



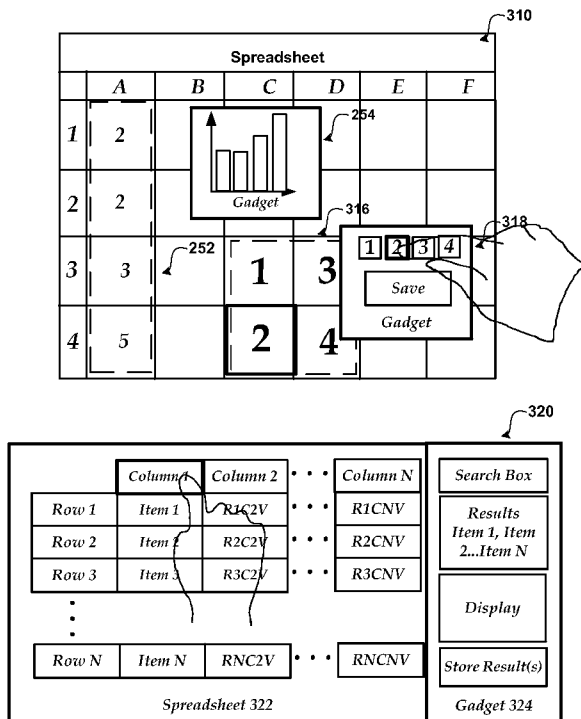
- (51) **International Patent Classification:**
G06F 3/048 (2006.01) G06F 9/44 (2006.01)
G06F 3/14 (2006.01)
- (21) **International Application Number:**
PCT/US2012/063133
- (22) **International Filing Date:**
2 November 2012 (02.11.2012)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
13/289,663 4 November 2011 (04.11.2011) US
- (71) **Applicant (for all designated States except US):** MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).
- (72) **Inventors:** WALDMAN, Lawrence; Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). PATEL, Keyur

Rahul; Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). PRISH, Shahar; Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). BURKE, Eoin James; Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). BATTAGIN, Daniel; Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US). SIMMONS, Jon Garrett; Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, Washington 98052-6399 (US).

- (81) **Designated States (unless otherwise indicated, for every kind of national protection available):** AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, 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,

[Continued on next page]

(54) Title: INTERACTION BETWEEN WEB GADGETS AND SPREADSHEETS



(57) **Abstract:** Gadgets integrate with spreadsheets and the spreadsheet calculation engine. A gadget is bound to a range of cells (one or more) within a sheet. When one or more of the cells of the bound range is accessed (e.g. selected, hovered over, edited, deleted, added), the gadget is notified. The gadget may use this information for updating a display rendered by the gadget and/or performing some other action. The gadget uses an Application Programming Interface (API) to create the binding, communicate with the spreadsheet, and perform actions in the spreadsheet. The gadget is a Named Item object within the spreadsheet such that the gadget may be viewed/displayed differently from other objects. Out of date requests from the gadget are attempted to be detected and prevented from being processed such that the gadget is not acting on the latest data.

Fig. 3



NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))

Published:

- with international search report (Art. 21(3))
- 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))

INTERACTION BETWEEN WEB GADGETS AND SPREADSHEETS

BACKGROUND

[0001] Spreadsheets are used for many different purposes, such as analyzing, managing, and sharing information. For example, a user may configure visualization tools to help the track and highlight important data trends. Many users and businesses attempt to add custom functionality to a spreadsheet application but programming these additions can be difficult and limited to the functionality exposed by the spreadsheet application.

SUMMARY

[0002] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

[0003] GadgetGadgets integrate with spreadsheets and the spreadsheet calculation engine. A gadget can be bound to a range of cells (one or more) and/or to an object within a sheet. For explanatory purposes, the term “bound to a range of cells” includes bound to at least one of: a cell and an object within the spreadsheet. When one or more of the cells of the bound range is accessed (e.g. selected, hovered over, edited, deleted, added), the gadget can be notified by an application programming interface (API) or other system. The gadget may use this information for updating a display rendered by the gadget and/or performing some other action. The gadget may render data obtained from the spreadsheet while the spreadsheet application protects the backend logic and calculation details in the spreadsheet itself to any unauthorized users viewing the gadget. The gadget uses an Application Programming Interface (API) to create the binding, communicate with the spreadsheet, and perform actions in the spreadsheet. The gadget can be a Named Item object within the spreadsheet such that the gadget may be viewed/displayed differently from other objects (e.g. selected for publication in a report, dashboard, ...). Some implementations of gadgets will attempt to detect and prevent out of date requests from being made such that the gadget or spreadsheet application are not acting on the latest data.gadgetgadget.

30

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] FIGURE 1 illustrates an exemplary computing device;

[0005] FIGURE 2 illustrates an exemplary system integrating gadgets with a spreadsheet and spreadsheet calculation engine;

[0006] FIGURE 3 shows an exemplary displays of gadgets and spreadsheets;

[0007] FIGURE 4 is a functional block diagram illustrating the generation of a named object view report across multiple object sources including gadgets;

[0008] FIGURE 5 shows a process for binding a gadget to a spreadsheet; and

[0009] FIGURE 6 shows a process for a gadget interacting with a spreadsheet.

5

DETAILED DESCRIPTION

[0010] Referring now to the drawings, in which like numerals represent like elements, various embodiments will be described. In particular, FIGURE 1 and the corresponding discussion are intended to provide a brief, general description of a suitable computing environment in which embodiments may be implemented.

10

[0011] Generally, program modules include routines, programs, components, data structures, and other types of structures that perform particular tasks or implement particular abstract data types. Other computer system configurations may also be used, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like.

15

Distributed computing environments may also be used where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

20

[0012] Referring now to FIGURE 1, an illustrative computer architecture for a computer 100 utilized in the various embodiments will be described. The computer architecture shown in FIGURE 1 may be configured as a mobile computing device (e.g. smartphone, notebook, tablet ...) or desktop computer and includes a central processing unit 5 ("CPU"), a system memory 7, including a random access memory 9 ("RAM") and a read-only memory ("ROM") 10, and a system bus 12 that couples the memory to the central processing unit ("CPU") 5.

25

30

[0013] A basic input/output system containing the basic routines that help to transfer information between elements within the computer, such as during startup, is stored in the ROM 10. The computer 100 further includes a mass storage device 14 for storing an operating system 16, application program(s) 24 such as a spreadsheet application and/or other application that works with tabular data, Web browser 25, files 27 (e.g. spreadsheets) and gadget manager 26 which will be described in greater detail below. The Web browser 25 is operative to request, receive, render, and provide interactivity with electronic content, such as Web pages, videos, documents, and the like. According to an

embodiment, the Web browser comprises the INTERNET EXPLORER Web browser application program from MICROSOFT CORPORATION.

5 [0014] The mass storage device 14 is connected to the CPU 5 through a mass storage controller (not shown) connected to the bus 12. The mass storage device 14 and its associated computer-readable media provide non-volatile storage for the computer 100. Although the description of computer-readable media contained herein refers to a mass storage device, such as a hard disk or CD-ROM drive, the computer-readable media can be any available media that can be accessed by the computer 100.

10 [0015] By way of example, and not limitation, computer-readable media may comprise computer storage media and communication media. Computer storage media includes volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, Erasable Programmable Read Only Memory
15 (“EPROM”), Electrically Erasable Programmable Read Only Memory (“EEPROM”), flash memory or other solid state memory technology, CD-ROM, digital versatile disks (“DVD”), or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by the computer 100.

