



(51) International Patent Classification:
G06F 9/44 (2006.01) **G06F 3/048** (2006.01)
G06F 3/14 (2006.01)

(21) International Application Number:
PCT/US2009/060891

(22) International Filing Date:
15 October 2009 (15.10.2009)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant (for all designated States except US):
HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. [US/US]; 11445 Compaq Center Drive W., Houston, TX 77070 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **SIMPSON, Edwin** [GB/GB]; Mail Stop T286, Filton Rd., Stoke Gifford, Bristol BS34 8QZ (GB). **RHODES, Michael** [GB/GB]; Bristol Filton B1 (BUK01) 2 T400, Filton Rd., Stoke Gifford, Bristol BS34 8QZ (GB).

(74) Agents: **DAKIN, Lloyd, Jr.** et al.; Hewlett-Packard Company, Intellectual Property Administration, 3404 E. Harmony Road, Mail Stop 35, Fort Collins, CO 80527-2400 (US).

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

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

[Continued on next page]

(54) Title: ZOOMING GRAPHICAL EDITOR

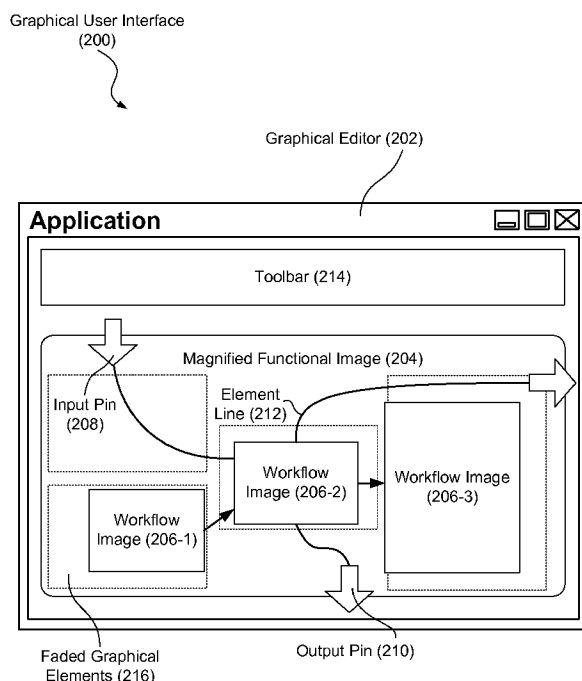


Fig. 2

(57) Abstract: A method for graphically editing a software application includes displaying a graphical user interface (100) for the software application to a user on a display device. The graphical user interface (100) includes at least one functional image (102) representing at least one functional component implemented by a processor executing the software application. The at least one functional component includes a plurality of workflow elements. The method further includes, in response to an action performed by the user, animatingly zooming in on the at least one functional image (102) in the graphical user interface (100), displaying to the user in a magnified view (204) of the at least one functional image (102) a plurality of workflow images (206) representing the workflow elements of the functional component, and allowing the user to modify the workflow elements by modifying the workflow images (206) within the magnified view to alter a functionality of the at least one functional component.



MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM,
TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW,
ML, MR, NE, SN, TD, TG).

— *as to applicant's entitlement to apply for and be granted
a patent (Rule 4.17(ii))*

Declarations under Rule 4.17:

— *as to the identity of the inventor (Rule 4.17(i))*

Published:

— *with international search report (Art. 21(3))*

ZOOMING GRAPHICAL EDITOR

5

BACKGROUND

10 **[0001]** Many software applications configured to interact with human users do so through a Graphical User Interface (GUI). A GUI is typically displayed to the user and exposes certain elements of the application's functionality to the user in an interactive manner. Often such a GUI is displayed to the user as a composite of multiple sections or "functional images,"
15 with each functional image modularly corresponding to one or more particular elements of functionality implemented by the application. For example, a GUI for an email application may include a first functional image for composing or reading one's email and a second functional image for browsing a list of email contacts. It is not uncommon for such a functional image to graphically
20 represent its particular functionality to the user interactively, for example through the use of icons, buttons, hyperlinks, pictures, diagrams, tables, graphs, and/or text.

[0002] Furthermore, it is often the case that the various functionalities, data, and content represented by such functional images interact with each
25 other in a way that is not always displayed directly to a user through the GUI. Typically this is done in an effort to preserve the advantages of application abstraction afforded by the GUI and to simplify the user's interaction with the application.

[0003] Nevertheless, users of such applications may differ widely in
30 their preferences for how the functional elements of an application work together as well as which functional elements are included in the GUI. Accordingly, some degree of customization is often a desirable attribute for a

software application. Still, many users do not have the necessary technical skills, or have the time to learn how to customize how an application works to their preferences, and many applications do not provide an intuitive way for users to customize certain aspects of the application functionality. Thus, some users who require application customization may be forced to take the costly and time consuming measure of hiring professional software engineers to make the desired changes. In some cases, they may choose not to hire professionals to make the desired changes and are thus left with inefficient editing tools.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The accompanying drawings illustrate various embodiments of the principles described herein and are a part of the specification. The illustrated embodiments are merely examples and do not limit the scope of the claims.

[0005] Fig. 1 is a diagram showing an illustrative graphical user interface displaying functional images, according to one embodiment of principles described herein.

[0006] Fig. 2 is a diagram showing an illustrative graphical user interface displaying workflow elements within a functional component, according to one embodiment of principles described herein.

[0007] Fig. 3 is a diagram showing an illustrative example application interface, according to one embodiment of principles described herein.

[0008] Fig. 4 is a diagram showing an illustrative example application interface after zooming in to edit a functional component, according to one embodiment of principles described herein.

[0009] Fig. 5 is a flowchart showing an illustrative method for graphically editing a software application, according to one embodiment of principles described herein.

[0010] Throughout the drawings, identical reference numbers designate similar, but not necessarily identical, elements.

DETAILED DESCRIPTION

[0011] As described above, many software applications utilize GUIs to dynamically interact with users. Often these GUIs include a number of functional images that modularly represent at least some aspects of the underlying functionality of the application. However, the users of such applications may differ widely in their preferences for how the various functional elements of an application work together. Accordingly, some degree of customization is often a desirable attribute for a software application. Still, many users do not have the necessary technical skills to customize how an application works to their preferences, and many applications do not provide an intuitive way for users to customize certain aspects of the application functionality. As such, it would be desirable to allow users to edit the relationships between functional elements in an intuitive manner within the GUI of the application itself.

