include an application configuration selected by the user. Further, the second working environment may include a default application configuration.

Further, according to the system, the controller may launch the assigned task, input user instance values entered by the user, and/or store the user instance values in accordance with a save option.

According to yet a further aspect of the present system, there is provided a computer program that may be stored on a computer readable memory medium, the computer program may be configured to system provide a UI to accomplish a task, the computer program may include a program portion configured to: access user profile information that include stored task information for one or more users of a plurality of users; assign a task of a plurality of tasks to a user of the plurality of users; determine whether the assigned task corresponds with the stored task
information for the user; set a working environment based upon the determination;
and render the set working environment using the UI. The program portion may also
be configured to render information to inform the user of the task. Moreover, the
program portion may be configured to store on a storage device information related
to the task in accordance with a selected save option of a plurality of save options.

Further, the program portion may be configured to: set a first working
environment when it is determined that the assigned task corresponds with the
stored task information for the user; and set a second working environment different
from the first working environment when it is determined that the assigned task does
not correspond with the stored task information for the user. Moreover, the first
working environment may include an application configuration selected by the user.
Further, the second working environment may include a default application
configuration. According to the computer program, the program portion may be
configured to: launch the assigned task, input user instance values entered by the
user, and/or store the user instance values in accordance with a save option.

According to yet a further aspect of the present system, there is discloses a
method for rendering a working UI to a user, the method may include one or more of
the acts of: receiving a business process flow comprising a series of successive
tasks; assigning a task of the series of successive tasks to the user; determining
whether the assigned task is to be performed for the first time; retrieving a default UI
template for the task when it is determined that the assigned task is to be performed
for the first time, the default UI template including applications for performing the
task; rendering the default UI template for the task; customizing the UI based on
input from the user; and saving the customized UI when the task is completed by the
first user.

The method may further include the acts of: determining whether the user
customized UI exists; retrieving the customized UI template when it is determined
that the customized UI exists; and/or rendering the user customize UI template. The
method may also include acts of requesting to save the customized UI according to a
save option of a plurality of save options; and saving the customized UI in
accordance with one of the save options of the plurality of save options. The
customized UI may further include a working environment which includes one or
more applications and work in progress (WIP) data which includes user entered information. According to the method, the plurality of save options may include first to third save options. With the first save option, the working environment is saved as task type information. Further, according to the second save option, the working environment is saved as a task type information and the WIP data is saved as task instance information corresponding to a task instance. Further, according to the third save option, the working environment and WIP data is saved as task instance information.

According to yet another aspect of the present system, there is provided a method to provide a user interface (UI) to one or more users of a plurality of users. The method may include one or more of the acts of: accessing task information assigned to a user of a plurality of users, the task information including task type information related to one or more task types and task instance information corresponding to a task instance of a corresponding task type; setting a working environment based upon the accessed task information; rendering the set working environment using the UI; customizing the working environment according to a user's selection; and saving the working environment in accordance with a save option of a plurality of save options. According to one save option of the plurality of save options, the task type information may be updated to correspond to the working environment. According to a further save option of the plurality of save options, the task type information may be updated to correspond with the working environment and the task instance information may be updated to correspond with work in progress (WIP) information comprising at least one user entry.

Further, in the method, according to a different save option of the plurality of save options, the task instance information may be updated to correspond with the working environment and work in progress (WIP) information including at least one user entry. The user entry may correspond with an information arrangement that may be changeable by the user, such as a window, a text box, a radio button box, a slider, a check box, a drop down list, etc., that may be displayed, for example, in an element of a graphical user interface (GUI) such as, for example, a widget (or control). The user entry may provide an interaction point for manipulation of a given
type of data. According the instance information related to the user input may include the user entry.

Accordingly, the present system may assemble tools (e.g., software applications, widgets, instance values, etc.) and may associate these tools with a specific task instance defining a working environment based on a BPM/workflow and/or an individual worker's personal configuration for each specific task instance.

BRIEF DESCRIPTION OF THE DRAWINGS:

The invention is explained in further detail, and by way of example, with reference to the accompanying drawings wherein:

FIG. 1 shows a UI in accordance with an embodiment of the present system;
FIG. 2A show a UI which corresponds with the task of FIG. 1;
FIG. 2B show a UI which corresponds with the task of FIG. 1;
FIG. 2C shows a working environment for a completed task according to an embodiment of the present system;
FIG. 2D shows a working environment for a manager according to an embodiment of the present system;
FIG. 3 shows a flow diagram that illustrates a process in accordance with an embodiment of the present system;
FIG. 4 shows a flow of an embodiment of the present system;
FIG. 5 shows a schematic of an embodiment of the present system; and
FIG. 6 shows a system 600 in accordance with another embodiment of the present system.

DETAILED DESCRIPTION OF THE PRESENT SYSTEM:

The following are descriptions of illustrative embodiments that when taken in conjunction with the following drawings will demonstrate the above noted features and advantages, as well as further ones. In the following description, for purposes of explanation rather than limitation, illustrative details are set forth such as architecture, interfaces, techniques, element attributes, etc. However, it will be apparent to those of ordinary skill in the art that other embodiments that depart from these details would still be understood to be within the scope of the appended claims. Moreover, for the
purpose of clarity, detailed descriptions of well known devices, circuits, tools, techniques and methods are omitted so as not to obscure the description of the present system. It should be expressly understood that the drawings are included for illustrative purposes and do not represent the scope of the present system. In the accompanying drawings, like reference numbers in different drawings may designate similar elements.

For purposes of simplifying a description of the present system, the terms "operatively coupled", "coupled" and formatives thereof as utilized herein refer to a connection between devices and/or portions thereof that enables operation in accordance with the present system. For example, an operative coupling may include one or more of a wired connection and/or a wireless connection between two or more devices that enables a one and/or two-way communication path between the devices and/or portions thereof. For example, an operative coupling may include a wired and/or wireless coupling to enable communication between a content server and one or more user devices. A further operative coupling, in accordance with the present system may include one or more couplings between two or more user devices, such as via a network source, such as the content server, in accordance with an embodiment of the present system.

The term rendering and formatives thereof as utilized herein refer to providing content, such as digital media, such that it may be perceived by at least one user sense, such as a sense of sight and/or a sense of hearing. For example, the present system may render a user interface on a display device so that it may be seen and interacted with by a user. Further, the present system may render audio visual content on both of a device that renders audible output (e.g., a speaker, such as a loudspeaker) and a device that renders visual output (e.g., a display). To simplify the following discussion, the term content and formatives thereof will be utilized and should be understood to include audio content, visual content, audio visual content, textual content and/or other content types, unless a particular content type is specifically intended, as may be readily appreciated.

The system, device(s), method, user interface, etc., described herein address problems in prior art systems. In accordance with an embodiment of the present system, a system, method, device, computer program, and interface for rendering a
UI for a users convenience. The UI may include one or more applications which are necessary to complete an assigned task. The method may include the act of accessing user profile information including stored task information for one or more users of a plurality of users. The method may also include the act of assigning a task of a plurality of tasks to a user of the plurality of users. The method may further include the act of determining whether the assigned task corresponds with the stored task information for the user. The method may also include the act of setting a working environment based upon the determination. Further, the method may include the act of rendering the set working environment using the UI. The task may be selected from a plurality of tasks. The method may also determine the tasks from a workflow. The method may also assign tasks to two or more users. In addition, the present system may collect other statistics related to the user and/or user device (e.g., an MS) in accordance with the present system, such as a relative time of an action, geo-location, network, detected content item, etc.

The user interaction with and manipulation of the computer environment is achieved using any of a variety of types of human-processor interface devices that are operationally coupled to the processor controlling the displayed environment. A common interface device for a user interface (UI), such as a graphical user interface (GUI) is a mouse, trackball, keyboard, touch-sensitive display, etc. For example, a mouse may be moved by a user in a planar workspace to move a visual object, such as a cursor, depicted on a two-dimensional display surface in a direct mapping between the position of the user manipulation and the depicted position of the cursor. This is typically known as position control, where the motion of the depicted object directly correlates to motion of the user manipulation.

An example of such a GUI in accordance with an embodiment of the present system is a GUI that may be provided by a computer program that may be user invoked, such as to enable a user to select and/or classify/annotate content. In accordance with a further embodiment, the user may be enabled within a visual environment, such as the GUI, to classify content utilizing a reduced description palette to simplify content analysis, presentation, sharing, etc., of separate content portions in accordance with the present system. To facilitate manipulation (e.g.,
content selection, annotation, sharing, etc.) of the content, the GUI may provide different views that are directed to different portions of the present process.

For example, the GUI may present a typical UI including a windowing environment and as such, may include menu items, pull-down menu items, pop-up windows, etc., that are typical of those provided in a windowing environment, such as may be represented within a Windows™ Operating System GUI as provided by Microsoft Corporation and/or an OS X™ Operating System GUI, such as provided on an iPhone™, MacBook™, iMac™, etc., as provided by Apple, Inc., and/or another operating system. The objects and sections of the GUI may be navigated utilizing a user input device, such as a mouse, trackball, finger, and/or other suitable user input. Further, the user input may be utilized for making selections within the GUI such as by selection of menu items, window items, radio buttons, pop-up windows, for example, in response to a mouse-over operation, and other common interaction paradigms as understood by a person of ordinary skill in the art.