20 [0016] According to various embodiments, computer 100 may operate in a networked environment using logical connections to remote computers through a network 18, such as the Internet. The computer 100 may connect to the network 18 through a network interface unit 20 connected to the bus 12. The network connection may be wireless and/or wired. The network interface unit 20 may also be utilized to connect to
25 other types of networks and remote computer systems. The computer 100 may also include an input/output controller 22 for receiving and processing input from a number of other devices, including a touch input device 28. The touch input device may utilize any technology that allows single/multi-touch input to be recognized (touching/non-touching). For example, the technologies may include, but are not limited to: heat, finger pressure,
30 high capture rate cameras, infrared light, optic capture, tuned electromagnetic induction, ultrasonic receivers, transducer microphones, laser rangefinders, shadow capture, and the like. According to an embodiment, the touch input device may be configured to detect near-touches (i.e. within some distance of the touch input device but not physically touching the touch input device). The touch input device 28 may also act as a display.

The input/output controller 22 may also provide output to one or more display screens, a printer, or other type of output device.

[0017] A camera and/or some other sensing device may be operative to record one or more users and capture motions and/or gestures made by users of a computing device.

5 Sensing device may be further operative to capture spoken words, such as by a microphone and/or capture other inputs from a user such as by a keyboard and/or mouse (not pictured). The sensing device may comprise any motion detection device capable of detecting the movement of a user. For example, a camera may comprise a MICROSOFT KINECT® motion capture device comprising a plurality of cameras and a plurality of
10 microphones.

[0018] Embodiments of the invention may be practiced via a system-on-a-chip (SOC) where each or many of the components/processes illustrated in the FIGURES may be integrated onto a single integrated circuit. Such a SOC device may include one or more processing units, graphics units, communications units, system virtualization units and
15 various application functionality all of which are integrated (or “burned”) onto the chip substrate as a single integrated circuit. When operating via a SOC, all/some of the functionality, described herein, may be integrated with other components of the computing device/system 100 on the single integrated circuit (chip).

[0019] As mentioned briefly above, a number of program modules and data files may
20 be stored in the mass storage device 14 and RAM 9 of the computer 100, including an operating system 16 suitable for controlling the operation of a networked personal computer, such as the WINDOWS 7® operating system from MICROSOFT CORPORATION of Redmond, Washington. According to one embodiment, the operating system is configured to include support for touch input device 23. According to another
25 embodiment, gadget manager 26 may be utilized to process some/all of the touch input that is received from touch input device 28.

[0020] The mass storage device 14 and RAM 9 may also store one or more program modules. In particular, the mass storage device 14 and the RAM 9 may store one or more application programs 24, such as a spreadsheet application. According to an embodiment,
30 the spreadsheet application is MICROSOFT EXCEL. In conjunction with the operation of the application, gadget manager 26 is used to integrate one or more gadgets with a spreadsheet. Gadget manager 26 may be configured as an application/process and/or as part of a cloud based multi-tenant service that provides resources (e.g. services, data ...) to

different tenants (e.g. MICROSOFT OFFICE 365, MICROSOFT SHAREPOINT ONLINE).

[0021] Generally, gadget manager 26 is configured to process input and calls that include a gadget and a spreadsheet. The gadgets may be first party and/or third party web extensions that use standard web technologies (e.g. XML, HTML, FLASH, JAVASCRIPT, ...) that are hosted by another server (e.g. 19) and associated with the spreadsheet application. Some gadgets may be integrated with a sheet's contents whereas other gadgets are separate from the sheet. The gadgets may be used online and/or offline with the spreadsheet and may be accessed through a link. A gadget is bound to a range of cells (one or more) within a sheet. When one or more of the cells of the bound range is accessed (e.g. selected, hovered over, edited, deleted, added), the gadget is notified. The gadget may use this information for updating a display rendered by the gadget and/or performing some other action. The gadget may render data obtained from the spreadsheet while the spreadsheet application protects the backend logic and calculation details in the spreadsheet itself to any unauthorized users viewing the gadget. The gadget uses an Application Programming Interface (API) 29 to communicate with the spreadsheet. The API 29 comprises functionality for binding the gadget to the spreadsheet, obtaining/putting content from/into the spreadsheet, changing a display of the spreadsheet, requesting a display of a spreadsheet user interface, and the like. According to an embodiment, the gadget is a Named Item object within the spreadsheet such that the gadget may be viewed/displayed differently from other objects (e.g. selected for electronic display/publication in a report, dashboard, ...). Out of date requests from the gadget are attempted to be detected and prevented from being processed such that the gadget is not acting on the latest data. Additional details regarding the operation of gadget manager 26 will be provided below.

[0022] FIGURE 2 illustrates an exemplary system integrating gadgets with a spreadsheet and spreadsheet calculation engine. As illustrated, system 200 includes application program 24, service 210, gadget manager 26, data store 240, gadget sites 260, and input device/display 202. According to an embodiment, display 202 is a touch screen device.

[0023] According to one embodiment, application program 24 is a spreadsheet application that is configured to receive input from a touch-sensitive input device 202. For example, gadget manager 26 may provide information to application 24 and/or service 210 in response to a user's finger (i.e. finger on hand 256) selection of a menu option, a gadget

component (e.g. selecting a bar in an interactive bar chart), selection of one or more cells of a spreadsheet (e.g. spreadsheet 250), selection of a column/row in the spreadsheet (e.g. selection 252), and the like.

5 [0024] As illustrated, service 210 is a cloud based and/or enterprise based service that is configured to provide services relating to one or more productivity applications (e.g. MICROSOFT EXCEL, MICROSOFT SHAREPOINT). Service 210 may also be configured as a client based application. Although system 200 shows a productivity service and/or a content management service, other services/applications may be configured to use slicers with tabular data. The functionality of application program 24
10 may be included within service 210.

[0025] As illustrated, service 210 is a multi-tenant service that provides resources 215 and services to any number of tenants (e.g. Tenants 1-N). According to an embodiment, multi-tenant service 210 is a cloud based service that provides resources/services 215 to tenants subscribed to the service and maintains each tenant's
15 data separately and protected from other tenant data. For example, service 210 may be configured to provide services corresponding to productivity applications (e.g. spreadsheet, content management, word processing, presentation, messaging ...), and the like.

[0026] Touch input system 200 as illustrated comprises a touch screen input device
20 202 that detects when a touch input has been received (e.g. a finger touching or nearly touching the touch screen). Any type of touch screen may be utilized that detects a user's touch input. For example, the touch screen may include one or more layers of capacitive material that detects the touch input. Other sensors may be used in addition to or in place of the capacitive material. For example, Infrared (IR) sensors may be used. According to
25 an embodiment, the touch screen is configured to detect objects that in contact with or above a touchable surface. Although the term "above" is used in this description, it should be understood that the orientation of the touch panel system is irrelevant. The term "above" is intended to be applicable to all such orientations. The touch screen may be configured to determine locations of where touch input is received (e.g. a starting point,
30 intermediate points and an ending point). Actual contact between the touchable surface and the object may be detected by any suitable means, including, for example, by a vibration sensor or microphone coupled to the touch panel. A non-exhaustive list of examples for sensors to detect contact includes pressure-based mechanisms, micro-

machined accelerometers, piezoelectric devices, capacitive sensors, resistive sensors, inductive sensors, laser vibrometers, and LED vibrometers.