[0012] In light of these and other considerations, the present specification discloses systems and methods in which a user can edit the relationships between functional elements of a software application directly within a GUI of the application. The GUI may include a graphical editor component that allows a user to modify how workflow elements associated with various functions of the application interact.

[0013] For example, a GUI of the application may present one or more functional images to a user, with each of the functional images representing an underlying functionality of the application. The user may be allowed to zoom in on a particular functional image, thereby triggering the launch of a graphical editor within the GUI. The graphical editor may display various workflow images overlaying the selected functional image, each of the workflow images representing a workflow element in the implementation of the functionality represented by the functional image. Having the workflow images overlaying the functional images upon launch of the graphical editor may give the user the impression that he or she is not launching a separate editor, but rather editing the application directly. The user may then manipulate the

workflow images to alter the way in which the underlying workflow elements interact with each other, thus changing the way the selected function operates. Workflow images may also be selectively added to or removed from the selected functional image to selectively add or remove workflow elements in the functionality represented by the functional image.

[0014] By animatingly zooming in on a functional image and displaying an overlaying graphical editor on the functional image representing the functionality to be edited and allowing the user to graphically alter the way in which workflow elements of the selected functionality interact, a much more intuitive method for customizing a software application is provided.

[0015] In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present systems and methods. It will be apparent, however, to one skilled in the art that the present apparatus, systems and methods may be practiced without these specific details. Reference in the specification to “an embodiment,” “an example” or similar language means that a particular feature, structure, or characteristic described in connection with the embodiment or example is included in at least that one embodiment, but not necessarily in other embodiments. The various instances of the phrase “in one embodiment” or similar phrases in various places in the specification are not necessarily all referring to the same embodiment.

[0016] Throughout the specification and in the appended claims, the term “functional image” in context with a graphical user interface is to be broadly interpreted to mean a graphical representation of elements used to perform a function within a computer application. The term “workflow images” in context with a graphical user interface is to be broadly interpreted to be a graphical representation of individual workflow elements used to form a functional component.

[0017] Throughout this specification and in the appended claims, the term “functional component” refers to a component of a software application that performs a particular function within that application. The term “workflow element” refers to a process or element that is part of a functional component.

[0018] Throughout this specification and in the appended claims, the term “graphical editor” refers to an aspect of a software application that allows a user to edit functions of particular application components by manipulating graphics through a graphical user interface.

5 **[0019]** Referring now to the figures, Fig. 1 is a diagram showing an illustrative graphical user interface (100) displaying functional images. According to one illustrative embodiment, a graphical user interface (100) includes a number of functional images (102-1, 102-2, 102-3) representing functional components. Each functional image includes a number of graphical
10 elements (104) allowing a user to interact with the function. Component lines (106) may be shown between functional images to indicate a relation between the underlying functional components. A toolbar (108) may provide various tools, functions, and options to a user.

[0020] A functional image (102-1, 102-2, 102-3) may include a group
15 graphical elements (104) including, but not limited to, images, buttons, hyperlinks, tables, graphs or pieces of text which allow a user to interact with a particular function within an application. One example of a functional component may be a chat function. The functional image representing the chat function may include a number of graphics to display various elements of the
20 chat function to a user and allow the chat function to receive input from the user.

[0021] Each graphical element presents a specific operation to a user of the graphical user interface (100). A number of underlying workflow elements provides the underlying functions of these graphical elements. Using
25 the chat function example, a graphical element (104) may be an icon or piece of text identifying another user to chat with. A graphical element (104) may also be a button allowing a user to send chats to other users.

[0022] Various functional components within an application may receive services from or provide services to other functional components. The
30 relation between different functional components may be illustrated by displaying component lines (106) between various functional images. The component lines (106) may indicate a relation between the underlying functions

of the functional images (102-1, 102-2, 102-3). In one embodiment, the component lines (106) may be hidden until an action performed by a user causes them to become visible. This action may include, but is not limited to, clicking a button or causing a mouse pointer to hover over a certain region on the user interface.

[0023] As mentioned above, an application embodying principles described herein may allow a user to open a particular function of the application for editing by selecting the functional image (102-1, 102-2, 102-3) associated with that function. The user may indicate this selection through any number of ways, including, but not limited to, positioning a cursor of the GUI over the desired functional image (102-1, 102-2, 103-3), clicking a button of a mouse or other user interface device, tapping on a touchpad, scrolling a mouse wheel, pressing a key or sequence of keys on a keyboard, and combinations thereof. In response to a selection of a functional image (102-1, 102-2, 102-3) by the user, the GUI may animatingly zoom in on the selected functional image (102-1, 102-2, 102-3).

[0024] Fig. 2 is a diagram showing an illustrative graphical user interface (200) after zooming in on a selected functional image (204). According to one illustrative embodiment, a graphical editor (202) may fade into view upon zooming in. The graphical editor of the present example displays a magnified view of the selected functional image (204), workflow images (206-1, 206-2, 206-3), overlaying faded graphical elements (216), an input pin (208), an output pin (210), element lines (212), and a toolbar (214).

[0025] When a user zooms in on a particular functional image, the magnified view of the selected functional image (204) may fill the screen. The graphical elements (216) associated with the zoomed functional image (204) may become faded, and workflow images (206-1, 206-2, 206-3) representing the workflow elements underlying the functionality of the functional component may appear over the faded graphical elements (216). In certain examples, the workflow images (206-1, 206-2, 206-3) appear over the graphical elements (216) in a manner representative of the relationship between the workflow

elements represented by the workflow images (206-1, 206-2, 206-3) and the graphical elements (216) of the selected functional image (204).

[0026] Examples of workflow elements that may be represented by workflow images (206-1, 206-2, 206-3) in this manner include, but are not limited to, interactive workflow elements, process workflow elements, and display workflow elements. An interactive workflow element may be a workflow element that receives some type of input from a user and initiates or otherwise manipulates application processes based on that input. Such input may be received through a corresponding graphical element within its assigned functional image. In a process workflow element, a process may be executed without displaying data to or receiving data from a user. A display workflow element may be used to display information or images to a user. In addition, documents and other data or content sources may be displayed as workflow elements. It will be readily apparent to those having skill in the art that the foregoing workflow elements are merely exemplary, and other types of workflow elements may be used according to principles described herein as may best suit a particular application of the principles.