Similar interfaces may be provided by a device having a touch sensitive screen that is operated on by an input device such as a finger of a user or other input device such as a stylus. In this environment, a cursor may or may not be provided since location of selection is directly determined by the location of interaction with the touch sensitive screen. Although the GUI utilized for supporting touch sensitive inputs may be somewhat different than a GUI that is utilized for supporting, for example, a computer mouse input, however, for purposes of the present system, the operation is similar. Accordingly, for purposes of simplifying the foregoing description, the interaction discussed is intended to apply to either of these systems or others that may be suitably applied.

FIGs. 1-2D will be discussed below to facilitate a discussion of illustrative embodiments of the present system. FIG. 1 shows a UI in accordance with an embodiment of the present system. A UI 100 may correspond with an initial working environment that is rendered the first time a task type is launched by a user. Accordingly, as FIG. 1 illustrates a working environment that corresponds with a task type that has been assigned to a user. One or more task instances have not yet been assigned to the user. Thereafter, the user may reconfigure the working environment as desired. The system may save the user reconfigured working
environment, for example, in a memory of the system, in accordance with a selected save option. Thereafter, when the task type is relaunched (e.g., launched a second time, etc.), the system may render the working environment in accordance with the selected save option. This process is illustrated with reference to FIGs. 2A-2B of which show UIs each having a working environment that corresponds with the task type of FIG. 1.

For example, with reference to FIG. 2A, a UI 200A having a working environment which corresponds with the working environment shown in UI 100 of FIG. 1 is shown. The UI 200A corresponds with the same task type (e.g., a loan application task type) as shown in the UI 100 of FIG. 1. However in the UI 200, the user has customized the working environment by, for example, selecting certain tools 106-1 - 106-7 (which may include applications (e.g., desktop and/or server applications), widgets, etc.) to use in a work area 106 that may be used to perform the task. These tools may be selected from areas which may include, for example, a recommended task area 108 and a user tools area 110 that may include recommended or user preferred tools. Further, the user has input work in progress data (WIP) such as "user input task data" which may include any user input instance information such as, for example, text or a change of setting (e.g., selecting a radio button, checking a box, etc.) that the user may enter in a user input entry area of a selected tool 106-x such as a text entry box, a radio button, etc. For example, the user may have entered WIP information 142 into a text entry area 140-T and/or may have selected/changed a setting of radio buttons 140-R, etc. After customizing the working environment and/or selecting/entering user input task data, the working environment of UI 200A may be saved by, for example, the system, in accordance with certain save options. Various save options will be described in more detail below. Thereafter, the task instance corresponding with the working environment of UI 200A may be shut by, for example, switching tasks, shutting the system down, logging off, selecting shut menu selection, etc. Accordingly, when the task is shut, the working environment corresponding with the task type or task instance may be closed following a save operation. Then, when a user desires to relaunch a previously saved working environment of a task type or instance, the user may select the corresponding task type or task instance and the system may download or
otherwise access the saved data corresponding with the working environment of the selected task type or task instance. As this data may have been saved in accordance with one or more save options, certain inputs and/or tools in the relaunched working environment may be the same or changed depending upon a selected save option. For example, as shown, in FIG. 2B, the user input WIP information 142 that was input in the UI 200A of FIG. 2A may not rendered when a task instance is relaunched (as shown in FIG. 2B) dependent upon a previous save option in the work environment of FIG. 2A. The instance information may include various instance information such as text entries (e.g., names, addresses, etc.) in a text or drop-down box, radio button selections, check box selections, and other instance information as may be typical in a windows-based environment. The information relating a work environment of a task type and/or task instance may be saved in a memory of the system in accordance with corresponding user information. Thus, although information of a working environment of a task type may be common to one or more user's, after a user modifies a working environment of a certain task type, the corresponding information of the working environment of the modified task type may be saved in accordance with corresponding user information. Thus, after a user modifies a working environment of, for example, a task type (that may be similar for one or more users), the user modified working environment of the task type is saved in accordance with corresponding user information such that when the user relaunches the task type, the user modified working environment may be rendered.

Thus, in accordance with an embodiment of the present system, multiple "instances" of the tools and/or applications may be created so that each user can modify their own tool and/or application instances. Accordingly, a working environment (e.g., of a task type of task instance) may be based upon different tool/application instances depending on the identity of the current user.

Further, the system may switch task (types or instances) and/or may present several security features which may be desirable in an enterprise environment. First, with regard to security features, the system may log all data to track user's habits.

Moreover, the system may block access to working environments associated with a task type or instance once the task type or instance, or a milestone associated with the task type or instance, has been completed. The type of data that may be blocked
may be set by the system and/or a manager of the system. For example, FIG. 2C, a
UI 200C shows having a working environment for a completed task instance "e.g.,
Morgan's Loan." Although the task instance shown in FIG. 2C is different from the
task instance shown in FIGs. 2A or 2B, it is of the same type as the task type shown
in FIG. 1A. Further, the user may launch the task instance of FIG. 2C from another
workplace environment such as, for example, the workplace environments of the task
instances shown in FIGs. 2A or 2B, by selecting a "Morgan" selection in another task
area 114. For example, if the user is in the working environment shown in FIG. 2B,
the user may select the "Morgan" selection in the other task area 114 to switch tasks
(e.g., from the Smith task instance to the Morgan task instance). Accordingly, upon
determining that the user has requested to switch to the Morgan task instance, the
system may switch task instances by launching the Morgan task instance.

Once a task type or instance is completed, the system may inform the use of a
tasks completion (e.g., see, 213, FIG. 2C) and/or prevent the user from accessing
certain data and/or tools related to the task type or instance. For example, assuming
that the Morgan task instance has been completed, the system may render certain
information (e.g., WIP data) that it deems may be accessible to a user (e.g., see,
area 211 of tool 206). However, information that is determined to be secure may be
encrypted (e.g., see, 209) or entirely restricted and will therefore not be rendered
absent a proper authorization. Further, the system may set a working environment of
a UI 200C so that certain tools, which were previously available to the user (e.g., in
an initial working environment of the task type or instance) are no longer available in
a rendered working environment once the task type or instance has been completed.
In other words, once the system determines that a task type or instance, or a
milestone in a task type or instance, has been completed, the system may associate
a new user interface with the task type or instance. Moreover, the system may select
and render certain tools to inform a user of a status of the task type or instance
and/or contact persons (e.g., see, 213 and 217). The user may then select a link,
such as link 219, to obtain more data regarding the task type or instance. Thus, the
present system may limit access to certain information once a task type or instance
or other milestone has been completed so as to enhance system security and ensure
that a user does not continue to work on a task after it has been completed. A
manager may also review this data or receive notification from the system once a
task type or instance or other milestone is completed.

FIG. 2D shows a working environment for a manager according to an
embodiment of the present system. The system may generate a working environment
based upon a hierarchical system. For example, one or more managers may have
access to certain information which their subordinates cannot access. Further, the
present system may associate certain information according to project, group, etc.
For example, a manager of an accounting group may access cost data, while a
manager of an engineering project may access design data, a manager of an IT
department may access networking data, and a manager of a shipping department
may access logistics data. The present system may also provide for collaboration by
providing tools that enable a user to share data among other users (e.g., see, tool
207). For example, a sharing tool 207 or a data link may be used to share
information among selected users.

Referring back to FIG. 1, the working environment of UI 100 may include a
GUI in accordance with an embodiment of the present system. The working
environment of the UI 100 may correspond with an initial UI and is provided in
accordance with a task type which may be assigned to a user by the present system.
The task type may be selected by the system from a plurality of tasks of a workflow
process. Further, the task type may correspond with a specific business process task
(BPT). The working environment of UI 100 may include one or more of windows,
frames, menus, title bars, applications, tasks, links, widgets, and/or features
(hereinafter "tools") that are selected by the system in accordance with the current
task so that the user may focus upon tasks in the current window. For example, UI
100 may include tools which may include one or more of a windows such as a main
window 102, a work area 106, a recommend task area 108, a user tools area 110, a
directory area 118, a time area 116, an efficiency area 112, a save area 120, an
other task area 114. Each of the tools may include, for example, a frame, menus,
and/or a title which describes the area. For example, the main window 102 may
include a title 102-T, the work area 106 may include a title 106-T, the recommend
task area 108 may include a title 108-T, the user tools area 110 may include a title
110-T, the directory area 118 may include a title 118-T, the time area 116 may
include a title 116-T, the efficiency area 112 may include a title 112-T, the save area 120 may include a title 120-T, and the other task area 114 may include a title 114-T. Each title may describe an appropriate area and may be initially set by the system in accordance with the BPT.

With respect to the main window 102 this window may include one or more of the work area 106, the recommend task area 108, the user tools area 110, the directory area 118, the time area 116, the efficiency area 112, the save area 120, and the other task area 114, if desired. Further, each of these areas may be maximized, minimized, and/or closed as may be typical in a windows operating environment.

The system may provide the capability to drag and drop tools into any tool so as to syndicate information distributed by different sources. As used herein, a frame located within another frame may be referred to as a subframe. Likewise, any window in another window may be referred to as a subwindow. The sources may include the one or more of applications including, for example, windows, menus, titlebars, applications, tasks, links, widgets, and/or features that are selected by the system in accordance with the task. For example, with respect to the work area 106, a plurality of tools such as applications or widgets such as for example, a browser application (e.g., IE, Firefox, etc.) 106-1, a credit check application 106-2, a word processor (e.g., MS Word) 106-3, a report application 106-4, a widget application 106-6, and/or other applications 106-N may operate together in the work area 106. Each of the applications 106-1 - 106-N may be linked to share data with each other. The links may be established by the system in accordance with the task and/or user selections. Thus, for example, information which is generated by the credit check application 106-2 and the word processor 106-3 may be sent to the report application 106-4 which may then prepare a corresponding report for later use. This report may then be stored by the system in a database or other memory.

