NAMED OBJECT VIEW OF ELECTRONIC DATA REPORT

BACKGROUND

Electronic document is sharing allows multiple users to access the same document. Some documents, such as spreadsheets, may include multiple sheets with each sheet including a large amount of data. A report may be generated from the document to present the data in an organized format. However, the report may confuse viewers if the report contains too much data. A viewer may need to search through the report to locate desired information, which can be laborious and time consuming.

SUMMARY

A named object view of a report is generated from an electronic data file. The file includes multiple objects. Each object is associated with a name. The objects to be published are identified in the file. A named object view of the report associated with the file is generated by publishing the identified objects. A viewer at a client is presented with the named object view of the report such that the viewer's attention is focused on the published objects.

In one embodiment, a report renderer converts the published objects into a format suitable for display on a web page such that each object that is identified for publishing may be individually rendered by a web application. The web-formatted version of the published objects is forwarded to a client via a network for display by a web browser. In another embodiment, the report renderer is implemented in the same application associated with the electronic data file such that the named object view of the report is generated at the client.

The invention may be implemented as a computer process, a computing system, or as an article of manufacture such as a computer program product or computer readable media. The computer program product may be a computer storage media readable by a computer system and encoding a computer program of instructions for executing a computer process. The computer program product may also be a propagated signal on a carrier readable by a computing system and encoding a computer program of instructions for executing a computer process.

These and various other features, as well as advantages, which characterize the present invention, will be apparent from a reading of the following detailed description and a review of the associated drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 is a computer system architecture diagram illustrating a computer system utilized in and provided by the various embodiments of the invention;

FIGURE 2 is a network architecture diagram illustrating an illustrative operating environment for embodiments of the invention;

FIGURE 3 is a functional block diagram illustrating the generation of a named object view of an electronic data report; and

FIGURE 4 is a process diagram showing aspects of an illustrative process for generating a named object view of an electronic data report.

DETAILED DESCRIPTION

Referring now to the drawings, in which like numerals represent like elements, various aspects of the present invention 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 of the invention may be implemented. While the invention will be described in the general context of program modules that execute on server and personal computer systems, those skilled in the art will recognize that the invention may also be implemented in combination with other types of computer systems and program modules.

Generally, program modules include routines, programs, components, data structures, and other types of structures that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments 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.

Referring now to FIGURE 1, an illustrative computer architecture for a computer 100 utilized in the various embodiments of the invention will be described. The computer architecture shown in FIGURE 1 illustrates a conventional desktop or laptop computer, including a central processing unit 110 ("CPU"), a system memory 120, including a random access memory ("RAM") 122 and a read-only memory ("ROM") 124, and a system bus 130 that couples the memory to the CPU 110. 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 124. The computer 100 further includes a mass storage device 140 for storing an operating system 142, application programs, and other program modules, which will be described in greater detail below.

The mass storage device 140 is connected to the CPU 110 through a mass storage controller (not shown) connected to the bus 130. The mass storage device 140 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, it should be appreciated by those skilled in the art that computer-readable media can be any available media that can be accessed by the computer 100.

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, EPROM, 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.

According to various embodiments of the invention, the computer 100 may operate in a networked environment using logical connections to remote computers through a network 150, such as the Internet. The computer 100 may connect to the network 150 through a network interface unit 160 connected to the bus 130. It should be appreciated that the network interface unit 160 may also be utilized to connect to other types of networks and remote computer systems. The computer 100 may also include an input/output controller 170 for receiving and processing input from a number of other devices, including a keyboard, mouse, or electronic stylus (not shown in FIGURE 1). Similarly, the input/output controller 170 may provide output to a display screen, a printer, or other type of output device.

As mentioned briefly above, a number of program modules and data files may be stored in the mass storage device 140 and RAM 122 of the computer 100, including the operating system 142 that is suitable for controlling the operation of a networked personal computer, such as the WINDOWS XP operating system from MICROSOFT CORPORATION of Redmond, Washington. The mass storage device 140 and the RAM 122 may also store one or more program modules. In particular, the mass storage device 140 and the RAM 122 may store a web browser application program 144. As known to those skilled in the art, the web browser application program 144 is operative to request, receive, render, and provide interactivity with electronic documents, such as a web page 146 that has been formatted using HTML. According to one embodiment of the invention, the web browser application program 144 comprises the INTERNET EXPLORER web browser application program from MICROSOFT CORPORATION. It should be

appreciated, however, that other web browser application programs from other manufacturers may be utilized to embody the various aspects of the present invention, such as the FIREFOX web browser application from the MOZILLA FOUNDATION.