[0027] In addition to the workflow images (206-1, 206-2, 206-3), at least one input pin (208) and/or at least one output pin (210) may also appear in the magnified functional image (204). An input pin (208) may indicate where information or services may be received from a separate function within the application. An output pin (210) may indicate where information and services are being sent to a separate function within the application. The user may have the ability to modify where information or services coming from an input pin (208) are directed. The user may also have the ability to modify the workflow elements to provide different information to the output pins (210). For example, the user may use a mouse to click on a particular input pin (208), output pin (210), or representation of data associated with the selected pin (208, 210) and drag with the mouse to associate the data from the selected pin (208, 210) with a different workflow image (206-1, 206-2, 206-3). Additionally or alternatively, the graphical editor (202) may provide means for the user to create additional input and output pins (208, 210) or to delete existing pins (208, 210). In one

embodiment, the input/output pins may be displayed at any time, regardless of whether the application is currently zoomed in on a functional image.

[0028] Element lines (212) may also be displayed to a user upon zooming in on a functional image. The element lines (212) may be displayed connecting workflow images (206) together. Element lines (212) may also link a workflow image (206) to an input pin (208) or an output pin (210). An element line (212) between two workflow images (206) may indicate a relation between the underlying workflow elements represented by the two workflow images (206). An example of a relation would be a flow of data. One workflow element may produce data which is then sent to be used by the other workflow element.

[0029] In one embodiment, when a user zooms in on a functional image, the toolbar (214) may be expanded to include additional functions and options for editing the application. A user may be able to use the toolbar (214) to perform tasks such as adding or removing workflow images (206), thus adding or removing functionality to the application or functional component.

[0030] The previous figures illustrate a broad description of an application embodying principles described herein. Fig. 3 and Fig. 4 illustrate a more detailed example of an application embodying graphical editing principles described herein.

[0031] Fig. 3 is a diagram showing an illustrative example application interface (300). According to one illustrative embodiment, an example of an application interface (300) embodying principles described herein may include a toolbar (302), a zoom button (304), an overview button (306), a “Reminders” functional image (308), a “What’s Happening” functional image (310), a “Documents” functional image (312), a “Members” functional image (314), input pins (316), output pins (318), and component lines (320).

[0032] As mentioned above, the toolbar (302) may provide standard functions and operations associated with various software applications. In one embodiment, a “zoom” button (304) may allow a user to zoom in on the selected functional image. If the application is currently zoomed in on a particular image, an “overview” button (306) may allow a user to zoom out to gain an overview of the functional images involved with the application.

[0033] The example application interface (300) shows example functional images representing underlying functional components of the application. One such example functional image is a “Reminders” image (308). The “Reminders” image (308) may represent a function that provides reminders to a user based on predefined settings as chosen by the user. The “Reminders” function may receive services from other functions as well as provide services to other functions.

[0034] An additional example of a functional image is a “What’s Happening” image (310). The “What’s Happening” image (310) may represent a functional component that provides a user with updates from other users as well as any other kind of suitable events the user desires to be informed about. Like other functional components, the “What’s Happening” function may receive services from other functions as well as provide services to other functions.

[0035] An additional example of a functional image is a “Documents” image (312). The “Documents” image (312) may represent a functional component that allows a user to browse documents and other files on either a local hard drive or a network hard drive which the user has permission to access. The “Documents” component may also allow a user to browse documents on a web-based service. Like other functional components, the “Documents” function may receive services from other functions as well as provide services to other functions.

[0036] An additional example of a functional image is a “Members” image (314). The “Members” image (314) may represent a functional component that allows a user to interact with members of a particular group. For example, the application may be an application used by an organization to allow its users to readily interact with other members of the organization. Like other functional components, the “Members” function may receive services from other functions as well as provide services to other functions.

[0037] In the example shown in Fig. 3, a component line (320) is displayed connecting the “Members” image (314) to the “What’s Happening” image (310). The component line (320) may be connected from an output pin (318) on the “Members” image (314) to an input pin (316) on the “What’s

Happening” image (310). The component line indicates that the “What’s Happening” functional component receives services from the “Members” functional component. These services may include current statuses of various members to be displayed in the “What’s Happening” image (310).

5 **[0038]** When a user desires to edit a particular function, the user may select the functional image (308, 310, 312, 314) associated with that function, thereby causing the application interface GUI to animatingly zoom in on the selected functional image (308, 310, 312, 314). Fig. 4 is a diagram showing an illustrative zoomed in example application interface (400) allowing a user to

10 graphically edit the underlying functional component. Fig. 4 illustrates an application interface when zoomed in on the “What’s Happening” image (310) illustrated in Fig. 3 and described above. According to one illustrative embodiment, the zoomed in example application interface (400) may include an additional toolbar (404) and a zoomed in functional image (402). On top of the

15 zoomed in functional image (402), number of workflow images may be displayed. These images may include a “Fetch Documents” image (406), a “Fetch Members” image (408), a “Fetch Interests” image (410), a “Filter Updates List” image (412), an “Order/Group” image (414), a “Create List” image (416), and a list preview (418). The application may also display workflow

20 element connector lines (420).

[0039] When a user performs an action instructing the GUI to zoom in on a functional image, that image becomes enlarged on the user’s screen. The animation of the zoom may use a variety of algorithms to graphically display a zooming process.

25 **[0040]** In one embodiment, when a user zooms in on a functional image, an additional toolbar (404) may appear. The additional toolbar (404) may provide additional functions and options relating specifically to graphically editing the application. For example, the toolbar (404) may include features that will allow a user to add or remove workflow images from the user interface

30 (400), thus adding or removing workflow elements from the functional component being edited. The toolbar (404) may also provide features that will

allow a user to make various changes to the workflow images, thus changing the functionality of the workflow elements represented by the workflow images.