The recommend task area 108 may include a list of one or more recommended tools such as GUIs, windows, menus, titlebars, applications, tasks, links, widgets, instances, and/or features (hereinafter features 108-F) that a user may select for use. For example, the user may copy or drag one or more of the features 108-F to another window (or frame) such as, for example, the work area
106, the efficiency area 112, etc. This process will be known as "selecting the tool." The user may also set one or more links to other windows, menus, titlebars, applications, tasks, links, widgets, data, values, and/or features in the GUI 100, as is known in the art. The user may copy or drag a tool to a certain area such as, for example, the main window 102 or another window in which the tool may be active.

The features 108-F may be selected in accordance with the BPT and/or user characteristics. For example, if the system determines that the user has a certain handicap, then certain accessibility options (e.g., a text to speech (TTS) and/or speech to text (STT) applications) may be provided.

The user tools area 110 may include tools such as one or more GUIs, windows, menus, titlebars, applications, tasks, links, widgets, instances, and/or features, that the system determines the user may need. For example, the system may determine that the user is likely to use one or more GUIs, windows, menus, titlebars, applications, tasks, links, widgets, instances, and/or features, based upon previous history. For example, the system may determine GUIs, windows, menus, titlebars, applications, tasks, links, widgets, instances, and/or features, that are most likely and/or most recently used by the user. The system may determine this using, for example, heuristic analysis, that may use as an input the times and/or types of GUIs, windows, menus, titlebars, applications, tasks, links, widgets, instances, and/or features, used by the user. As heuristic analysis is known in the art, for the sake of clarity, it will not be discussed. Moreover, a user preference list associated with the user may also be used to determine the contents of the user tools area 110.

The directory area 118 may include a directory which corresponds with the current process or task. For example, a project directory may show groups, leaders, tasks, etc. The current user's (e.g., "John") name may be highlighted. The user may select certain elements in the directory for more information. For example, selecting "Group 2" may expand the directory tree and/or show one or more of a hierarchy of this group, tasks, completion levels, expected and/or actual completion times of tasks, names of members, etc. This information may be useful when navigating through groups of employees and/or scheduling tasks. The system may use the expected and/or actual completion times to schedule new tasks or provide updates to others (e.g., managers, directors, persons who are working on a task which depends
upon another task being performed/completed, etc.). The system may set a security
level for each user. Accordingly, the system may provide certain information (e.g.,
group information, employment statistics for a plurality of workers such as pay,
security level, efficiency/productivity, etc.) to a manager, while providing an employee
with project work related information so that a worker can accomplish a certain task.
Thus, each user may have a certain security level and the system may determine
what the user is allowed to see based on a tool's assigned security level.

The time area 116 may include various times which may be displayed for the
user's convenience. However, this information may also be used by the system to
schedule tasks (e.g., task types, task instances, etc.), determine efficiencies of
persons/groups, for accounting, etc. The system may also save this information in a
database in accordance with one or more of user information (e.g., user identification
(ID)), group information, task information, etc., for later use.

The efficiency area 112 may be used to display a user's and/or group's
efficiency. Efficiency information may also be presented in a graphic form for a
user's convenience. The information in this area may be calculated by the system
(e.g., using information from the time area, directory area, etc.), and may be stored in
a database in accordance with one or more of user information (e.g., user
identification (ID)), group information, task information, etc., for later use.

The other tasks area may 114 may display a list of other tasks that may be
assigned to the user or which the user may have finished. Accordingly, the user may
select another task in order to switch to that task. For example, in the current window
shown in FIG. 2A, if the user wishes to switch from "Smith's" loan to "Morgan's" loan,
all the user would have to do is select (e.g., by highlighting, clicking on, etc.) Morgan.
This process may be known as switching working environments and is more clearly
illustrated with reference to FIG. 2C where a working environment corresponding with
the Morgan loan task is shown.

For the sake of clarity, the various UI elements, windows, menus, titlebars,
applications, tasks, links, widgets, instances, and/or features of the UI 100 may be
referred to as tools. When the user selects a tool, the system may remember this
selection using various methods as will be described below with regard to the save
area 120. Each tool may have one or more initial settings or instances which may be
set by the system in accordance with, for example, the process. For example, using
the initial setting, a menu may have certain submenus, a widget such as, for
example, a text entry box, may be blank or have certain text entered, a radio box
may have certain buttons selected, and a browser may have a default home page.

Thereafter, the user may change the default settings or instances. For example, a
user may add/change text in a widget such as a text entry box, may change/select a
default web page of a browser, or add data to a word processor, a spreadsheet, etc.
This information may be referred to as user state information which may include, for
example, instance values that may be entered by the user.

The save area 120 may be used to select and/or save various tasks in
accordance with the present system. The save area may include one or more save
options such as, for example, three saving levels: a level one (L1) 126, a level two
(L2) 124, and a level three (L3) 122, save selections. Further, the present system
may provide a user-defined save option. These selections may be used for
preserving for tasks and the working environment of UI 100 (e.g., including sets of
application tools) of the tasks. Further, a "save now" selection may be used to save
and/or exit a task depending upon configuration. The L1 126 save option may be
available at all times. However, the L2 124 and L3 122 save options may be
disabled depending upon state. For example, with reference to FIG. 1, a task
(instance) has not been assigned to a user yet. Accordingly, the L2 124 and L3 122
save options may be disabled. In the disabled state, a save option may be removed
from a GUI or dehighlighted. With regard to FIG. 1 and the L1 126 save option, this
save option may be available when a task instance has not yet been assigned to a
user. However, absent an assigned task instance, the L2 124 and L3 122 save
options may be disabled by the system. As shown in the other task area 114 of FIG.
1, two task types (i.e., "Loan application" and "House Appraisal") are shown.
However a task instance (e.g., Smith, Morgan, Pierpont, Jones, Daley, etc.) as
shown in FIGs. 2A, has not yet been assigned to the user. Accordingly, as a task
instance does not exist, the user may only select "Loan Application" or "House
Appraisal" task type selections. For example, if the user selects "Loan Application,"
task type selection in the working environment of FIG. 1, the working environment
may change to a "Loan Application" default working environment. For example, a
default working environment may include a work area 106 that may be empty (i.e., it may have no tools) or the system could set some default tool(s) (e.g., tools 106-x, etc.). The user may customize the default working environment by, for example, resizing tools, linking tools, linking data, adding tools from various areas such as, for example, the recommended task area 106, the user tools area 108, etc. When the user finishes customizing the default working environment, the user may save a corresponding configuration (e.g., workplace environment) by selecting an option to save this configuration. For example, the user may select a "save now" menu item (which will perform a save in accordance with a selected save option (e.g., a save option having a highlighted menu item) of the save options L1-L3, etc. Accordingly, in the present example, when selecting the "save now" menu item, a save will be performed in accordance with the L1 126 save option which is the only save option that is available. Accordingly, in accordance with the L1 126 save option, the working environment (e.g., the set of tools, the configuration of the tools, etc.) may be saved by the system and linked with a corresponding task type such as, for example, the "Loan Application" task type. However, no data such as, for example, "work-in-progress" (WIP) data will be saved. Similarly, if the user selects the "House Appraisal" task type from the working environment of FIG. 1, a "House Appraisal" default working environment may be rendered by the system for the user's convenience. The "House Appraisal" default working environment may be specific to the "House Appraisal" task type. Again, the user may repeat the same procedures as described above with respect to the "Loan Application" task type, to customize a working environment corresponding to the "House Appraisal" task type according to the user's desired configuration.

Now, assuming the system (e.g., the BPM/Workflow system) has started assigning tasks, a task instance may be assigned to a user. For example, the user may be assigned with Smith's loan application which has a task instance of "Loan Application" type - Mr. Smith's Loan application (the Smith task instance). Accordingly, the system may display the Smith task instance in the other task area 114 under the "Loan Application" task type. Now that a task instance has been assigned to the user, the system may enable the L2 (124) & L3 (122) save options for the user's or the system's selection. Accordingly, the user may then select the
Smith task instance and a customized working environment corresponding to the "Loan Application" task type (that may have been previously configured and saved by a user) may be rendered (e.g., see, work area 106, FIG. 2B). The user may then enter data (e.g., WIP data) for processing the Smith loan application.

The user may also add new tools onto the workplace environment such as is shown in the work area 106 of FIG. 2A. When the user desires to shut down or switch working environments, the user may select one of the save options (e.g., L1, L2, and L3) to perform a save operation. With regard to the L2 124 save option, this option may be selected by the user to preserve a set of tools without the WIP data. For example, when the user selects the L2 124 save option (or selects the save now menu item while the L2 124 save option is highlighted), the system may preserve a set of tools (without WIP data) (in a similar fashion as to the L1 126 save option), and link the preserved set of tools with a corresponding task type (e.g., the "Loan Application" task type, etc.). Thus, if the user selects the L2 124 save option while in the working environment shown in FIG. 2A, the system may: 1) preserve the set of tools in the work area 106 (without WIP data 142), and link the preserved set of tools with the task type that the current task instance (e.g., Smith) belongs to i.e., the "Loan Application" task type; and 2) preserve WIP data 1420 (e.g., Smith's data) in the work area 106 and link it with the current task instance (e.g., the "Smith" task instance as shown in the other task area 114 of FIG. 2A).