[0027] Gadget manager 26 is configured to process the received touch input from touch screen 202. For example, a user may select one or more cells (e.g. selection 252) within spreadsheet 250 to bind with a gadget (e.g. gadget 254). In the current example, a user selected cells 252 to identify the range of cells to bind with gadget 254.

[0028] Gadgets comprise web content that may be configured in the same manner as other content on the web. Generally, gadgets are configured to display information at a glance and provide easy access to frequently used tools. For example, a gadget 254 may be created to display an interactive bar chart that may obtain data from the spreadsheet (e.g. from cells 252) as well as to place data within the spreadsheet. For example, a user may select a bar of the bar chart and drag the bar to a new value. In the example illustrated, a user has selected the bar on the far right that corresponds to the cell value at location A4. After the user has adjusted the bar, the value at A4 is updated by the gadget to the spreadsheet (e.g. the current value as determined by the gadget is 5). Not only can the gadget show content that is obtained from the spreadsheet, the gadget can cause the display of the spreadsheet to change. In the current example, when gadget 254 is shown, the cells associated with the gadget are distinguished from the other cells (e.g. a border around the associated cells, the cells highlighted, the formatting of the cells changing, and the like).

[0029] Gadgets may be configured to perform almost any desired operation. For example, the gadget may be configured to display stock quotes, update headline, display slide shows, interact with the spreadsheet data (e.g. setting values/obtaining values).

[0030] Gadgets may be built using different programming technologies, such as: XML, HTML, JavaScript, ADOBE FLASH, and the like. Gadgets may run on a network site (e.g. gadget web site 260), a local machine, and/or may be contained within the spreadsheet. The site may be hosted by a third party and/or hosted by service 210. The gadgets may be used online and/or offline with the spreadsheet and may be accessed through a link.

[0031] Gadgets integrate with spreadsheets and the spreadsheet calculation engine 222. Some gadgets may be integrated with a sheet's contents whereas other gadgets are separate from the sheet. As discussed, a gadget may be bound to a range of cells (one or more) within a sheet.

[0032] When one or more of the cells of the bound range is interacted with (e.g. selected, hovered over, edited, deleted, added), the gadget is notified. The gadget may use this information for updating a display rendered by the gadget and/or performing some other action.

5 [0033] The gadget uses an Application Programming Interface (API) 230 to communicate with the spreadsheet. The API 230 comprises functionality for binding the gadget to the spreadsheet, obtaining/putting content from/into the spreadsheet, changing a display of the spreadsheet, requesting a display of a spreadsheet user interface, and the like.

10 [0034] Gadgets are treated as an object of the spreadsheet along with the other objects of a spreadsheet (tables, charts, range, ...). According to an embodiment, the gadget may also be a Named Item object within the spreadsheet such that the gadget may be viewed/displayed differently from other objects (e.g. selected for publication in a report, dashboard, ...). (See FIGURE 4 and related discussion).

15 [0035] The gadget is configured to communicate asynchronously with the spreadsheet. As such, a call from the gadget to the spreadsheet may be out of date (e.g. the data referred to by the call is not the current data). When a call is received that is out of date, the gadget is notified and the call is not processed such that the gadget is not acting on data which it may not be aware of.

20 [0036] Data store 240 may store spreadsheets that may be associated with one or more gadgets. The user may open a file/workbook from a user interface associated with the spreadsheet application and/or some other application.

[0037] A gadget may display data differently depending on a user's access permissions. For example, the spreadsheet may restrict access to data within the spreadsheet and/or the gadget may specify restrictions. For example, views that the user does not have access to are not shown within the gadget, but views the user does have access are displayed within the gadget. The gadget may render data obtained from the spreadsheet while the spreadsheet application protects the backend logic and calculation details in the spreadsheet itself to any unauthorized users viewing the gadget.

25 [0038] FIGURE 3 shows an exemplary displays of gadgets and spreadsheets.

[0039] As illustrated, display 310 shows a spreadsheet that is associated with two gadgets, gadget 254 and gadget 318. Gadget 254 is the gadget as described in FIGURE 2 and is bound to selected range 252.

[0040] Gadget 318 is a gadget that allows a user to display values and enter values into the spreadsheet. For example, gadget 318 shows entry boxes for entering values that may be placed within the spreadsheet. In the current example, the user has entered the values 1, 2, 3 and 4 for entry into the bound location 316 after the user selects the save
5 option. In the current example, the gadget uses the API to adjust a view of the spreadsheet in response to a current selection of an entry box (e.g. the user is entering the value 2 in the entry box causing the corresponding cell location to be highlighted). The user may also change a value within the spreadsheet causing the gadget to redisplay and update the value. In other words, changes made to the gadget may be shown in the spreadsheet and
10 changes made to the spreadsheet may be shown in the gadget.

[0041] Display 320 shows a spreadsheet 322 with a gadget 324 displayed in a task pane near the display of the spreadsheet. In the current example, gadget 324 is configured to search content obtained from outside of the spreadsheet and provide, display and store results from the search within the spreadsheet. For example, the gadget 324 may use on or
15 more search services to search for source from one or more locations. A user may decide to store one or more of the returned results in spreadsheet 322. In response to a user selecting to store results by selecting the store results button, the spreadsheet may display a user interface selection method and/or place the user within a touch mode that allows a user to select where in the spreadsheet that they would like the results stored. For
20 example, in response to an API call to allow touch input to select a location, the user selects column 1 as the location to store the content. In response to the selection, the spreadsheet returns the location to gadget 324 such that it may place the results within the determined location.

[0042] FIGURE 4 is a functional block diagram illustrating the generation of a
25 named object view report across multiple object sources including gadgets. The named object view of the report may be generated from different electronic object sources. As shown in the diagram, the electronic data object sources are obtained from a workbook, such as workbook 400, a gadget 401 and from data source 403. The electronic object sources, however, may be any file/data source that may be associated with an object. For
30 example, one or more objects that are created from a web page, a common data source may be displayed within the named object view. The object may or may not be stored within an object source when it is obtained and displayed within a report. For example, the object may be a newly created object that is not yet stored within a electronic file, but is stored within a memory of a computing device.

[0043] Workbooks may include multiple objects on different sheets. The objects may be any item in the workbooks that can be individually selected or manipulated. For example, the objects in workbook 400 may include a table 402, a gadget 404, a single data cell 406 and a bar graph 408. Other objects may also be included within workbook 400.

5 As illustrated, gadget 401 includes a pie chart 410, a named range 412, an image 414, and a pivot table 416. Data source 403 includes a text box 418, and a line graph 420. Some of the objects may use data from a common data source. For example, bar graph 408 in workbook 400, pie chart 410 in gadget 401, and line graph 420 from data source 403 may be created from the same data source. In one embodiment, an object may be defined to
10 encompass multiple smaller objects. For example, an object may correspond to multiple pages of a workbook. In another embodiment, an object may include dynamic information such that the size of the object changes over time. For example, the named range 412 may correspond to a range of cells that are contained within a workbook (e.g. workbook 400 and/or some other workbook (not shown)). The range of cells may increase or decrease in
15 size in response to spreadsheet processing. The named range may correspond to a range from other data sources. For example, a web page may contain a range of values that change over time.