[0041] As mentioned above, when a user zooms in on a functional image, the graphical elements of that image may fade and workflow images may appear over the zoomed functional image (402). The workflow images represent workflow elements that are involved with the operations of the functional component represented by the zoomed in functional image. Fig. 4 illustrates several workflow images representing workflow elements which may be involved with the operations of the “What’s Happening” function described in Fig. 3.

[0042] One example of a workflow image is a “Fetch Documents” image (406). The “Fetch Documents” image (406) may represent a workflow element which receives a data feed from the “Documents” function. The “Fetch Documents” workflow element may provide services to a filtering workflow element which will be discussed in more detail below.

[0043] An additional example of a workflow image is a “Fetch Members” image (408). The “Fetch Members” image (408) may represent a workflow element which receives a data feed from the “Members” function. The “Fetch Members” workflow element may provide services to a filtering workflow element which will be discussed in more detail below.

[0044] An additional example of a workflow image is a “Fetch Interests” image (410). The “Fetch Interests” image (410) may represent a workflow element which receives a data feed from an internet site determined by a user. For example, a user may choose to receive data from a Really Simple Syndication (RSS) feed for news or sports information. The “Fetch Interests” workflow element may provide services to a filtering workflow element which will be discussed in more detail below. In one embodiment, the user may use the additional toolbar (404) to make adjustments to the “Fetch Interests” workflow image. These changes may include what sources from which “interests” may be pulled.

[0045] An additional example of a workflow image is a “Filter Updates” image (412). The “Filter Updates” image (412) may represent a

workflow element which processes the data received from a multiple of other workflow elements. The “Filter Updates” workflow element may filter the data received from other services and determine which data to pass on to a “Create List” workflow element.

5 **[0046]** An additional example of a workflow image is a “Create List” image (416). The “Create List” image (416) may represent a workflow element which takes all of the received filtered data and creates a list of “what’s happening” and displays it in a manner which is readily viewed by a user. In one embodiment, the user may use the additional toolbar (404) to make
10 changes to the “Create List” image which will change how the displayed list is formatted.

[0047] An additional example of a workflow image is an “Order/Group” image (414). The “Order/Group” image may represent a workflow element which gives specific properties as chosen by a user to the create list workflow
15 element. In one embodiment, a user may have the option to make changes to the “order/group” workflow image which will affect how the “what’s happening” list is ordered or grouped. It may be the case that the workflow does not originally contain the order/group workflow element. This workflow element could be added by the user using the additional toolbar (404).

20 **[0048]** In one embodiment, a list preview (418) may display to a user, a preview of the “What’s Happening” list. The preview of the list may be instantly updated as the user makes changes to the workflow elements within the user interface (400). This may give the user a better idea of how the changes will affect the application without having to exit the graphical editor.

25 **[0049]** Workflow element lines (420) may be displayed between the workflow images to indicate a relation between the workflow elements represented by the workflow images. For example, a workflow element line may be displayed between the “Fetch Documents” image (406) and the “Filter Updates” image (412). This workflow element line (420) indicates that the
30 “Fetch Documents” workflow element provides services to the “Filter Updates” workflow element. In a further example, a workflow element line may connect an input pin to the “Fetch Members” image (408). This workflow element line

may indicate that the “Fetch Members” workflow element is receiving services from a source which is external to the “What’s Happening” functional component which is currently being edited.

5 **[0050]** In one embodiment, a user may further zoom in on a workflow image to gain deeper level access to the application. For example, a user could zoom in on a workflow image and the code used in the image could be displayed and provide the user with a means to alter the code if desired. This option may be useful for technically skilled users who wish to further customize an application to their preferences.

10 **[0051]** An advantage of a graphical editor embodying principles described herein is that it gives a user an intuitive method for editing an application. Other applications which are configured to allow a user to edit them typically jump to a separate screen or separate interface upon entering an editing mode. Other graphical editors may lay out workflow images with
15 different sizes and positions than the underlying functional images. These approaches make it difficult for the users to correlate certain workflow elements with the actual functions they are providing to the application. By providing a graphical editor which overlays the functions it is editing, the user can more intuitively understand how the changes being made may affect the application.
20 It will also make it easier for a user to locate the functions they wish to edit. In addition, the zooming aspect provides the intuitive feel of “looking under the hood” regarding specific functions within the application.

[0052] Fig. 5 is a flowchart showing an illustrative method (500) for zooming in to edit an application. According to one illustrative embodiment, a
25 graphical user interface for a software application is displayed (step 502) to a user on a display device, the graphical user interface may include at least one functional image representing at least one functional component implemented by a processor executing the software application. The at least one functional component may include a plurality of workflow elements. In the graphical
30 interface, at least one component connector line between two functional images may be displayed (step 504) to a user. The at least one component connector indicates a relation between the two functional images. In response to an

action performed by the user, the at least one functional image in the graphical user interface is animatingly zoomed (step 506) in on. In the graphical interface, a plurality of workflow images representing workflow elements of the functional component may be displayed (step 508) to the user. The user may
5 be allowed (step 510) to modify the workflow elements by modifying the workflow images to alter functionality of the at least one functional component. In a graphical user interface, at least one element connector line between the plurality of workflow images may be displayed (step 512) to the user. The element connector line may indicate a relation between the workflow elements.

10 **[0053]** In sum, a graphical editor allows a user to zoom in on a functional image of an application. Upon zooming in, graphical editing tools appear as well as workflow images representing underlying workflow elements of the functional component currently being edited. A graphical editor embodying principles described herein provides a more intuitive method for
15 graphically editing an application to suit the user's preferences.

[0054] The preceding description has been presented only to illustrate and describe embodiments and examples of the principles described. This description is not intended to be exhaustive or to limit these principles to any precise form disclosed. Many modifications and variations are possible in light
20 of the above teaching.