Thus, if the user adds new tools to work on the Smith task instance, the newly added tools may be used by other task instances of the same task type (e.g., of the "Loan application" task type). Accordingly, the system may automatically perform a save according to the L1 126 save option (e.g., saving a current tool set of a working environment to a corresponding task type) as well as saving WIP data 142 corresponding with the task instance (as opposed to the task type). In other words, the L2 save option is similar to the L1 save option with the addition of a task data save. Thus, by incorporating the features of the L1 save option with a task data save, in a single operation, user productivity may be enhanced. Further, the system may keep track of the Smith instance belonging to the "Loan application" task type so that it may perform an L2 type save.
With regard to the L3 122 save option, this save option may be selected when the user may not want changes in a particular task instance to be used in other task instances of the same task type. In other words, the user may not want to save changes for a current task instance to be used for other task instances of the same task type. For example, considering that the user-added tools in work area 106 of FIG. 2A may be intended (by the user) for the Smith instance loan application only, the user may then select the L3 save option. When the L3 save option is selected, the system may: 1) preserve the set of tools of the work area 106 without WIP data 142 and link the preserved set of tools with the current task instance i.e., the Smith task instance (as opposed to the task type as in the case of L1 126 & L2 124 save options); and 2) preserve Smith's data of the work area 106 and link it with the "Smith" task instance (e.g., see, the task instance in the other task area 114).

For example, to illustrate the difference between the L2 and the L3 save options, let's say that the user selects the L2 save option after adding additional tools and entering data for the Smith task instance (i.e., the Smith loan application). Thereafter, the BPM/workflow assigns the user another "Loan Application" task instance such as a Morgan task instance corresponding with Morgan's loan application. When the user selects and opens the Morgan task instance, an updated default working environment (e.g., including the additional tools added by the user when saving the Smith instance using an L2 124 save option) will be shown in the work area 106. However, as the L2 save option was performed during the Smith task instance, WIP data 142 was not saved for other tasks instances (e.g., the Morgan instance. Accordingly, prior WIP data from the Smith instance will not be rendered for a working environment of the Morgan task instance. On the other hand, assuming that the user selects the L3 save option (when in the Smith instance) instead of the L2 save option after adding additional tools (e.g., to a default working environment) and entering data for Smith (e.g., WIP data 142), thereafter when the user switches to the Morgan task instance, the original default working environment (without the additional tools or WIP data added with the Smith instance) will be rendered.

To summarize the different operations performed by the L1, L2 and L3 save options:
L1: saving task type <> working environment
L2: L1 + saving task instance <> WIP data
L3: saving task instance <> WIP data <> working environment

Thus, with respect to the L1 126 save option, this save option may be selected to preserve a set of selected tools of a working environment with a corresponding task type (e.g., by linking the working environment with the corresponding task type) however, WIP data which may include, for example, state information, will not be saved by the system. For example, this save option may be selected to preserve a set of selected tools of a current working environment when, for example, the user switches task instances, switches working environments, or logs off or shuts down the working environment. Then, when the task instance is re-launched, the previous workplace environment including the user selected tools will reappear as they previously did in previous working environment before the task was previously shut down. However, according to the L1 126 save option, an archive of tools may be stateless and may not save WIP information such as user state information which may include user input instance values (e.g., text in a text entry box, radio button selections, check box selections, etc. as is common in a Windows-type environment) entered or selected by user when using a tool of the working environment. Thus, only the layout, position and/or type of tool may be preserved. However, WIP (e.g., state information which may include data which the tools handle) may not be preserved. Thus, in accordance with the L1 126 save option, WIP of a task instances that were being processed by the tools of the work environment may not be stored (with task type or task instance information) when the L1 126 save operation is performed. (e.g., when the user switches tasks or the current working environment system is switched, logged off or shut down). For the sake of clarity, the process of switching, shutting down, or logging off tasks may be commonly referred to as switching working environments. According to the present embodiment, when a user switches working environments, the system may perform a save in accordance with a highlighted save option. Thus, the user does not have to expressly perform a save (e.g., a L1 126, L2 124, or L3 122) save in order to save
the current working environment and/or WIP data (if required by a corresponding save option) before switching working environments.

Thus, according to the L1 126 save option, if the current working environment is accidentally switched (e.g., due to an unintentional shutdown, reset, etc.), the system may not be able to revert back to the state of the work before the current working environment was switched. For example, a worker (e.g., a user) may be assigned a task of processing a loan application for customer "Smith" known as a Smith task instance. With the L1 126 save option selected, this worker can select certain tools (e.g., a loan calculator, a credit report searcher, a spreadsheet, a web browser, etc.) that the worker would like to use, and place and/or resize the tools in any location on the UI 100 so as to form a current working environment as illustrated by UI 200A in FIG. 2A. The system, in accordance with the L1 126 save option, may then preserve this working environment (i.e., the environment configured by the worker when the working environment was switched) with a task type of the Smith instance (e.g., of "Loan Application" instance). Thereafter, when the user relaunches the same task type, the system may bring back a similar working environment (e.g., a loan calculator, a credit report searcher, a spreadsheet, with the same size windows, etc.), that was previously stored. However, any WIP information 142 such as user input instance values which may include, for example, information inserted into, for example, text entry boxes, etc., (e.g., see, 142, FIG. 2A) of the previous working environment, will be lost and/or not used when the working environment is rendered again by relaunching the same task type (e.g., the Smith task). For example, compare user entry areas 140 (which may include user entry areas such as a user text entry field 140-T, a radio button field 140-R, etc.) in FIGs. 2A and 2B. In FIG. 2B, the WIP data 142 has been set to a user-defined initial (or default) state. Thus, once the user starts to work on customer Smith's loan application (i.e., on the Smith instance), when the L1 save option is highlighted, in order not to lose data due (e.g., WIP data 142) to various circumstances such as, for example, a user switching task instances, etc., the user may finish the current task instance (e.g., the Smith loan application—the user may also click on a "complete task" menu item to complete a task) before switching task items or the user may highlight and/or select the L2 save option to save the WIP data. However, when task instance is switched and
thereafter re-launched, in accordance with the L1 126 save option, the working environment of the closed task will always start in accordance with the "initial" (e.g., default) state rather than in accordance with a last or previous state which may include WIP data which may include user input instance values. Thus, the "initial" state may include the tools and configuration however, it may not include WIP data which may include user input task data such as user input instance values such as, for example, Smith's name and associated data entered by the user. The L1 126 save option may be set (e.g., highlighted) by a user or by the system, as desired. Further, the L1 126 save option may be used to prevent stale information from being used. For example, by setting the L1 126 save option, the user may be prevented from locking in an outdated loan interest rate from a previous day when a working environment is switched. Thus, for example, if a user leaves a task open for a period of time that is greater than a threshold value (e.g., 1 or more days), the system may set the L1 save option and inform a user that the data is stale. Thus, a user may be prevented from locking in a loan with a stale date and/or interest rate. The user may override the save option that was set by the system by selecting another save option.

With respect to the L2 124 save option, this save option may be used to preserve a set of selected tools of a working environment with a certain task type and may save WIP data (which may include state information) for a current task instance in association with the task instance. Thus, if the current task instance is switched, when the user returns to this task instance at a later time, the system may re-launch the task instance with the selected tools and the WIP data as previously input before the task instance was relaunched. Accordingly, the working environment saved under the L2 124 save option may be considered semi-stateful.

Thus, in accordance with the L2 124 save option, the current working environment may preserved in association with a task type of the current task instance and WIP data 142 of the current working environment may be preserved in association with the task instance. Thus, the system may maintain state information about the task instances being processed when a save in accordance with the L2 124 save option is performed. Accordingly, when the working environment of the current task instance (e.g., the Smith instance) is saved in accordance with the L2 124 save option, and thereafter switched, the system may be able to bring back to the previous state of the
work (i.e., the previous working environment including the WIP data before the previous working environment was switched) when the task instance is relaunched.

Thus, in accordance with the level-two (L2) 124 save option, the state of the working environment that is preserved may be considered static. In other words, according to an embodiment of the present system, only one state (i.e., the current task instance) may be preserved per in association with a working environment of a task instance. For example, a worker may be assigned to process a task instance such as a loan application for Smith (i.e., Smith's loan) with the level two (L2) 124 save option active. The user may open this instance (i.e., the Smith instance) and preserve the working environment and associated WIP data using an L2 124 save. Accordingly, the system may save the working environment (without the WIP data) in association with a task type and may save the WIP data in association with the task instance. However, if this worker switches to another task instance of the same task type such as a loan processing task for another customer (e.g., customer Morgan's loan which corresponds to the Morgan instance), when an L2 124 save option is selected, the system may save the working environment of the Morgan instance in association with the task type (e.g., Loan Application) and may save the WIP data in association with the current task instance (e.g., the Morgan instance). This save option may be selected to preserve a set of selected tools of a working environment with a corresponding task type however, WIP data which may include, for example, state information, will not be saved by the system in this example.