[0044] The report may be generated automatically in response to an event (e.g. a user opening a workbook, selecting a Named Object View (NOV) option within a user
20 interface, performing a search, and the like). For example, an author user or administrative user may access one or more of the workbook, web page, and/or a data source from a server to generate a report 430. Alternatively, a workbook, web page, and/or a data source may be accessed at a client. The objects to publish in the report may be automatically/manually selected. For example, the user may select all/portion of the
25 objects to publish in the report. For example, the user may manually select gadget 404 in workbook 300, the named range 412 in gadget 401 and the text box 418 and the line graph 420 in data source 403 to appear in the report 430. Bar graph 408 and pie chart 410 may be automatically selected to be published as they are created using common data (e.g. the same data source). The user may open a file/workbook from a publisher interface. In one
30 embodiment, names associated with the objects are provided in the publisher interface. The user may select the objects to publish from the list. In one embodiment, objects may be selected for publication from multiple workbooks.

[0045] The user may select the objects for inclusion in the report 430 by name such that each published object may be easily identified. In one embodiment, the published

object names may be automatically generated. For example, the table 402 may be automatically identified as “table”, “table1”, or some other unique name. The user may rename the automatically generated name. In another embodiment, the user defines a name for the published object. For example, the gadget 404 may be defined as “Payroll Gadget”. Likewise, the named range 412 may be defined as “my data”.

[0046] The selected objects are published in the report 430. For example, the gadget (“Payroll Gadget”) 406, the named range (“my data”) 412, the text box 418, and the line graph 420 are published in the report 430. The unselected portions of the workbook 400, gadget 401 and data source 403 are not published in the report 430.

[0047] According to an embodiment, the object sources that utilize a same data source are searched to determine the objects to obtain from the object sources. For example, many object sources may create objects (e.g. tables, charts, graphs) using the same data source. In the current example, the bar graph 408 and pie chart 410 are created using the same data. A user may desire to see all of the objects created using the data source. The data source may be automatically/manually selected. For example, the data source(s) may be determined from the selected object(s) and/or the data source may be determined from input from a user (e.g. through a user interface element). The user may then select a NOV mode the user may view the selected objects by cycling through different views that are obtained from the data source.

[0048] Another example includes where users may view a web page and alter the data, creating unique and interesting views specific to their interpretation of the data. The users may want to share those views, especially in today’s socially connected internet environment. In this example, assume that a user has friends who each have made multiple views using the data from the web page. Each of these friends may have one or more object sources (e.g. files) that are used to store the different views. One or more of the views may also be stored in a memory. When a report is generated to show a view created from the web page, the different object sources that include the views created by the different users may also be searched for the similar objects. The objects from the different object sources are obtained and are included in the report. When cycling through the various views of the objects, a user may see all of the different views without having to individually access each object source.

[0049] According to an embodiment, the permissions of the object sources are followed. Views that the user does not have access are not shown, but views the user does have access are displayed. For example, when a user is viewing a display from the Payroll

Gadget 406 they will not have access to payroll views that they are not authorized to view (e.g. views of other users payroll information).

5 **[0050]** A user at the client is presented with a named object view of the report 430. According to an embodiment, the report does not distinguish where an object is obtained from. According to another embodiment, a user may query the object to determine where it is located. For example, a user may right click on a displayed object to find out its location and/or open the object and the associated object source from where it was obtained.

10 **[0051]** A report renderer converts the published objects into a Markup Language (ML) when requested. For example, one of the object sources (e.g. workbook 400) is accessed from a server. The ML version of the published objects is forwarded to a client for display via a web browser. In one embodiment, a published object is rendered on a user interface 450. For example, the user interface 450 may be a web part. A web part is a modular unit of information that has a single purpose. Web parts form the basic building
15 blocks of many web pages. More than one web part may be simultaneously displayed on a dashboard at the client such that different published objects may be viewed side-by-side. For example, the user may select the text box 418 and the line graph 420 for display on different web parts because the text box 418 elaborates on data presented in the line graph 420.

20 **[0052]** A tool bar that includes the names of the published objects may also be rendered on the user interface 450. The tool bar may be used for accessing specific published objects. For example, the tool bar is a drop down menu 460 that includes a list of the names of the published objects. Specifically, the drop down menu 360 includes selectable entries for “Payroll Gadget”, “my data”, text box, and line graph. The other
25 objects may also be included within the user interface. The named object view of the report is rendered on the user interface 550 with the first published object (i.e., “Payroll Gadget”). The other published objects may be rendered at the user interface 450 in response to user input. For example, the user may select an entry from the drop down menu 460 such that the user interface 450 refreshes and the published object
30 corresponding to the selected entry is rendered on the user interface 450. Thus, each published object may be individually displayed to the user at the client. The named object view of the report focuses the user’s attention on the published objects. The non-published objects of the object sources are not displayed to the user.

[0053] Referring now to FIGURES 5 and 6, an illustrative process for creating and using gadgets with spreadsheets will be described. When reading the discussion of the routines presented herein, it should be appreciated that the logical operations of various embodiments are implemented (1) as a sequence of computer implemented acts or
5 program modules running on a computing system and/or (2) as interconnected machine logic circuits or circuit modules within the computing system. The implementation is a matter of choice dependent on the performance requirements of the computing system implementing the invention. Accordingly, the logical operations illustrated and making up the embodiments described herein are referred to variously as operations, structural
10 devices, acts or modules. These operations, structural devices, acts and modules may be implemented in software, in firmware, in special purpose digital logic, and any combination thereof.

[0054] FIGURE 5 shows a process for binding a gadget to a spreadsheet.

[0055] After a start operation, process 500 flows to operation 510, where an API is
15 provided for interaction between the gadget and the spreadsheet. The API may provide a variety of functions. For example, the API may provide functions for: binding the gadget to the spreadsheet, obtaining/putting content from/into the spreadsheet, changing a display of the spreadsheet, requesting a display of a spreadsheet user interface, storing data associated with the gadget within the spreadsheet file but not in the cells of the
20 spreadsheet, and the like.

[0056] Moving to operation 520, a range of cells is selected to bind to the gadget. The selection may be a user selection of one or more cells within a spreadsheet and/or a selection made through the API. For example, a user may select one or more cells of the spreadsheet using touch input, a user interface associated with the spreadsheet, and/or
25 provide a name of an object, a range of cells (e.g. A1:A9, B1:F3, table 1, a name of a range, a name of an item, and the like) within a call associated with the API. According to an embodiment, the selection of the range uses touch input. For example, a user may perform a gesture to select one or more cells of a spreadsheet. Additionally, according to some embodiments, the user may only have to place their finger (or some other object)
30 near the surface of the touch surface for touch input to be received and detected. The cells may include cells within one or more rows and one or more columns of the spreadsheet. The selection may be used to identify the range to bind with the gadget.

[0057] Transitioning to operation 530, the binding between the gadget and the spreadsheet is established. The binding may specify a unique name for the gadget, a link to the gadget, and the range of cells of the spreadsheet to be associated with the gadget.

5 [0058] Moving to operation 540, the binding information is stored. According to an embodiment, the binding information is stored within the spreadsheet such that when the spreadsheet file is copied, the gadget and binding information move with the copy. According to another embodiment, the binding may be stored at a different location such that the gadget is not part of the spreadsheet file.

10 [0059] The process then flows to an end operation and returns to processing other actions.

[0060] FIGURE 6 shows a process for a gadget interacting with a spreadsheet.

15 [0061] After a start operation, process 600 flows to operation 610, where a determination is made as to when the range that is bound to the gadget is interacted with (e.g. one or more cells within the spreadsheet is selected, hovered over, edited, deleted, added).

[0062] Moving to operation 620, the gadget is notified of the interaction. According to an embodiment, the notification provides the gadget that the bound range has changed. The notification may include other information. For example, the notification may include the specified cell(s) that have been interacted with and/or the changed data.