CLAIMS

WHAT IS CLAIMED IS:

- 5 1. A method for graphically editing a software application, the method comprising:
- displaying a graphical user interface (100) for said software application to a user on a display device, said graphical user interface (100) comprising at least one functional image (102) representing at least one functional component
- 10 implemented by a processor executing said software application, said at least one functional component comprising a plurality of workflow elements;
- in response to an action performed by said user, animatingly zooming in on said at least one functional image (102) in said graphical user interface (100);
- 15 displaying to said user in a magnified view (204) of said functional image (102) a plurality of workflow images (206) representing said plurality of workflow elements of said functional component; and
- allowing said user to modify said plurality of workflow elements by modifying said workflow images (206) within said magnified view (204) to alter a
- 20 functionality of said at least one functional component.
2. The method of claim 1, further comprising displaying to said user in said graphical user interface (100) at least one component connector line (106) between two said functional images (102), said at least one component
- 25 connector line (106) indicating a relation between said two functional images (102).
3. The method of claim 2, in which said at least one component connector line (106) is connected to at least one of said functional images (102) in said
- 30 graphical user interface (100) through one of: a displayed input pin (208) and a displayed output pin (210).

4. The method of any preceding claim, further comprising displaying to said user in said graphical user interface (100) at least one element connector line (212) between said plurality of workflow images (206), said element connector line (212) indicating a relation between said plurality of workflow elements.

5

5. The method of any preceding claim, further comprising, allowing said user to modify an input or an output to said functional component with said graphical user interface (100).

10

6. The method of any preceding claim, in which said action comprises at least one of: using a scrolling wheel on a mouse, double clicking on a region within a functional component interface, and clicking a button within a toolbar (214).

15

7. A system for graphically editing a software application comprising:
a processor; and
a display device communicatively coupled to said processor;
in which said processor is configured to, while executing said software application:

20

display on said display device a graphical user interface (100) comprising at least one functional image (102) representing a functional component of said software application, said functional component comprising a plurality of workflow elements;

25

responsive to a user action, cause said graphical user interface to animatingly zoom in on said at least one functional image (102);

display in a magnified view (204) of said at least one functional image (102) a number of workflow images (206) representing said plurality of workflow elements; and

30

allow said user to modify said plurality of workflow elements by modifying said workflow images (206) within said magnified view (204) to alter a functionality of said at least one functional component.

.

8. The system of claim 7, in which said graphical user interface (100) further comprises at least one component connector line (106) between two functional images (102), said at least one component connector line (106) indicating a relation between said two functional images (102).

9. The system of claim 7 or 8, in which said at least one component connector line (106) is connected to at least one of said functional images (102) in said graphical user interface (100) through one of: a displayed input pin (208) and a displayed output pin (210).

10. The system of any of claims 7-9, in which said processor is further configured to display at least one element connector line (212) in said graphical user interface (100) between said workflow images (206), said at least one element connector line (212) indicating a relation between said plurality of workflow elements.

11. The system of any of claims 7-10, in which said processor is further configured to allow a user to modify an input or an output to said functional component.

12. The system of any of claims 7-11, in which said action comprises at least one of: using a scrolling wheel on a mouse, double clicking on a region within a functional component interface, and clicking a button within a toolbar (214).

13. A computer program product for graphically editing a software application, said computer program product comprising:

a computer readable storage medium having computer readable code embodied therewith, said computer readable program code comprising:

computer readable program code configured to:

display a graphical user interface (100) for said software application to a user, said graphical user interface (100) comprising at least one

functional image (102) representing at least one functional component implemented by a processor executing said software application, said at least one functional component comprising a plurality of workflow elements;

allow a user to perform an action causing said graphical user interface (100) to animatingly zoom in on said at least one functional image (102);

display to said user in a magnified view (204) of said at least one functional component a plurality of workflow images (206) representing said plurality of workflow elements; and

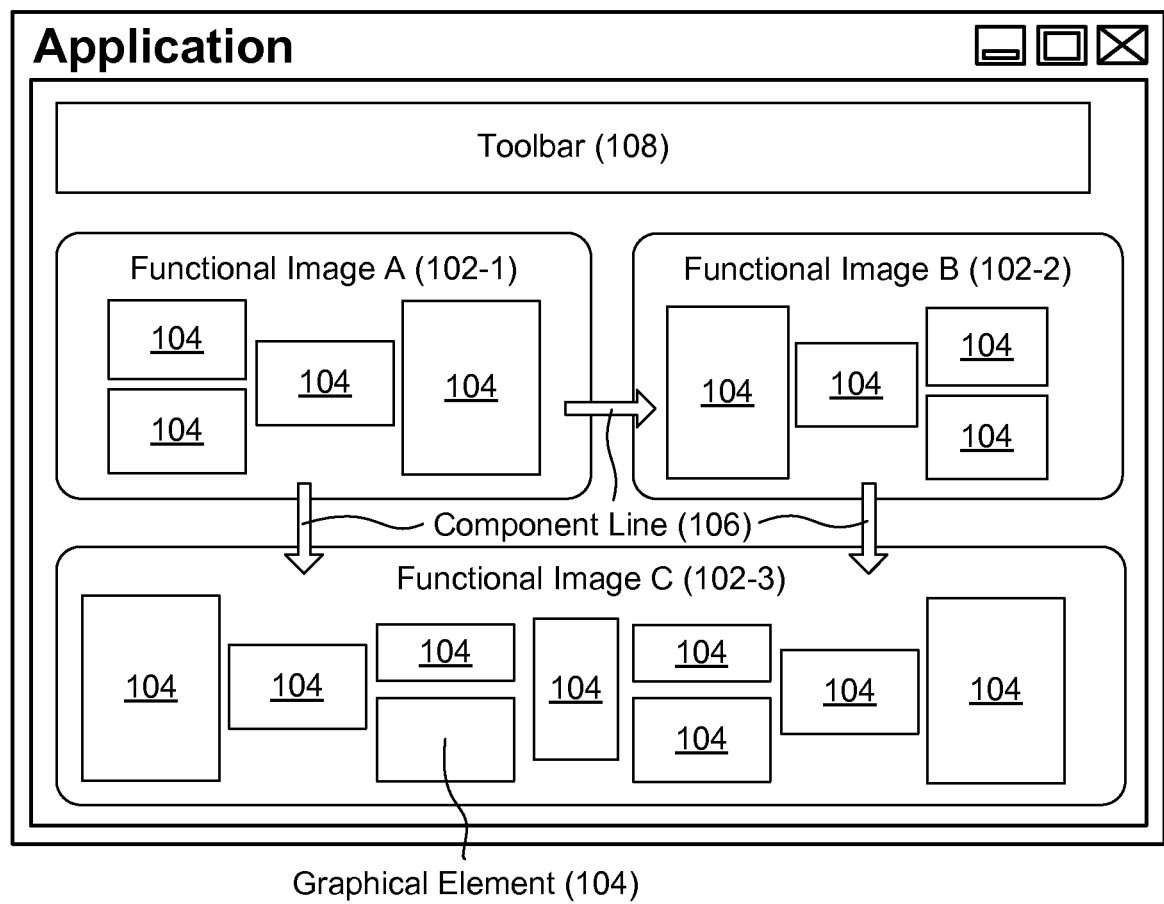
allow said user to modify said plurality of workflow elements by modifying said workflow images (206) within said magnified view to alter a functionality of said at least one functional component.