As will be described in greater detail below, the web page 146 may comprise a named object view of a report associated with electronic data from an application program 148. The application program 148 may comprise the EXCEL spreadsheet application program from MICROSOFT CORPORATION or another spreadsheet application program from another manufacturer. It should be appreciated that although the embodiments of the invention described herein are presented in the context of a spreadsheet application program, the invention may be utilized in other types of application programs that support electronic data reporting. For instance, the embodiments of the invention described herein may be utilized within a word processing application program, a presentation application program, a drawing or computer-aided design application program, or a database application program in order to allow the rendering of and interaction with a document without requiring a dedicated application program. Additional details regarding the process for generating a named object view of a report are be provided below with respect to FIGURES 2-4.

Referring now to FIGURE 2, a network architecture diagram will be described that illustrates an operating environment for the various embodiments of the invention. As shown in FIGURE 2, the computer 100 is connected to the network 150. Also connected to the network 150 is an application server 210. The application server 210 comprises a server computer which may contain some or all the conventional computing components described above with respect to FIGURE 1. Additionally, the application server 210 is operative to execute a web server application for receiving and responding to requests for documents stored at or accessible to the application server 210. Moreover, the application server 210 is operative to receive and respond to requests for pages generated by a web application 220. It should be appreciated that the web application 220 may comprise code executable at the application server 210, executable code for communicating with other computers, and may include templates, graphics, audio files, and other content known to those skilled in the art.

The web application 220 is operative to provide an interface to a user of the computer 100 to view a named object view of a report associated with electronic data accessible via the network 150. In particular, the web application 220 utilizes a server application programming interface ("API") 222. According to this embodiment of the invention, the server API 222 is operative to enable communication with a spreadsheet calculation server 230. The spreadsheet calculation server 230 is operative to execute a spreadsheet calculation server program 232. The spreadsheet calculation server program 232 comprises an executable program for retrieving and

calculating a named object view of a report associated with electronic data. The electronic data may be any file such as a workbook 242 stored in a data store 240. It should be appreciated that in the embodiments of the invention described herein, the spreadsheet calculation server 230 may include many of the conventional hardware and software components discussed above with respect to FIGURE 1.

The computer 100 may transmit a request to the application server 210 for a named object view of a report of the workbook 242 within the context of the web browser application 144. In response to such a request, the web application 220 communicates with the spreadsheet calculation server 230 through the server API 222. In particular, the web application 220 requests from the spreadsheet calculation server 230 the requested named object view of the report.

The spreadsheet calculation server program 232 loads the appropriate workbook 242 from the data store 240 and processes the workbook 242. The workbook 242 is processed as discussed in detail below. Briefly, the spreadsheet calculation server 230 identifies the objects to be published in the workbook 242. The spreadsheet calculation server 230generates the named object view of the report associated with the workbook 242 by publishing the identified objects. The report renderer 224 converts the published objects into HTML such that each object that is identified for publishing may be individually rendered by the web application 220. The application server 210 forwards the HTML version of the published objects to the client 100 for display by a web browser via the network 150. A user at the client 100 may then view the published objects of the workbook 242.

FIGURE 3 is a functional block diagram illustrating the generation of a named object view of an electronic data report. The named object view of the report is generated from an electronic data file. As shown in the diagram, the electronic data file is a workbook 300 associated with a spreadsheet application. However, the electronic data file may be any file that may be associated with a report.

The workbook 300 includes multiple objects on different pages. The objects may be any item in the workbook 300 that can be individually selected or manipulated. For example, the objects may include a table 302, a paragraph 304, a single data cell 306, a bar graph 308, a pie chart 310, a named range 312, an image 314, a pivot table 316, a text box 318, and a line graph 320. In one embodiment, an object in the workbook 300 may be defined to encompass multiple smaller objects. For example, an object may correspond to multiple pages of the workbook 300. In another embodiment, an object may include dynamic information such that the size of the object changes over time. For example, the named range 312 may correspond to a range of cells in the workbook 300. The range of cells may increase or decrease in size in response to spreadsheet processing.