20 [0063] Transitioning to operation 630, a call is received from the gadget. According to an embodiment, the call is received through an API call. As discussed herein, the call may relate to: binding the gadget to the spreadsheet, obtaining/putting content from/into the spreadsheet, changing a display of the spreadsheet, requesting a display of a spreadsheet user interface, and the like

25 [0064] Moving to decision operation 640, a determination is made as to whether the request is out of date. For example, a change may have been made to the spreadsheet between a time when the gadget made the call and a time when the spreadsheet received the call. In another example, a spreadsheet application (e.g. at a network location) may have multiple users simultaneously editing a workbook. When the call is out of date, the process moves to operation 650. When the call is not out of date, the process flows to
30 operation 660.

[0065] At operation 650, a notification is sent to the gadget that informs the gadget that the request is out of date and may not be associated with the most current data in the spreadsheet. In response to the notification, the gadget may obtain the latest data from the

spreadsheet and resubmit the request and/or perform some other operation (e.g. cancel the operation).

5 [0066] At operation 660, the operation is performed. The operation may relate to a variety of different actions. For example, the operation may be obtaining/putting content from/into the spreadsheet, changing a display of the spreadsheet, requesting a display of a spreadsheet user interface, identifying the gadget for publishing, and the like.

[0067] Flowing to operation 670, the display(s) may be updated. The spreadsheet display may be updated and/or the gadget display may be updated.

10 [0068] The process then flows to an end operation and returns to processing other actions.

[0069] The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Since many embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.

CLAIMS

WHAT IS CLAIMED IS:

1. A method for integrating a gadget with a spreadsheet, comprising:
providing an Application Programming Interface (API) for the gadget to
5 communicate with the spreadsheet that comprises functions for: obtaining content from
one or more cells of the spreadsheet; and placing content into one or more cells of the
spreadsheet;
determining a range of cells of the spreadsheet to bind to the gadget,
wherein the range of cells comprises one or more cells of the spreadsheet;
10 establishing a binding between the range of cells of the spreadsheet and the
gadget;
storing the binding within the spreadsheet;
determining an interaction with the range of cells;
automatically providing a notification to the gadget in response to the
15 interaction;
receiving a call from the gadget using the API; and
performing an operation involving the spreadsheet that relates to the
received call.
2. The method of Claim 1, further comprising exposing the gadget as a named
20 item within the spreadsheet such that the gadget can be displayed differently from other
objects within the spreadsheet.
3. The method of Claim 1, wherein determining the interaction to the range of
cells comprises determining at least one of: when a cell within the range of cells is deleted;
when a cell within the range of cells is added; when a cell within the range of cells is
25 edited; when a cell within the range of cells is hovered over; and when a cell within the
range of cells is selected.
4. The method of Claim 1, wherein the API further comprises a function for
changing the display of one or more cells of the spreadsheet in response to an action
occurring on the gadget and creating a binding and displaying a selection user interface for
30 selecting at least one of: content from the sheet; a location within the sheet; and a type of
content.
5. The method of Claim 1, further comprising in response to an interaction
with the range of cells in the spreadsheet automatically updating a display of the gadget.

6. The method of Claim 1, wherein the binding is automatically updated in response to a change made to the spreadsheet that changes a size of the range of cells.

7. The method of Claim 1, wherein receiving the call from the gadget using the API comprises determining when the call is out of date; and when the call is out of
5 date preventing the call from being executed; and informing the gadget of the out of date request.

8. A computer-readable medium having computer-executable instructions for integrating a gadget with a spreadsheet, comprising:

10 providing an Application Programming Interface (API) for the gadget to communicate with the spreadsheet that comprises functions for: obtaining content from one or more cells of the spreadsheet; placing content into one or more cells of the spreadsheet; creating a binding; and displaying a selection user interface for selecting at least one of: content from the sheet; a location within the sheet; and a type of content;

15 determining a range of cells of the spreadsheet to bind to the gadget, wherein the range of cells comprises one or more cells of the spreadsheet;

establishing a binding between the range of cells of the spreadsheet and the gadget;

storing the binding within the spreadsheet;

20 determining an interaction with the range of cells that is at least one of: when a cell within the range of cells is deleted; when a cell within the range of cells is added; when a cell within the range of cells is edited; when a cell within the range of cells is hovered over; when a cell within the range of cells is changed indirectly; when a cell within the range is bound to another cell that changes; when the cell within the range of cells that includes a calculation that is updated; and when a cell within the range of cells is
25 selected;

automatically providing a notification to the gadget in response to the interaction;

receiving a call from the gadget using the API; and

30 performing an operation involving the spreadsheet that relates to the received call.

9. A system for integrating a gadget with a spreadsheet, comprising:
a display;
a processor and a computer-readable medium;

an operating environment stored on the computer-readable medium and executing on the processor; and

a process configured to perform actions, comprising:

5 providing an Application Programming Interface (API) for the gadget to communicate with the spreadsheet that comprises functions for: obtaining content from one or more cells of the spreadsheet; placing content into one or more cells of the spreadsheet; creating a binding; and displaying a selection user interface for selecting at least one of: content from the sheet; a location within the sheet; and a type of content;

10 determining a range of cells of the spreadsheet to bind to the gadget, wherein the range of cells comprises one or more cells of the spreadsheet;

establishing a binding between the range of cells of the spreadsheet and the gadget;

storing the binding within the spreadsheet;

15 exposing the gadget as a named item within the spreadsheet such that the gadget is publishable in a report;

determining an interaction with the range of cells that is at least one of: when a cell within the range of cells is deleted; when a cell within the range of cells is added; when a cell within the range of cells is edited; when a cell within the range of cells is hovered over; and when a cell within the range of cells is selected;

20 automatically providing a notification to the gadget in response to the interaction;

receiving a call from the gadget using the API; and

25 performing an operation involving the spreadsheet that relates to the received call.

10. The system of Claim 9, wherein receiving the call from the gadget using the API comprises determining when the call is out of date; and when the call is out of date preventing the call from being executed; and informing the gadget of the out of date request.