With respect to the level-three L3 122 save option, this save option may be selected to preserve a set of user selected tools of a working environment and WIP data with a corresponding task instance. Accordingly, when a task instance is saved under the L3 122 save option, the set of selected tools as well as the WIP data is stored (e.g., in a memory of the system) in association with the task instance. Accordingly, when the user re-launches the task instance, the previously saved working environment (including the set of selected tools) with the WIP data associated with saved working environment may be rendered for the user's convenience. Accordingly, the working environment saved under the L3 122 save option may be considered is fully-stateful, as both the working environment and the WIP may be preserved. Accordingly, using the L3 122 save option, a worker (i.e., a
user) may concurrently process several tasks of the different instances and the same
type and preserve a working environment and WIP data for each of the task
instances when switching tasks. For example, the worker may be assigned two tasks
of the same task type i.e., a first task instance (i.e., the Smith instance) to process a
loan application for customer A (i.e., Smith) and a second task instance to process a
loan application for customer B (i.e., Morgan) at the same time. Accordingly, the
worker may alternate between working environments associated with each task
instance while the system may store a previous state (including WIP data) for each
task instance as the user switches between task instances. Accordingly, when this
worker selects customer A's loan application task instance, the system will launch the
Smith instance and render working environment including corresponding WIP data
(e.g., WIP data for Smith). Likewise, when the worker opens customer B's loan
application task instance, the system will launch the Morgan instance and render a
working environment including corresponding WIP data (e.g., WIP data for the
Morgan).

When storing the working environment associated with a task, the system may
form preservation data and store this data in a memory such as, for example, a
database of the system. The system may save data when a working environment is
switched, shut down, logged off, at periodic intervals and/or at other times. For
example, the system may log all user inputs for later use so that in case the system
shuts down unexpectedly (or freezes) the working environment and WIP data may be
saved. The log of all user inputs may be processed and stored in the preservation
data. Further, the preservation data may, dependent upon the save option selected,
include WIP data corresponding with a certain task. Thus, the system may determine
which save option was selected (e.g., L1, L2, L3, etc.) and store the user information
in accordance with the selected save option. The save option may be set in
accordance with a task by the system and/or the user. For example, a system
manage may set the save option. However, it is also envisioned that the user may
set the save option.

A task assignment function according to the present system will now be
described. The task assignment function may assign task types and/or task
instances to selected user's (e.g., individuals, groups, etc.) and/or track the progress
of a specific task instance as it is performed and/or completed. Task types and/or task instances may be assigned based upon various data. Each user may be assigned a task inbox by the system. One or more user's may belong to a group. A group leader may also assign tasks to members of his group. When a task is assigned to a user, the user may be notified by the system through a user's task inbox the other task area 114, FIG. 1). A newly assigned task instance may include an identifier (e.g., "Smith") that may be highlighted and/or may include expected completion times, priority codes, etc., so that a user may be properly informed of the task instance. When a user selects a task instance and opens it (e.g., by clicking on a corresponding identifier), the system may provide a UI upon which a working environment specific to the task instance is rendered. The working environment may include various software application tools running or available for the user's selection. Accordingly, there may be a set of pre-defined tools in the user interface panel. Each working environment may be specific to a related task type or task instance that the user has been assigned. A user may freely add new tools onto the working environment as desired. Further, a user may link different tools together so that data may be exchanged among tools. Further, this data may be shared in a global level with other task types and/or task instances of the same user or other users so as to form a collaborative computing/data sharing environment. Thus, a user may output data from a tool that is active in a user's working environment which corresponds with a task type or task instance such that this data may be used by other task types or task instances of the same or other users. Accordingly, a user may set an output of a first tool to be an input to another tool, so when a tool generates output data, the other tool may automatically receive the output data and use this data for processing. Thereafter, this data may then be output to the first tool, stored for later use (e.g., in a memory of the system), or may be output to a further tool for yet further processing. The system may store data generated by any tool in a memory of the system, as desired. The system may also communicate with the tools, the users, etc., via a network.

With respect to tools, users may add new tools onto the working environment (e.g., into the work area 106 or other areas) according to system settings. A set of (e.g., pre-configured) tools may be available from the tool library of the system.
These tools may be displayed for a user's selection in, for example, a user tools area (e.g., see, 110). Users may select the tools they desire to use for a certain task. After setting up a preferred working environment, users may decide to save the working environment in accordance with various save options (e.g., L1, L2, L3, and/or user defined (UD) -type save operations). Further, according to the UD save option, a user may configure items to save according to the user's preference. The system may also select a default save option. After selecting a save options, the system may preserve the user's preferred working environment (e.g., tools, layout of the tools, association among the tools, instance data, etc.) for this particular user and for this particular task, in accordance with the selected save option. The system may store information related to a certain task type and/or task instance in a memory (e.g., a database of the system) for later use by the user and/or others. After a task has been shut down or switched, the system may relaunch a working environment of the task type and/or task instance in accordance with the stored information.

Each task instance which the system assigns to user may belong to a task type (or class) [1:3-ok]. The task instance may be used by the system to select a working environment for a task of a given type or class. Thus, when the system assigns a new task instance of a task type or class that is the same as, or similar to, a previously assigned task type or class (which has been previously assigned to the user), the system may set a working environment which corresponds with the previously assigned task type or task class [1:4-ok]. Accordingly, when the user launches the newly assigned task instance, the system may render a working environment which is the same as, or similar to, a working environment of a previously assigned task type or task class. The system may select a save option (e.g., L1, L2, L3, etc.) to associate with a new task based upon, for example, system settings, user settings, task type, task class, task members, etc. [1:5-ok]

A saved working environment may be used by the system to generate a working environment that is may be associated with a new task which is to be assigned to a user. For example, when a user builds a working environment for a "collecting customer data" task type, and saves this information in accordance with a save option (e.g., L1, L2, and L3) this information may be saved by the system (e.g., in a memory of the system) in accordance with the save option. Thereafter, when
assigning new tasks, the system may determine that the new task type corresponds with a saved task type. Accordingly, the system may then assign a default working environment that may be the same as, or similar to, a working environment of a previously saved task of the same task type or class. Thus, for example, when a user is assigned with a task which has a task type that corresponds with the "collect customer data" task type, the system may assign a working environment which corresponds with the previously-stored "collect customer data" task type. [16-ok] For another task type such as an "approving customer request," the system may render a different working environment that is associated with this task type. Thus, when task types or task instances are assigned to a user, the system may refer to existing task types or task instances, etc. in order to select a working environment for use with the task.

When a user starts using the various tools on a working environment to perform a task instance, the user can choose to save work using one or more different save options. The save options may function to save a working environment that the user has configured and may save and/recall WIP data and/or user input task data in accordance with the selected save option. The saved working environment, WIP data and/or user input task data may be associated with the task type and/or task instance. This user can then close, the working environment (e.g., by switching tasks, logging off, shutting down the computer, etc.). When user returns to a closed task (e.g., by selecting the task from, for example, the user's task inbox), the system may relaunch the task and open the saved working environment.

The system may also distinguish different working environments for different task types. Thus, when the system determines that a task belongs to a certain type or task class (e.g., collect customer data, mortgage sales, mortgage cold calls, collections, human resources, accounts receivable, etc.), the system may assign the same or a similar working environment to tasks of the same or similar task type or task class. Thus, for example, if a user is assigned several different task instances each of which is associated with a different customer (e.g., customers Smith, Morgan, Johnson, Doe, Fitzpatrick, etc.) and has the same task type (e.g., "collect customer data"), the system may associate similar working environments for each of these task instances. However, data such as, names, etc. may be different in each
task and, thus, the task instances may be different. Thus, the user's task inbox (e.g., see, the other task area 114) may have different task instances for the tasks of the same type (e.g., "collect customer data") listed under a common task type or task class. Once a user customizes the task types or task instances (e.g., initial task instances), then these task instances may be referred to as a user-customized task type or task instance. The user may then select one of these tasks to launch and use the same set of tools in each working environment that corresponds with the same task type. Accordingly, the user may be presented with a common working environment for each task of the same type. When the user processes a task instance corresponding to each of the different customers, the user and/or the system may save a workplace environment and/or WIP data according to a predetermined save option.

FIG. 3 shows a flow diagram that illustrates a process 300 in accordance with an embodiment of the present system. The process 300 may be performed using one or more computers communicating over a network. The process 300 can include one of more of the following acts. Further, one or more of these acts may be combined and/or separated into sub-acts, if desired. In operation, the process may start during act 301 and then proceed to act 302.

In act 302, the process may parse workflow information of, for example, a BPM. As workflow information is well known in the art, it will not be further discussed. The process may receive the workflow data from a memory of the present system. After completing act 302, the process may continue to act 304.

In act 304, the process may load and then process group/worker (user) information to determine available users. The group/worker information may be downloaded from a memory (e.g., a group/worker database) of the present system. The group/worker information may include information related to the capabilities of one or more workers. For example, information related to the capabilities of a worker may include information such as, a current status of a worker, a group in which the worker belongs (e.g., collections), education of a worker, previous performance data of the worker (0 absences, 98% efficiency, etc.), ability of a worker, etc. For example, this data may include information indicative of a worker's ability to perform
tasks of certain types, classes, etc. After completing act 304, the process may continue to act 306.

In act 306, the process may extract tasks from the workflow information.