A user may access the workbook 300 from a server to generate a report 330. Alternatively, the workbook 300 may be accessed at a client. The user selects objects to publish in the report. For example, the user may select the single data cell 306, the named range 312, the text box 318, and the line graph 320 to appear in the report 330. The user may open the workbook 300 from a publisher interface. In one 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.

Each object to be published is associated with a unique name. The user may select the objects for inclusion in the report 330 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 302 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 single cell data 306 may be defined as "special number". Likewise, the named range 312 may be defined as "my data".

The selected objects are published in the report 330. For example, the single data cell ("special number") 306, the named range ("my data") 312, the text box 318, and the line graph 320 are published in the report 330. The unselected portions of the workbook 300 are not published in the report 330.

A viewer at the client is presented with a named object view of the report 300. A report renderer converts the published objects into HTML when the workbook 300 is accessed from a server. The HTML 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 350. For example, the user interface 350 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 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 viewer may select the text box 318 and the line graph 320 for display on different web parts because the text box 318 elaborates on data presented in the line graph 320.

A tool bar that includes the names of the published objects may also be rendered on the user interface 350. The tool bar may be used for accessing specific published objects. For example, the tool bar is a drop down menu 360 that includes a list of the names of the published objects. Specifically, the drop down menu 360 includes selectable entries for "special number", "my data", text box, and line graph. The named object view of the report is rendered on the user interface 350 with the first published object (i.e., "special number"). The other published objects may be rendered at the user interface 350 in response to user input. For example, the viewer may

select an entry from the drop down menu 360 such that the user interface 350 refreshes and the published object corresponding to the selected entry is rendered on the user interface 350. Thus, each published object may be individually displayed to the viewer at the client. The named object view of the report focuses the viewer's attention on the published objects. The non-published objects of the workbook 300 are not displayed to the viewer.

Referring now to FIGURE 4, an illustrative routine will be described illustrating a process for generating a named object view of a report associated with electronic data. When reading the discussion of the routine presented herein, it should be appreciated that the logical operations of various embodiments of the present invention are implemented (1) as a sequence of computer-implemented acts or 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 in FIGURE 4, and comprising the embodiments of the present invention described herein are referred to variously as operations, structural devices, acts or modules. It will be recognized by one skilled in the art that these operations, structural devices, acts and modules may be implemented in software, in firmware, in special purpose digital logic, and any combination thereof without deviating from the spirit and scope of the present invention as recited within the claims attached hereto.

FIGURE 4 is a process diagram showing aspects of an illustrative process for generating a named object view of a report associated with electronic data. Processing begins at a start block where electronic data is stored as a file in a data store. A report of the file is created by a user and rendered to a viewer at a client as discussed in detail below.

The file is retrieved from the data store and opened at block 400. The file includes multiple objects. The objects may be any item in the file that can be individually selected or manipulated. The objects may be associated with static or dynamic information.

Moving to block 410, the objects in the file are named. In one embodiment, an object is automatically assigned a name. For example, two images may be assigned the names "image1" and "image2" respectively. In another embodiment, the name of an object is user-defined. For example, a single cell or a named range in a spreadsheet file may be defined with a name that communicates meaning to the user. The name of any object may also be renamed by the user.

Transitioning to block 420, a selection of objects to publish in the report is received. In one embodiment, the user selects the objects to publish from a list of the named objects. In another embodiment, the user may navigate the file and select each object to be published directly

from the file. In yet another embodiment, the user may select the objects to publish from multiple files.

Proceeding to block 430, the named object view of the report is generated using the published objects. Any objects in the file that are not selected for publication are not included in the report.

Continuing to decision block 435, a determination is made whether the named object view of the report is to be displayed on a web page. The named object view of the report is displayed on a web page when the file is retrieved from the data store via a server, in which case processing proceeds to block 440. The named object view of the report is not displayed on a web page when the file is retrieved from the data store at the client, in which case processing proceeds at block 455.

Continuing to block 440, a report renderer converts the published objects to a format suitable for display on a web page. For example, the report renderer may convert the published objects to HTML or XML. Advancing to block 450, the web-formatted version of the published objects is rendered at the client for display by a web browser. A viewer at the client is presented with a named object view of the report. In one embodiment, the first published object is rendered on a web part. The user may navigate to the other published objects such that one published object at a time is rendered in the web part.