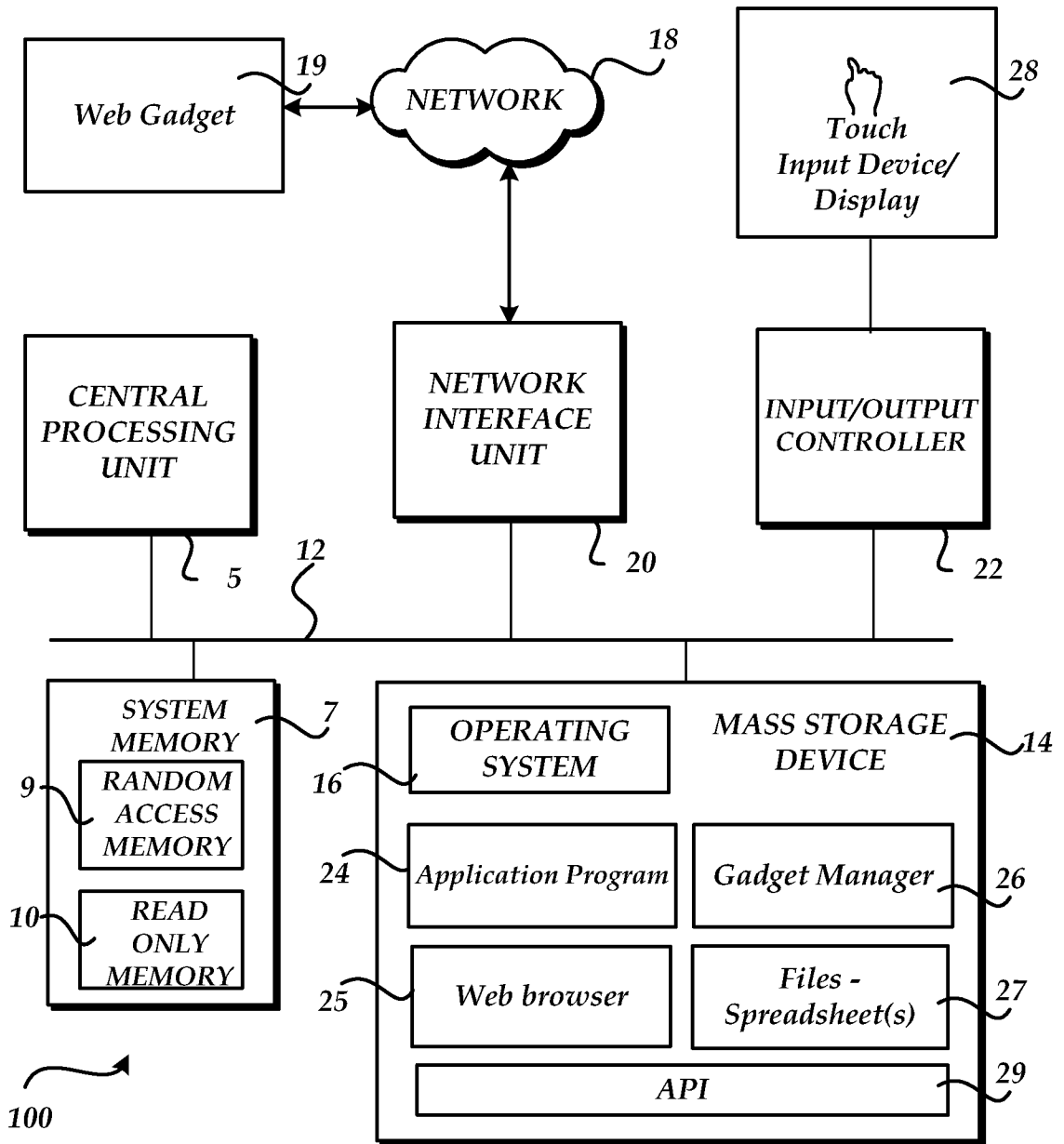


Fig. 1

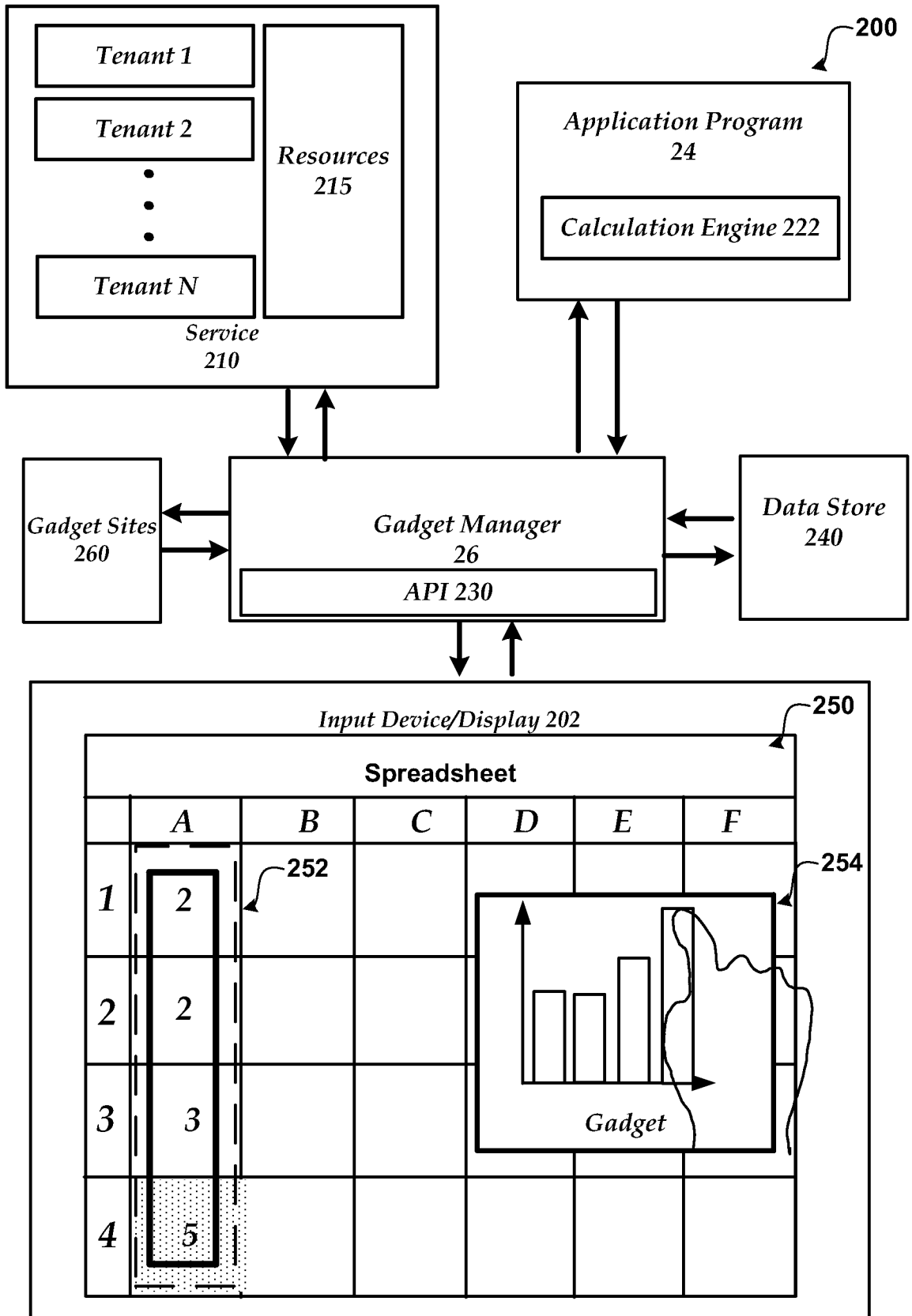


Fig. 2

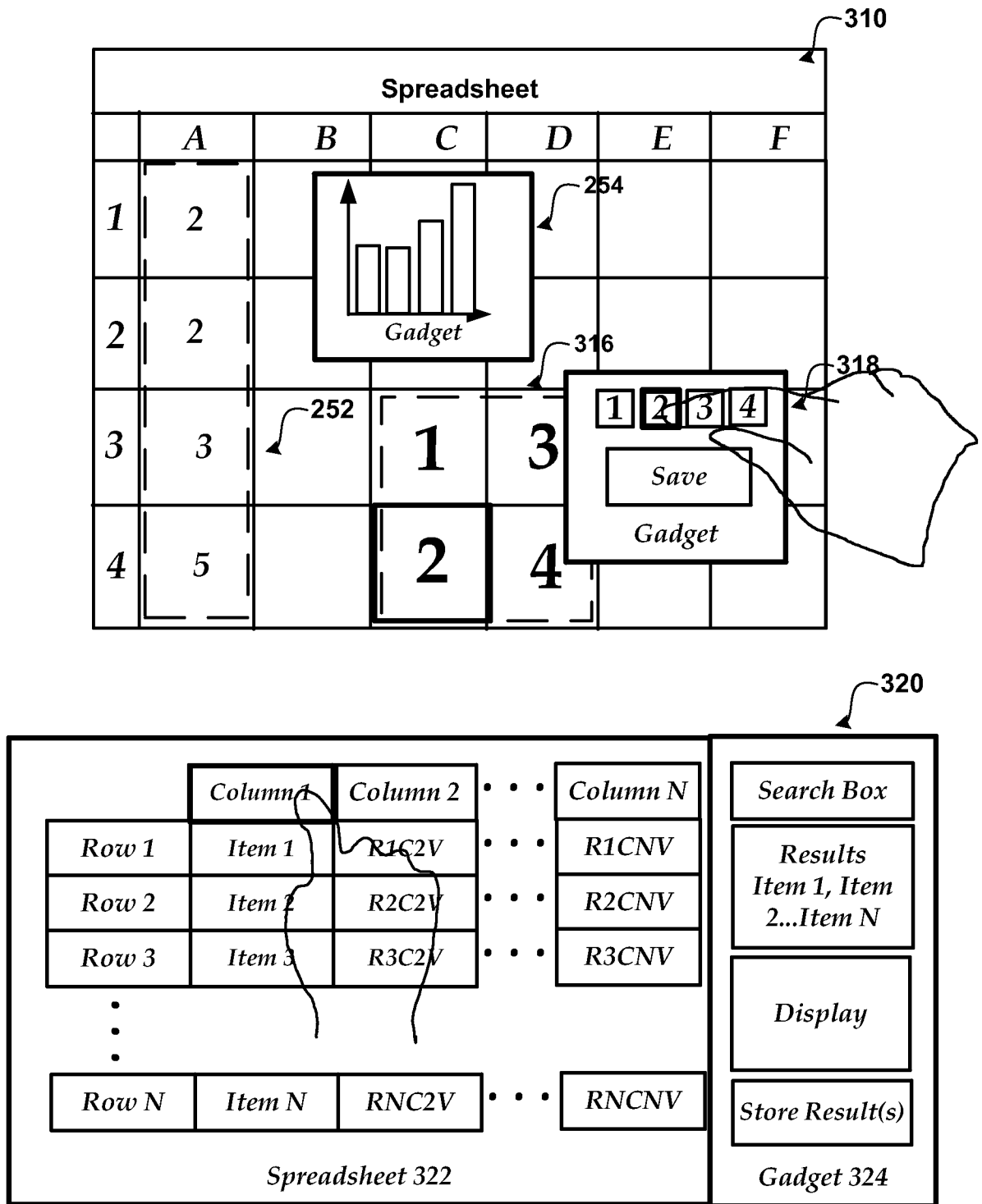


Fig. 3

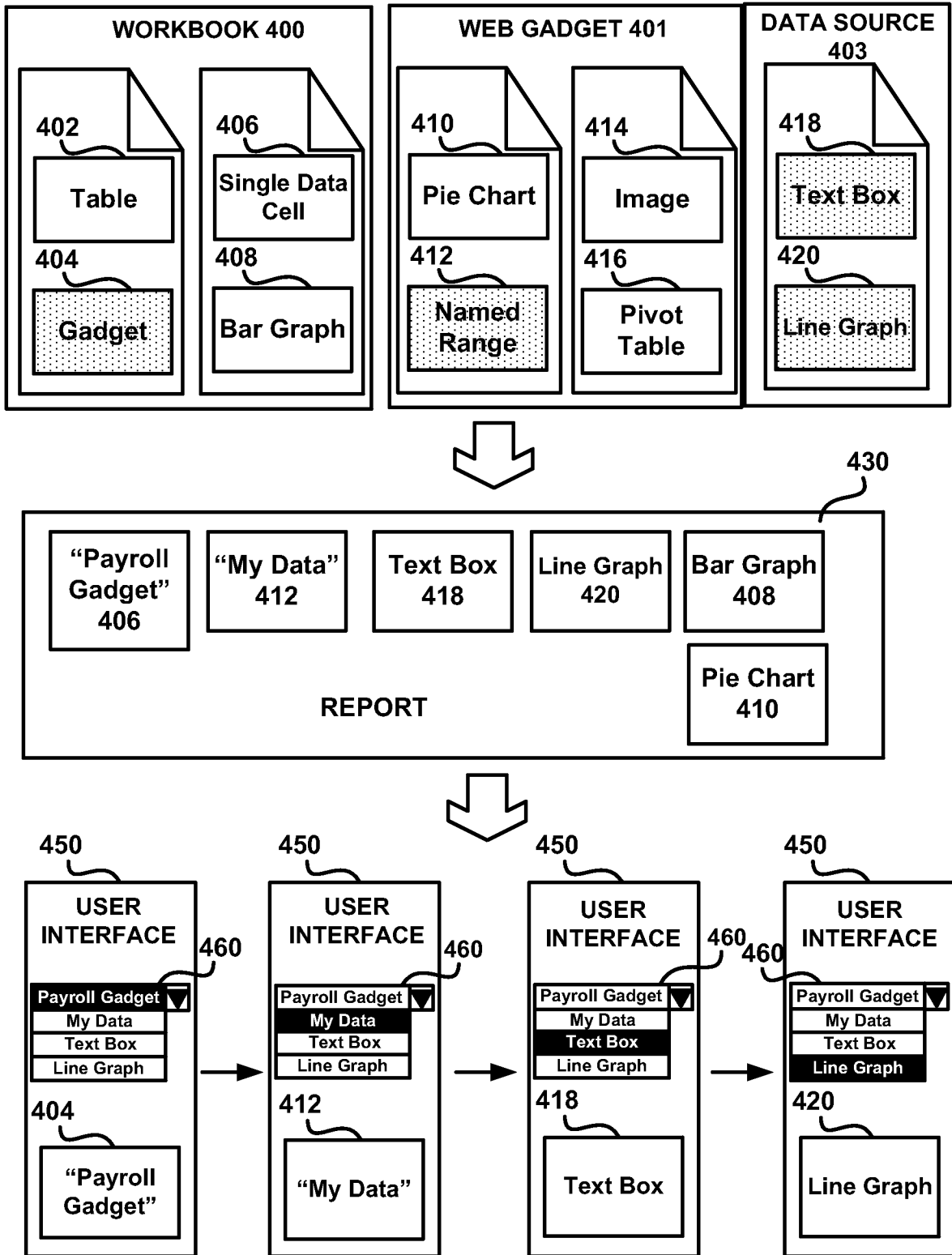


Fig. 4

5/6

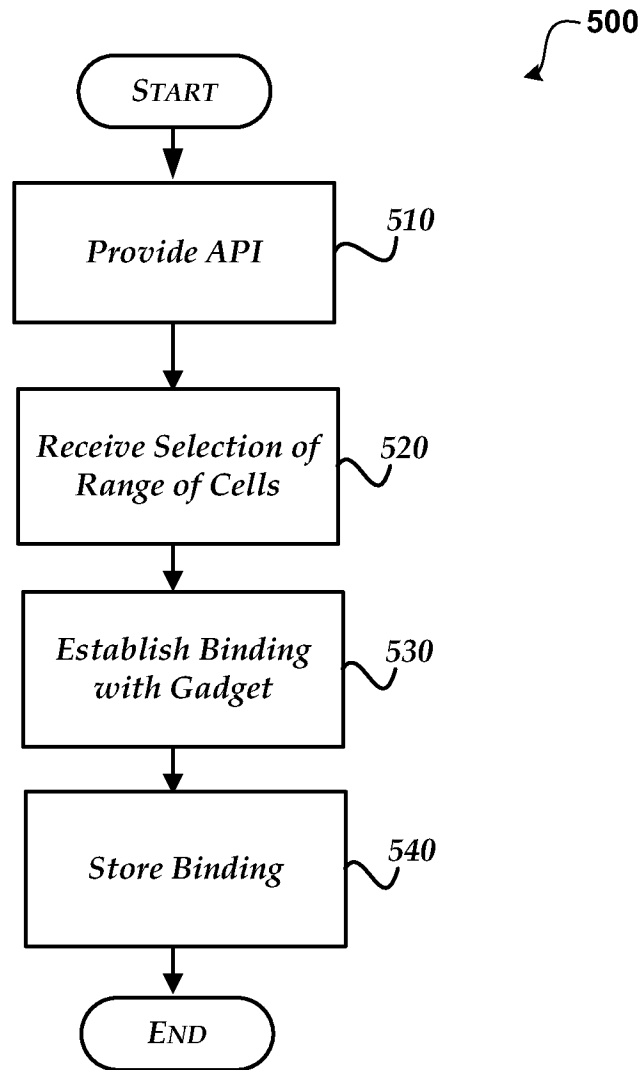


Fig. 5

6/6

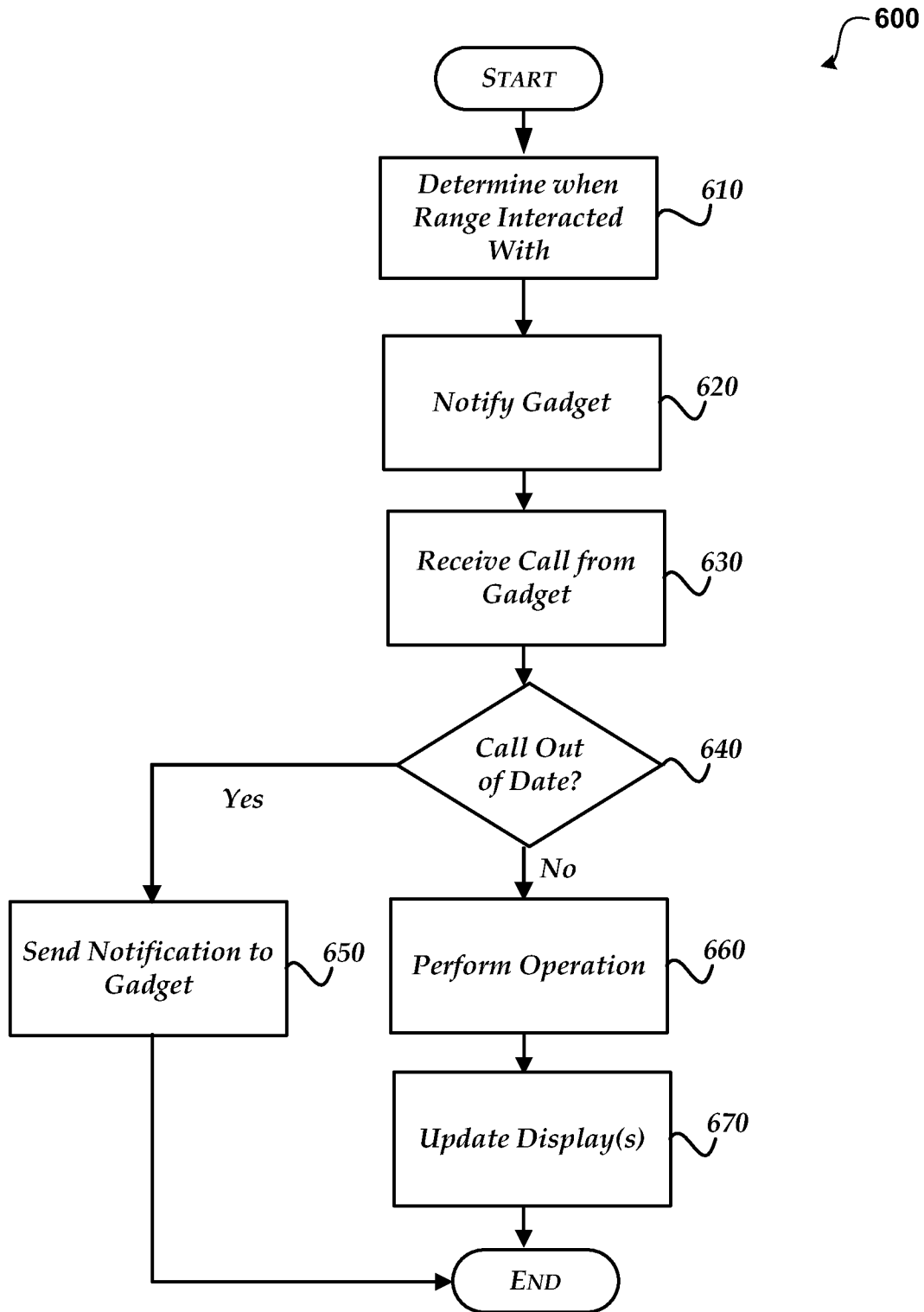


Fig. 6

A. CLASSIFICATION OF SUBJECT MATTER**G06F 3/048(2006.01)i, G06F 3/14(2006.01)i, G06F 9/44(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 3/048; G06F 15/16; G06F 9/44; G06F 17/00; G06F 17/60

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) & Keywords: gadget, spreadsheet, data, edit, and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2010-0077344 A1 (GAFFNEY, EAMON et al.) 25 March 2010 See abstract; paragraphs [0011],[0046]; claim 1; and figure 3.	1-10