In act 308, the process may match extracted tasks types, classes, and/or instances to available groups and/or users. For example, the process may determine a type of each task and match this information to the group/worker information. Thereafter, the process may determine which users (e.g., workers) are best matches for each task type or class. Thus, the process may, for example, match workers which are deemed to be capable of performing certain tasks types with corresponding task types. The process may also determine a current load on a worker, before matching a worker with a task type, a task instance, etc. For example, if a worker is determined to have a load which is beyond a threshold level, the system may not match the worker to a task type, a task instance, etc. The process may match a current task type by selecting workers who have previously performed tasks of the same type with a predetermined efficiency (e.g., greater than 60%). After completing act 308, the process may continue to act 310.

In act 310, the process may assign tasks to users. Each user may be assigned a reasonable number of tasks. If there are more tasks than available users can handle, the system may not assign some tasks. Accordingly, these tasks may be considered unassigned tasks. The unassigned tasks may be assigned at a later time such as, for example, when the user completes another task. After completing act 310, the process may continue to act 312.

In act 312, the process may determine whether a task instance that is assigned to a user has a task type that corresponds with a stored task type. The stored task type may correspond with a task type such as, for example, a "Loan Application," "House Appraisal," "collect customer data," or other task types. The process may make this determination using any suitable method. For example, the process may query a task type database which may contain stored task types. Thereafter, the task type database may send results of the query back to the process. However, it is also envisioned that the process may use a list of stored task types for each user. Accordingly, if the process determines that a task instance that is assigned to a user has a task type that corresponds with a stored task type, the
process may continue to act 314. However, if the system determines that a task instance that is assigned to a user has a task type that does not correspond with a stored task type, the process may continue to act 320. The process may perform this determination for each task instance that is assigned. However, it is also envisioned that the process may assign a default task type to a task instance if assigned.

In act 314, the process may assign a workplace environment (WE) that corresponds with the stored task type of the assigned task instance or task type. After completing act 314, the process may continue to act 318.

In act 318, the process may inform the user of the task type and/or task instance that was assigned to the user. For example, the process may send a notification of the task type and/or task instance to a user's task inbox (e.g., other task area 114). Thereafter, a user may select this notification for more information and/or to launch the task type and/or task instance. After completing act 318, the process may continue to act 322 where it ends.

In act 320, the process may assign a default workplace environment to the task type. After completing act 320, the process may continue to act 318. The default workplace environment may be used when, for example, a task type does not correspond with any saved task types.

After a task instance is assigned, the process may monitor the progress of the task instance and update data regarding one or more task types, instances, etc. For example, efficiency data may be updated according to data compiled by the system that relates to the task instance. When the system determines that a task instance has been completed or has met a certain milestone (e.g., 90% completed), the system may change a status of the task instance (e.g., from pending to completed or almost complete). This information may be stored in a task progress database for use by, for example, the process, managers, etc. The system may output the task instance progress (e.g., 90% complete, 0% complete, etc, using, for example, a task progress bar in the workplace environment. When the process determines that a user has completed a task instance, the process may assign a new task instance to the user. The process may select the new task instance from, for example, the unassigned task instances, or may select a task instance from a task instance which was assigned to another user (thus, reassigning a task instance). For example, the
process may determine that a certain user has not begun a certain task instance and may therefore re-assign this task to another user who has just finished a task instance. Further, if the process determines that a user has an assigned task instance that is almost completed, but the worker will not be in for 2 weeks (e.g., due to a holiday), the process may re-assign this task instance (e.g., a work-in-progress) to another user. Thus, the process may also interface with a human-resources database (e.g., a time accounting database, etc.) to determine if and/or when a worker will be available so that the system may assign/reassign tasks accordingly.

[18-ok] FIG. 4 is a diagram of an information flow according to an embodiment of the present system. The system 400 may include one or more tables such as one or more task tables 402, one or more task instance tables 404, one more working environment tables 406, and one or more work in progress tables 408. The information included in the tables may be used by the system 400 to determine a working environment and/or work-in-progress (WIP) data to use as will be explained elsewhere. The tables may be stored by the system 400 in a memory of the system and/or may be combined with each other, if desired. The data in the one or more tables 402-408 illustrate information that may be used by the system 400 in order to determine working environment/WIP data to render in a workplace environment for a particular task type/instance. According to the L1 save option, each task type will be associated with one working environment. According to the L2 save option, each task instance will be associated with one WIP data. Accordingly, the L2 save option may perform all the acts of the L1 save option and may associate each task instance with WIP data as well. According to the L3 save option, each task instance will be associated with one WIP and one working environment.

FIG. 5 shows a schematic of an embodiment of the present system. A system 500 may include or more of a BPM management portion 502, a wireless server 504, one or more base stations (BSs) BS-1 and BS-2, a network 506, an external storage 510, user devices 508-1 - 508-n, global positioning system (GPS) transmitter(s) 522, and mobile stations (MSs) MS-1 - MS-m. One or more of these components may be combined with each other, if desired.

Each of the BSs (BS-1, BS-2) may include a corresponding service area SA-1 and SA-2, respectively, and may communicate with one or more MSs in a
corresponding service area using any suitable method (e.g., TDMA, CDMA, GSM, Bluetooth, WiMax, WiFi, etc.) as is known in the art.

The working environment management portion 502 may include one or more of a BPM/Workflow component 512, a browser/desktop (BD) execution environment 514, a co-ordination component 518, one or more storage devices 516, and a controller 520. The controller 520 may include one or more controllers one or more of which may be located locally or remotely from the others. The controller 520 may control the overall operation of the working environment management 502.

The storage 516 may store one or more of operating tools of the present system (e.g., applications, widgets, instance data, etc.), data generated by the present system, and/or user information (e.g., user preferences, user identification (ID), user passwords, user logins, employment information, etc.), BPM information, task information, etc. The storage 516 may include any suitable storage device or devices such as, for example, a read only memory (ROM), a random access memory (RAM), an optical memory, a flash memory, a disc storage, a SAN, etc.

The BPM workflow component 512 may be responsible for, among other things, executing business processes workflows and managing the assignment of tasks to one or more users. For example, the BPM workflow component 512 may perform one or more tasks such as, for example, parse workflow information, load/retrieve worker information from the storage 516, extract tasks from the workflow, and/or match tasks (e.g., task types, classes, instances, etc.) to available users (workers, etc.), and assign one or more tasks (e.g., task types, classes, instances, etc.) to one or more users.

The browser/desktop execution environment 514 may provide allow different tools such as, for example, applications, widgets, etc., user interfaces (UI), etc. to run independently on a common area. The tools may be initially assigned by the system 502 or selected by the user. As the ability to run different widgets or application UIs in a common window is within the art (e.g., see, iGoogle™ or Netvibes™), for the sake of clarity, it will not be discussed in detail herein.

The coordination component 518 may operate under the control of the controller and synchronize and/or coordinate the functions between the BPM/Workflow component 512 and the browser/desktop execution environment 514.
The system 500 may assign tasks (e.g., task types, classes, instances, etc.) to individuals (e.g., user's) and/or provide a personalized working environment and dynamically associate tasks (e.g., task types, classes, instances, etc.) which are assigned to individuals with the personalized working environment for individuals.

The system 500 may receive location information regarding the one or more MSs (Ms-1 - MS-m) and assign tasks (e.g., task types, classes, instances, etc.) to user's based upon a geophysical location of an MS (e.g., a user's MS). For example, in a mobile environment, a user of an MS may be a traveling. The system 500 may, determine the geophysical location of the user's MS and assign a task (e.g., including task types and/or instances, classes, etc.) to the user based upon the determined position. For example, if the system 500 determines that a task (e.g., task types, classes, instances, etc.) may require a user to be located at a certain position (e.g., at a certain location or address), the process may assign this task (e.g., task types, classes, instances, etc.) to, a user who may be able to reach the certain position at a given time (e.g., a nearest user, etc.). Accordingly, the system may include least-traveling-distance functions, and the like, to determine how to assign one or more tasks (e.g., task types, classes, instances, etc.). The system 500 may determine a geophysical location using any suitable method such as, for example, GPS transmissions, assisted GPS (A-GPS), triangulation, base station ID, etc. The system 500 may also await a user's confirmation when assigning tasks (e.g., task types, classes, instances, etc.) and may reassign tasks (e.g., task types, classes, instances, etc.) or notify certain user's (e.g., managers), if, for example, a confirmation for a certain task instance is not received from a user within a predetermine period of time.

The external storage 510 may store information similarly to the storage 516 and/or may include any suitable storage device or devices as described elsewhere. For example, the external storage 510 may store one or more external applications, etc., that may be used by the present system. The external storage 510 may communicate with other components via any suitable method such as, for example, the network 506.

The user devices 508-1 - 508-n may include any suitable computational device which may provide a user interface. For example, the user devices 508-1 -
508-n may include workstations, etc. The user devices 508-1 - 508-n may communicate with each other and/or the working environment management 502 any suitable method. For example, the user devices 508-1 - 508-n may communicate with each other and/or the working environment management 502 using one or more networks such as, for example, the network 506. The user devices 508-x may include any suitable device for rendering a UI of the present system.

The working environment management 502 may communicate with the one mobile stations MS-1 - MS-m via the wireless server 504 and corresponding stations BS-1 and BS-2. However, a wired connection may also be used. The mobile stations (MS-1 - MS-m) may include similar functions as provide by the user devices 508-x. Accordingly, the mobile stations (MS-1 - MS-m) may render a UI for a user's convenience.