14. The computer program product of claim 13, in which said graphical user interface (100) further comprises at least one component connector line (106) between two said functional images (102), said at least one component connector line (106) indicating a relation between said two functional images (102).

15. The computer program product of claim 13 or 14, in which said computer readable program code is further configured to display to said user through said graphical editor, at least one element connector line (212) between said plurality of workflow images (206) representing said plurality of workflow elements, said at least one element connector line (212) indicating a relation between said workflow elements.

1/5

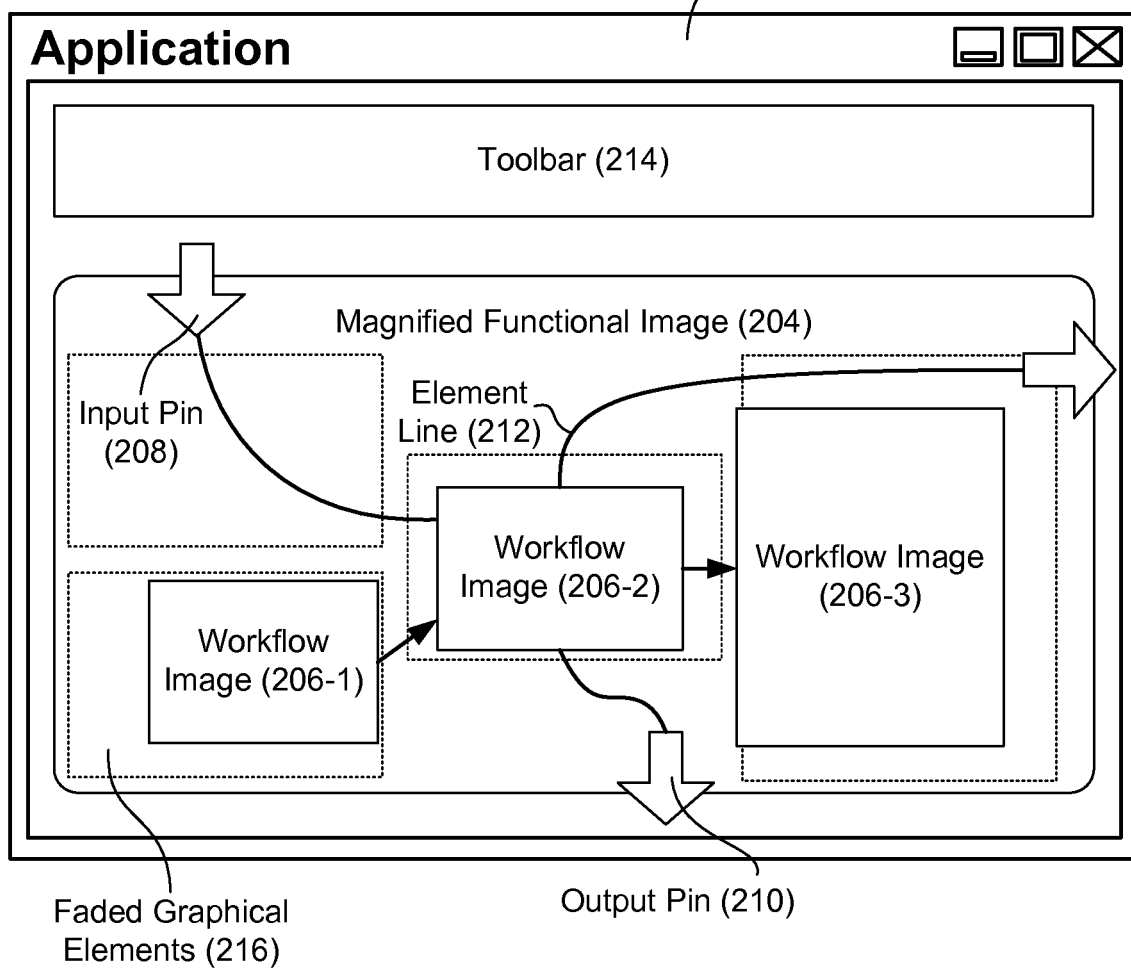
Graphical User Interface
(100)

**Fig. 1**

2/5

Graphical User Interface
(200)

Graphical Editor (202)