A	US 2008-0307385 A1 (DREILING, ALEXANDER et al.) 11 December 2008 See abstract; paragraphs [0029]-[0030]; and figure 1.	1-10
A	EP 1367514 A1 (FRAMTIDSFORUM I & M AB) 03 December 2003 See abstract and paragraph [0095].	1-10
A	US 2010-0318890 A1 (BILLHARZ, ALAN et al.) 16 December 2010 See paragraph [0056] and figure 8.	1-10
A	US 2008-0294751 A1 (DREILING, ALEXANDER) 27 November 2008 See paragraph [0033] and figure 2.	1-10

 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

20 MARCH 2013 (20.03.2013)

Date of mailing of the international search report

22 MARCH 2013 (22.03.2013)

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
189 Cheongsu-ro, Seo-gu, Daejeon Metropolitan
City, 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

LEE, Dong Yun

Telephone No. 82-42-481-8734



INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2012/063133

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2010-0077344 A1	25.03.2010	None	
US 2008-0307385 A1	11.12.2008	CN 101373431 A EP 2003573 A1 JP 2008-305411 A JP 5086183 B2	25.02.2009 17.12.2008 18.12.2008 14.09.2012
EP 1367514 A1	03.12.2003	None	
US 2010-0318890 A1	16.12.2010	None	
US 2008-0294751 A1	27.11.2008	AT 525702 T CN 101312462 A EP 1995668 A1 EP 1995668 B1 JP 2008-293502 A JP 4842297 B2 US 7702754 B2	15.10.2011 26.11.2008 26.11.2008 21.09.2011 04.12.2008 14.10.2011 20.04.2010