The network 506 may include, any suitable network such as, may include one or more of a wide area network (WAN), a local area network (LAN), a proprietary network, the Internet, an intranet, etc. The mobile stations (MS-1 - MS-m) may be coupled to the network 506.

Accordingly, the present system provides a personalized working environment for users who have tasks (e.g., task types, classes, instances, etc.) assigned to them. These tasks (e.g., task types, classes, instances, etc.) may be automatically determined and/or assigned by an information technology (IT) system such as a BPM or workflow system. This working environment may include various applications and/or features needed to perform the task (e.g., task types, classes, instances, etc.) which was assigned to the user. The system may also keep track of tasks, task classes, task types, task instances, working environments, and/or WIP data and a relationship among them for a user's convenience. Accordingly, workplace efficiency may be enhanced.

FIG. 6 shows a system 600 in accordance with a further embodiment of the present system. The system 600 includes a user device 690 that has a processor 610 operationally coupled to a memory 620, a rendering device 630, such as one or more of a display, speaker, etc., a user input device 670 and a content server 680 operationally coupled to the user device 690. The memory 620 may be any type of device for storing application data as well as other data, such as task data, instance
data, workflow data, places, audio wallpapers, content, reaction indications, tallied reaction indications, comments, graphing data, such as heat map data, heat line graph data, heat comment graph data, play lists, recommended content, etc. The application data and other data are received by the processor 610 for configuring the processor 610 to perform operation acts in accordance with the present system. The operation acts include controlling at least one of the rendering device 630 to render one or more of the GUIs and/or to render content. The user input 670 may include a keyboard, mouse, trackball or other devices, including touch sensitive displays, which may be stand alone or be a part of a system, such as part of a personal computer, personal digital assistant, mobile phone, converged device, or other rendering device for communicating with the processor 610 via any type of link, such as a wired or wireless link. The user input device 670 is operable for interacting with the processor 610 including interaction within a paradigm of a GUI and/or other elements of the present system, such as to enable web browsing, content selection, such as provided by left and right clicking on a device, a mouse-over, pop-up menu, etc., such as provided by user interaction with a computer mouse, etc., as may be readily appreciated by a person of ordinary skill in the art.

In accordance with an embodiment of the present system, the rendering device 630 may operate as a touch sensitive display for communicating with the processors 610 (e.g., providing selection of a web browser, a Uniform Resource Locator (URL), portions of web pages, etc.) and thereby, the rendering device 630 may also operate as a user input device. In this way, a user may interact with the processor 610 including interaction within a paradigm of a UI, such as to support content selection, input of reaction indications, comments, etc. Clearly the user device 690, the processor 610, memory 620, rendering device 630 and/or user input device 670 may all or partly be portions of a computer system or other device, and/or be embedded in a portable device, such as a mobile station (MS), mobile telephone, personal computer (PC), personal digital assistant (PDA), converged device such as a smart telephone, etc.

The system and method described herein address problems in prior art systems. In accordance with an embodiment of the present system, the user device 690, corresponding user interfaces and other portions of the system 600 are provided
for browsing content, selecting content, providing reaction indications, reaction indication palettes, etc., and for transferring the content and reaction indications, tallied reaction indications, etc., between the user device 690 and the content server 680.

The methods of the present system are particularly suited to be carried out by a computer software program, such program containing modules corresponding to one or more of the individual steps or acts described and/or envisioned by the present system. Such program may of course be embodied in a computer-readable medium, such as an integrated chip, a peripheral device or memory, such as the memory 620 or other memory coupled to the processor 610.

The computer-readable medium and/or memory 620 may be any recordable medium (e.g., RAM, ROM, removable memory, CD-ROM, hard drives, DVD, floppy disks or memory cards) or may be a transmission medium utilizing one or more of radio frequency (RF) coupling, Bluetooth coupling, infrared coupling etc. Any medium known or developed that can store and/or transmit information suitable for use with a computer system may be used as the computer-readable medium and/or memory 620.

Additional memories may also be used. The computer-readable medium, the memory 620, and/or any other memories may be long-term, short-term, or a combination of long-term and short-term memories. These memories configure processor 610 to implement the methods, operational acts, and functions disclosed herein. The operation acts may include controlling the rendering device 630 to render elements in a form of a UI and/or controlling the rendering device 630 to render other information in accordance with the present system.

The memories may be distributed (e.g., such as a portion of the content server 680) or local and the processor 610, where additional processors may be provided, may also be distributed or may be singular. The memories may be implemented as electrical, magnetic or optical memory, or any combination of these or other types of storage devices. Moreover, the term "memory" should be construed broadly enough to encompass any information able to be read from or written to an address in the addressable space accessed by a processor. With this definition, information on a network is still within memory 620, for instance, because the processor 610 may
retrieve the information from the network for operation in accordance with the present system. For example, a portion of the memory as understood herein may reside as a portion of the content server 680. Further, the content server 680 should be understood to include further network connections to other devices, systems (e.g., servers), etc. While not shown for purposes of simplifying the following description, it is readily appreciated that the content server 680 may include processors, memories, displays and user inputs similar as shown for the user device 690, as well as other networked servers, such as may host web sites, etc. Accordingly, while the description contained herein focuses on details of interaction within components of the user devices 690, it should be understood to similarly apply to interactions of components of the content server 680.

The processor 610 is capable of providing control signals and/or performing operations in response to input signals from the user input device 670 and executing instructions stored in the memory 620. The processor 610 may be an application-specific or general-use integrated circuit(s). Further, the processor 610 may be a dedicated processor for performing in accordance with the present system or may be a general-purpose processor wherein only one of many functions operates for performing in accordance with the present system. The processor 610 may operate utilizing a program portion, multiple program segments, or may be a hardware device utilizing a dedicated or multi-purpose integrated circuit.

In accordance with an embodiment of the present system, the rendering device 630 may operate as a touch sensitive display for communicating with the processors 610 (e.g., providing selection of a web browser, a Uniform Resource Locator (URL), portions of web pages, etc.) and thereby, the rendering device 630 may also operate as a user input device. In this way, a user may interact with the processor 610 including interaction within a paradigm of a UI, such as to support content selection, input of reaction indications, comments, etc. Clearly the user device 690, the processor 610, memory 620, rendering device 630 and/or user input device 670 may all or partly be portions of a computer system or other device, and/or be embedded in a portable device, such as a mobile telephone, personal computer (PC), personal digital assistant (PDA), converged device such as a smart telephone, etc.
The system and method described herein address problems in prior art systems. In accordance with an embodiment of the present system, the user device 690, corresponding user interfaces and other portions of the system 600 are provided for rendering tasks, browsing content, selecting content, providing reaction indications, reaction indication palettes, etc., and for transferring the information related to the tasks, content and reaction indications, tallied reaction indications, etc., between the user device 690 and the content server 680.

The methods of the present system are particularly suited to be carried out by a computer software program, such program containing modules corresponding to one or more of the individual steps or acts described and/or envisioned by the present system. Such program may of course be embodied in a computer-readable medium, such as an integrated chip, a peripheral device or memory, such as the memory 620 or other memory coupled to the processor 610.

The computer-readable medium and/or memory 620 may be any recordable medium (e.g., RAM, ROM, removable memory, CD-ROM, hard drives, DVD, floppy disks or memory cards) or may be a transmission medium utilizing one or more of radio frequency (RF) coupling, Bluetooth coupling, infrared coupling etc. Any medium known or developed that can store and/or transmit information suitable for use with a computer system may be used as the computer-readable medium and/or memory 620.

Additional memories may also be used. The computer-readable medium, the memory 620, and/or any other memories may be long-term, short-term, or a combination of long-term and short-term memories. These memories configure processor 610 to implement the methods, operational acts, and functions disclosed herein. The operation acts may include controlling the rendering device 630 to render elements in a form of a UI and/or controlling the rendering device 630 to render other information in accordance with the present system.

The memories may be distributed (e.g., such as a portion of the content server 680) or local and the processor 610, where additional processors may be provided, may also be distributed or may be singular. The memories may be implemented as electrical, magnetic or optical memory, or any combination of these or other types of storage devices. Moreover, the term "memory" should be construed broadly enough
to encompass any information able to be read from or written to an address in the addressable space accessed by a processor. With this definition, information on a network is still within memory 620, for instance, because the processor 610 may retrieve the information from the network for operation in accordance with the present system. For example, a portion of the memory as understood herein may reside as a portion of the content server 680. Further, the content server 680 should be understood to include further network connections to other devices, systems (e.g., servers), etc. While not shown for purposes of simplifying the following description, it is readily appreciated that the content server 680 may include processors, memories, displays and user inputs similar as shown for the user device 690, as well as other networked servers, such as may host web sites, etc. Accordingly, while the description contained herein focuses on details of interaction within components of the user devices 690, it should be understood to similarly apply to interactions of components of the content server 680.

The processor 610 is capable of providing control signals and/or performing operations in response to input signals from the user input device 670 and executing instructions stored in the memory 620. The processor 610 may be an application-specific or general-use integrated circuit(s). Further, the processor 610 may be a dedicated processor for performing in accordance with the present system or may be a general-purpose processor wherein only one of many functions operates for performing in accordance with the present system. The processor 610 may operate utilizing a program portion, multiple program segments, or may be a hardware device utilizing a dedicated or multi-purpose integrated circuit.

Accordingly, the present system may dynamically assign job tasks to an individual or a group of individuals and then, dynamically associate the job tasks assigned to individuals with a personalized working environment.