**Fig. 2**

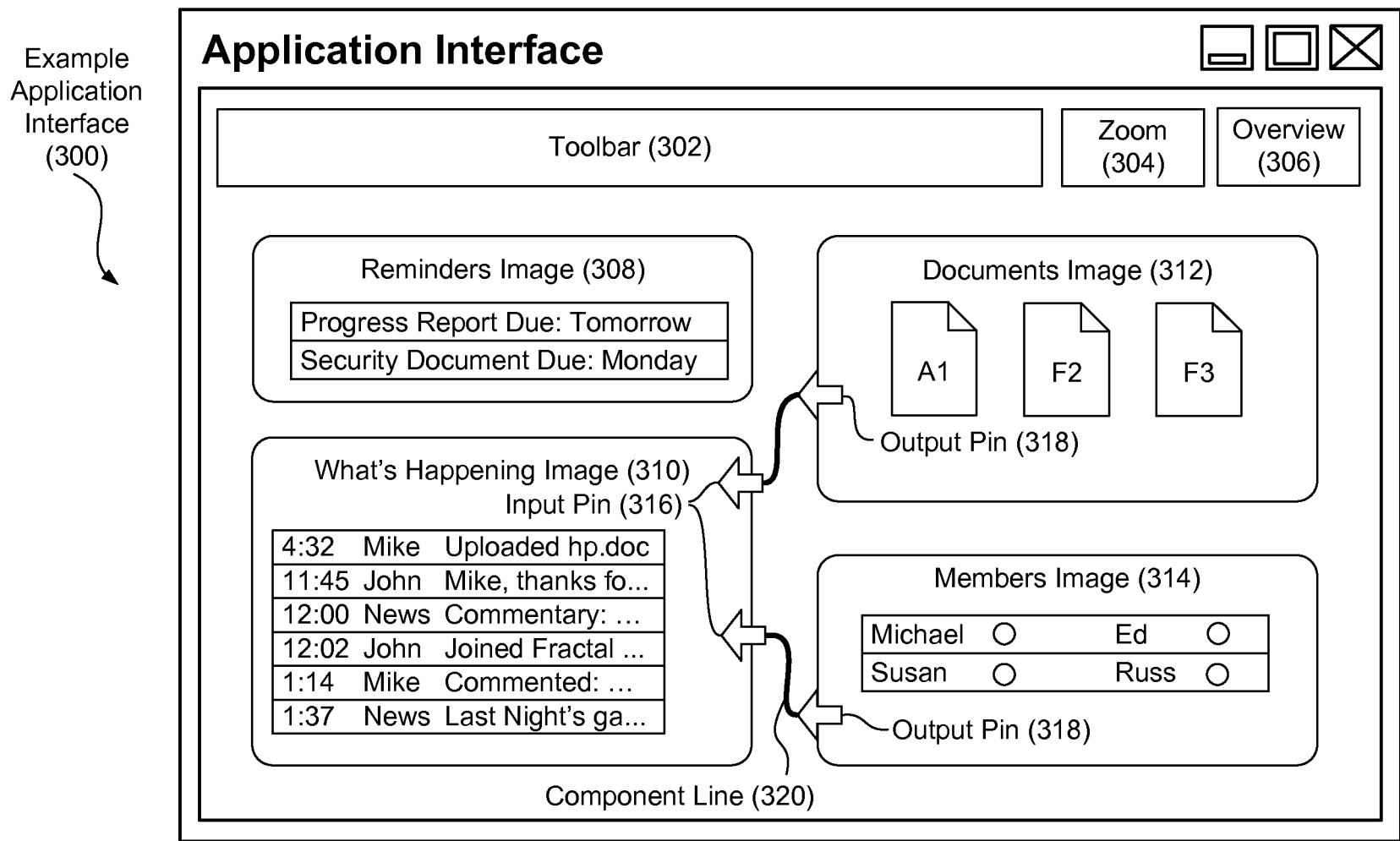


Fig. 3

Zoomed In
Example
Application
Interface
(400)