Proceeding to block 455, the published objects are directly rendered at the client. A viewer at the client is presented with a named object view of the report. In one embodiment, the first published object is rendered in a user interface at the client. The user may navigate to the other published objects such that one published object at a time is rendered in the user interface.

Moving to block 460, a tool bar is rendered at the client. The tool bar includes the names of the published objects such that the viewer may navigate to specific published objects. In one embodiment, the tool bar is a drop down menu of a list of the names of the published objects. The drop down menu is rendered in the web part or the user interface.

Transitioning to block 470, a particular published object selection is received at the tool bar from the viewer. For example, the viewer may select a published object from the drop down menu. Proceeding to block 480, the selected published object is rendered at the client. For example, the web part or the user interface refreshes in response to the viewer input such that the selected published object is displayed on the web part or the user interface. Thus, the viewer's attention is focused on the published objects in the report. The unpublished file data is not presented to the viewer and may be disregarded. Processing then terminates at an end block.

Based on the foregoing, it should be appreciated that the various embodiments of the invention include a method, system, apparatus, and computer-readable medium for generating a

named object view of a report associated with electronic data. 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.

5

WE CLAIM:

1. A method for generating a named object view of a report associated with a file, the method comprising:

receiving a request for the report;

retrieving the file associated with the requested report from a data store;

receiving a selection of objects in the file, wherein the selected objects are published;

generating the report with the published objects; and

rendering the report with the published objects at a client.

2. The method of Claim 1, wherein receiving a request for the report further comprises receiving the request for the report at a server.

3. The method of Claim 1, further comprising converting the published objects into a format suitable for rendering at the client by a web browser, wherein the report is rendered at the client by the web browser.

4. The method of Claim 1, wherein rendering the report further comprises rendering each published object individually at the client such that the published objects are displayed on the client one at a time.

5. The method of Claim 1, wherein the file is associated with a spreadsheet application.

6. The method of Claim 5, wherein at least one object comprises a named range.

7. The method of Claim 1, further comprising assigning a unique name to each object.

8. The method of Claim 7, wherein assigning a unique name further comprises assigning a unique name to an object automatically.

9. The method of Claim 7, wherein rendering the report further comprises rendering the report with a tool bar, wherein the tool bar identifies the published objects by name.

10. The method of Claim 9, further comprising displaying a particular published object in a user interface at the client when the name associated with the particular published object is selected from the tool bar.

11. A system for generating a named object view of a report associated with a file, the system comprising:

a client, wherein a request for the report is generated at the client;

a data store, wherein the file associated with the requested report is stored in the data store; and

a server coupled to the client and the data store, wherein the server is arranged to:

receive the request for the report from the client;

retrieve the file associated with the requested report from the data store;

receive a selection of objects in the file, wherein the selected objects are published;

generate the report with the published objects; and

render the report with the published objects at the client.

12. The system of Claim 11, wherein the server is further arranged to convert the published objects into a format suitable for rendering at the client.

13. The system of Claim 11, wherein the server is further arranged to render each published object individually at the client such that the published objects are displayed on the client one at a time.

14. The system of Claim 11, wherein the server is further arranged to assign a unique name to each object in the file.

15. The system of Claim 14, wherein the objects are selected for publication by name.

16. The system of Claim 14, wherein the server is further arranged to render the report with a tool bar, wherein the tool bar identifies the published objects by name.

17. The system of Claim 16, wherein the server is further arranged to display a particular published object in a web part at the client when the name associated with the particular published object is selected from the tool bar.

18. A computer-readable medium having computer-executable instructions stored thereon which, when executed by a computer, will cause the computer to perform a method for generating a named object view of a report associated with a file, the method comprising:

receiving a request for the report;

retrieving the file associated with the requested report from a data store;

assigning a unique name to each object in the file;

receiving a selection of objects in the file, wherein the selected objects are published, and wherein the objects are selected by the corresponding unique name;

generating the report with the published objects;

rendering the report with the published objects, wherein the report lists the unique names assigned to the published objects;

receiving a selection of the unique name of a particular published object; and

displaying the particular published object corresponding to the selected unique name, wherein the particular published object is displayed in a user interface at the client.

19. The computer-readable medium of Claim 18, further comprising converting the published objects into a format suitable for rendering at the client by a web browser, wherein the report is rendered at the client by the web browser.

20. The computer-readable medium of Claim 18, wherein assigning a unique name further comprises assigning a unique name to an object automatically.