Finally, the above discussion is intended to be merely illustrative of the present system and should not be construed as limiting the appended claims to any particular embodiment or group of embodiments. For example, the present system may be utilized to determine the presence of distinguishing information such as a watermark and push content which corresponds with the detected watermark and/or the user profile information. The present system may be provided in a form of a
content rendering device, such as a MS. A further embodiment of the present
system may provide a UI that operates as a browser extension, such as a rendered
browser toolbar, that can build a content rendering playlist, such as a video playlist.
In addition, the present system may push predetermined content while a user is
browsing the Internet.

Thus, while the present system has been described with reference to
exemplary embodiments, including user interfaces, it should also be appreciated that
numerous modifications and alternative embodiments may be devised by those
having ordinary skill in the art without departing from the broader and intended spirit
and scope of the present system as set forth in the claims that follow. Further, while
exemplary user interfaces are provided to facilitate an understanding of the present
system, other user interfaces may be provided and/or elements of one user interface
may be combined with another of the user interfaces in accordance with further
embodiments of the present system.

The section headings included herein are intended to facilitate a review but
are not intended to limit the scope of the present system. Accordingly, the
specification and drawings are to be regarded in an illustrative manner and are not
intended to limit the scope of the appended claims.

In interpreting the appended claims, it should be understood that:

a) the word "comprising" does not exclude the presence of other elements
or acts than those listed in a given claim;

b) the word "a" or "an" preceding an element does not exclude the
presence of a plurality of such elements;

c) any reference signs in the claims do not limit their scope;

d) several "means" may be represented by the same item or hardware or
software implemented structure or function;

e) any of the disclosed elements may be comprised of hardware portions
(e.g., including discrete and integrated electronic circuitry), software portions (e.g.,
computer programming), and any combination thereof;

f) hardware portions may be comprised of one or both of analog and
digital portions;
g) any of the disclosed devices or portions thereof may be combined together or separated into further portions unless specifically stated otherwise;

h) no specific sequence of acts or steps is intended to be required unless specifically indicated; and

i) the term "plurality of" an element includes two or more of the claimed element, and does not imply any particular range of number of elements; that is, a plurality of elements may be as few as two elements, and may include an immeasurable number of elements.
What is claimed is:

1. A method of providing a user interface (UI) comprising acts of:
   accessing user profile information including stored task information for one or more users of a plurality of users;
   assigning a task of a plurality of tasks to a user of the plurality of users;
   determining whether the assigned task corresponds with the stored task information for the user;
   setting a working environment including a plurality of applications based upon the determination; and
   rendering the set working environment using the UI.

2. The method of claim 1, further comprising an act of informing the user of the task before a user can launch the task.

3. The method of claim 1, further comprising an act of storing information related to the task in accordance with a selected save option of a plurality of save options.

4. The method of claim 1, wherein the act of setting comprises:
   setting a first working environment when it is determined that the assigned task corresponds with the stored task information for the user; and
   setting a second working environment different from the first working environment when it is determined that the assigned task does not correspond with the stored task information for the user.

5. The method of claim 4, wherein the first working environment comprises an application configuration selected by the user.

6. The method of claim 4, wherein the second working environment comprises a default application configuration.
7. The method of claim 1, further comprising acts of:
   launching the assigned task;
   inputting, by the user, instance values; and
   storing the instance values in accordance with a save option when the task is shut down.

8. A system which provides a user interface (UI), the system comprising:
   a controller which:
      accesses user profile information that includes stored task information for one or more users of a plurality of users,
      assigns a task of a plurality of tasks to a user of the plurality of users,
      determines whether the assigned task corresponds with the stored task information for the user, and
      sets a working environment based upon the determination; and
   a rendering device which renders the set working environment using the UI.

9. The system of claim 8, wherein the controller renders on the rendering device information to inform the user of the task.

10. The system of claim 8, further comprising a storage device which stores information related to the task in accordance with a selected save option of a plurality of save options.

11. The system of claim 8, wherein the controller:
   sets a first working environment when it is determined that the assigned task corresponds with the stored task information for the user; and
   sets a second working environment different from the first working environment when it is determined that the assigned task does not correspond with the stored task information for the user.
12. The system of claim 11, wherein the first working environment comprises an application configuration selected by the user.

13. The system of claim 11, wherein the second working environment comprises a default application configuration.

14. The system of claim 8, wherein the controller:
   launches the assigned task;
   inputs user instance values entered by the user; and
   stores the user instance values in accordance with a save option.

15. A computer program stored on a computer readable memory medium, the computer program configured to system provide a user interface (UI) to accomplish a task, the computer program comprising:
   a program portion configured to:
   access user profile information that include stored task information for one or more users of a plurality of users;
   assign a task of a plurality of tasks to a user of the plurality of users;
   determine whether the assigned task corresponds with the stored task information for the user;
   set a working environment based upon the determination; and
   render the set working environment using the UI.

16. The computer program of claim 15, wherein the program portion is configured to render information to inform the user of the task.

17. The computer program of claim 15, wherein the program portion is configured to store on a storage device information related to the task in accordance with a selected save option of a plurality of save options.

18. The computer program of claim 15, wherein the program portion is configured to:
set a first working environment when it is determined that the assigned task corresponds with the stored task information for the user; and
set a second working environment different from the first working environment when it is determined that the assigned task does not correspond with the stored task information for the user.

19. The computer program of claim 18, wherein the first working environment comprises an application configuration selected by the user.

20. The computer program of claim 18, wherein the second working environment comprises a default application configuration.

21. The computer program of claim 15, wherein the program portion is configured to:
   launch the assigned task;
   input user instance values entered by the user; and
   store the user instance values in accordance with a save option.

22. A method for rendering a working user interface (UI) to a user, the method comprising acts of:
   receiving a business process flow comprising a series of successive tasks;
   assigning a task of the series of successive tasks to the user;
   determining whether the assigned task is to be performed for the first time;
   retrieving a default UI template for the task when it is determined that the assigned task is to be performed for the first time, the default UI template comprising applications for performing the task;
   rendering the default UI template for the task;
   customizing the UI based on input from the user; and
   saving the customized UI when the task is completed by the first user.

23. The method of claim 22, further comprising the acts of:
   determining whether the user customized UI exists;
retrieving the customized UI template when it is determined that the
customized UI exists; and
rendering the user customized UI template.

24. The method of claim 22, further comprising acts of:

requesting to save the customized UI according to a save option of a plurality
of save options; and

saving the customized UI in accordance with one of the save options of the
plurality of save options.

25. The method of claim 24, where the customized UI comprises a working
environment which comprises one or more applications and work in progress (WIP)
data which comprises user entered information.

26. The method of claim 25, wherein according to:

a first save option of the plurality of save options, the working environment is
saved as task type information;

according to a second save option of the plurality of save options, the working
environment is saved as a task type information and the WIP data is saved as task
instance information corresponding to a task instance; and

according to a third save option of the plurality of save options, the working
environment and WIP data is saved as task instance information.

27. A method of providing a user interface (UI) comprising acts of:

accessing task information assigned to a user of a plurality of users, the task
information comprising at least one of task type information related to one or more
task types and task instance information corresponding to a task instance of a
corresponding task type;

setting a working environment based upon the accessed task information;
rendering the set working environment using the UI;

customizing the working environment according to a user's selection; and
saving the customized working environment in accordance with a save option
of a plurality of save options.
28. The method of claim 27, wherein according to one save option of the plurality of save options, the task type information is updated to correspond to the working environment.

29. The method of claim 28, according to a further save option of the plurality of save options, the task type information is updated to correspond with the working environment and the task instance information is updated to correspond with work in progress (WIP) information comprising at least one user entry.

30. The method of claim 29, wherein according to a different save option of the plurality of save options, the task instance information is updated to correspond with the working environment and work in progress (WIP) information comprising at least one user entry.
START

PARSE WORKFLOW

LOAD GROUP/WORKER INFO.

EXTRACT TASKS FROM WORKFLOW

MATCH TASKS TO AVAILABLE USERS

ASSIGN TASKS TO USERS

SAME TYPE?

ASSIGN WORKPLACE ENVIRONMENT (WE)

ASSIGN DEFAULT WE

INFORM USER OF ASSIGNED TASKS

END

FIG. 3
**INTERNATIONAL SEARCH REPORT**

**INV. G06Q10/00**

**A. CLASSIFICATION OF SUBJECT MATTER**

**ADD.**

According to International Patent Classification (IPC) or to both national classification and IPC

**B. RELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

G06Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

**EPO-Internal , WPI Data**

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No</th>
</tr>
</thead>
</table>

Further documents are listed in the continuation of Box C

See patent family annex

* Special categories of cited documents

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

Date of the actual completion of the international search

20 May 2010

Date of mailing of the international search report

31/05/2010

Name and mailing address of the ISA/
European Patent Office, P B 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel (+31-70) 340-2040,
Fax (+31-70) 340-3016

Authorized officer

Fiorenzo Catalano, M
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Patent document cited in search report</td>
<td>Publication date</td>
<td>Patent family member(s)</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-----------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>WO 2007103806 A1</td>
<td>13-09-2007</td>
<td>NONE</td>
</tr>
<tr>
<td>US 2004039627 A1</td>
<td>26-02-2004</td>
<td>CA 2427215 A1</td>
</tr>
</tbody>
</table>