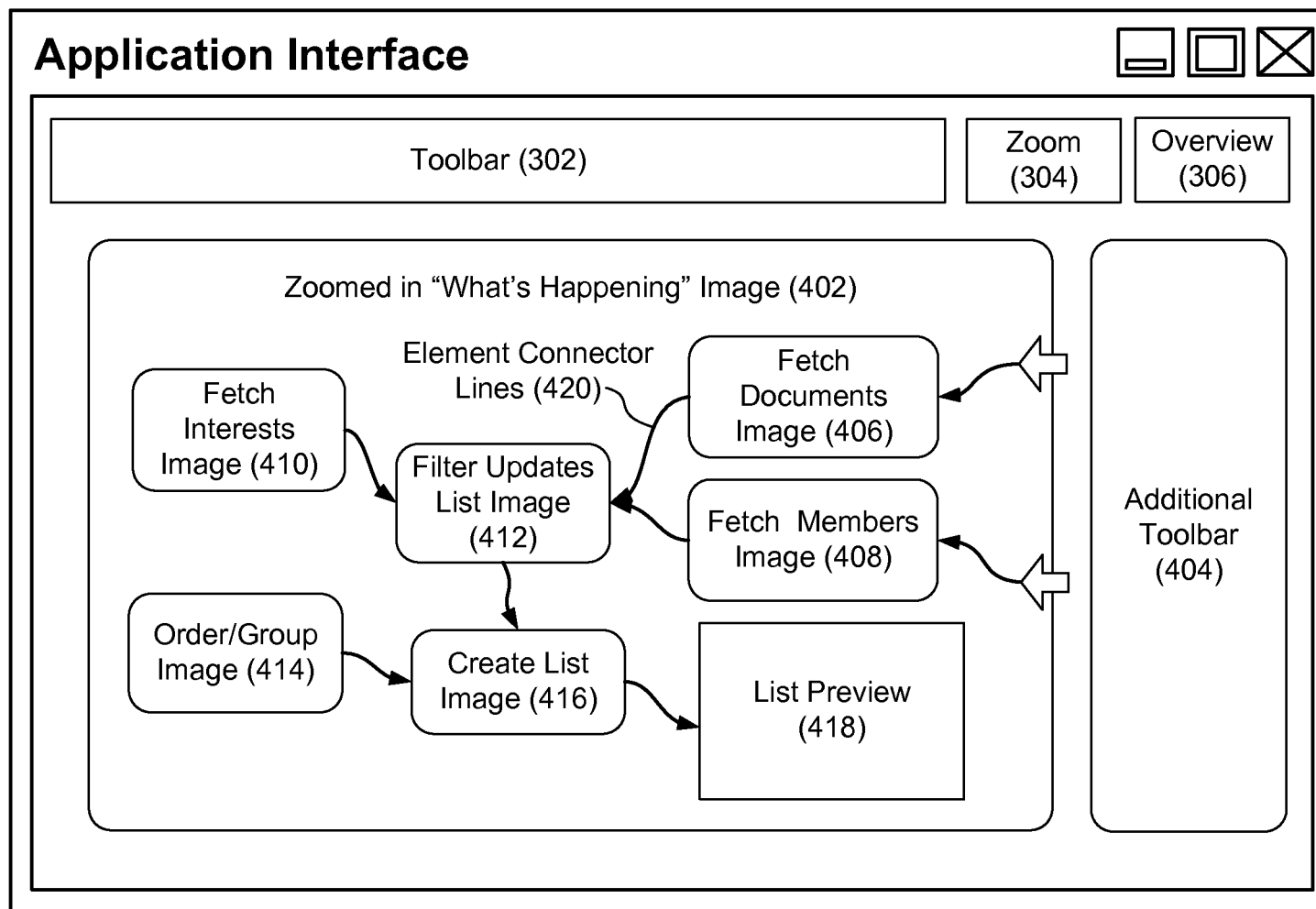
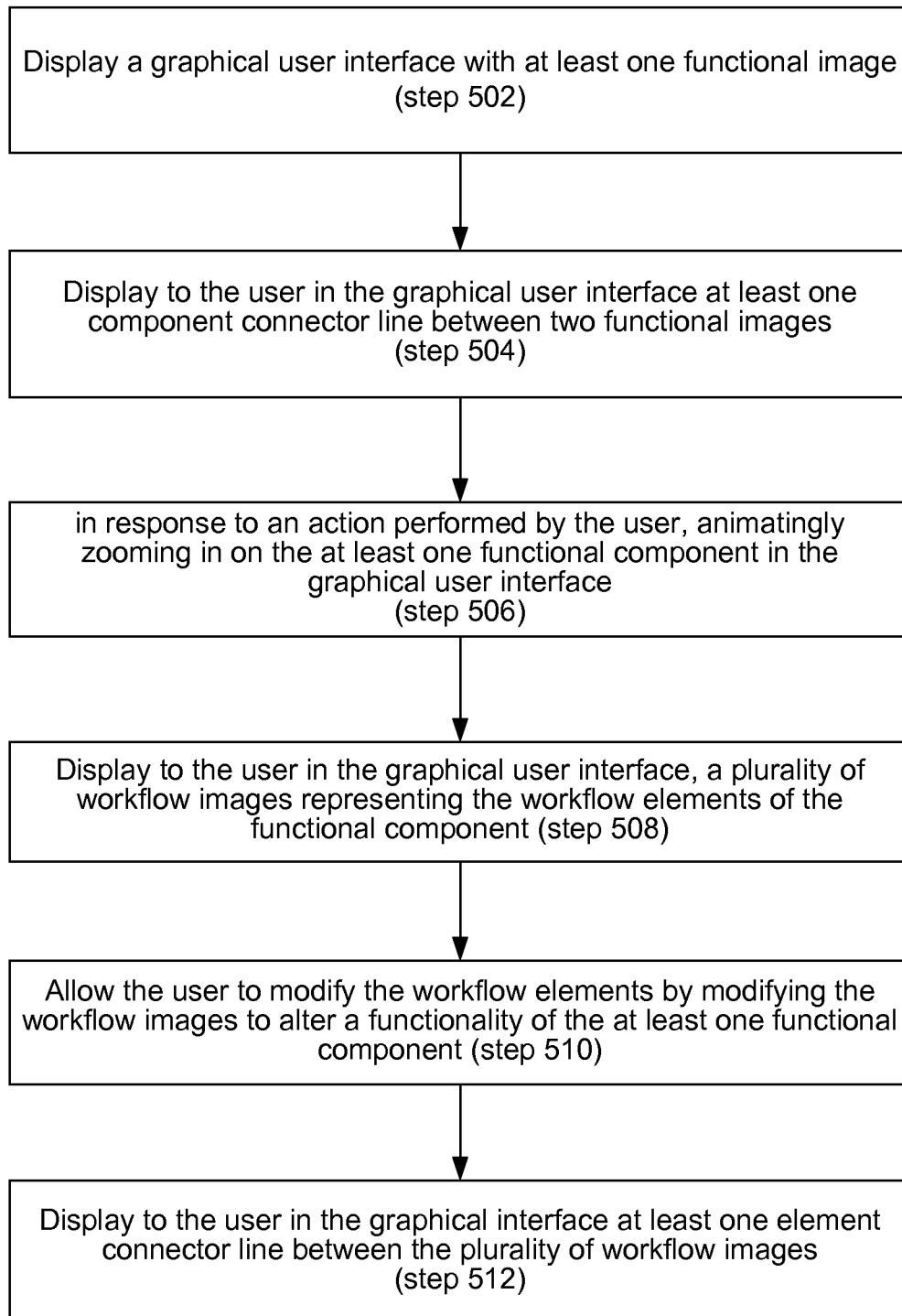


Fig. 4

5/5

500

**Fig. 5**

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2009/060891**A. CLASSIFICATION OF SUBJECT MATTER****G06F 9/44(2006.01)i, G06F 3/14(2006.01)i, G06F 3/048(2006.01)i**

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F 9/44; G06F 3/048; G06F 3/14; G06T 3/40

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

Korean utility models and applications for utility models

Japanese utility models and applications for utility models

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

eKOMPASS(KIPO internal) NDSL, Google & Keywords: GUI, graphical user interface, zoom*, magnif*, maxim*, edit, workflow*, function*

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2008-046849 A (FUJITSU LTD.) 28 February 2008 See paragraphs [8][9][10][45]~[52].	1-15
A	US 2009-0254867 A1 (FAROUKI KARIM et al.) 08 October 2009 See figures 3,4; paragraphs [37][52].	1-15
A	US 2007-0245300 A1 (CHAN BENJAMIN et al.) 18 October 2007 See figure 1; paragraphs [13][46]~[49].	1-15
A	US 6037939 A1 (KASHIWAGI KOUICHI et al.) 14 March 2000 See Summary.	1-15



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 application or patent 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 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

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

06 JULY 2010 (06.07.2010)

Date of mailing of the international search report

07 JULY 2010 (07.07.2010)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
Government Complex-Daejeon, 139 Seonsa-ro, Seo-
gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

YOON, Hye Sook

Telephone No. 82-42-481-8370



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2009/060891Patent document
cited in search reportPublication
datePatent family
member(s)Publication
date

JP 2008-046849 A

28.02.2008

None

US 2009-0254867 A1

08.10.2009

None

US 2007-0245300 A1

18.10.2007

None

US 6037939 A1

14.03.2000

JP 03-247036 B2

02.11.2001

JP 03-278329 B2

15.02.2002

JP 09-091114 A

04.04.1997

JP 09-097150 A

08.04.1997

JP 3247036 B2

15.01.2002

JP 3278329 B2

30.04.2002